UNIVERSITÀ
DEGLI STUDI
DI PADOVA

Sede Amministrativa: Università degli Studi di Padova
Dipartimento di Matematica

Scuola di Dottorato di Ricerca in Scienze Matematiche
Indirizzo: Matematica Computazionale
Ciclo: XXVI

REGULARIZATION TECHNIQUES BASED ON KRYLOV SUBSPACE METHODS FOR ILL-POSED LINEAR SYSTEMS

Direttore della Scuola: Ch.mo Prof. Paolo Dai Pra

Coordinatore d'Indirizzo: Ch.ma Prof.ssa Michela Redivo Zaglia

Supervisore: Dott. Paolo Novati

Dottoranda: Silvia Gazzola

Riassunto

Questa tesi è incentrata sulle tecniche di regolarizzazione per problemi lineari discreti malposti e di grandi dimensioni. Molteplici applicazioni fisiche ed ingegneristiche sono modellate da questo genere di problemi che, in ambito continuo, sono spesso formulati mediante equazioni integrali di Fredholm di prima specie con nucleo regolare. Più precisamente, queste equazioni modellano i cosiddetti problemi inversi, cioè problemi in cui la causa di un effetto osservato deve essere ricostruita. Una volta discretizzati, questi problemi si presentano come sistemi lineari, la cui matrice dei coefficienti è fortemente malcondizionata e il cui vettore dei termini noti è affetto da qualche perturbazione (spesso chiamata rumore). In questo contesto, risolvere direttamente il sistema lineare discretizzato produrrebbe una soluzione priva di significato, in quanto pesantemente dominata da errori; inoltre, a causa delle grandi dimensioni del sistema, tale procedimento potrebbe risultare infattibile, perchè computazionalmente troppo costoso. Pertanto, qualche forma di regolarizzazione deve essere applicata in modo da poter calcolare una approssimazione fisicamente significativa della soluzione esatta del problema trattato: regolarizzare significa appunto sostituire il sistema lineare con un problema ad esso collegato ma avente migliori proprietà numeriche.

La prima parte di questa tesi (Capitolo 1) offre una panoramica sui problemi inversi e descrive brevemente le loro proprietà nel continuo. Quindi, nel discreto, vengono esaminati i più comuni metodi di regolarizzazione basati su una qualche fattorizzazione della matrice del sistema.

La restante parte della tesi riguarda le tecniche di regolarizzazione iterative che consistono nell'applicazione di metodi di Krylov: questo tipo di regolarizzazione è particolarmente appropriato quando devono essere risolti sistemi lineari di grandi dimensioni. Più precisamente, nel Capitolo 2, viene proposta un'accurata descrizione dei metodi di Krylov più popolari nell'ambito della regolarizzazione: storicamente, i primi metodi ad essere utilizzati a tale scopo sono stati quelli legati alle equazioni normali e le proprietà regolarizzanti di molti di essi sono già state analizzate. Per quanto riguarda i metodi basati sull'algoritmo di Arnoldi, la situazione è differente: nella maggior parte dei casi, le loro proprietà regolarizzanti non sono ancora state rigorosamente studiate. Pertanto, sempre nel Capitolo 2, viene proposta un'analisi originale delle proprietà approssimanti dell'algoritmo di Arnoldi nel caso in cui esso venga impiegato per la risoluzione di sistemi lineari malposti: l'obbiettivo di questa analisi è di fornire maggiori spiegazioni riguardo all'utilizzo dei metodi basati sull'algoritmo di Arnoldi per la regolarizzazione.

I risultati più significativi presentati nella tesi riguardano la classe dei metodi di tipo Arnoldi-Tikhonov, introdotti per la prima volta una decina di anni fa e descritti nel Capitolo 3. L'approccio di tipo Arnoldi-Tikhonov consiste nel risolvere, mediante un metodo iterativo basato sull'algoritmo di Arnoldi, un problema regolarizzato tramite il metodo di Tikhonov. Rispetto ad un approccio regolarizzante puramente iterativo, i metodi di tipo Arnoldi-Tikhonov sono in grado di calcolare soluzioni approssimate più accurate, in quanto all'interno del procedimento iterativo di tipo Arnoldi-Tikhonov possono essere facilmente incorporate alcune informazioni

sul comportamento e sulla regolarità della soluzione. Fra i maggiori problemi aperti legati all'utilizzo dei metodi di tipo Arnoldi-Tikhonov figurano la ricerca di metodi efficienti per la scelta dei parametri di regolarizzazione e la scelta di opportune matrici di regolarizzazione. Le problematiche relative alla scelta dei parametri sono trattate nel Capitolo 4, dove vengono derivate due nuove tecniche che possono essere utilizzate congiuntamente ai metodi di tipo Arnoldi-Tikhonov; sempre nel Capitolo 4 viene descritta una nuova estensione del metodo di Arnoldi-Tikhonov al caso della regolarizzazione di Tikhonov a più parametri. Infine, nel Capitolo 5, vengono presentate due innovative ed efficienti strategie per approssimare la soluzione di problemi regolarizzati nonlineari: più precisamente, i termini di regolarizzazione inizialmente definiti utilizzando la norma 1 o il funzionale di Variazione Totale (TV) sono approssimati mediante opportune matrici di regolarizzazione che vengono aggiornate adattivamente durante le iterazioni del metodo di Arnoldi-Tikhonov.

In generale, nel corso della trattazione, vengono illustrati i risultati di molteplici esperimenti numerici, con l'obbiettivo di mostrare il comportamento dei nuovi metodi proposti e di confrontarli con quelli già esistenti.

Abstract

This thesis is focussed on the regularization of large-scale linear discrete ill-posed problems. Problems of this kind arise in a variety of applications, and, in a continuous setting, they are often formulated as Fredholm integral equations of the first kind, with smooth kernel, modeling an inverse problem (i.e., the unknown of these equations is the cause of an observed effect). Upon discretization, linear systems whose coefficient matrix is ill-conditioned and whose right-hand side vector is affected by some perturbations (noise) must be solved. In this setting, a straightforward solution of the available linear system is meaningless because the computed solution would be dominated by errors; moreover, for large-scale problems, solving directly the available system could be computationally infeasible. Therefore, in order to recover a meaningful approximation of the original solution, some regularization must be employed, i.e., the original linear system must be replaced by a nearby problem having better numerical properties.

The first part of this thesis (Chapter 1) gives an overview on inverse problems and briefly describes their properties in the continuous setting; then, in a discrete setting, the most well-known regularization techniques relying on some factorization of the system matrix are surveyed.

The remaining part of the thesis is concerned with iterative regularization strategies based on some Krylov subspaces methods, which are well-suited for large-scale problems. More precisely, in Chapter 2, an extensive overview of the Krylov subspace methods most successfully employed with regularizing purposes is presented: historically, the first methods to be used were related to the normal equations and many issues linked to the analysis of their behavior have already been addressed. The situation is different for the methods based on the Arnoldi algorithm, whose regularizing properties are not well understood or widely accepted, yet. Therefore, still in Chapter 2, a novel analysis of the approximation properties of the Arnoldi algorithm when employed to solve linear discrete ill-posed problems is presented, in order to provide some insight on the use of Arnoldi-based methods for regularization purposes.

The core results of this thesis are related to class of the Arnoldi-Tikhonov methods, first introduced about ten years ago, and described in Chapter 3. The Arnoldi-Tikhonov approach to regularization consists in solving a Tikhonov-regularized problem by means of an iterative strategy based on the Arnoldi algorithm. With respect to a purely iterative approach to regularization, Arnoldi-Tikhonov methods can deliver more accurate approximations by easily incorporating some information about the behavior of the solution within the reconstruction process. In connection with Arnoldi-Tikhonov methods, many open questions still remain, the most significant ones being the choice of the regularization parameters and the choice of the regularization matrices. The first issues are addressed in Chapter 4, where two new efficient and original parameter selection strategies to be employed with the Arnoldi-Tikhonov methods are derived and extensively tested; still in Chapter 4, a novel extension of the Arnoldi-Tikhonov method to the multi-parameter Tikhonov regularization case is described. Finally, in Chapter 5, two efficient and innovative schemes to approximate the solution of nonlinear regularized problems are presented: more precisely, the regularization terms originally defined by the 1-

norm or by the Total Variation functional are approximated by adaptively updating suitable regularization matrices within the Arnoldi-Tikhonov iterations.

Along this thesis, the results of many numerical experiments are presented in order to show the performance of the newly proposed methods, and to compare them with the already existing strategies.

Acknowledgements

At the end of my exciting experience as a Ph.D. student, there are many people I'd like to thank for the important role they played in my professional and personal growth.

First of all, I'd like to express my deep gratitude to my advisor, Paolo Novati: he's a wonderful person and collaborating with him has been an excellent experience. When I arrived in Padua, he introduced me to the study of regularization methods and, during the last three years of my academic life, he constantly stimulated and helped me to direct my research. In particular I'd like to thank him for always encouraging me, for trusting me, for providing me helpful advices, and for giving me the opportunities and the freedom to attend many meetings, conferences and to spend some time abroad.

I'm very grateful to James Nagy: he kindly accepted to host me at Emory University for four months; this experience has been awesome and unforgettable. I'd like to thank him for his generous welcome, for proposing and working with me on a very interesting application of the techniques I've been studying so far, for further motivating and addressing me, and for supporting me in many ways even after my period in the United States.

Among the people I met at the Math Department of Padua, I'd like to thank Prof. Michela Redivo Zaglia, for helping me in many circumstances, especially for advising me about many worthwhile conferences and courses, and for providing the necessary bureaucratic background of many activities. I'd like to thank also Maria Rosaria Russo, for passing me some useful codes and softwares and for helping me in understanding and using them. I'd like to thank Marco Donatelli, for inviting me to attend some useful doctoral courses at the University of Insubria.

I count myself fortunate to have met some outstanding professors willing to listen to me and to give me further motivations that had an influence on my work: I'd like to mention Michele Benzi, Dianne O'Leary and Lothar Reichel.

Looking back at my years as a master student in Parma, I'd like to thank Alessandra Aimi: without her encouragement I wouldn't probably have thought of going on with my academic career.

I'm grateful to many friends I have in Padua: Gabriella, Marco, Federico, Stefano, Giulio, Anna, Stefano, Daniele, Paolo, Simone, Vittorio, Alice, Giulia, Cecilia, Giovanni, Gabriele, Juan Miguel, Martino, Marisa, Manuela. Most of them are present or former Ph.D. students and, in some circumstances, they have been acting as a second family away from home. I also mention some friends that I met in Atlanta and that helped making my experience even more enjoyable: I'd like to thank Sebastian, Luca, Hana, Scott, Ryan and Ida.

Last but not least, I'd like to express my most profound gratitude to my family: I thank my father Francesco and my mother Orestina for their everlasting love, for the education they gave me, for their unlimited support, and for letting me free to pursue my own inclinations during all my rational life. I thank my brother Paolo, for sharing with me his passion for music, and for being (in rather hidden ways!) proud of his older sister.

Finally, I thank you, Marco: there could never be enough words to express my gratitude for your support, your understanding, your encouragements. Thanks for always believing in me (more than I often do!), for giving me the strength to go on in many situations and for being near me even now that we live thousands of miles away; wherever I'll be from now on, I hope not to be too far away from you.

Contents

Introduction

This thesis is focussed on the study of regularization techniques for large-scale linear discrete ill-posed problems: these kind of problems arise in many fields of sciences and engineering; basically, every time that a cause of an observed effect should be recovered, one is dealing with an inverse problem. Solving inverse problems is not straightforward: from a numerical point of view, the main difficulty is that the solution of an inverse problem is very sensitive to the perturbations affecting the data. Since the data are typically recovered experimentally and through measurements, one cannot assume to work with exact quantities and the solutions that one obtains applying some standard techniques (both direct and iterative) are typically meaningless. Because of the important applications that involve the solution of an inverse problem (just to list some of them: seismic inversion, gravity surveying, back heat conduction, signal processing, medical imaging) and because of the inherent difficulties encountered in the process of approximating a meaningful solution, the study of inverse problems constitutes a very active field of research. The solution of an inverse problem requires some regularization. Regularizing means substituting the available problem with a related one that has better numerical properties. The first works on regularization date back to the late-fifties or early-sixties (we cite, in chronological order, [107, 100, 121, 119, 41]) and, at present, several different regularization techniques have been developed and have been proved to be very effective when employed to solve many classes of inverse problems coming from different applications. The first works on regularization essentially propose a direct or a variational approach to regularization, which has to be applied on the continuous or on the discrete problem. An efficient implementation of the direct regularization techniques on the discretized problem typically requires some standard factorizations of the system matrix to be computed.

In this thesis we propose various new strategies to deal with the regularization of large-scale problems: in general, in this framework, we cannot assume any factorizations of the coefficient matrix to be available, because of their high computational cost. Therefore, in these cases, just an iterative approach to regularization is possible. Many efficient iterative regularization methods have been proposed during the last twenty years: most of them are based on a purely iterative approach (regularization by early termination of the iterations), but some more sophisticated techniques merging a variational and an iterative approach to regularization have been considered, as well. In particular, when iteratively projecting the full-dimensional discrete problem onto Krylov subspaces of increasing dimension, i.e., when adopting Krylov subspace methods, one can typically achieve a good approximation of the desired solution in just a few iterations. This is due to the fact that, usually, the first basis vectors of a Krylov subspace can reproduce the basic features of the exact solution; however, when too much iterations are performed, the reconstructed solution begins to degenerate because of the noise components that dominate it. An alternative form of iterative regularization consists in considering the full-dimensional Tikhonov-regularized problem and in projecting it onto Krylov subspaces of increasing dimension: this is the basic idea behind the Arnoldi-Tikhonov (AT) method. The benefit of adopting an approach like this is that more accurate and stable solutions can be recovered by solving the projected regularized problems instead of just projecting the available system: for instance, by adopting the Arnoldi-Tikhonov strategy, one can more easily enforce

some additional regularity constraints into the reconstructed solution. The goal of this thesis is to derive some alternatives to the existing Arnoldi-Tikhonov methods, in order to improve their efficiency and reliability and to solve a broader class of problems. In particular, the main contributions of this thesis are concerned with:

- the issue of incorporating a general regularization matrix into the classical AT scheme;
- the issue of choosing a regularization parameter at each iteration of the AT method, as well as a stopping criterion for the iterations;
- the issue of adapting the AT approach to work when the norm of the regularization and fit-to-data terms in Tikhonov regularization are different from the standard 2-norm ones.

We believe that the Arnoldi-Tikhonov method is a very powerful strategy that can be employed to regularize large-scale problems originating in many applications; since it was first derived quite recently in [19], some work can still be done in improving and generalizing it, and in studying its theoretical properties.

Throughout this thesis, we work directly on some discretization of a linear continuous inverse problem, i.e., we are just concerned with the solution of ill-conditioned linear systems. Moreover, we assume that the considered linear model is known exactly and that just the right-hand side vector of the linear system is affected by some perturbation. In this thesis, in addition to describing some original methods, we try to provide some background material on regularization and on iterative projection methods, in order to gain some more insight into the newly proposed methods and to give an unifying framework for the iterative regularization techniques.

The main application we consider in this thesis is the deblurring and denoising of images: this application is particularly important in astronomy (when recovering images taken from telescopes and satellites) and in medicine (when performing diagnosis by images). However, we underline that we take into account just test problems whose solutions are known: in this way we are able to assess the quality of the obtained restorations and to perform some comparisons with already existing methods. Anyway, we believe that the data considered can quite truthfully reproduce what happens in real-life problems. All the computations have been executed using Matlab 7.10 with 16 significant digits on a single processor computer Intel Core i3-350M.

We now summarize the contents of each chapter:

Chapter 1. We introduce some models of linear inverse problems that can be described by Fredholm integral equations of the first kind. We describe some important properties, typical of the continuous models, that are "inherited" by the corresponding discrete problems. We briefly address some discretization issues and we introduce some core factorization methods that are employed to describe and analyze discrete ill-posed problems. Finally we list some test problems that are used throughout the thesis and we review the most well-known and basic forms of regularization, with a particular emphasis on Tikhonov regularization.

Chapter 2. We describe the basic ideas behind projection methods, providing in this way a common framework for Krylov subspace methods. Iterative methods based on the Arnoldi and Lanczos bidiagonalization algorithms are briefly introduced. An important part of this chapter is devoted to the derivation of some convergence and approximation properties of the Arnoldi algorithm when applied to discrete ill-posed problems: in particular we analyze the behavior of the Krylov basis vectors, we give some estimates on the approximation of the SVD of the original matrix by means of the SVD of the projected matrices, along with some estimates on the convergence of Arnoldi-based iterative methods and, in particular, of the GMRES method. Finally the regularization properties of some iterative methods are explained and the class of the hybrid methods is introduced.

Chapter 3. In the first part of this chapter we describe the class of the Arnoldi-Tikhonov methods, starting from the most basic one that can work in connection with standard form Tikhonov regularization. In the second part of this chapter we survey some extensions of the AT methods that have been proposed to work in connection with general form Tikhonov regularization; we also derive an original strategy that can work with generic regularization matrices.

Chapter 4. This chapter is focussed on the choice of the regularization parameter when performing Arnoldi-Tikhonov regularization. First of all, we analyze how some popular parameter choice strategies have been adopted to work with the most well-regarded regularization methods. We focus on the discrepancy principle, the L-curve criterion and the Generalized Cross validation method. Then, most of these strategies are adapted to work in connection with the Arnoldi-Tikhonov method. In particular, in the case of the discrepancy principle, we propose an original scheme, called secant update method, that has been derived starting from a linear approximation of the discrepancy principle and that leads to an efficient parameter update strategy to be employed at each step of the Arnoldi algorithm and to be used as a stopping criterion. Finally, we deal with the so-called multi-parameter Tikhonov method: this method has been introduced to allow more regularization matrices into the regularization process, forcing in this way a potentially more accurate reconstruction. We extend the Arnoldi-Tikhonov method to work in connection with multi-parameter regularization and we extend the secant update method to act simultaneously as a stopping criterion and as an update formula to properly set the values of the regularization vector at each iteration.

Chapter 5. We describe some extensions of the Arnoldi-Tikhonov method that can successfully approximate the solution of Tikhonov-like regularization methods having the fit-to-data and the regularization terms evaluated in a generic p-norm. Depending on the regularization matrices that one wishes to employ, we propose two approaches. The first one is based on the idea of involving particular regularizers into the definition of the Krylov subspace: since the regularizers are updated at each iterations, the approximate solutions belongs to flexible Krylov subspaces. The second one is based on suitable restarts of the Arnoldi algorithm.

The material in this thesis extends and completes the methods described in the published or submitted papers [38, 39, 37, 40] (listed in chronological order).

Chapter 1

An overview of inverse problems and regularization

Inverse problems are ubiquitous in many areas of science and engineering. Every time we want to recover the cause of an observed consequence, or reconstruct the internal configuration of a system given its recovered effects, we are dealing with an inverse problem. These problems are very sensitive with respect to perturbations in the available data. Since, typically, the data are affected by some measurement errors, there is no hope to recover a meaningful approximation of the solution directly from the original system and some sort of regularization must be employed. Regularizing means substituting the available problem with a related one having better numerical properties (in particular, it should be more robust with respect to perturbations in the data). In Section 1.1 we explain the basic analytic features of linear inverse problems. Then, in Section 1.2, we describe some common test problems that are used through all the thesis to illustrate and to validate the proposed algorithms; particular emphasis is posed on image restoration problems. Finally, in Section 1.3 we extensively survey the most well-established regularization methods, classifying them either as direct or iterative. In this way we provide the ground to describe, during the following chapters, the original contributions of this thesis. This chapter is based on the excellent introductions provided in [48, 55, 59, 64].

1.1 Linear inverse ill-posed problems

The concept of ill-posed problem goes back to Hadamard [45] that, when working on problems in mathematical physics, defined a linear problem as well-posed if it simultaneously satisfies the following three requirements:

- the problem has a solution (existence);
- there is only one solution (uniqueness);
- the solution depends continuously on the data (stability).

If just one of the above requirements is violated, the problem is said ill-posed. As far as the first or the second conditions are violated, one can still try to slightly reformulate the problem or add some additional requirements. However, when the third requirement does not hold, arbitrarily small perturbation of the data can produce arbitrarily large perturbation in the solution (for an immediate understanding of this phenomenon, see Figure 1.1.7); in this case one tries to remedy by substituting the original problem with a nearby one that is less sensitive to perturbation: this process is called regularization. The regularized problem can then be solved by standard numerical techniques. Hadamard was mistaken in believing that ill-posed problems lack of physical meaning, since they arise in many areas of science and engineering,

basically every time that one is interested in recovering the internal structure of a system from its measured behavior, i.e., when dealing with inverse problems. Some examples of inverse problems include medical imaging (such as computerized tomography), seismic inversion and geophysical prospecting, image deblurring and signal processing.

1.1.1 Analytical description

In the following we exclusively focus on linear inverse problem that can be formulated as Fredholm integral equations of the first kind, which can be generically written as

$$\int_\Omega \hat{k}(s,t)f(t)dt = g(s), \tag{1.1.1}$$

where $\hat{k} \in L^2(\Omega \times \Omega)$ and $s, t \in \Omega \subset \mathbb{R}^q$. In the above equation, both the kernel $\hat{k}$ and the right-hand side function g are assumed to be known, the only unknown being the function f; $\hat{k}$ models the process that acts on an unknown input f and produces a known output g. Indeed, recalling what is said at the beginning of Section 1.1, we have to assume g to be available just approximatively, since it is typically affected by approximation or rounding errors due, for instance, to some inaccuracies of the measuring device. Therefore the available function is of the form

$$g = g^{ex} + \epsilon \tag{1.1.2}$$

where g^{ex} is the ideally exact output of the system and ϵ is an unknown perturbation. Assuming that Ω is compact, let us introduce the linear compact operator $K : L^2(\Omega) \to L^2(\Omega)$ defined by

$$Kf = \int_\Omega \hat{k}(\cdot,t)f(t)dt\,;$$

we assume K to be injective (therefore we define $K^{-1} : \mathcal{R}(K) \to L^2(K)$). The topology we consider is the usual L^2 one, equipped with the distance, norm, and scalar product defined by

$$d(z,w) = \|z-w\|, \quad \|x\| = (x,x)^{1/2}, \quad (x,y) = \int_\Omega x(t)y(t)dt.$$

Since K is compact, it admits a singular value expansion (SVE), i.e. there exists a non-increasing sequence of nonnegative singular values $\{\sigma_j\}_{j\geq 1}$ that quickly converges to zero, with corresponding singular function u_j, v_j such that

$$Kv_j = \sigma_j u_j\,; \tag{1.1.3}$$

in particular, $\{v_j\}_{j\geq 1}$ is an orthonormal basis of $L^2(\Omega)$. Expanding f and g with respect to the v_j's and the u_j's (accordingly) and using the relation (1.1.3) we get

$$f = \sum_{j=1}^{\infty}(f,v_j)v_j, \quad g = \sum_{j=1}^{\infty}(g,u_j)u_j = \sum_{j=1}^{\infty}\sigma_j(f,v_j)u_j. \tag{1.1.4}$$

It is well known that applying the operator K has a smoothing effect on f, in the sense that high-frequency components and edges are leveled by the integration: in particular, the higher the oscillations, the more severe the dumping. This phenomenon is theorized by the Riemann-Lebesgue lemma: let us consider the function $f_p(t) = \sin(2\pi pt)$, $p = 1, 2, \ldots$; then

$$g_p(s) = \int_\Omega \hat{k}(s,t)f_p(t)dt \to 0 \quad \text{for} \quad p \to \infty.$$

Since the the parameter p controls the frequency of the function f_p, we can state that the higher frequencies are damped by K and the corresponding function g_p is smoother than f_p. We also remark that

1. the smoother the kernel $\hat{k}$, the faster the singular values $\{\sigma_j\}_{j\geq 1}$ decay to zero. Specifically, if the derivatives of $\hat{k}$ of order $0, 1, \ldots, r$ exist and are continuous, then the σ_j's decay approximatively as $j^{-r-1/2}$, i.e., $\sigma_j = O(j^{-r-1/2})$; if $\hat{k}$ is analytical, then the decay is exponential, i.e., $\sigma_j = O(\rho^j)$, $0 < \rho < 1$. In particular, in [68, Definition 2.42] the following classification is proposed: if there exists $\alpha > 0$ such that $\sigma_j = O(j^{-\alpha})$, then α is called degree of ill-posedness and the problem is mildly ill-posed if $0 < \alpha \leq 1$, moderately ill-posed if $\alpha > 1$; the problem is severely ill-posed if $\sigma_j = O(e^{-\alpha j})$;

2. the smaller the σ_j, the more oscillating the corresponding singular functions u_j and v_j.

Looking at the expressions in (1.1.4) we understand that the above remarks further describe the smoothing effect of K.

Unfortunately, in this setting, there are no guarantees that K^{-1} is a continuous operator. Indeed, we should expect the inverse problem to amplify the frequencies: in particular, the higher the frequency, the more severe the amplification. The functions g that can be inverted are fully characterized by the following

Picard condition. A function $g \in L^2(\Omega)$ given by

$$g = \sum_{j=1}^{\infty} (g, u_j) u_j$$

belongs to $\mathcal{R}(K)$ if and only if

$$\sum_{j=1}^{\infty} \left(\frac{(g, u_j)}{\sigma_j} \right)^2 < \infty. \tag{1.1.5}$$

The necessary condition is quite natural to derive: if $g \in \mathcal{R}(K)$, directly from the relations (1.1.3) and (1.1.4) we get

$$f = \sum_{j=1}^{\infty} \frac{(g, u_j)}{\sigma_j} v_j.$$

To better understand the role of the Picard condition we remark that, being g an L^2 function, we already know that the coefficients (g, u_j) tend to zero faster than $j^{-1/2}$; the Picard condition requires something more since, if $g \in \mathcal{R}(K)$, then the coefficients (g, u_j) tend to zero faster than $\sigma_j j^{-1/2}$ (recall that $\sigma_j \to 0$). In practice, dealing with functions g of the form (1.1.2) and even assuming $\|\epsilon\| < \|g^{ex}\|$, we cannot expect the perturbation ϵ to satisfy the Picard condition. Therefore $g \notin \mathcal{R}(K)$ and, if we naively try to recover f by inverting K, we inevitably obtain an useless result $\hat{f}$ with extremely large norm. Even in a continuous framework, we need to employ a regularization method to recover a stable solution that approximates the desired f. Indeed, the most basic requirement of a regularization method is that $\hat{f} \to f$ as $\|\epsilon\| \to 0$ (cf. [120, Sect. II.1]).

Concluding this section, we mention that there are important cases of Fredholm integral equations whose associated operator is not compact: we cite the inverse Laplace transformation and the convolution equations. The latter are a special case of equation (1.1.1), obtained considering $\hat{k}(s, t) = h(s - t)$, i.e.,

$$\int_{\Omega} h(s - t) f(t) dt = g(s). \tag{1.1.6}$$

The corresponding operators have a continuum instead of countably many singular values, which nevertheless accumulate at the origin. However, the above discussion basically applies also to these problems; we are not going into the details because the target of this thesis is to discuss regularization on the discrete problems. In the following sections we propose some examples showing the discrete analogous of the features just described.

1.1.2 Discretization "matters"

The first step to obtain a computer approximation of the problem (1.1.1) is to discretize it, in such a way that a system of linear algebraic equations is obtained. There are basically two ways to achieve this:

- **collocation methods**. A quadrature formula (for instance the midpoint rule, cf. [101, Chapter 9]) is employed to formally rewrite the left-hand side of equation (1.1.1) as

$$\sum_{j=1}^{n} w_j \hat{k}(s, t_j) f(t_j) + E_n(s), \quad j = 1, \ldots, n$$

  Then the quadrature error $E_n(s)$ is discarded and the collocation requirement that the above function must be equal to the right-hand side of (1.1.1) at k selected points is enforced, i.e.,

$$\sum_{j=1}^{n} w_j \hat{k}(s_i, t_j) \widetilde{f}(t_j) = g(s_i), \quad i = 1, \ldots k.$$

  In the above equation, we introduced the function $\widetilde{f}$ to underline that, neglecting the contribution $E_n(s)$, the exact values $f(t_j)$ are no longer taken into account. The above equation leads to the problem

$$\min_{x \in \mathbb{R}^n} \|b - Ax\|_2, \quad \text{where} \quad \begin{array}{rcl} A_{i,j} & = & w_j \hat{k}(s_i, t_j) \\ x_j & = & \widetilde{f}(t_j) \\ b_i & = & g(s_i) \end{array} ;$$

- **projection methods**. We focus on the Petrov-Galerkin method. Let us consider an approximation of f and g of the form

$$f(t) = f_n(t) + E_f(t) = \sum_{i=1}^{n} x_i \phi_i(t), \quad g(s) = g_n(s) + E_g(s) = \sum_{j=1}^{n} b_j \psi_j(s),$$

  where $\{\phi_i(s)\}_{i=1,\ldots,n}$, $\{\psi_i(t)\}_{i=1,\ldots,n}$ are two sets of basis functions for two suitable finite dimensional subspaces of $L^2(\Omega)$. After substituting the above expressions into equation (1.1.1), in order to determine the unknown coefficients x_i, $i = 1, \ldots, n$, we impose the residual

$$\sum_{i=1}^{n} x_i \int_{\Omega} \hat{k}(s, t) \phi_i(t) dt - \sum_{i=1}^{n} b_i \psi_i(s)$$

  to be orthogonal to the space $\text{span}\{\psi_1, \ldots, \psi_n\}$. In this way we obtain the system

$$Ax = b, \quad \text{where} \quad \begin{array}{rcl} A_{i,j} & = & (\psi_i, K\phi_j) \\ b_i & = & (\psi_i, g) \end{array} , \; i, j = 1, \ldots, n.$$

  An important aspect of projection methods is that, if the subspaces are suitably chosen, they themselves have inherent regularization properties. Indeed, if the discretization is too coarse, then the finite dimensional problem will be fairly well-conditioned but its solution will be a rough approximation of the exact one, being affected by a large discretization error; on the contrary, if the discretization is too fine, then the ill-posedness of the original problem is "inherited" into the finite dimensional one and its solution is meaningless because of the high sensitiveness to the perturbations. We mention in passing that the class of regularization methods we are focussing on, consist in projecting once more the finite dimensional problems into suitably chosen Krylov subspaces (cf. Sections 2.2 and 3.1).

We remark that, at the discretization stage, there are indeed two philosophies: "first regularize than discretize" and "first discretize than regularize". While we are exclusively concerned with the latter, the former consists in applying some regularization (for instance Tikhonov regularization, cf. Section 1.3) directly on the continuous problem. Finally we underline that, from a strictly mathematical point of view, every nonsingular matrix automatically has a continuous inverse. However, provided that a sufficiently fine discretization has been considered, the ill-posedness of the continuous problem is reproduced by the huge conditioning of the system matrix (see Section 1.1.3), and the finite dimensional problem will suffer of the same instability phenomena as the ones described in the previous section. For this reasons the originating linear systems are often referred to as discrete ill-posed problems.

From the above discussion we see that the discretized problem has the form

$$\min_x \|b - Ax\| \quad \text{or} \quad Ax = b, \tag{1.1.7}$$

depending on the size and the properties of the involved coefficient matrix. To keep the discussion general, in this setting we take into account also rectangular matrices, even if the original contributes of this thesis are focussed on the case $A \in \mathbb{R}^{n\times n}$, since they are all related to use of Krylov subspace methods (cf. Section 2.3) for regularization purposes. Recall that, in the continuous setting, we were concerned with perturbations of the right-hand side of (1.1.1); analogously, in this framework, we have to pay attention to the so-called noise e that affects the vector b. Let us assume that there exists an exact solution x^{ex} that satisfies

$$x^{ex} = \arg\min_x \|b^{ex} - Ax\| \quad \text{or} \quad Ax^{ex} = b^{ex}; \tag{1.1.8}$$

the discrete counterpart of the relation (1.1.2) is given by the following relation

$$b = b^{ex} + e, \quad \|e\| < \|b^{ex}\|, \tag{1.1.9}$$

which links the ideally exact vector b^{ex} in (1.1.8) and the available vector b in (1.1.7). In the following we assume that the only errors are confined in the right-hand side of (1.1.7) even if, to be more precise, also the errors that affect the matrix A should be taken into account; the norm of the perturbation e is denoted by ε, i.e. $\varepsilon = \|e\|$. We furthermore assume that the noise is white Gaussian, meaning that all the elements of the vector e are drawn from the same Gaussian distribution with zero mean and standard deviation ν. This is equivalent to saying that the covariance matrix for e is a scaled identity, i.e., $\mathrm{Cov}(e) = \nu^2 I$. The ratio

$$\widetilde{\varepsilon} = \frac{\|e\|}{\|b^{ex}\|} = \frac{\varepsilon}{\|b^{ex}\|}$$

is called noise level.

1.1.3 Some matrix analysis tools

In this section we introduce and we define some classical factorizations and estimates that are extremely useful to understand and analyze discrete ill-posed problems of the form 1.1.7. Unless otherwise stated, the vector norm (and the induced matrix norm) that we consider is always the 2-norm, even when not explicitly pointed out. As a matter of notation, the letter I always denotes the identity matrix; sometimes, to better underline the size n of the identity matrix, we use the notation I_n. Given the introductory nature of this chapter, we are just going to state the main results, skipping proofs and secondary remarks.

Singular Value Decomposition (SVD)

The SVD is probably the most powerful tool to analyze the ill-posedness of discrete problems and can be regarded as a finite dimensional version of the SVE introduced in Section 1.1.1. Let A be a $k \times n$ matrix. Then

$$A = U\Sigma V^T, \tag{1.1.10}$$

where the matrices $U = [u_1, \ldots, u_k] \in \mathbb{R}^{k\times k}$ and $V = [v_1, \ldots, v_n] \in \mathbb{R}^{n\times n}$ are orthogonal, and $\Sigma = \mathrm{diag}(\sigma_1, \ldots, \sigma_p) \in \mathbb{R}^{k\times n}$, $p = \min\{k, n\}$. The columns u_i's and v_i's of U and V are called left and right singular vectors, respectively; the nonnegative scalars σ_i's are called singular values and are ordered nonincreasingly. The SVD of A is strongly linked to the eigenvalue decomposition of A^TA and AA^T: in particular the singular values are the square roots of the eigenvalues. A variant of this decomposition that is meaningful for rectangular matrices with $k > n$ is the so-called "thin" SVD, where $U \in \mathbb{R}^{k\times n}$ has orthonormal columns, and $\Sigma \in \mathbb{R}^{n\times n}$ is still diagonal. In connection with discrete ill-posed problems we often have that:

1. the singular values σ_i gradually decay and cluster at zero, with no evident gap between two consecutive ones (cf. Figure 1.1.3, frame **(a)**);

2. the singular vectors u_i and v_i tend to have more sign changes (i.e., more oscillations) as i increases (cf. Figure 1.1.4, first row, and Figure 1.1.5).

We can immediately see that the above properties are analogous to the SVE ones listed in Section 1.1.1; this is a first clue that the singular values and vectors of the matrix A are approximations of the singular values and functions of the operator K. Indeed, if we assume that the continuous problem has been discretized by the Galerkin method using two sets of orthonormal basis functions $\{\phi_i(s)\}_i$ and $\{\psi_i(t)\}_i$, $i = 1, \ldots, n$, some precise estimates regarding the convergence of the SVD of the matrix A to the SVE of the corresponding operator K can be derived: in [51], the author proves that, if $\|A\|_F$ (the Frobenius norm of A) increasingly better approximates $\|K\|$, then the singular values of A increasingly better approximate n singular values of K and the functions

$$\breve{u}_j(s) = \sum_{i=1}^{n} u_{ij}\phi_i(s) \quad \text{and} \quad \breve{v}_j(s) = \sum_{i=1}^{n} v_{ij}\psi_i(s), \quad j = 1, \ldots, n$$

increasingly better approximate the left and right singular functions of K, respectively.

Regarding the second property of the above list, in [63] some interesting derivations are made in order to compare the behavior of the singular vectors and the discrete Fourier basis: starting from some analytical considerations to be made in the continuous setting and that involve the (infinite dimensional) matrix B defined by

$$B_{h,j} = \left| \left(u_j(s), \frac{e^{\hat{\imath}hs}}{\sqrt{2\pi}} \right) \right|, \quad h = -\infty, \ldots, \infty,\ j = 1, \ldots, \infty,\ \hat{\imath} = \sqrt{-1},$$

one can state that the singular functions behave similarly to the Fourier functions, meaning that singular functions $u_j(s)$ with small indices j correspond to low frequencies and singular functions $u_j(s)$ with large indices j correspond to high frequencies. Upon discretization, let us consider the unitary matrix F that represents the Discrete Fourier Transform (DFT) and let us form the product F^HU: in this way, we basically consider the finite dimensional analogous of the matrix B. The plots in Figure 1.1.1, show that the largest entries in the matrix F^HU typically have the generic shape of a "rotated V", meaning that the frequency of the vectors u_j's essentially increases with j.

Many interesting quantities linked to the matrix A can be defined starting from the SVD. For instance, we can define the rank of A as

$$\mathrm{rk}(A) = \max\{i = 1, \ldots, p \,:\, \sigma_i \neq 0\}.$$

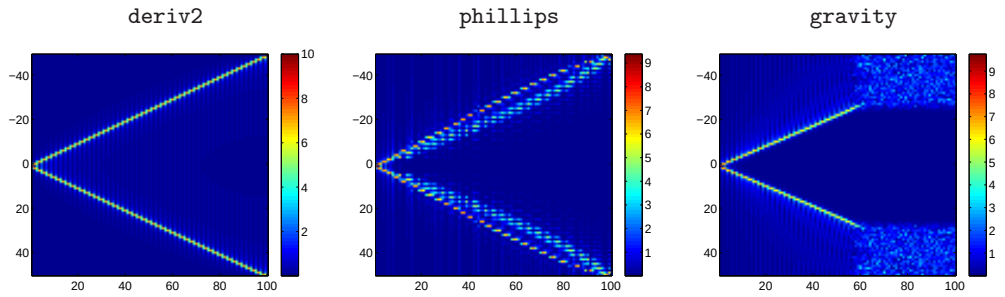

Figure 1.1.1: Plots of the modulus of the elements of the matrix $F^H U$. The size of the test problems is 100: on the horizontal axis we mark the indices of the left singular vectors and on the vertical axis we mark the frequency values. For the problem `deriv2`, we know that the singular functions are indeed the Fourier basis functions (cf. Section 1.2.1), and so the rightmost frame represents the ideal situation. We also remark that the strange behavior of the coefficients for the problem `gravity` after index 60 is due to the fact that the singular values following the 60-th one lie around the machine precision level (cf. Figure 1.1.3) and so the corresponding singular vectors carry unreliable information and they should be discarded.

Again, the best rank-r approximation of the matrix A in the 2-norm is the matrix A_r defined as

$$A_r = U_r \Sigma_r V_r^T, \tag{1.1.11}$$

where U_r and V_r are obtained considering just the first r columns of U and V, respectively, and $\Sigma_r = \mathrm{diag}(\sigma_1, \ldots, \sigma_r)$; the relation $\sigma_{r+1} = \|A - A_r\|$ holds. Finally, when dealing with a rectangular or a singular matrix, we can define the Moore-Penrose generalized inverse (or, more simply, pseudoinverse) as

$$A^\dagger = \sum_{i=1}^{\mathrm{rk}(A)} v_i \sigma_i^{-1} u_i^T \tag{1.1.12}$$

As a consequence, the solution of the problems (1.1.7) can be written as

$$x = \sum_{i=1}^{\mathrm{rk}(A)} \frac{u_i^T b}{\sigma_i} v_i. \tag{1.1.13}$$

Of course, the above expressions incorporate the case of a square nonsingular matrix A: indeed, in this case, one should just consider that $\mathrm{rk}(A) = n$. For a proof of the existence of the SVD and for some additional properties linked to the SVD, we refer to [42]. Now we rewrite the decomposition (1.1.10) in a way closer to (1.1.3), that is,

$$A v_i = \sigma_i u_i \quad \text{and} \quad A^T u_i = \sigma_i v_i, \quad i = 1, \ldots, p = \min\{k, n\}.$$

Thanks to the above relations we can state that multiplication by the matrix A still have a smoothing effect, since the oscillatory singular vectors correspond to the small singular values (cf. Figure 1.1.2). Moreover, as i increases, we can regard the right singular vector v_i as approximately belonging to the null space of the columns of A and the left singular vector u_i as approximately belonging to the null space of the rows of A. The matrix A is therefore nearly rank-deficient and its numerical null space is spanned by vectors with many sign changes.

Directly from definition (1.1.10) we see that $\|A\|_2 = \sigma_1$ and therefore the 2-norm condition number of A can be defined through the SVD in the following way:

$$\mathrm{cond}_2(A) = \|A\|_2 \|A^\dagger\|_2 = \frac{\sigma_1}{\sigma_{\mathrm{rk}(A)}}.$$

The clustering of the singular values at 0 implies that the condition number of A may be huge. A classical result in perturbation theory states that, referring to the notations introduced in

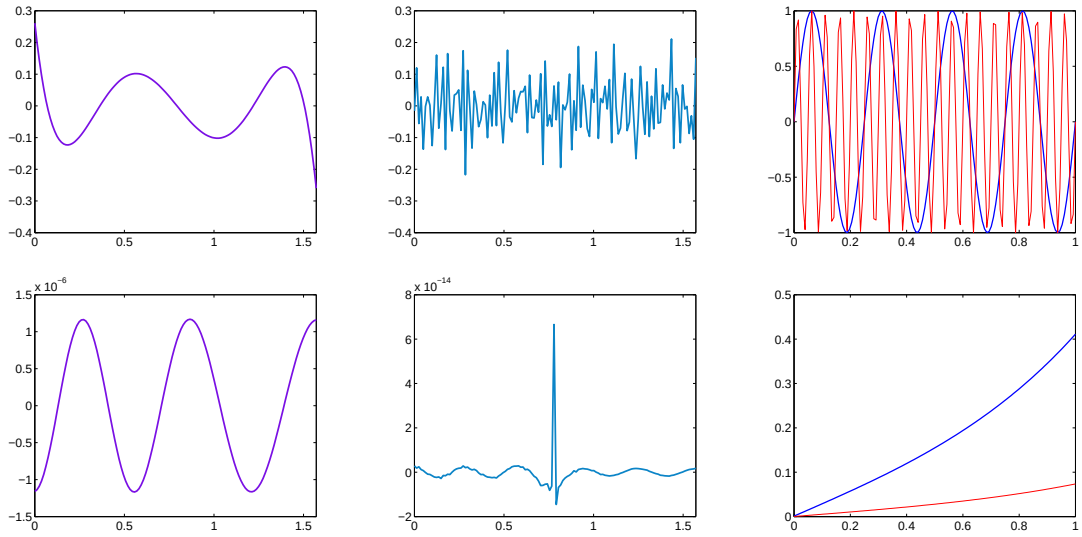

Figure 1.1.2: Smoothing effect of the multiplication by the matrix A: more precisely, in the lower row we show the results obtained multiplying by A the vectors displayed in the upper row. In the upper leftmost frame we consider the left singular vector u_6 of A, in the upper middle frame we consider the left singular vector u_{12} of A, in the upper rightmost frame we consider the functions $\sin(8\pi t)$ (ticker line) and $\sin(40\pi t)$ (finer line), $t \in [0, 1]$. These quantities are associated to the problem `baart` of size 128. Looking at the rightmost frames we can claim that a sort of "discrete Riemann-Lebesgue lemma" holds.

(1.1.7), (1.1.8) and (1.1.9),

$$\frac{\|x^{ex} - x\|_2}{\|x^{ex}\|_2} \leq \mathrm{cond}_2(A) \frac{\|e\|_2}{\|b^{ex}\|_2}. \tag{1.1.14}$$

The above estimate describes the amplifications of the errors that affect the solution of (1.1.7) when the matrix A is bad-conditioned. Of course this is just an upper bound, but experience says that, in the case of ill-posed problems, it is quite tight. At this point, we fully understand that the instability of the continuous problem is reflected into the ill-conditioning of the matrix associated to the discretized linear system.

The cost of MATLAB's algorithm to compute the SVD is $O(k^2 n)$ flops for a dense $k \times n$ matrix, with $k \geq n$. This suggests that, despite the fact that the SVD is still a powerful theoretical tool to estimate the behavior of the system and to express the regularized solutions in an efficient way (as it is evident from the derivations of Section 1.3 and Chapter 4), we cannot assume the SVD to be available when dealing with large-scale systems as, for instance, the ones arising in image deblurring problems (cf. Section 1.2.2). Therefore, from a computational point of view, when dealing with large-scale problems one needs to develop regularization methods that do not rely on the knowledge of the SVD.

Generalized Singular Value Decomposition (GSVD)

The GSVD of a matrix pair (A, L) was originally introduced in [122] and generalizes the SVD of A in two ways: first of all, when $L = I_n$, the GSVD of (A, I_n) is the SVD of A; furthermore, the square roots of the generalized eigenvalues of $(A^T A, L^T L)$ are essentially the generalized singular values of (A, L). Assuming that $A \in \mathbb{R}^{k \times n}$ and $L \in \mathbb{R}^{q \times n}$ with $k \geq n \geq q$, that L is of full rank, and that the intersection of the null-spaces of A end L is trivial, i.e., $\mathcal{N}(A) \cap \mathcal{N}(L) = \{0\}$, the GSVD of (A, L) is given by the following decompositions:

$$A = \widetilde{U} \begin{pmatrix} \widetilde{\Sigma} & 0 \\ 0 & I_{n-q} \end{pmatrix} X^{-1} \quad \text{and} \quad L = \widetilde{V} \begin{pmatrix} \widetilde{M} & 0 \end{pmatrix} X^{-1}, \tag{1.1.15}$$

where the columns of $\widetilde{U} = [\widetilde{u}_1, \ldots, \widetilde{u}_n] \in \mathbb{R}^{k\times n}$ are orthonormal, $\widetilde{V} = [\widetilde{v}_1, \ldots, \widetilde{v}_q] \in \mathbb{R}^{q\times q}$ is orthogonal, $X = [x_1, \ldots, x_n] \in \mathbb{R}^{n\times n}$ is nonsingular, $\widetilde{\Sigma} = \mathrm{diag}(\widetilde{\sigma}_1, \ldots, \widetilde{\sigma}_q) \in \mathbb{R}^{q\times q}$, and $\widetilde{M} = \mathrm{diag}(\widetilde{\mu}_1, \ldots, \widetilde{\mu}_q) \in \mathbb{R}^{q\times q}$. Both the $\widetilde{\sigma}_i$'s and the $\widetilde{\mu}_i$'s are nonnegative, they are ordered such that

$$0 \leq \widetilde{\sigma}_1 \leq \cdots \leq \widetilde{\sigma}_q \leq 1, \quad 1 \geq \widetilde{\mu}_1 \geq \cdots \geq \widetilde{\mu}_q > 0,$$

and they are normalized such that $\widetilde{\sigma}_i^2 + \widetilde{\mu}_i^2 = 1$, $i = 1, \ldots, q$. The generalized singular values of (A, L) are defined as

$$\gamma_i = \frac{\widetilde{\sigma}_i}{\widetilde{\mu}_i}, \quad i = 1, \ldots, q\,,$$

and therefore appear in nondecreasing order. If $q > n$, then we should first compute the QR factorization of L, i.e., $L = QR$ where $R \in \mathbb{R}^{q\times q}$, then compute the factorizations (1.1.15) for the matrix pair (A, R), and finally take

$$A = \widetilde{U}\widetilde{\Sigma}X^{-1} \quad \text{and} \quad L = (Q\widetilde{V})\widetilde{M}X^{-1}. \tag{1.1.16}$$

The GSVD is a very useful tool when dealing with Tikhonov regularization in general form (cf. Section 1.3); we again refer to Section 1.3 to have a list of the matrices L that are typically employed in this setting: all of them are approximations of some derivative operator. Here we note that, if $q < n$, the null space of L is nontrivial and is spanned by the last $n - q$ columns of X.

Some theoretical results that link the SVDs of A and L, and the GSVD of (A, L), allow to state that usually, when dealing with ill-posed discrete problems, if the matrix L is well conditioned, then the matrix X is approximately as well conditioned as L; again, the generalized singular values gradually converge to zero, in a reverse order with respect to the usual singular values (cf. Figure 1.1.3, frame **(b)**); finally, the x_i's exhibit more oscillations as i decreases (cf. Figure 1.1.4, second row).

To keep the dissertation simpler, the next theoretical considerations are made focussing exclusively on the SVD; being aware of the analogous behavior of the singular values (vectors) and the generalized singular values (vectors), we can tacitly assume that, when a matrix pair (A, L) must be considered, each statement regarding the SVD is also true for the GSVD, provided that the u_i's have been replaced by the $\widetilde{u}_i$'s, the v_i's by the x_i's, and that the σ_i's by the γ_i's.

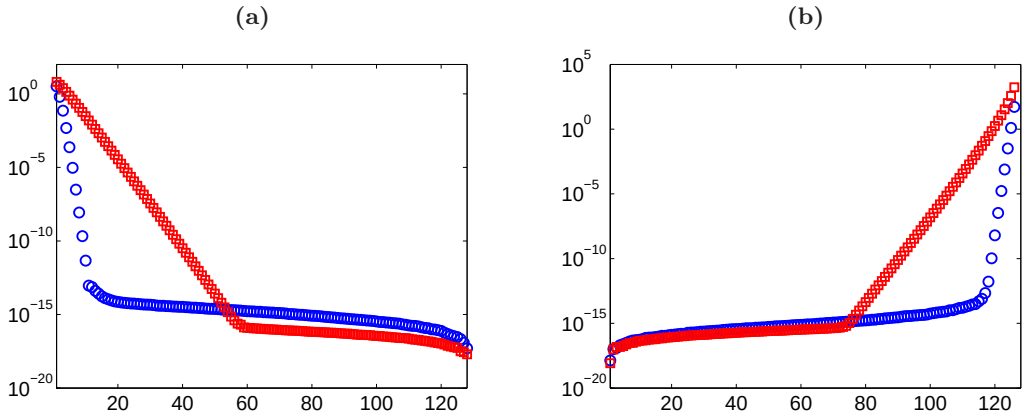

Figure 1.1.3: **(a)** decay of the singular values for the test problems `baart` (circles) and `gravity` (squares) , both of size 128; we can clearly see that, starting from index 20 for `baart` and index 60 for `gravity`, the singular values level off due to the influence of finite-precision arithmetic. **(b)** increasing behavior of the generalized singular values for the matrix pairs (A, D_2), where A is the matrix associated to `baart` (circles) and `gravity` (squares), and D_2 is defined in (1.3.18); we can clearly see that up to index 105 for `baart` and index 70 for `gravity`, the generalized singular values are leveled off due to the influence of finite-precision arithmetic.

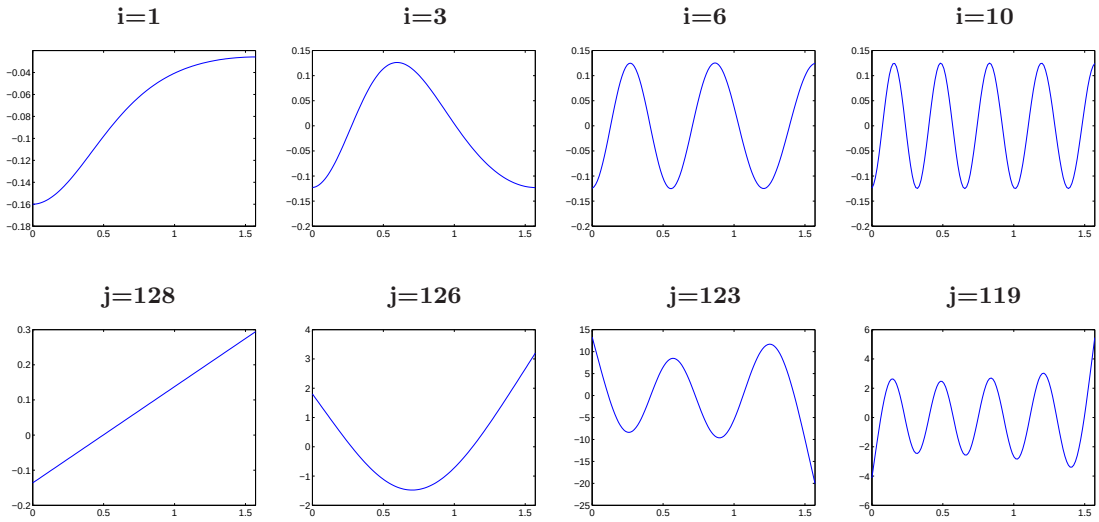

Figure 1.1.4: First row: increasing oscillations of the right singular vectors of the matrix $A \in \mathbb{R}^{128\times 128}$ associated to the test problem `baart` for increasing values of the index i. Second row: increasing oscillations in the right generalized singular vectors of the matrix pair (A, D_2) associated to the test problem `baart`, for decreasing values of the index j.

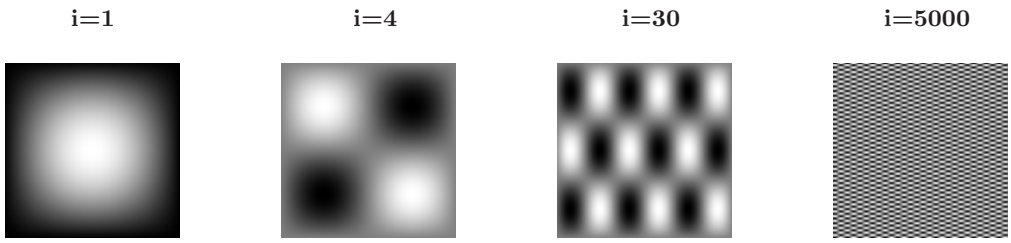

Figure 1.1.5: Increasing oscillations of the right singular vectors of the matrix modeling a symmetric separable Gaussian blur with $\alpha_1 = \alpha_2 = 2$ (cf. Section 1.2.2). The singular vectors are displayed as two-dimensional 200×200 arrays and the effect of this kind of blur on an ideally exact image can be appreciated in Figure 1.1.7, leftmost and median lower frames.

1.1.4 The Discrete Picard Condition

As already addressed in the previous section (cf. the estimate (1.1.14)), when inverting the coefficient matrix A to solve the linear system (1.1.7), we should expect the difficulties encountered in the continuous case. In particular, looking directly at the expression (1.1.13) and keeping in mind the behavior of the singular values and vectors, we can understand that the decay of the coefficients $|u_i^T b|$'s (sometimes referred to as Fourier coefficients) relative to the decay of the σ_i's plays a central role in the approximation of the solution. Indeed, under the additional assumptions considered in [53], the decay of the quantities $|u_j^T b|$ even determines the quality of the approximate solution that can be attained employing some direct regularization method (Section 1.3.1).

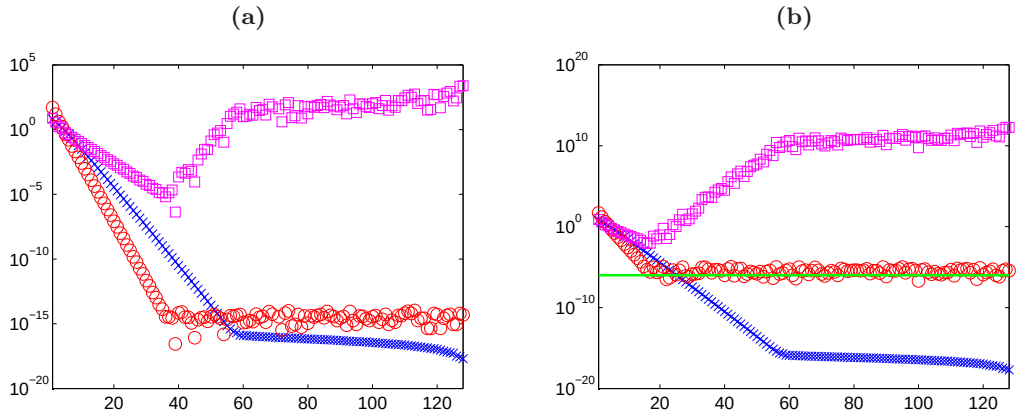

Figure 1.1.6: Picard plots for the test problem `gravity` of size 128. In both frames, on the horizontal axis we report the values of the index i; the scalars σ_i's, $|u_i^T b|$'s and $|u_i^T b|/\sigma_i$'s are denoted by a cross, a circle and a square, respectively. **(a)** we assume to deal with the exact right-hand side b^{ex}. We can clearly see that, except for a few outliers, the scalars $|u_i^T b|$'s decay faster than the σ_i's; however, we should discard the components of indices greater than 38 since, for these values, the Fourier coefficients are close to the machine precision. **(b)** we assume that the right-hand side b is affected by a noise whose level is $\widetilde{\varepsilon} = 10^{-6}$. In this case, starting from index 18, the coefficients $|u_i^T b|$ level off around the threshold 10^{-6} (highlighted by a horizontal line).

The Picard plot is a tool used to determine whether considering an approximate solution of the form (1.1.13) is meaningful or not: it consists in a plot of the coefficients $|u_j^T b|$ together with the singular values σ_i versus the values of i (two examples of Picard plots are displayed in Figure 1.1.6). Looking at the Picard plot we can immediately see that, if the Fourier coefficients decay similarly to the singular values, the coefficients of the linear combination (1.1.13) are "well-behaved" and the norm of computed solution is limited. With respect to the continuous case we should remark that, from a strictly mathematical point of view, the norm of the solution of the discrete problem is always bounded; however, in practice, it could become so huge to be useless. Moreover, in the discrete case, particular attention should be paid to the rounding errors introduced when computing the SVD of the matrix A: for instance (cf. [53]), those singular values that lie below the threshold $\varepsilon_m \sigma_1$ (where ε_m is the machine precision) should be regarded as numerically zero and the corresponding singular vectors should be discarded (cf. again Figure 1.1.6, frame **(a)**). These considerations can be summarized in the following

Discrete Picard Condition (DPC)[53]. The vector b^{ex} in (1.1.8) satisfies the DPC if, on the average, the Fourier coefficients $|u_i^T b^{ex}|$ decay to zero faster than the corresponding numerically nonzero singular values.

As stated above, the decay of the Fourier coefficients should be regarded as a local phenomenon: in practice, to check this, we can both visually inspect the Picard plot and employ the so-called

moving geometrical mean, i.e., evaluate the decrease of the quantities

$$\sigma_i^{-1} \left(\prod_{j=i-l}^{i+l} |u_j^T b| \right)^{1/(2l+1)}, \quad i = l+1, \ldots, n-l,$$

where l is typically a small integer; we remark that some possible numerically zero coefficients $|u_j^T b|$ have to be neglected in the computation of the moving geometrical mean.

More in general, when we consider data that are affected by noise, the Fourier coefficients level off at a plateau determined by the variance of the noise: the higher the noise, the earlier the level off. In this situation, the approximate solution computed by formula (1.1.13) is completely dominated by the contributions corresponding to the smallest singular values, i.e., it is highly oscillatory (cf. Figure 1.1.6, frame **(b)**, and Figure 1.1.7). Indeed, separating the contributions of the ideally exact right-hand side of (1.1.8) and of the noise in (1.1.13), we obtain the expression

$$x = \sum_{i=1}^{\mathrm{rk}(A)} \frac{u_i^T b^{ex}}{\sigma_i} v_i + \sum_{i=1}^{\mathrm{rk}(A)} \frac{u_i^T e}{\sigma_i} v_i.$$

Assuming that the unperturbed problem satisfies the DPC, the contribution of the first sum is admissible; the degenerating term is the second sum: of course, the vector e does not fulfill the DPC and, excluding the first elements of the sum, the remaining ones rapidly increase and their contribution prevails in the recovered solution x, making it meaningless (cf. Figure 1.1.7). The second sum is often referred to as inverted noise. In order to compute an admissible solution, the predominant oscillating components should be filtered out: this is precisely our goal when we apply a regularization method to the discrete problem. In particular, because of the location of the plateau of the $|u_j^T b|$'s, one can only hope to keep the coefficients lying above the noise level.

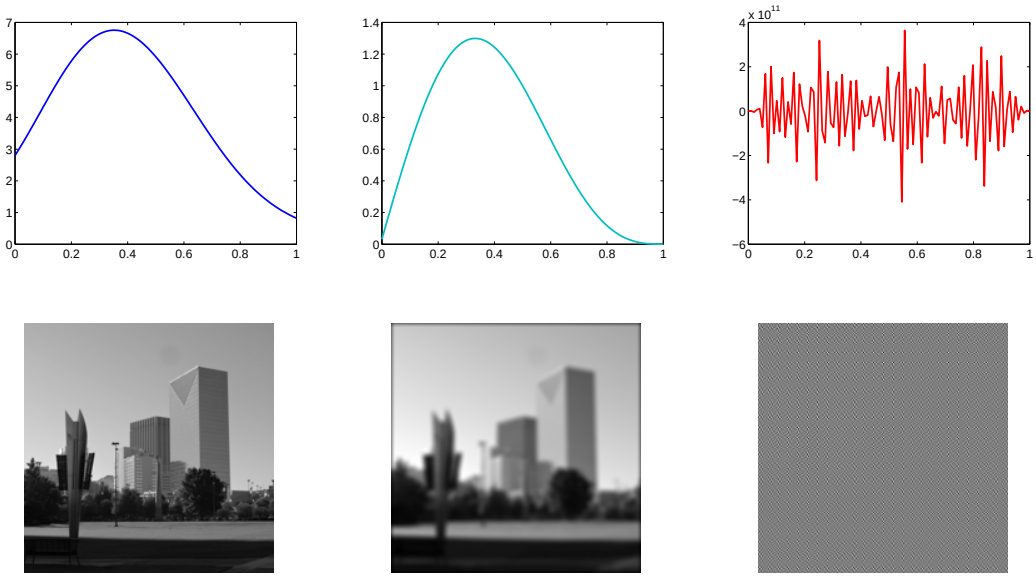

Figure 1.1.7: Examples of the difficulties encountered in naively solving ill-posed problems. In both rows, from left to right, we show the ideally exact solution, the available data and the solution obtained naively inverting the coefficient matrix. In the first row we consider the `gravity` problem of size 128, the data are affected by a noise level $\widetilde{\varepsilon} = 10^{-6}$. In the second row we consider the deblurring and denoising of a 200×200-pixel image: the blurring matrix is computed using the function `blur`, whose parameters are `q=4` and `s=2`; the noise level is $\widetilde{\varepsilon} = 10^{-3}$.

1.2 Test Problems

In this section we describe some test problems that are used in the next chapters to assess the performance of the newly proposed algorithms. In Section 1.2.1, starting from the information given in [58], we briefly address some problems taken exclusively from the package Regularization Tools, originally proposed in [54] and further extended in [56, 57]. All the considered linear systems have been obtained discretizing Fredholm integral equations of the first kind, with $\Omega \subset \mathbb{R}$: most of them model some physically meaningful inverse problems, although some of them are artificial ill-posed problems that don't have any physical counterpart. In Section 1.2.2 we focus on image deblurring and denoising problems: we explain the model used to deal with imaging problems, that is still based on Fredholm integral equations of the first kind, with $\Omega \subset \mathbb{R}^2$. Although a test problem that models a particular kind of blur is provided in [56], we perform many tests on images employing the routines taken from the package Restore Tools [90], which has been specifically written to implement in an efficient and user-friendly way some classical image-processing operations.

For all the test problems we assume the system matrix A to be available, as well as the exact solution x^{ex}. The exact right-hand side is often obtained by taking $b^{ex} = Ax^{ex}$. We obtain a perturbed right-hand side corresponding to a given noise level (denoted by `nl`) using the MATLAB's instructions

```
e = randn(m,1); e = (nl*norm(b_ex)/norm(e))*e; b = b_ex + e;
```

Typically, the range of the noise levels considered is between 10^{-3} and 10^{-1}.

1.2.1 Problems from *Regularization Tools*.

- `baart`. Discretization of an artificial Fredholm integral equation of the first kind by means of the Galerkin method, where

$$\hat{k}(s,t) = \exp(s\,\cos t), \quad g^{ex}(s) = 2\frac{\sin s}{s}, \quad s \in \left[0, \frac{\pi}{2}\right].$$

  The integration interval is $[0, \pi]$ and the solution is the function $f(t) = \sin t$.

- `deriv2`. Discretization of a Fredholm integral equation of the first kind by means of the Galerkin method. The kernel is the Green's function for the second derivative, i.e.,

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

  The solution and the available vector are respectively given by

$$f(t) = t, \quad g^{ex}(s) = \frac{s^3 - s}{6}.$$

  For this particular problem we know the analytical expression of the singular values and the singular functions:

$$\sigma_i = (i\pi)^{-2}, \quad u_i(t) = \pm\sqrt{2}\sin(i\pi t), \quad v_i(s) = \mp\sqrt{2}\sin(i\pi s).$$

  According to the classification given in Section 1.1.1, the problem is mildly ill-conditioned. Both s and t belong to $[0, 1]$. The resulting matrix is symmetric.

- `foxgood`. Discretization of a Fredholm integral equation of the first kind by means of the midpoint quadrature rule, where

$$\hat{k}(s,t) = \left(s^2 + t^2\right)^{\frac{1}{2}}, \quad g^{ex}(s) = \frac{1}{3}\left((1+s^2)^{\frac{3}{2}} - s^3\right), \quad f(t) = t.$$

  Both s and t belong to $[0,1]$. The resulting matrix is symmetric. This is an artificial severely ill-posed problem that does not satisfy the Discrete Picard Condition .

- `gravity`. Discretization of a Fredholm integral equation of the first kind modeling a one-dimensional gravity surveying problem.

$$f(t) = \sin(\pi t) + \frac{1}{2}\sin(2\pi t)$$

  is a mass distribution located at depth $d = 0.25$. The kernel is given by

$$\hat{k}(s,t) = d\left(d^2 + (s-t)^2\right)^{-\frac{3}{2}}.$$

  $g^{ex}(s)$ represents the vertical component of the gravity field measured at the surface and it is directly computed as $b = Ax$. Both s and t belong to $[0,1]$. The problem is discretized using the midpoint quadrature rule, leading to a symmetric Toeplitz matrix.

- `i_laplace`. Discretization of the inverse Laplace transformation by means of Gauss-Laguerre quadrature. We consider the following functions:

$$\hat{k}(s,t) = \exp(-st), \quad f(t) = \exp(-t/2), \quad g^{ex}(s) = \frac{1}{s + 1/2};$$

  both t and s belong to $[0,\infty)$.

- `phillips`. Discretization of Fredholm integral equation of the first kind , by means of the Galerkin method, where

$$\hat{k}(s,t) = \phi(s-t), \quad f(t) = \phi(t), \text{ where } \phi(x) = \begin{cases} 1 + \cos(\pi x/3), & |x| < 3 \\ 0, & x > 3 \end{cases}.$$

  As a consequence,

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

  Both s and t belong to $[-6,6]$. The resulting matrix is symmetric.

- `shaw`. Discretization of a Fredholm integral equation of the first kind by means of a quadrature method.

$$\hat{k}(s,t) = (\cos(s) + \cos(t))^2 \left(\frac{\sin(u)}{u}\right)^2, \quad u = \pi\left(\sin(s) + \sin(t)\right)$$

  is the point spread function (cf. Section 1.2.2) for an infinitely long line. We know this problem to be severely ill-posed, since $\sigma_i = O(e^{-2i})$. The solution f is given by

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

  Both s and t belong to $[-\pi/2, \pi/2]$. The resulting matrix is symmetric; the right-hand side of (1.1.1) is directly computed as $b = Ax$.

1.2.2 Image restoration problems

Image deblurring and denoising is the process of reconstructing an approximation of an image from blurred and noisy measurements. Grayscale images (the only kind of images that we consider) are mathematically represented by two-dimensional arrays with integer entries, called pixels. The value of each pixel characterizes the grayscale intensity of a tiny portion of the overall image: the value 0 corresponds to a black pixel, the value 255 corresponds to a white pixel. Typically a small image has around 256^2 pixels, while a large high-resolution image can count up to 10^6 pixels. We inevitably have to deal with blurred images every time we take a picture using a camera; more significantly, the recorded astronomical, microscopical or medical images are often affected by serious blurring that prevents a good understanding of the object being portrayed. The blur can be originated by many causes: for instance, the camera lenses could be out of focus, some turbulence could occur, some motion could happen. We can model the image deblurring and denoising problem using a Fredholm integral equation of the first kind (1.1.1), whose kernel function $\hat{k}$ represents the blurring operation; in this setting f is the ideally exact scene one wants to capture, g is the recorded blurred and noisy image (cf. Figure 1.1.7) and both the variables s and t are two-dimensional vectors. When the kernel $\hat{k}$ is of the form $\hat{k}(s,t) = \hat{k}(s-t)$, i.e., when we are dealing with deconvolution problems (cf. Section 1.1.1), the blur is called spatially invariant. In this case, the kernel $\hat{k}$ can be defined in two ways:

- **analytically**. Sometimes, the knowledge of the physical process that causes the blur provides an analytical expression for $\hat{k}$. For instance, the motion blur and the out-of-focus blur can be described in this way. We just characterize the Gaussian blur, which models the distortions caused by atmospheric turbulence: in its most general form, the kernel $\hat{k}$ is given by a Gaussian function

$$\hat{k}(x_1, x_2) = \gamma \exp\left(-\frac{1}{2} [x_1\ x_2] \begin{bmatrix} \alpha_1^2 & \rho^2 \\ \rho^2 & \alpha_2^2 \end{bmatrix}^{-1} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \right), \tag{1.2.1}$$

  where $x = (x_1, x_2) \in \mathbb{R}^2$ and the normalization factor is

$$\gamma = \frac{1}{\left(2\pi\sqrt{\alpha_1^2\alpha_2^2 - \rho^4}\right)}$$

  The parameters (α_1, α_2) and ρ determine the width and the orientation of the Gaussian function, respectively. Due to the exponential decay of the values of $\hat{k}$ away from zero, during the discretization process it is natural to truncate $\hat{k}$.

- **experimentally**. Once a point source (i.e., an ideally single white pixel) has been localized, an image of it is generated: the resulting picture, where the intensity of the single white pixel has been distributed over the neighboring pixels, allows us to define the action of the blur (directly in a discrete setting). For instance, in astronomical imaging the point source could be represented by an isolated white star.

The above description explains why, in this setting, the kernel $\hat{k}$ is often called Point Spread Function (PSF); for some examples of different kind of blurs, we refer to Figure 1.2.1. When deriving a linear system of the kind (1.1.7), suitable boundary conditions describing the behavior of the scene outside the recorded image should be imposed in order to avoid ringing artifacts near the boundaries of the restored image. Since the blurring process is described by an integral equation, it is natural to expect that the values of boundary pixels are affected by the values assigned to the pixels immediately outside the recorded array; these conditions enter the definition of the system matrix. Among the most classical boundary conditions we list the zero, periodic, reflecting, replicating ones (see Figure 1.2.2). More recently-introduced boundary

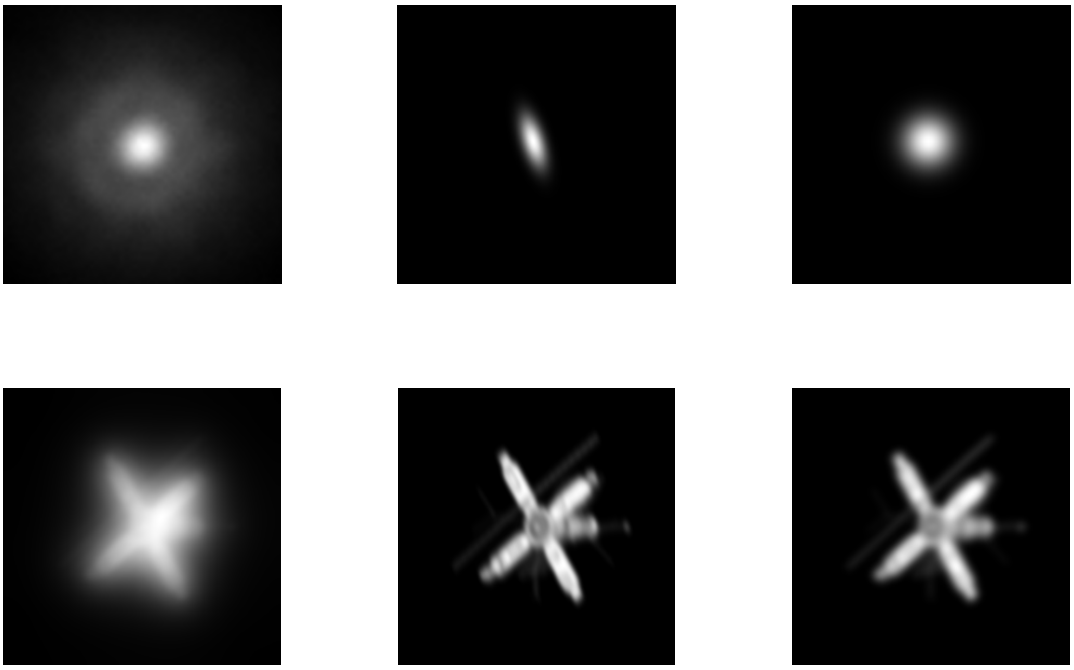

Figure 1.2.1: Examples of different kind of blurs: in the upper row we show a blowup of different point spread functions, and in the lower row we show the effect of the corresponding blurs applied to a well-known test image made available in [90]. Leftmost frames: example of a PSF experimentally derived; it reproduces the atmospheric blur that one encounters when taking images from a ground-based telescope and belongs to a set of test data that was developed at the US Air Force Phillips Laboratory, New Mexico (cf. [90]). Median frames: Gaussian PSF with parameters $\alpha_1 = 4$, $\alpha_2 = 2$ and $\rho = 2$. Rightmost frames: Gaussian PSF with parameters $\alpha_1 = \alpha_2 = 4$ and $\rho = 0$.

conditions include the anti-reflective [27] and the synthetic ones [33]. In the image restoration framework, noise can typically arise from some interferences in the mechanical recording device (modeled by the so-called Poisson noise) and readout noise arising in the analog-to-digital conversion (modeled by Gaussian white noise); in the following only Gaussian white noise will be taken into account. In order to write down a system of the form (1.1.7), the two-dimensional arrays so far considered should be converted into one-dimensional vectors: one way to achieve this is to stack the columns of the two-dimensional matrix $X \in \mathbb{R}^{N\times N}$ in order to form a one-dimensional vector $x \in \mathbb{R}^n$, $n = N^2$. The above operation is denoted by "vec": we write $x = \text{vec}(X)$ and $X = \text{vec}^{-1}(x)$. Finally, in order to arithmetically manipulate the image arrays, we should transform their original integer entries into double precision floating point.

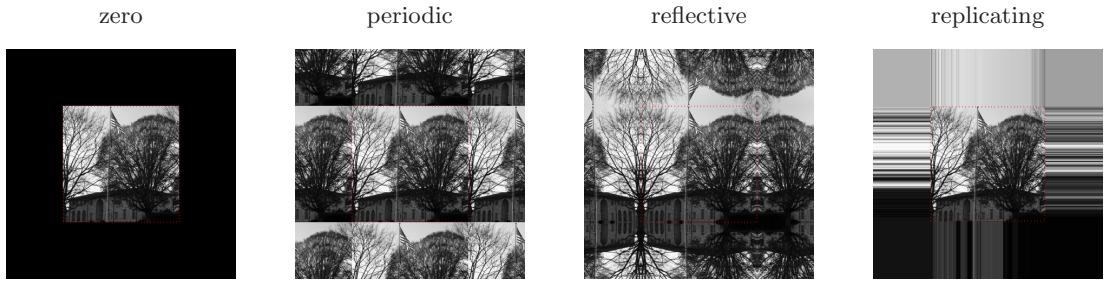

Figure 1.2.2: Examples of some different boundary conditions. The most natural boundary conditions for this kind of image seem the reflecting ones.

Because of the conversion from 2D to 1D arrays, the resulting linear systems are typically of huge dimensions, even for small images. The coefficient matrix should be scaled so that $A^T\mathbf{1} = \mathbf{1}$, where $\mathbf{1}$ denotes a vector whose entries are all equal to 1: this is equivalent to the physical assumption that the operator represented by A conserves energy (cf. [92]). Often the matrices associated to some kind of blur and some boundary conditions can be represented using some compact data structure: we suggest two meaningful examples. The first one is given by a Gaussian blur (1.2.1) with parameters $\rho = 0$ and $\alpha_1 = \alpha_2$ (cf. Figure 1.2.1, upper rightmost frame) that gives rise to a symmetric and separable blur, i.e., the horizontal and the vertical components of the blur can be separated and the blurring matrix can be decomposed into the Kronecker product of two matrices corresponding to the two one-dimensional blurring components. The second example is given by a blur with periodic boundary conditions that gives rise to a block circulant matrix A_{BCCB} with circulant blocks: efficient computations with this kind of matrix can be achieved exploiting the connection between its spectral decomposition and the discrete Fourier transform, implemented by a Fast Fourier Transform (FFT) algorithm. Moreover, if we take zero boundary conditions, the resulting matrix is block Toeplitz A_{BTTB} with Toeplitz blocks. The function `blur(N,q,s)` from [56] computes the matrix A_{BTTB} of size $n =$`N`^{2}, in the case of a Gaussian blur with parameters $\alpha_1 = \alpha_2 =$`s`, $\rho = 0$, and truncated after `q` components (therefore `q` is the half-bandwidth of each Toeplitz block).

Images can be easily and efficiently manipulated using some MATLAB's built-in functions and the routines collected in its Image Processing Toolbox (IPT). As said at the beginning of this section, in many circumstances we use the routines collected in [90], which allow us to efficiently perform the basic operations on the available linear system (1.1.7) by exploiting the particular structure of the involved matrices.

Direct regularization methods (Section 1.3.1) have been classically employed to solve linear systems whose matrices have a well-defined structure. However, more recently, iterative or hybrid methods have been successfully considered. With respect to the direct ones, the latter can successfully handle a wider class of imaging problems. For instance, they are still efficient when the system matrix has been obtained imposing some peculiar boundary conditions (as done in [33]) or considering an approximation of a spatially variant blur (as done in [88]).

Moreover, some additional constraints, such as nonnegativity (recall that the pixels' values are nonnegative), can be naturally incorporated into an iterative scheme to enhance the quality of the reconstruction (cf. [14, 92]). Finally, to improve the performance of some iterative methods, particular types of preconditioners can be considered (cf. [23, 50, 91]). We refer to [8] for a recent summary of many iterative methods applied to image restoration problems.

1.3 Survey of some basic regularization techniques

As extensively explained in Sections 1.1.3 and 1.1.4, when dealing with linear systems coming from the discretization of linear ill-posed problems, the solutions obtained by straightforwardly solving the available systems (1.1.7) are meaningless, since they are dominated by high-oscillating components originated by the noise that affects the available data. As already stated in the introduction of this chapter, in order to recover a meaningful solution, one has to impose some sort of smoothness on the computed solution; this is achieved by applying a regularization method, i.e. by substituting the available system with a nearby problem, whose solution is less affected by the unwanted noise components.

Each regularization method is characterized by a regularization parameter α, that specifies the amount of regularization to be imposed; depending on the context, the regularization parameter can be a positive real number $\lambda \in \mathbb{R}^+$, a natural number $m \in \mathbb{N}$, a couple $(m, \lambda_m) \in \mathbb{N} \times \mathbb{R}^+$, a vector $(\lambda_1, \dots, \lambda_k) \in (\mathbb{R}^+)^k$ or $(m, \lambda_1, \dots, \lambda_k) \in \mathbb{N} \times (\mathbb{R}^+)^k$. Keeping our dissertation general, we can say that the regularized solution of a linear discrete ill-posed problem (1.1.7) is a vector $x_\alpha \in \mathbb{R}^n$ obtained computing

$$x_\alpha = A_\alpha^\sharp b, \tag{1.3.1}$$

where $A_\alpha^\sharp$ denotes the regularized inverse of the coefficients matrix A; the subscript α highlights the dependence of the regularization operator (and therefore of the regularized solution) on the regularization parameter. Different choices of $A_\alpha^\sharp$ define different regularization methods: in the following, we give an explicit expression of $A_\alpha^\sharp$ for each considered regularization method.

1.3.1 Direct Regularization Methods

In general, direct regularization methods are based on the knowledge of some canonical factorization of the matrix A: in this way, the regularity requirements are explicitly incorporated into the formulation of the problem. Of course, recalling the analysis performed in the previous sections, we immediately understand that the basis given by the right (generalized) singular vectors is the most natural one to express the regularized solution. Therefore, in this section, we assume the (G)SVD to be available. The computational effort required by a direct regularization method can be quite accurately estimated a priori; because of these estimates, we can state that regularizing an unstructured large-scale system by applying a direct method is not computationally affordable. Every direct regularization method (and also many iterative ones) can be expressed in the form of filtered (G)SVD expansion, i.e.,

$$x_\alpha = V \Phi_\alpha \Sigma^\dagger U^T b \quad \left(x_\alpha = X \begin{pmatrix} \Phi_\alpha \widetilde{\Sigma}^\dagger & 0 \\ 0 & I_{n-q} \end{pmatrix} \widetilde{U}^T b \right). \tag{1.3.2}$$

For all the methods described in this section, the matrix Φ_α is diagonal, i.e., $\Phi_\alpha = \operatorname{diag}(f_1^\alpha, \dots, f_n^\alpha)$ and the regularized solution can be explicitly expressed as

$$x_\alpha = \sum_{i=1}^{n} f_i^\alpha \frac{u_i^T b}{\sigma_i} v_i \quad \left(x_\alpha = \sum_{i=1}^{q} f_i^\alpha \frac{\widetilde{u}_i^T b}{\widetilde{\sigma}_i} x_i + \sum_{i=q+1}^{n} \widetilde{u}_i^T b x_i \right). \tag{1.3.3}$$

The scalars f_i^α's are called filter factors. For this reason, regularization methods are often referred to as spectral filtering methods. Considering the expressions (1.3.2) and (1.3.3), it is

evident that the regularizing effect of a spectral filtering method is enclosed in the matrix Φ_α. Different expressions of the filter factors define different regularization methods and the amount of regularization is specified by the parameter α: however, all the regularization methods have in common that for small (generalized) singular values the filter factors are almost zero, and for large (generalized) singular values the filter factors are close to 1 (cf. Figure 1.3.1).

Tikhonov Regularization

Tikhonov regularization method [120], in its most general form, prescribes to determine an approximation x_λ of the exact solution by computing

$$x_\lambda = \arg \min_{x \in \mathbb{R}^n} \left\{ \|b - Ax\|^2 + \lambda \|L(x - x^*)\|^2 \right\}, \tag{1.3.4}$$

where $L \in \mathbb{R}^{q \times n}$ is called regularization matrix, $\lambda \in \mathbb{R}^+$ is called regularization parameter and $x^* \in \mathbb{R}^n$ is an initial guess for the solution (if it is not available, one simply takes $x^* = 0$). The basic idea behind Tikhonov regularization method is to search for some x_λ providing at the same time a small residual $\|b - Ax_\lambda\|$ and a moderate value for the penalty function $\|L(x_\lambda - x^*)\|$. The first term $\|b - Ax_\lambda\|$ is often called fit-to-data term and its size determines how the regularized solution x_λ fits the initial problem: intuitively, this term should not be too small, since the solution would be close to the unregularized one of (1.1.7); on the other side, this term should not be too big, otherwise the solution would badly fit the initial model. The penalization term $\|L(x - x^*)\|$, often referred to as regularization term, measures the regularity of the solution. Of course the definition of this term depends on the choice of L but, in general, we can say that an approximate solution dominated by noise components has a large regularization term; by including this term into the Tikhonov functional to be minimized, we avoid it being too big and therefore we force regularity. The weight to be assigned to the regularization term is specified by λ: the larger λ, the smaller the regularization term in the minimization process; the smaller λ, the larger the regularization term in the minimization process. Therefore, under-estimating λ can lead to highly oscillatory approximations of x^{ex} (under-smoothing), while over-estimating λ can cause the reconstructed solution to be too smooth (over-smoothing).

Before giving some details about the choice of the regularization term in Tikhonov regularization, we derive some equivalent formulation of the problem (1.3.4). With a simple change of variable, i.e., defining $\delta = x - x^*$ and performing some elementary algebraic manipulations on the problem (1.3.4), we obtain the following penalized least square problem

$$\delta_\lambda = \arg \min_{\delta \in \mathbb{R}^n} \left\{ \|r^* - A\delta\|^2 + \lambda \|L\delta\|^2 \right\}, \tag{1.3.5}$$

where $r^* = b - Ax^*$. The solution x_λ of (1.3.4) is then recovered by taking $x_\lambda = x^* + \delta_\lambda$. Employing some properties of the 2-norm we can rewrite (1.3.5) as the following regularized least square problem

$$\delta_\lambda = \arg \min_{\delta \in \mathbb{R}^n} \left\| \begin{pmatrix} A \\ \sqrt{\lambda} L \end{pmatrix} \delta - \begin{pmatrix} r^* \\ 0 \end{pmatrix} \right\|, \quad x_\lambda = x^* + \delta_\lambda. \tag{1.3.6}$$

Moreover, we obtain a third formulation equivalent to (1.3.5) by writing the normal equations associated to (1.3.6)

$$(A^T A + \lambda L^T L)\delta_\lambda = A^T r^*, \quad x_\lambda = x^* + \delta_\lambda. \tag{1.3.7}$$

A different way to derive the equivalent formulations (1.3.6) and (1.3.7) is to consider that the Tikhonov functional in (1.3.5) is differentiable and convex: computing its differential and putting it equal to zero leads to the solution of the normal equations (1.3.7); (1.3.6) is the least squares problem associated to the above mentioned normal equations. Directly from problem (1.3.7) we can derive the expression

$$A_\lambda^\sharp = (A^T A + \lambda L^T L)^{-1} A^T \tag{1.3.8}$$

for the regularized inverse in the case of Tikhonov regularization method.

Among the equivalent formulations (1.3.5)-(1.3.7), the regularized least squares one (1.3.6) is to be preferred, for reasons of computational convenience and numerical stability. An efficient algorithm based on some preliminary transformations of (1.3.4) and on some orthogonal transformation of the matrix A was already proposed in [31]. We remark that, in the case of Tikhonov regularization, the knowledge of the SVD of A (GSVD of (A, L)) is not demanding, but the computations are much easier if it is available, especially when it comes to the choice of the regularization parameter. When employing Tikhonov method, the basic distinction is between standard form regularization ($L = I$) and general form regularization ($L \neq I$). In the following, to keep the notations simpler, we assume that $x^* = 0$.

Standard form regularization When the regularization matrix in (1.3.4) is the identity matrix, Tikhonov regularization is said to be in standard form. Therefore, the regularized solution x_λ is obtained by solving

$$\min_{x \in \mathbb{R}^n} \left\{ \|b - Ax\|^2 + \lambda \|x\|^2 \right\}. \tag{1.3.9}$$

Following the derivations just explained, solving the above problem is equivalent to solving

$$\min_{x \in \mathbb{R}^n} \left\| \begin{pmatrix} A \\ \sqrt{\lambda} I_n \end{pmatrix} x - \begin{pmatrix} b \\ 0 \end{pmatrix} \right\| \quad \text{or} \quad (A^T A + \lambda I_n) x_\lambda = A^T b. \tag{1.3.10}$$

The basic idea behind the choice of $L = I_n$ is that an approximate solution dominated by noise is characterized by high-frequencies components having large amplitudes, which cause a huge 2-norm; by including the 2-norm of the solution vector into the Tikhonov functional, we force the regularized solution to be less dominated by noise. We remark that the choice $L = I_n$ is the simplest one, both from a theoretical and a computational point of view, and it is always quite valid: however, in some situations, the quality of the computed reconstructions is poor, if compared to the one obtained applying more advanced forms of regularization (cf. the next paragraph).

To better understand the regularizing effect of Tikhonov method and the impact of the choice of λ, we rewrite the regularized solution x_λ as filtered SVD expansion (1.3.3). Substituting the SVD of A (1.1.10) into the normal equations (1.3.10) and exploiting the orthogonality of the involved matrices, we obtain

$$x_\lambda = V(\Sigma^T \Sigma + \lambda I_n)^{-1} \Sigma^T U^T b \quad \text{or} \quad x_\lambda = \sum_{i=1}^{n} \frac{\sigma_i^2}{\sigma_i^2 + \lambda} \frac{u_i^T b}{\sigma_i} v_i. \tag{1.3.11}$$

Directly from the above definition, we derive the following expression for the standard form Tikhonov filter factors

$$f_i^\lambda = \frac{\sigma_i^2}{\sigma_i^2 + \lambda}, \quad i = 1, \ldots, n. \tag{1.3.12}$$

Looking at the above expression, the regularizing effect of Tikhonov method is even more evident. Assuming that $\lambda = \bar{\lambda}$ is fixed, the filter factors $f_i^{\bar{\lambda}}$ corresponding to singular values $\sigma_i \gg \bar{\lambda}$ are almost 1, while the filter factors $f_i^{\bar{\lambda}}$ corresponding to singular values $\sigma_i \ll \bar{\lambda}$ are almost $\sigma_i^2/\bar{\lambda}$, i.e. they are almost zero. Of course, choosing $\lambda < \bar{\lambda}$ results in more filter factors close to 1 and choosing $\lambda > \bar{\lambda}$ results in more filter factors close to 0 (cf. Figure 1.3.1). Indeed, considering $\lambda \to 0$ we obtain the unregularized solution, while considering $\lambda \to \infty$ we obtain the zero vector. Recalling the increasingly oscillating behavior of the singular vectors as the corresponding singular values decrease (cf. Section 1.1.3), we have a further insight into the regularizing properties of Tikhonov method: the oscillating components of the solution are gradually suppressed, while the smoother components are gradually kept (cf. Figures 1.3.2 and 1.3.3). After this analysis it is evident that a proper value for the regularization parameter λ should be such that $\sigma_n < \lambda < \sigma_1$.

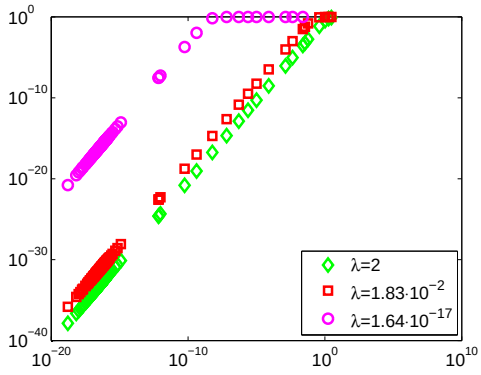

Figure 1.3.1: Logarithmic plot of the filter factors for standard form Tikhonov regularization versus the singular values, for three values of the regularization parameter λ. The considered test problem is `shaw` of size 128.

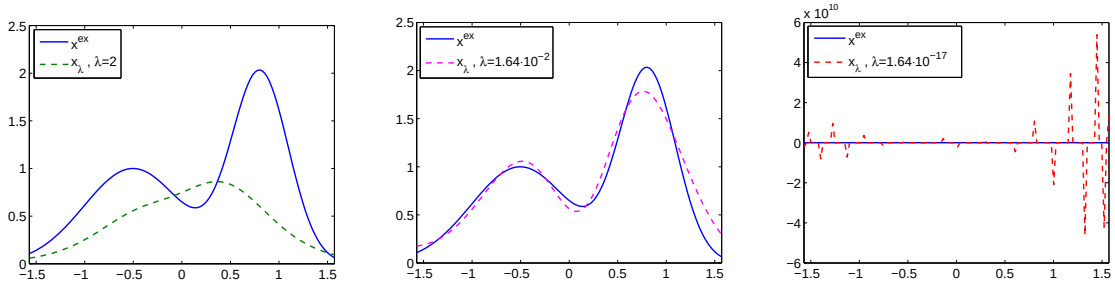

Figure 1.3.2: Solution of the test problem `shaw` of size 128 by means of standard Tikhonov regularization method; the noise level is $\widetilde{\varepsilon} = 10^{-2}$. The employed regularization parameters are ones considered in Figure 1.3.1 (i.e., from the left; $\lambda = 2$, $\lambda = 1.64 \cdot 10^{-2}$, $\lambda = 1.64 \cdot 10^{-17}$): we can clearly see that the solution displayed in the leftmost frame is over-regularized, the solution displayed in the middle frame is accurate, the solution displayed in the rightmost frame is under-regularized.

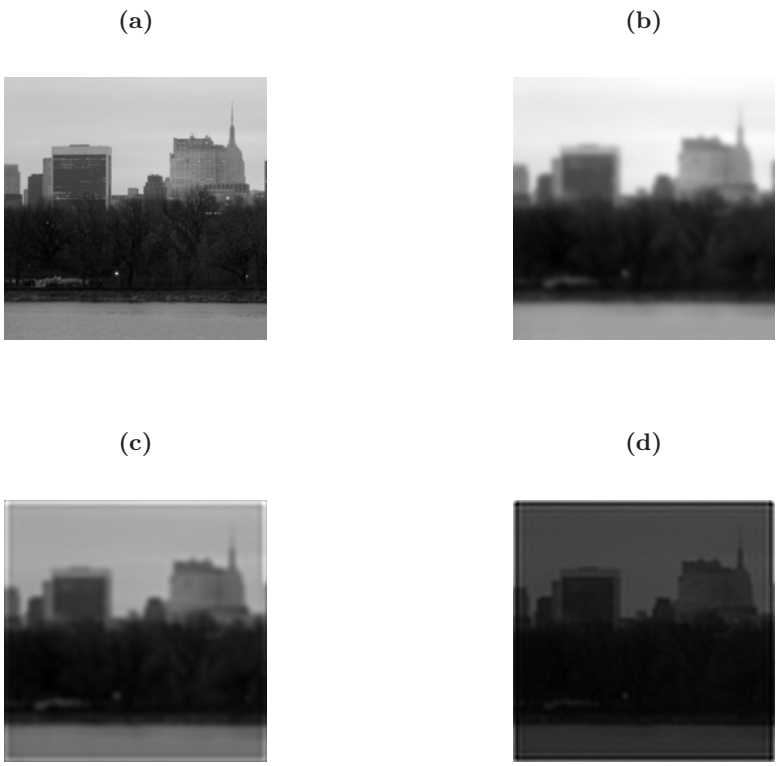

Figure 1.3.3: **(a)** Exact image. **(b)** Image corrupted by a symmetric Gaussian blur with $\sigma = 2$ and by white noise whose level is 10^{-2}. **(c)** Over-smooth reconstruction by standard form Tikhonov regularization. **(d)** Under-smooth reconstruction by standard form Tikhonov regularization.

General form regularization When the regularization matrix in (1.3.4) is not the identity matrix, Tikhonov regularization is said to be in general form. If some particular features or some properties of the desired exact solution are known in advance, the best choice is to build suitable regularization matrices L that are able to enhance them. We remark that a general form problem can always (at least theoretically) be transformed into standard form, by applying some transformations involving the so-called A-weighted generalized pseudoinverse $L_A^\dagger$ of the operator L,

$$L_A^\dagger = \left(I - (A(I - L^\dagger L))^\dagger A\right) L^\dagger, \tag{1.3.13}$$

cf. [32, 61]. The main assumption when performing this kind of regularization is that $\mathcal{N}(A) \cap \mathcal{N}(L) = \{0\}$ or, equivalently, $\text{rk}([A^T\ L^T]^T) = n$: in this way the regularized solution computed by Tikhonov method is unique. Even if the regularized solution depends on the choice of the matrix L, we still indicate it by x_λ, not to overload the notations. In the following we provide a list of the most used regularization matrices: in general they are all defined starting from a continuous derivative operator and they have a nontrivial subspace containing smooth (low-frequency) functions. In this way the regularization term $\|L \cdot\|$ is actually a seminorm and the matrix L acts as a high-pass filter, meaning that, if $Ly \neq 0$, then y is an irregular vector having a high-frequency (i.e. noisy) appearance. If one has a good intuition about some features of the desired reconstructed solution, a regularization matrix L whose null space contains vectors having the above mentioned features should be employed when performing Tikhonov regularization in general form (cf. Figure 1.3.4). On the other side, one should also be sure that the null-space of L is not too large, otherwise too many components of the vector x_λ would be unaffected by regularization.

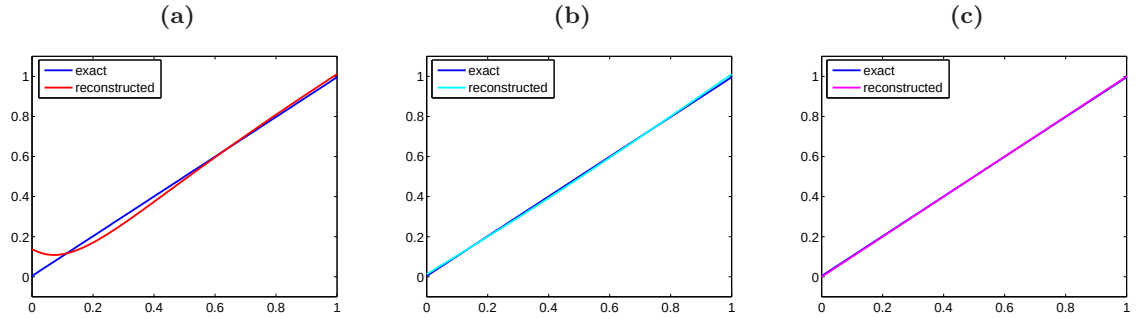

Figure 1.3.4: Different reconstructions of the solution of the test problem `foxgood` of size 128, having $\widetilde{\varepsilon} = 10^{-2}$. For this problem, $x^{ex} \in \mathcal{N}(D_2)$. **(a)** Reconstruction by standard Tikhonov regularization; $\lambda = 7.25 \cdot 10^{-2}$, relative error $\|x_\lambda - x^{ex}\|/\|x^{ex}\| = 5.3537 \cdot 10^{-2}$. **(b)** Reconstruction by general Tikhonov regularization with $L = D_2$; $\lambda = 10$, relative error $\|x_\lambda - x^{ex}\|/\|x^{ex}\| = 1.0990 \cdot 10^{-2}$. **(c)** Reconstruction by general Tikhonov regularization with $L = D_2$; $\lambda = 10^4$, relative error $\|x_\lambda - x^{ex}\|/\|x^{ex}\| = 4.0423 \cdot 10^{-3}$. We remark that, when x^{ex} completely belongs to the null space of the regularization matrix (as in this example), better results are obtained by giving more weight to the regularization term in the minimization process.

We derive an expression of the regularized solution x_λ analogous to the one in (1.3.11), the only difference being that for Tikhonov regularization in general form the GSVD of (A, L) is involved and therefore the filtering is defined with respect to the generalized singular values and vectors. Starting from the regularized solution expressed by the normal equations (1.3.7), substituting the decompositions (1.1.15) or (1.1.16), and exploiting the properties of the matrices therein involved, we easily obtain

$$x_\lambda = \sum_{i=1}^{q} \frac{\widetilde{\sigma}_i^2}{\widetilde{\sigma}_i^2 + \lambda \widetilde{\mu}_i^2} \frac{\widetilde{u}_i^T b}{\widetilde{\sigma}_i} x_i + \sum_{i=q+1}^{n} (\widetilde{u}_i^T b) x_i \tag{1.3.14}$$

if $q < n$, or

$$x_\lambda = \sum_{i=1}^{n} \frac{\widetilde{\sigma}_i^2}{\widetilde{\sigma}_i^2 + \lambda \widetilde{\mu}_i^2} \frac{\widetilde{u}_i^T b}{\widetilde{\sigma}_i} x_i \tag{1.3.15}$$

if $q \geq n$. Therefore, the filter factors for Tikhonov regularization in standard form are

$$f_i^\lambda = \frac{\widetilde{\sigma}_i^2}{\widetilde{\sigma}_i^2 + \lambda \widetilde{\mu}_i^2}, \quad i = 1, \ldots, \min\{q, n\}. \tag{1.3.16}$$

Remembering that, for $q \leq n$, we assume L to be of full rank (cf. Section 1.1.3), we can divide both the numerator and the denominator of f_i^λ in (1.3.16) by $\widetilde{\mu}_i^2$: in this way we obtain an expression formally identical to (1.3.12) where, instead of i-th singular value σ_i, the i-th generalized singular value $\gamma_i = \widetilde{\sigma}_i / \widetilde{\mu}_i$ appears. The behavior of the filter factors for general form Tikhonov regularization is analogous to the one explained in the previous section for standard form Tikhonov regularization: if $\lambda \gg \gamma_i$ (i.e., for small i's), then $f_i^\lambda \simeq 0$; if $\lambda \ll \gamma_i$ (i.e., for large i's), then $f_i^\lambda \simeq 1$ (remember that the generalized singular values are ordered nondecreasingly). Looking at the expressions in (1.3.14) and (1.3.15) we have a further proof that the components laying in the null space of L are unregularized. Indeed, in (1.3.14), the contributions given by the vectors x_i, $i \geq q+1$, are unaffected by the choice of λ; in (1.3.15), if $\mu_k = 0$, then the contributions given by the vectors x_i, $i \geq k$, are unaffected by the choice of λ.

We conclude this section by listing the main regularization matrices commonly used, along with some variations. As already addressed, most of them are scaled finite differences approximations of some derivative operator. One of the most basic regularization matrix is

$$D_1 = \begin{pmatrix} 1 & -1 & & \\ & \ddots & \ddots & \\ & & 1 & -1 \end{pmatrix} \in \mathbb{R}^{(n-1)\times n}, \tag{1.3.17}$$

which discretizes the one-dimensional first derivative. The null space of this matrix is given by the constant vectors, i.e.,

$$\mathcal{N}(D_1) = \left\{ [1, 1, \ldots, 1]^T \right\} \in \mathbb{R}^n,$$

consistently with the fact that the first derivative of a constant function is the zero function. Another very popular choice for L is the discretized one-dimensional second derivative

$$D_2 = \begin{pmatrix} 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \end{pmatrix} \in \mathbb{R}^{(n-2)\times n}; \tag{1.3.18}$$

again, consistently with the continuous case, the null-space is given by constant and linearly increasing (or decreasing) vectors, i.e.,

$$\mathcal{N}(D_2) = \left\{ [1, 1, \ldots, 1]^T, [1, 2, \ldots, n]^T \right\} \in \mathbb{R}^n.$$

The list could continue including matrices that represent discretizations of the third, fourth etc. one-dimensional derivatives. The matrices defined so far do not force any kind of boundary condition: anyway, if the behavior of the solution is known at the boundary of its domain, one can try to recover a more accurate reconstruction by incorporating the prescribed boundary conditions into the regularization matrix. For instance, if we are dealing with zero boundary conditions, the matrices

$$D_1^0 = \begin{pmatrix} 1 & & & \\ 1 & -1 & & \\ & \ddots & \ddots & \\ & & 1 & -1 \\ & & & -1 \end{pmatrix} \in \mathbb{R}^{n\times n} \tag{1.3.19}$$

and

$$D_2^0 = \begin{pmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ & & & 1 & -2 \end{pmatrix} \in \mathbb{R}^{n \times n}, \tag{1.3.20}$$

which are square modifications of the ones defined in (1.3.17) and (1.3.18), can enforce the reconstructed solution to be zero at its boundary, since the only vector belonging to their null space is the zero vector (trivial null space). Many other variants of the basic matrices (1.3.17) and (1.3.18) have been considered in literature: among the most recently studied ones, we refer to [22, 28, 29, 106]. We remark that most of the matrices obtained by modifying the standard rectangular ones coming from the discretization of some derivative operator are square: working with square regularization matrices is particularly advantageous when one wants to apply some iterative regularization method based on the Arnoldi algorithm (cf. Sections 2.2, 2.3, and 3.2).

We now turn our attention to the definition of suitable regularization matrices that can be regarded as discrete versions of some partial derivative: such matrices are useful when we are dealing with multi-dimensional problems, as for instance the ones arising in the image reconstruction case. Recalling the model described in Section 1.2.2 we have that, working with the two-dimensional array X, the first and second derivatives of the columns of X (i.e., in the vertical direction of the image) are given by

$$D_1 X \quad \text{and} \quad D_2 X,$$

where D_1 and D_2 are defined in (1.3.17) and (1.3.18). Analogously, we can consider the first and the second derivatives of the rows of X (i.e., in the horizontal direction of the image) by taking

$$X D_1^T \quad \text{and} \quad X D_2^T.$$

Working with the one-dimensional array $x = \text{vec}(X)$ and exploiting some properties of the Kronecker product, we obtain

$$\text{vec}(D_1 X I_N) = (I_N \otimes D_1)x \quad \text{and} \quad \text{vec}(I_N X D_1^T) = (D_1 \otimes I_N)x$$

(analogously for the matrix D_2), and therefore we can define the first and the second derivative in the vertical and horizontal directions as

$$D_1^v = I_N \otimes D_1 \in \mathbb{R}^{(n-N)\times n}, \qquad D_1^h = D_1 \otimes I_N \in \mathbb{R}^{(n-N)\times n}, \tag{1.3.21}$$

$$D_2^v = I_N \otimes D_2 \in \mathbb{R}^{(n-2N)\times n}, \qquad D_2^h = D_2 \otimes I_N \in \mathbb{R}^{(n-2N)\times n}. \tag{1.3.22}$$

Therefore, if one wants to uniformly act on the horizontal and the vertical components of a perturbed image, the regularization matrices

$$D_1^{hv} = \begin{pmatrix} D_1^h \\ D_1^v \end{pmatrix} \in \mathbb{R}^{2(n-N)\times n} \tag{1.3.23}$$

and

$$D_2^{hv} = \begin{pmatrix} D_2^h \\ D_2^v \end{pmatrix} \in \mathbb{R}^{2(n-2N)\times n} \tag{1.3.24}$$

can be employed; we remark that the matrix (1.3.24) can be regarded as a discrete version of the two-dimensional Laplacian operator.

Finally, we mention a method used to build square regularization operators L having a given null space: this strategy has been theorized in [85] and we employ it to perform some numerical tests in later chapters. Since the null space of the regularization operator has a core role in

the quality of the obtained reconstruction, we could expect that, in some circumstances, this strategy to define regularization operators is very successful (besides being also computationally convenient). Given $M \in \mathbb{R}^{n\times\ell}$, whose column vectors reproduce the features of the solution one wants to preserve, we compute the QR factorization

$$M = WR, \quad W \in \mathbb{R}^{n\times\ell},\ R \in \mathbb{R}^{\ell\times\ell}$$

and we take, as regularization matrix,

$$L := I_n - WW^T \in \mathbb{R}^{n\times n}. \tag{1.3.25}$$

In this way the null space of L is spanned by the orthonormal columns of W.

TSVD

Truncated Singular Value Decomposition (TSVD) is perhaps the most basic form of regularization. Recalling the discussion in Section 1.1.3, we realize that the extremely large error components in the unregularized solution expressed with respect to the right singular vector basis are associated to the smallest singular values, i.e. ${}^{u_i^T b}/_{\sigma_i} \simeq {}^{u_i^T e}/_{\sigma_i}$ for large i's. However, if the DPC holds, the components corresponding to the largest singular values are not affected by errors, i.e., ${}^{u_i^T b}/_{\sigma_i} \simeq {}^{u_i^T b^{ex}}/_{\sigma_i} = v_i x^{ex}$ for small i's. Therefore, the basic idea behind TSVD is to define the regularized solution x_m by just suppressing the components corresponding to the smallest singular vectors, i.e.,

$$x_m = \sum_{i=1}^{m} \frac{u_i^T b}{\sigma_i} v_i, \quad m \le \mathrm{rk}(A). \tag{1.3.26}$$

Referring to the expression (1.3.3), in the TSVD case the filter factors are simply defined as

$$f_i^m = \begin{cases} 1 & \text{if } i \le m \\ 0 & \text{otherwise} \end{cases}. \tag{1.3.27}$$

The scalar $m \in \mathbb{N}$ is called truncation index and acts as a regularization parameter: if m is chosen too small, i.e. if too few components are retained, the regularized solution x_m is over-smoothed; on the contrary, if m is chosen too big, i.e. if too many component are retained, the regularized solution x_m is under-smoothed (cf. Figure 1.3.5). A first elementary way to choose the truncation index is given by a visual inspection of the Picard plot (cf. Section 1.1.4): when the coefficients $u_i^T b$ level off, it means that the components of the approximate solution are dominated by noise.

The TSVD regularized solution x_m can be equivalently defined as the minimum 2-norm solution of the following least squares problem

$$\min_{x\in\mathbb{R}^n} \|b - A_m x\|, \tag{1.3.28}$$

where A_m is the best rank-m approximation of A defined in (1.1.11). We remark that the bigger m the worse the conditioning of A_m; on the other side, the bigger m the closer A_m to A. Therefore, when choosing the truncation parameter, we should try to balance the smoothness of the recovered solution and the "proximity" to the original problem. Writing the solution of the above least squares problem as $x_m = A_m^\dagger b$, we straightforwardly obtain that the regularized inverse associated to the TSVD method is

$$A_m^\sharp = A_m^\dagger = V_m \Sigma_m^{-1} U_m^T. \tag{1.3.29}$$

Many variations of the basic TSVD (1.3.26) have been proposed. We briefly describe the so-

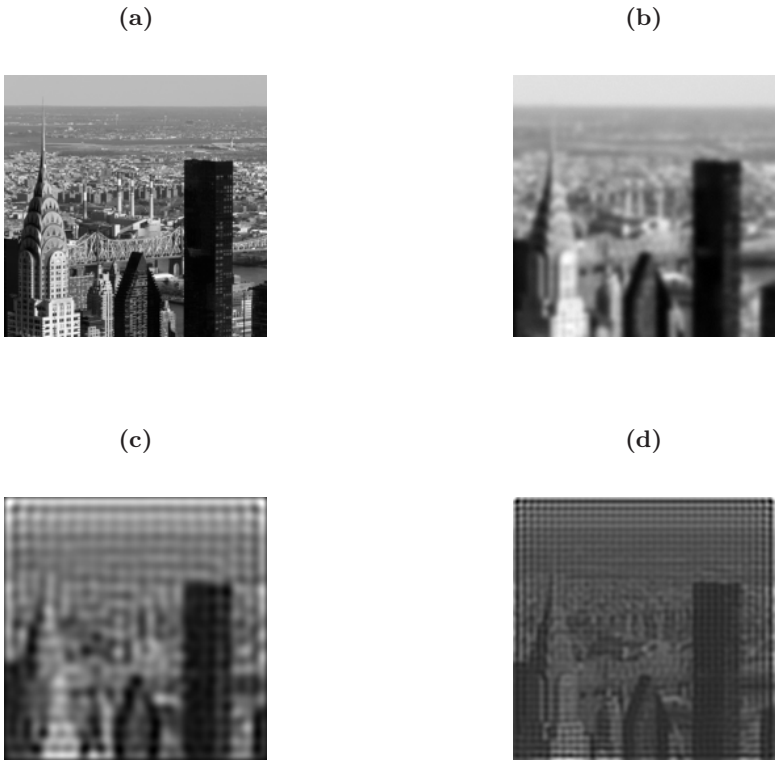

Figure 1.3.5: Images restored by the TSVD method. **(a)** Exact image. **(b)** Image corrupted by a symmetric Gaussian blur with $\sigma = 2$ and by white noise whose level is 10^{-2}. **(c)** Over-smooth reconstruction. **(d)** Under-smooth reconstruction.

called Selective SVD (SSVD), first introduced in [111]. The basic idea behind the SSVD is that the approximate solution of (1.1.7) is degraded because of the behavior of the coefficients $u_i^T b$'s (since the available vector b does not satisfy the DPC), which are not even guaranteed to be monotonically decreasing. Therefore, the filtering should be defined looking at the sequence $\{u_i^T b\}_{1 \le i \le n}$: once a threshold τ_F has been set, the SSVD regularized solution is obtained by computing

$$x_{\tau_F} = \sum_{|u_i^T b| > \tau_F} \frac{u_i^T b}{\sigma_i} v_i.$$

In the case of SSVD, the regularization parameter τ_F is continuous, even if it is used to define a truncation.

Another effective variant of method (1.3.26) is obtained by replacing the SVD of A by the GSVD (1.1.15) of the matrix pair (A, L), for a suitable L (for instance, L can be one of the matrices described in the previous paragraph), cf. [52]. This method is named truncated GSVD (TGSVD), for obvious reasons; the regularized solution is defined by

$$x_m = \sum_{i=1}^{m} \frac{\widetilde{u}_i^T b}{\widetilde{\sigma}_i} x_i, \quad m \le \min\{\mathrm{rk}(A), q\}.$$

There are many theoretical justifications for this choice of involving the GSVD: first of all, when expressing the regularized solution, the basis vectors are the right generalized singular vectors of (A, L), and therefore the basis is affected by the presence of L. Moreover, as explained above, the TSVD solution can be obtained by solving the 2-norm least squares problem (1.3.28): the TGSVD is obtained by solving the same problem evaluated by the seminorm $\|L \cdot\|$. Finally, expressing the solution x_m in terms of the GSVD of (A, L) is mathematically equivalent to expressing the solution x_m in terms of the SVD of the matrix $AL_A^\dagger$, where $L_A^\dagger$ is the A-weighted generalized pseudoinverse defined in (1.3.13).

Further versions of the TSVD have been obtained by slightly modifying the filter factors f_i^m's, in order to partially fill-in the gap between 0 and 1: we refer, for instance, to [36].

1.3.2 Iterative Methods

As already addressed in the previous sections, the regularization methods described so far are unfeasible for many large-scale ill-posed problems, the main difficulty being the cost of computing the SVD. The only alternative to practically deal with large-scale problems is to apply an iterative method. Starting from an initial guess for x^{ex}, the approximate solution x_{m+1} at the end of the $(m+1)$-th step of a generic iterative method is obtained by performing some computations involving the matrices A, A^T (sometimes just A) and the previous approximation x_m. The total computational cost of an iterative method depends on the cost of each matrix-vector multiplications by A and A^T, on the number of multiplications needed at each iteration, and on the total number of iterations to be run to obtain a suitable approximation of x^{ex}. Often the matrices associated to ill-posed problems are sparse and sometimes they are just known implicitly: dealing with this kind of matrices by only performing matrix-vector products is computationally very attractive. On the downside, purely iterative methods are affected by the so-called semiconvergence phenomenon (cf. Figure 2.6.1), i.e., during the early iterations, the intermediate solution x_m converges towards x^{ex} (over-smoothing phase), but then, during the later iterations, x_m starts to diverge (under-smoothing phase). In other words, at the beginning of the iterative process, the approximate solution is regularized, but rapidly becomes dominated by noise components. For this reason, when regularizing by an iterative method, one should be sure to provide a reliable stopping criterion to conveniently stop the iterations before the approximate solution results too corrupted. Therefore, in this setting, the number of iterations m acts a s a regularization parameter.

In this section we just describe some very basic iterative methods, which have classically been employed with regularizing purposes; each of them are expressed by the following update formula

$$x_{m+1} = x_m + \omega_m d_m, \tag{1.3.30}$$

where the vector $d_m \in \mathbb{R}^n$ is called step direction and the scalar $\omega_m \in \mathbb{R}$ is called step length; different choices of these quantities lead to different iterative method. We remark that the following iterative methods are just listed for the sake of completeness, since in this thesis we are never going to apply them. The next chapter introduces a more general approach to iterative regularization methods.

Standard Landweber method

The Landweber method (sometimes also referred to as Richardson method) prescribes to choose, in formula (1.3.30), a stationary $\omega_m = \omega$ and $d_m = A^T(b - Ax_m)$, i.e., the $(m+1)$-th approximation is obtained by computing

$$x_{m+1} = x_m + \omega A^T(b - Ax_m). \tag{1.3.31}$$

If we choose $\omega \in (0, 2/\sigma_1^2)$, convergence is guaranteed; a common choice for the step parameter is $\omega = 1/\|A\|_1\|A\|_\infty$. It can be shown that Landweber method can be interpreted as a SVD filtering method (1.3.3) whose filter factors at the m-th iteration are given by

$$f_i^m = 1 - \left(1 - \omega\sigma_i^2\right)^m, \quad i = 1, \dots, n.$$

Looking at the above expression we see that $f_i^m \simeq m\omega\sigma_i^2$ for small singular values σ_i's and therefore the decay rate is similar to the Tikhonov filter factors one (1.3.12). Moreover, as the number of iterations m increases, the filter factors are shifted toward smaller singular values (we can easily see this by fixing a scalar $\bar{k}$ and studying for which index i we have $f_i^m = \bar{k}$, for different values of m); therefore, going on with the iterates, more SVD components are included and this gives a further explanation of the semiconvergent behavior observed when applying Landweber method. In practice Landweber method is seldom used for regularization purposes, because of its very slow convergence.

Preconditioned Landweber method

To partially remedy the slow convergence of the classical Landweber method, one can resort to preconditioning. In a classical sense, a preconditioner is a matrix P chosen such that it is "close" to A (of course, the term "close" should be made more precise and depends on the context), it is inexpensive to invert, and a fixed iterative method is faster on AP^{-1} or $P^{-1}A$ than on A itself. We distinguish between right and left preconditioning: in the former case, A is replaced by $\widetilde{A} = AP^{-1}$ and x is replaced by $\widetilde{x} = Px$; in the latter case, A is replaced by $\widetilde{A} = P^{-1}A$ and b is replaced by $\widetilde{b} = P^{-1}b$. By applying some particular preconditioners we obtain the following well-known slightly more advanced methods:

Cimmino method. Cimmino method is derived by choosing the diagonal matrix

$$P = \operatorname{diag}\left(\frac{\|a_1^T\|}{\sqrt{c_1}}, \frac{\|a_2^T\|}{\sqrt{c_2}}, \dots, \frac{\|a_n^T\|}{\sqrt{c_n}}\right)$$

as a left preconditioner for the system (1.1.7), where $a_i^T = A^T e_i \in \mathbb{R}^n$, $i = 1, \dots, n$ is the i-th row of the matrix A, and c_i, $i = 1, \dots, n$ are positive constants. Of course, the above defined matrix P is easily invertible and, when the quantities relative to the left-preconditioned system are inserted into the scheme (1.3.31), we obtain the following update formula

$$x_{m+1} = x_m + \omega A^T M^{-1}(b - Ax_m), \tag{1.3.32}$$

where $M = PP^T$. The choice of the step parameter ω is affected by the preconditioning, too: to guarantee convergence, we should set $\omega \in (0, \|M^{-1}A^TA\|)$; common choices for this parameter are $\omega = 1/\|M^{-1}\|\|A\|_1\|A\|_\infty$ or $\omega = \sum_{i=1}^n c_i$.

SSOR methods. SSOR (symmetric successive over-relaxation) method is derived by first considering a splitting of the matrix A^TA defined by

$$A^TA = T + D + T,$$

where D is the diagonal part of A^TA and T is the strictly lower triangular part of A^TA. By employing the matrix

$$P_\omega = \sqrt{\frac{1}{2-\omega}} D^{-\frac{1}{2}}(D + \omega T)^T.$$

as a right preconditioner for the system (1.1.7) and by applying Landweber method (1.3.31) to it, we obtain the following update scheme

$$x_{m+1} = x_m + M_\omega^{-1}A^T(b - Ax_m), \tag{1.3.33}$$

where $M_\omega = P^TP$. As it is evident looking at the above expression, the parameter ω is defined implicitly into the matrix M_ω. While ω is usually chosen greater than 1 when dealing with well-posed problems, for ill-posed problems $\omega \ll 1$ is a better choice: in this way we assure some stability in the preconditioner but, at the same time, the convergence could still be very slow.

Steepest Descendent

We very briefly describe the steepest descent method as an example of iterative method that can be expressed in the form (1.3.30) and whose step size is adaptively defined at each iteration. First of all we remark that solving the problems (1.1.7) is mathematically equivalent to finding the minimum of the quadratic functional

$$\psi(x) = \frac{1}{2}x^TA^TAx - x^TA^Tb,$$

and that the step direction $d = A^T(b - Ax)$ is indeed the direction of the negative gradient of $\psi(x)$. The steepest descendent method is derived by choosing, at each step m, the negative gradient of $\psi(x_m)$ as step direction d_m, and the minimum of $\psi(x)$ in the direction d_m as step length ω_m, i.e.,

$$\omega_m = \arg\min_\omega \psi(x_m + \omega d_m) = \frac{\|d_m\|^2}{\|Ad_m\|^2}.$$

An analysis of the steepest descent method applied to ill-posed problems can be found in [89].

Chapter 2

Projection methods and Krylov subspace methods

In this chapter we introduce the general framework of projection methods, recalling the basic principles underlying them. Krylov subspace methods are particularly important cases of projection methods: indeed, the regularization methods described later in this thesis are based on projecting the original discrete ill-posed problem or the Tikhonov regularized problem onto suitable Krylov subspaces and then solving a sequence of reduced-dimension regularized problems. When introducing some of the Krylov subspace methods that are employed as iterative regularization methods, our goal is to provide some basic information, useful to better understand their regularizing properties; therefore, this chapter is not meant to give an exhaustive descriptions of different Krylov subspace methods. Our presentation is mainly based on [115], but some ideas are also taken from [8, 11, 42, 55, 59]. We also add a new analysis of the convergence and approximation properties of Krylov subspaces. This chapter is organized as follows: Section 2.1 is concerned with general projection methods; in Sections 2.2 we define the Krylov subspaces and we introduce the Arnoldi algorithm; in Section 2.3 we briefly describe the FOM, GMRES and CG methods; in Section 2.4 we introduce the Lanczos bidiagonalization procedure, along with the well-known equivalent CGLS and LSQR methods that work with both the coefficient matrix and its transpose; in Section 2.5 some theoretical estimates regarding the convergence of the Arnoldi algorithm, the convergence of the GMRES method and the approximation of the SVD are presented; finally, in Section 2.6, the regularizing properties of some iterative methods based on Krylov subspaces are described.

2.1 Projection methods

Projection methods aim at extracting an approximation of the solution of a linear system

$$Ax = b, \quad A \in \mathbb{R}^{n\times n} \tag{2.1.1}$$

from a given subspace. In general, once two m-dimensional subspaces $\mathcal{P}$ and $\mathcal{Q}$ of $\mathbb{R}^n$ have been fixed, a projection technique onto the subspace $\mathcal{P}$ and orthogonal to the subspace $\mathcal{Q}$ is a process that finds an approximate solution $\widetilde{x}$ to (2.1.1) by imposing the conditions

$$\widetilde{x} \in \mathcal{P} \quad \text{and} \quad b - A\widetilde{x} \perp \mathcal{Q}. \tag{2.1.2}$$

For obvious reasons, the subspace $\mathcal{P}$ is called search subspace, or subspace of candidate approximants. Since m constraints should be imposed to determine a vector in $\mathcal{P}$, m independent orthogonality conditions are considered: for this reason, the subspace $\mathcal{Q}$ is called subspace of constraints. A first classification of projection methods is derived just looking at the choice of $\mathcal{P}$ and $\mathcal{Q}$: if $\mathcal{P} = \mathcal{Q}$, then the method is called orthogonal; if $\mathcal{P} \neq \mathcal{Q}$, then the method is

called oblique. Conditions (2.1.2) are often referred to as Petrov-Galerkin conditions (or just Galerkin conditions, in the orthogonal case). If an initial guess x^* for the solution x^{ex} of (2.1.1) is available and we want to incorporate it into the iterative process, then we should find an approximation

$$\widetilde{x} \in x^* + \mathcal{P} \quad \text{such that} \quad b - A\widetilde{x} \in \mathcal{Q}. \tag{2.1.3}$$

Once we have rewritten the approximate solution as $\widetilde{x} = x^* + \delta$ and we have considered the initial residual $r^* = b - Ax^*$, conditions (2.1.3) can be equivalently expressed as

$$\delta \in \mathcal{P} \quad \text{and} \quad (r^* - A\delta, w) = 0 \quad \forall\, w \in \mathcal{Q}, \tag{2.1.4}$$

where (x_1, x_2) denotes the Euclidean scalar product of two vectors x_1, $x_2 \in \mathbb{R}^n$. Projection methods provide an unifying framework for most well-known iterative methods: in this setting, an iterative process is obtained by employing a sequence of projections defined updating the subspaces $\mathcal{P}$, $\mathcal{Q}$ and the initial guess x^* at each iteration. A matrix representation of a generic projection method can be derived by fixing two $n \times m$ matrices M and N whose columns span the subspaces $\mathcal{P}$ and $\mathcal{Q}$, respectively. In this way, the first condition in (2.1.4) can be trivially rewritten as

$$\widetilde{x} = x^* + My, \quad y \in \mathbb{R}^m.$$

Substituting the above expression in the second one of conditions (2.1.4), we get the following linear system

$$N^T AMy = N^T r^*.$$

If the $m \times m$ matrix $N^T AM$ is nonsingular (for instance, nonsingularity is guaranteed if A is positive definite and $\mathcal{P} = \mathcal{Q}$, or if A is nonsingular and $\mathcal{Q} = A\mathcal{P}$), then we can recover an approximation $\widetilde{x}$ of the solution of the original system (2.1.1) by taking

$$\widetilde{x} = x^* + M(N^T AM)^{-1} N^T r^*. \tag{2.1.5}$$

The previous derivations can be summarized in the following Algorithm 1.

Algorithm 1: Generic projection method

Input: A, b, x^*
For $m = 1, 2, \ldots$ until convergence

1. Choose two m-dimensional subspaces $\mathcal{P}$ and $\mathcal{Q}$.
2. Choose two matrices $M \in \mathbb{R}^{n \times m}$ and $N \in \mathbb{R}^{n \times m}$ such that their columns span the spaces $\mathcal{P}$ and $\mathcal{Q}$, respectively.
3. Take $r^* = b - Ax^*$.
4. Compute $y = (N^T AM)^{-1} N^T r^*$.
5. Update $x^* = x^* + My$.

Take $\widetilde{x} = x^*$.

Focussing on the case of orthogonal projection methods ($\mathcal{P} = \mathcal{Q}$) and oblique projection methods with $\mathcal{P} = A\mathcal{Q}$, we state two important characterizations and optimality properties that are used in the following sections.

Proposition 1. *Let A be symmetric positive definite and $\mathcal{P} = \mathcal{Q}$. Then a vector $\widetilde{x}$ is the result of an orthogonal projection onto $\mathcal{P}$ if and only if*

$$\|x^{ex} - \widetilde{x}\|_A = \min_{x \in x^* + \mathcal{P}} \|x^{ex} - x\|_A ,$$

where $\|z\|_A = (z, z)_A^{1/2} = (Az, z)^{1/2}$ is the A-norm of a generic vector $z \in \mathbb{R}^n$.

Proposition 2. *Let A be square, and assume that $\mathcal{Q} = A\mathcal{P}$. Then a vector $\widetilde{x} \in x^* + \mathcal{P}$ is the result of an oblique projection method onto $\mathcal{P}$ orthogonally to $\mathcal{Q}$ with starting vector x^* if and only if*

$$\|b - A\widetilde{x}\| = \min_{x \in x^* + \mathcal{P}} \|b - Ax\|.$$

The proofs of the above properties are omitted, since can be found in [115] and are quite immediate by using the properties of projectors. Indeed, projection methods are inevitably linked to projectors (or projection operators): a projector P is an idempotent (i.e., $P^2 = P$) linear mapping from $\mathbb{R}^n$ to itself. By definition, also $(I - P)$ and P^T are projectors. Given two m-dimensional subspaces $\mathcal{P}'$ and $\mathcal{Q}'$ of $\mathbb{R}^n$, such that no nonzero vector of $\mathcal{P}'$ is orthogonal to $\mathcal{Q}'$, a projection operator onto $\mathcal{P}'$ orthogonal to $\mathcal{Q}'$ is uniquely defined by requiring that, for any $x \in \mathbb{R}^n$,

$$Px \in \mathcal{P}' \quad \text{and} \quad (I - P)x \perp \mathcal{Q}'. \tag{2.1.6}$$

There are several other equivalent ways to characterize a projector; however, from the requirements (2.1.6) it is quite evident that a projector is such that

$$\mathcal{R}(P) = \mathcal{P}' \quad \text{and} \quad \mathcal{N}(P) = \mathcal{R}(I - P) = (\mathcal{Q}')^{\perp}.$$

P is said to be an orthogonal projector onto $\mathcal{P}'$ if $\mathcal{P}' = \mathcal{Q}'$; otherwise P is said oblique. Once a matrix M' whose columns span the space $\mathcal{P}'$, and a matrix N' whose columns span the space $\mathcal{Q}'$ have been fixed, the projector P can be expressed in matrix form as $P = M'(N'^T M')^{-1} N'^T$.

An orthogonal projector P onto $\mathcal{P}'$ has the two following important optimality property:

$$\|Px\| \leq \|x\| \quad \text{and} \quad \min_{y \in \mathcal{P}'} \|x - y\| = \|x - Px\|, \quad \forall x \in \mathbb{R}^n. \tag{2.1.7}$$

The vector Px appearing in the above formulas is said orthogonal projection of x onto $\mathcal{P}'$.

After having introduced some basic properties of the projector operators, we focus again on projection methods with $\mathcal{Q} = \mathcal{P}$ and $\mathcal{Q} = A\mathcal{P}$, and we reinterpret them in terms of projector. Assuming, as before, that $\mathcal{P} = \mathcal{Q}$ and that A is symmetric positive definite, we consider the error vectors

$$\delta^* = x^{ex} - x^* \quad \text{and} \quad \widetilde{\delta} = x^{ex} - \widetilde{x} = \delta^* - \delta.$$

We can easily prove that $\widetilde{r} = b - A\widetilde{x} = r^* - A\delta = A\widetilde{\delta}$; imposing $\widetilde{r}$ to be orthogonal to $\mathcal{P}$ we can derive that

$$(\delta^* - \delta, w)_A = 0 \quad \forall w \in \mathcal{P}.$$

Thanks to the previous relation we understand that δ is the orthogonal projection (with respect to the A inner product) of δ^* onto the subspace $\mathcal{P}$, i.e. $\delta = P_A \delta^*$ (P_A denotes the orthogonal projection operator onto $\mathcal{P}$). Therefore $\widetilde{\delta} = (I - P_A)\delta^*$ and, thanks to the optimality property (2.1.7), $\|\widetilde{\delta}\| \leq \|\delta^*\|$, which agrees with Proposition 1. Analogously, let us assume that $\mathcal{Q} = A\mathcal{P}$ and consider the residual vectors

$$r^* = b - Ax^* \quad \text{and} \quad \widetilde{r} = b - A\widetilde{x} = r^* - A\delta.$$

Since, by definition of oblique projection method,

$$(r^* - A\delta, w) = 0 \quad \forall w \in A\mathcal{P},$$

we can state that $A\delta$ is the orthogonal projection of r^* onto the subspace $A\mathcal{P}$, i.e. $\delta = P\delta^*$. Therefore $\widetilde{r} = (I - P)r^*$ and, thanks to the optimality property (2.1.7), $\|\widetilde{r}\| \leq \|r^*\|$, which agrees with Proposition 2.

Before concluding this section, we briefly comment on the quality of the approximation $\widetilde{x}$ attained by employing a projection method to solve the system (2.1.1). It is quite natural to expect that if no vector in $\mathcal{P}$ is "close" to x^{ex}, then the projection method badly perform;

indeed, denoting by P the orthogonal projection onto $\mathcal{P}$, directly from the estimate in (2.1.7) we obtain

$$\|(I - P)\, x^{ex}\| \leq \|x^{ex} - \widetilde{x}\|\,,$$

that is a lower bound on the error obtained by applying a projection method. One can also prove that, if $\mathcal{P}$ is invariant under A (i.e., $A\mathcal{P} \subset \mathcal{P}$), $x^* = 0$, and $b \in \mathcal{P}$, then the solution obtained from any projection method onto $\mathcal{P}$ is exact.

2.2 Krylov subspaces and the Arnoldi algorithm

Let $C \in \mathbb{R}^{n\times n}$ and $d \in \mathbb{R}^n$. The Krylov subspace $\mathcal{K}_m(C, d)$ with respect to C and d is defined by

$$\mathcal{K}_m(C, d) = \text{span}\{d, Cd, \ldots, C^{m-1}d\},\ m \geq 1. \tag{2.2.1}$$

Since in the following we employ Krylov subspaces to find an approximate solution of the linear system (2.1.1), it is natural to consider Krylov subspaces defined with respect to the coefficient matrix A and the right-hand-side vector b or the initial residual vector r^*: therefore, from now on, we make our considerations using the matrix A and the vector v, which can be either b or r^*, depending on the contexts. Directly from definition (2.2.1), we can state that $\mathcal{K}_m(A, v)$ is the subspace of all the vectors w of $\mathbb{R}^n$ that can be written as

$$w = p_{m-1}(A)v,$$

where p_{m-1} is a polynomial of degree at most $m-1$, i.e. $p_{m-1} \in \mathbb{P}_{m-1}$ ($\mathbb{P}_{m-1}$ is the space of all the polynomials whose degree does not exceed $(m-1)$). The link between Krylov subspaces and polynomials is useful to establish some basic properties of Krylov subspaces.

After defining the minimal polynomial of v relative to A as the monic polynomial p of minimal degree such that $p(A)v = 0$, and the grade of v relative to A (denoted by "$\text{grade}_A(v)$") as the degree of p (cf. [83, Chapter 7]), we can state the following proposition about the dimension of the Krylov subspace $\mathcal{K}_m(A, v)$.

Proposition 3. *The Krylov subspace $\mathcal{K}_m(A, v)$ is of dimension m if and only if*

$$m \leq \text{grade}_A(v);$$

$\mathcal{K}_{\text{grade}_A(v)}(A, v)$ is invariant under A and $\mathcal{K}_m(A, v) = \mathcal{K}_{\text{grade}_A(v)}(A, v)$ for $m \geq \text{grade}_A(v)$.

Recalling the estimates derived at the end of the previous section, we can state that every projection method method onto the subspace $\mathcal{K}_m(A, v)$ with $m \geq \text{grade}_A(v)$ is exact.

The Arnoldi algorithm [1] is a procedure employed to compute an orthonormal basis of $\mathcal{K}_m(A, v)$. There exist many variants of the Arnoldi algorithm: the simplest one (reported in Algorithm 2) is based on the standard Gram-Schmidt orthonormalization process.

Algorithm 2: Arnoldi algorithm (Standard Gram-Schmidt)

Input: A, v
Normalize v: $w_1 = {}^{v}/_{\|v\|}$.
For $j = 1, 2, \ldots, m$

1. For $i = 1, \ldots, j$: compute $h_{i,j} = (Aw_j, w_i)$.
2. Compute $w = Aw_j - \sum_{i=1}^{j} h_{i,j} w_i$.
3. Define $h_{j+1,j} = \|w\|$.
4. If $h_{j+1,j} = 0$ stop; else take $w_{j+1} = {}^{w}/_{h_{j+1,j}}$.

Starting from Algorithm 2, we can straightforwardly derive the following equalities

$$AW_m = W_m H_m + h_{m+1,m} w_{m+1} e_m^T \quad (2.2.2)$$
$$= W_{m+1} \bar{H}_m, \quad (2.2.3)$$

where $W_{m+1} = [w_1, ..., w_{m+1}] \in \mathbb{R}^{n\times(m+1)}$ has orthonormal columns that span the Krylov subspace $\mathcal{K}_{m+1}(A, v)$, and $w_1 = v/\|v\|$. The matrices

$$\bar{H}_m = \begin{pmatrix} h_{1,1} & h_{1,2} & h_{1,3} & \dots & \dots & h_{1,m} \\ h_{2,1} & h_{2,2} & h_{2,3} & \dots & \dots & h_{2,m} \\ & h_{3,2} & h_{3,3} & \dots & \dots & h_{3,m} \\ & & \ddots & \ddots & & \vdots \\ & & & \ddots & \ddots & \vdots \\ & & & & h_{m,m-1} & h_{m,m} \\ & & & & & h_{m+1,m} \end{pmatrix} \in \mathbb{R}^{(m+1)\times m}$$

and $H_m \in \mathbb{R}^{m\times m}$ (obtained by discarding the last row of $\bar{H}_m$) are upper Hessenberg. Directly from (2.2.2) we can also write

$$W_m^T A W_m = H_m. \quad (2.2.4)$$

The Arnoldi algorithm terminates as soon as $h_{m+1,m} = 0$, which means that an invariant subspace of A has been computed (this is true, at least, in exact arithmetic). In the following, we assume that m, the number of total iterations, is sufficiently small so that no breakdown happens and the above decompositions always exist; this hypothesis is perfectly meaningful in the regularization framework, since typically just very few iterations are needed to find a good approximation of the desired solution.

From now on, to keep the notations simple, we use the letter c to indicate either the vector

$$c = \|b\|e_1 \quad \text{or} \quad c = \|r^*\|e_1, \quad (2.2.5)$$

whose length should be clear from the context. In practice, we denote by c the right-hand side vector associated to the projected systems obtained considering either the relations (2.2.2) or (2.2.3).

A drawback of the Arnoldi procedure described in Algorithm 2 is that the vectors w_i's soon loose their orthogonality, i.e. the procedure is not numerically very stable. To avoid this degradation, more sophisticated versions of the algorithm are often considered: we cite the implementation based on the Modified Gram-Schmidt procedure and the one based on Householder orthogonalization. In the following we use the Householder-Arnoldi algorithm exclusively to check some theoretical estimates (cf. Section 2.5); when considering the Arnoldi algorithm for regularization purposes, we just employ its simplest version based on Standard Gram-Schmidt orthonormalization. This choice is motivated by the fact that the problems we are considering are intrinsically affected by some perturbations and we do not aim for a "perfect" reconstruction of the exact solution: we are just content when the dominating features of the exact solution are recovered and we commit an error that is comparable to the one that affected the data; to achieve this, the orthogonality of the Krylov basis vectors is not crucial, in the sense that it suffices to require

$$w_i^T w_j = \delta_{i,j} \pm p_{i,j}, \quad i, j = 1, \dots, m,$$

where $\delta_{i,j}$ is the Kronecker delta ($\delta_{i,j} = 0$ if $i \neq j$, $\delta_{i,j} = 1$ if $i = j$), and $\|p_{i,j}\| \ll \|e\|$. Moreover, the available vector b is affected by noise that is spread over the matrices W_m and $\bar{H}_m$.

If the matrix A is symmetric, the relations in the Arnoldi algorithm (Algorithm 2) simplify, giving rise to a three-term recurrence. Indeed, if A is symmetric, then the matrix H_m is

symmetric (recall relation (2.2.4)) and, at the same time, upper Hessenberg: the only matrix that satisfies both these requirements is a symmetric tridiagonal H_m, i.e.

$$h_{i,j} = 0,\ 1 \leq i < j-1, \quad \text{and} \quad h_{j,j+1} = h_{j+1,j},\ j = 1, \ldots, m.$$

In this case, the tridiagonal matrix H_m is usually denoted by

$$T_m = \begin{pmatrix} \alpha_1 & \beta_2 & & & \\ \beta_2 & \alpha_2 & \beta_3 & & \\ & \ddots & \ddots & \ddots & \\ & & \beta_{m-1} & \alpha_{m-1} & \beta_m \\ & & & \beta_m & \alpha_m \end{pmatrix} \in \mathbb{R}^{m \times m},$$

and the symmetric version of Arnoldi algorithm (Algorithm 2) is called Lanczos tridiagonalization (or, sometimes, symmetric Lanczos algorithm) [77]: we summarize it in Algorithm 3.

Algorithm 3: Lanczos tridiagonalization (Standard Gram-Schmidt)

Input: A symmetric, v
Normalize v: $w_1 = v/\|v\|$.
Set $\beta_1 = 0$, $w_0 = 0$.
For $j = 1, 2, \ldots, m$

1. Compute $\alpha_j = (Aw_j, w_j)$
2. Take $w = Aw_j - \alpha_j w_j - \beta_j w_{j-1}$.
3. Compute $\beta_{j+1} = \|w\|$.
4. If $\beta_{j+1} = 0$ stop; else define $w_{j+1} = w/\beta_{j+1}$.

Like in the Arnoldi algorithm case, severe loss of orthogonality happens during the process and more numerically stable algorithms have been developed to partially remedy it. Moreover, relations (2.2.2)-(2.2.4) still hold, provided that we formally substitute H_m and $\bar{H}_m$ by T_m and $\bar{T}_m$ ($\bar{T}_m$ is the $(m+1) \times m$ matrix obtained by appending the row $\beta_{m+1} e_m^T$ to the matrix T_m).

2.3 Krylov subspace methods

As said in the introduction to this chapter, Krylov subspace methods are particular and very important cases of projection methods: at the m-th iteration, the approximate solution x_m of (2.1.1) belongs to the subspace $x_0 + \mathcal{K}_m(A, r_0)$ defined in Section 2.2 and the residual r_m is orthogonal to some m-dimensional subspace $\mathcal{Q}_m$. In this setting we denote the initial quantities by the subscript 0 (instead of the asterisk employed so far): the reason behind this choice is that, in an iterative framework, we can regard the initial quantities as obtained at the 0-th iteration. Different choices of a basis for the subspace $\mathcal{K}_m(A, r_0)$ as well as different choices for the subspace $\mathcal{Q}_m$ determine different Krylov subspace methods. Many Krylov subspace methods proposed in literature are indeed mathematically equivalent, meaning that in exact arithmetic they would deliver the same solution. If no breakdown occurs (cf. again Section 2.2), then at each iteration the dimensions of the search and the constraints subspaces increase by 1.

Looking at Krylov subspace methods from the point of view of the theory of approximation, and recalling the link between Krylov subspaces and polynomials (Section 2.2) we can state that finding an approximate solution of (2.1.1) reduces to finding a suitable $q_{m-1} \in \mathbb{P}_{m-1}$ such that

$$A^{-1}b \simeq x_0 + q_{m-1}(A) r_0.$$

2.3.1 The FOM

The FOM (Full Orthogonalization Method, sometimes also referred to as Arnoldi's method for linear systems) is an orthogonal projection method that, given an initial guess x_0, at the m-th iteration takes $\mathcal{P} = \mathcal{Q} = \mathcal{K}_m(A, r_0)$. Recalling the derivations made in Section 2.1 for a general orthogonal projection method, the m-th approximation x_m computed by the FOM is such that $x_m = x_0 + W_m y_m$ ($y_m \in \mathbb{R}^m$), and $b - Ax_m \perp \mathcal{K}_m(A, r_0)$, where W_m is the orthonormal basis for $\mathcal{K}_m(A, r_0)$ generated by the Arnoldi algorithm (Algorithm 2). Using relation (2.2.2), the orthogonality condition can be rewritten as

$$\underbrace{W_m^T A W_m}_{H_m} y_m = \underbrace{W_m^T r_0}_{\|r_0\| e_1}.$$

Therefore, the m-th approximation obtained by the FOM can be computed as

$$x_m = x_0 + W_m y_m, \quad \text{where} \quad y_m = (H_m)^{-1} c, \tag{2.3.1}$$

and the vector c is defined in (2.2.5). The FOM method is summarized in Algorithm 4.

Algorithm 4: FOM

Input: A, b, x_0
Compute $\rho_0 = b - Ax_0$ and normalize ρ_0: $w_1 = \rho_0 / \|\rho_0\|$.
For $j = 1, 2, \ldots, m$

1. Perform one step of the Arnoldi algorithm (Algorithm 2) with input A, ρ_0, and update the matrices H_m and W_m.
2. If $h_{j+1,j} = 0$ compute $y_j = (H_j)^{-1} c$, $x_j = x_0 + W_j y_j$ and return.

Compute $y_m = (H_m)^{-1} c$ and $x_m = x_0 + W_m y_m$.

The m-th residual vector $\rho_m = b - Ax_m$ relative to the FOM can be straightforwardly computed employing relation (2.2.2) and using the equalities in (2.3.1). We have

$$\begin{aligned} b - Ax_m &= \rho_0 - AW_m y_m = W_m(c - H_m y_m) - h_{m+1,m} w_{m+1} e_1^T y_m \\ &= -h_{m+1,m}(e_1^T y_m) w_{m+1}. \end{aligned}$$

In particular, the norm of the m-th FOM residual is

$$\|\rho_m\| = h_{m+1,m} \left| e_m^T y_m \right|. \tag{2.3.2}$$

2.3.2 The GMRES method

The GMRES (Generalized Minimum Residual) method [116] is a highly regarded Krylov subspace method that can be introduced in two equivalent ways. In the framework of the projection methods, the GMRES method is defined as an oblique projection method having $\mathcal{P} = \mathcal{K}_m(A, r_0)$ and $\mathcal{Q} = A\mathcal{K}_m(A, r_0)$, where x_0 is an initial approximation for the solution. Once a basis for $\mathcal{K}_m(A, r_0)$ and a basis for $A\mathcal{K}_m(A, r_0)$ are provided, the approximate solution at the m-th step of the GMRES method can be recovered thanks to the general formula (2.1.5). An orthonormal basis W_m for $\mathcal{K}_m(A, r_0)$ can be easily generated running the Arnoldi algorithm (Algorithm 2). Concerning a basis for $A\mathcal{K}_m(A, b)$, the derivation is more cumbersome: substituting the "skinny" SVD of $\bar{H}_m$ (cf. Section 1.1.3), here denoted by $\bar{H}_m = \bar{U}\bar{\Sigma}\bar{V}^T$ (where $\bar{U} \in \mathbb{R}^{(m+1)\times m}$, $\bar{V} \in \mathbb{R}^{m\times m}$ have orthonormal columns, $\bar{\Sigma} = \mathrm{diag}(\bar{\sigma}_1, \ldots, \bar{\sigma}_m) \in \mathbb{R}^{m\times m}$), into the decomposition (2.2.3) we get

$$AW_m = W_{m+1}\bar{U}\bar{\Sigma}\bar{V}^T.$$

Let us define $Q = W_{m+1}\bar{U} \in \mathbb{R}^{n\times m}$: the matrix Q has orthonormal columns that span $A\mathcal{K}_m(A, r_0)$; indeed,

$$\begin{aligned} A\mathcal{K}_m(A, r_0) &= \text{span}\{Ab, \ldots, A^m b\} = \text{span}\{Aw_1, \ldots, Aw_m\} \\ &= \mathcal{R}(AW_m) = \mathcal{R}(W_{m+1}\bar{U}). \end{aligned}$$

Substituting the basis just determined into equation (2.1.5), we obtain

$$\begin{aligned} x_m &= x_0 + W_m(Q^T AW_m)^{-1}Q^T r_0 \\ &= x_0 + W_m(Q^T W_{m+1}\bar{H}_m)^{-1}\bar{U}^T \underbrace{\|r_0\|e_1}_{c} \\ &= x_0 + W_m(\bar{U}^T\bar{U}\bar{\Sigma}\bar{V}^T)^{-1}\bar{U}^T c \\ &= x_0 + W_m \underbrace{\bar{V}^T\bar{\Sigma}^{-1}\bar{U}^T}_{\bar{H}_m^\dagger} c = x_0 + W_m \underbrace{\bar{H}_m^\dagger c}_{y_m}, \end{aligned}$$

where we have exploited the relation (2.2.3) and the definition of pseudoinverse given in Section 1.1.3. Recalling the link between pseudoinverse and least squares problem, we can equivalently define

$$y_m = \arg\min_{y\in\mathbb{R}^m} \left\|c - \bar{H}_m y\right\|. \tag{2.3.3}$$

Thanks to Proposition 2, we can immediately state that the GMRES method, at the m-th step, minimizes the residual norm over all the vectors in $x_0 + \mathcal{K}_m(A, r_0)$.

Alternatively, we can derive the GMRES method by requiring that

$$x_m = x_0 + W_m y \tag{2.3.4}$$

and that the above mentioned optimality property holds, i.e.,

$$\|b - Ax_m\| = \min_{x\in x_0+\mathcal{K}_m(A,r_0)} \|b - Ax\|. \tag{2.3.5}$$

Taking $x = x_0 + W_m y$ in the right-hand side of the above equation and exploiting the factorization (2.2.3) along with the properties of the matrices thereby involved, we can write

$$\|b - Ax\| = \|r_0 - AW_m y\| = \left\|W_{m+1}\left(\|r_0\|e_1 - \bar{H}_m y\right)\right\|.$$

Therefore, at each step m, the approximation x_m is obtained by solving the reduced-dimension least squares problem (2.3.3) and by defining x_m as in (2.3.4). We summarize the GMRES method just derived in Algorithm 5.

Algorithm 5: GMRES method

Input: A, b, x_0
Compute $r_0 = b - Ax_0$ and normalize r_0: $w_1 = {}^{r_0}/_{\|r_0\|}$.
For $j = 1, 2, \ldots, m$

1. Perform one step of the Arnoldi algorithm (Algorithm 2) with input A, r_0 and update the matrices $\bar{H}_m$ and W_{m+1}.
2. If $h_{j+1,j} = 0$ compute $y_j = (H_j)^{-1}c$, $x_j = x_0 + W_j y_j$ and return.

Compute $y_m = \arg\min_{y\in\mathbb{R}^m} \|c - \bar{H}_m y\|$, $x_m = x_0 + W_m y_m$.

Thanks to the optimality property of the residual of the GMRES method and recalling that, if no breakdown happens in the Arnoldi algorithm, $\mathcal{K}_m(A, r_0) \subset \mathcal{K}_{m+1}(A, r_0)$, we can state

$$\|r_{m+1}\| \leq \|r_m\|. \tag{2.3.6}$$

Moreover, recalling the interpretation of the Krylov subspaces in terms of polynomials (cf. Section 2.2), condition (2.3.4) can be reformulated as

$$x_m = x_0 + p_{m-1}(A)r_0, \quad p_{m-1} \in \mathbb{P}_{m-1}.$$

Analogously, the residual $r_m = r_0 - Ax_m$ at the m-th step of the GMRES method can be expressed as

$$r_m = q_m(A)r_0, \quad \text{where} \quad q_m(t) = 1 - tp_{m-1}(t) \in \mathbb{P}_m^0 = \{p \in \mathbb{P}_m \,:\, p(0) = 1\}.$$

By simply expressing (2.3.6) in terms of polynomials, we can state that the GMRES residual satisfies the following optimality property

$$\|r_m\| = \|q_m(A)r_0\| \leq \|\hat{q}_m(A)r_0\|, \quad \forall\, \hat{q}_m \in \mathbb{P}_m^0. \tag{2.3.7}$$

2.3.3 The CG method

If A is a symmetric positive definite matrix, the CG (Conjugate Gradient) method is a well-known orthogonal projection method that, given an initial guess x_0 for the solution, takes $\mathcal{P} = \mathcal{Q} = \mathcal{K}_m(A, r_0)$. Therefore the CG method is mathematically equivalent to the FOM (recall the discussion in Section 2.3.1). The CG algorithm is derived by initially rewriting the FOM algorithm (Algorithm 4) employing Lanczos tridiagonalization (Algorithm 3). Summarizing the most important steps of the derivation of the CG algorithm, one can start by considering the LU factorization of T_m, i.e.,

$$x_m = x_0 + W_m(T_m^{-1}c) = x_0 + \underbrace{W_m U_m^{-1}}_{P_m}\underbrace{L_m^{-1}c}_{z_m};$$

by updating the LU factorization at each step m, an update formula of the type

$$x_m = x_{m-1} + \gamma_m p_m \tag{2.3.8}$$

is recovered for the solution. One can prove that the residual vectors r_i's are orthogonal to each other, and that the auxiliary vectors p_i's form an A-conjugate set, i.e., $(Ap_i, p_j) = \delta_{i,j}$. The most popular version of the CG method is retrieved by "reversing" the previous derivations: basically one seeks a solution of the form (2.3.8) and imposes the orthogonality conditions for the r_m's, and the A-conjugacy conditions for the p_m's. In Algorithm 6 we summarize the procedure obtained by rearranging the computations in a convenient way.

Algorithm 6: CG method

Input: A, b, x_0
Compute $r_0 = b - Ax_0$ and take $p_0 = r_0$.
For $j = 0, 1, \ldots,$ until convergence

1. Compute $\gamma_j = {}^{(r_j, r_j)}/_{(Ap_j, p_j)}$.
2. Update $x_{j+1} = x_j + \gamma_j p_j$.
3. Update $r_{j+1} = r_j - \gamma_j Ap_j$.
4. Compute $\delta_j = {}^{(r_{j+1}, r_{j+1})}/_{(r_j, r_j)}$.
5. Update $p_{j+1} = r_{j+1} + \delta_j p_j$.

Since the CG method satisfies the hypothesis of Proposition 1, we can immediately state that the CG method, at the m-th step, minimizes the error A-norm over all the vectors in $x_0 + \mathcal{K}_m(A, r_0)$.

2.4 Lanczos bidiagonalization, CGLS and LSQR

The projection methods so far analyzed require the system matrix A to be square; when this condition is not satisfied and we have to solve least squares problems of the form

$$\min_{x\in\mathbb{R}^n} \|b - Ax\|, \quad A \in \mathbb{R}^{k\times n},\ k \geq n, \tag{2.4.1}$$

a common approach is to consider the normal equations

$$A^T Ax = A^T b \tag{2.4.2}$$

associated to (2.4.1).

Many specific Krylov subspace methods involving $\mathcal{K}_m(A^T A, v)$ and approximating the solution x^{ex} of the original problem (2.4.1) have been developed. In the following, we briefly address the CGLS and LSQR methods, which are indeed mathematically equivalent: given an initial guess x_0 for x^{ex}, at the m-th step they both compute an approximation x_m such that

$$x_m \in x_0 + \mathcal{K}_m(A^T A, A^T r_0)$$

and

$$\|x^{ex} - x_m\|_{A^T A} = \min_{x\in x_0+\mathcal{K}_m(A^T A, A^T r_0)} \|x^{ex} - x\|_{A^T A}\,. \tag{2.4.3}$$

In particular, after some simple manipulations, condition (2.4.3) can be rewritten as

$$\|b - Ax_m\| = \min_{x\in x_0+\mathcal{K}_m(A^T A, A^T r_0)} \|b - Ax\|\,, \tag{2.4.4}$$

which is an optimality property for the norm of the residual of the original problem (2.4.1).

Of course, even if A is square, one could decide to solve the equivalent system (2.4.2) instead of (2.1.1). In this situation, thanks to the characterization given in Proposition 2, we can regard both the CGLS and the LSQR methods as oblique projection methods onto $\mathcal{K}_m(A^T A, A^T r_0)$ orthogonal to $A\mathcal{K}_m(A^T A, A^T r_0)$. The optimality property (2.4.4) is analogous to the GMRES one, coherently with the fact that also GMRES is an oblique projection method (cf. Section 2.3.2); GMRES differs from the CGLS and LSQR methods because of the subspaces chosen for the approximation.

We remark that a drawback of the transition to normal equations is the squaring of the condition number, i.e. $\mathrm{cond}(A^T A) = \mathrm{cond}(A)^2$.

The CGLS (sometimes also called CGNR) method is essentially the CG method (Section 2.3.3) applied to the normal equations (2.4.2), except for some modifications aimed at making explicitly available both the original residual $r_m = b - Ax_m$ and the residual of the normal equations $z_m = A^T r_m$ at each iteration m. We summarize the computation required by the CGLS method in Algorithm 7.

The LSQR method [98, 99] is based on the Lanczos bidiagonalization (LBD) algorithm (sometimes also referred to as Golub-Kahan bidiagonalization (GKB)), which iteratively computes the factorization

$$A = VBW^T, \quad \text{where} \quad V^T V = I_n,\ W^T W = I_n,\ B = \begin{pmatrix} \mu_1 & & & \\ \nu_2 & \mu_2 & & \\ & \ddots & \ddots & \\ & & \nu_n & \mu_n \end{pmatrix}.$$

Exploiting the orthonormality of the columns of V and W, the relations

$$AW = VB \quad \text{and} \quad A^T V = WB^T \tag{2.4.5}$$

Algorithm 7: CGLS method

Input: A, b, x_0
Compute $r_0 = b - Ax_0$, $z_0 = A^T r_0$ and take $p_0 = r_0$.
For $j = 0, 1, \ldots,$ until convergence

1. Compute $q_j = Ap_i$
2. Take $\gamma_j = {\|z_j\|^2}/{\|q_j\|^2}$.
3. Update $x_{j+1} = x_j + \gamma_j p_j$.
4. Update $r_{j+1} = r_j - \gamma_j q_j$.
5. Update $z_{j+1} = A^T r_{j+1}$.
6. Take $\delta_j = {\|z_{j+1}\|^2}/{\|z_j\|^2}$.
7. Update $p_{j+1} = z_{j+1} + \delta_j p_j$.

Algorithm 8: Lanczos bidiagonalization algorithm

Input: A, b, x_0.
Compute $r_0 = b - Ax_0$.
Initialize: $\nu_1 = \|r_0\|$, $v_1 = {r_0}/{\nu_1}$.
Initialize: $\hat{w}_1 = A^T v_1$, $\mu_1 = \|\hat{w}_1\|$, $w_1 = {\hat{w}_1}/{\mu_1}$.
For $j = 2, \ldots, m+1$

1. Compute $\hat{v}_j = Aw_{j-1} - \mu_{j-1} v_{j-1}$.
2. Set $\nu_j = \|\hat{v}_j\|$.
3. Take $v_j = {\hat{v}_j}/{\nu_j}$.
4. Compute $\hat{w}_j = A^T v_k - \nu_j w_{j-1}$.
5. Set $\mu_j = \|\hat{w}_j\|$.
6. Take $w_j = {\hat{w}_j}/{\mu_j}$.

are immediately derived from the above decomposition. If we consider equations (2.4.5) in a column-wise fashion, the two main recursion formulas to be employed in the LBD method are recovered. In Algorithm 8 we better explain the computations involved into Lanczos bidiagonalization procedure.

Algorithm 8 can be expressed in matrix form by the following relations

$$A^T V_{m+1} = W_m \bar{B}_m^T + \mu_{m+1} w_{m+1} e_{m+1}^T, \tag{2.4.6}$$

$$A W_m = V_{m+1} \bar{B}_m, \tag{2.4.7}$$

where $V_{m+1} = [v_1, \ldots, v_{m+1}] \in \mathbb{R}^{n\times(m+1)}$, $W_m = [w_1, \ldots, w_m] \in \mathbb{R}^{n\times m}$ have orthonormal columns, and $\bar{B}_m \in \mathbb{R}^{(m+1)\times m}$ is the lower bidiagonal matrix obtained by taking the leading $(m+1)\times m$ submatrix of B. We can easily see

$$\mathcal{R}(W_m) = \mathcal{K}_m(A^T A, A^T r_0) \quad \text{and} \quad \mathcal{R}(V_m) = \mathcal{K}_m(AA^T, r_0),$$

i.e., the sets of vectors $\{w_i\}_{i=1,\ldots,m}$ and $\{v_i\}_{i=1,\ldots,m}$ form an orthonormal basis for $\mathcal{K}_m(A^T A, A^T r_0)$ and $\mathcal{K}_m(AA^T, r_0)$, respectively.

Once Lanczos bidiagonalization algorithm and the matrix relations steaming from it have been established, the derivation of the LSQR method is very similar to the second derivation of the GMRES method presented in Section 2.3.2. Indeed, assuming that $x_0 = 0$, it suffices to impose the optimality property (2.4.4), i.e.,

$$x_m = \arg \min_{x\in\mathcal{R}(W_m)} \|b - Ax\|. \tag{2.4.8}$$

Substituting $x = W_m y$, $y \in \mathbb{R}^m$, in (2.4.8), using relation (2.4.7), and exploiting some properties of the involved matrices, we get that the m-th approximate solution delivered by the LSQR method is the vector $x_m = W_m y_m$, where y_m solves the following projected least squares problem

$$\min_{y\in\mathbb{R}^m} \|\bar{B}_m y - \underbrace{\|b\| e_1}_{c}\|. \tag{2.4.9}$$

Without going too much into the details, we just remark that the LSQR method solves each reduced-dimension subproblem (2.4.9) by updating of the QR factorization of the matrix $\bar{B}_m$ in a computationally convenient and reliable way. Finally, we remark that, when the right-hand side b is affected by errors, the matrices $\bar{B}_m$ and W_m are affected by errors, too; we refer to [66] for an exhaustive study of this phenomenon.

2.5 Some theoretical estimates

We fix some notations that are often used in the remaining part of this chapter.

Notations. We denote the best rank m approximation of the matrix A by

$$A_m = U_m \Sigma_m V_m, \tag{2.5.1}$$

exactly as done in equation (1.1.11) of the previous chapter. The SVD of the $m \times m$ matrix H_m defined in (2.2.2) is expressed by

$$H_m = U^{(m)} \Sigma^{(m)} V^{(m)T}, \tag{2.5.2}$$

where $U^{(m)}$, $V^{(m)} \in \mathbb{R}^{m\times m}$ are orthogonal, and $\Sigma^{(m)} = \mathrm{diag}(\sigma_1^{(m)}, \ldots, \sigma_m^{(m)})$; similarly, the SVD of the $(m+1)\times m$ matrix $\bar{H}_m$ defined in (2.2.3) is expressed by

$$\bar{H}_m = \bar{U}^{(m)} \bar{\Sigma}^{(m)} \bar{V}^{(m)T}, \tag{2.5.3}$$

where $\bar{U}^{(m)} \in \mathbb{R}^{(m+1)\times(m+1)}$, $\bar{V}^{(m)} \in \mathbb{R}^{m\times m}$ are orthogonal, and $\bar{\Sigma}^{(m)} = \mathrm{diag}(\bar{\sigma}_1^{(m)}, \ldots, \bar{\sigma}_m^{(m)}) \in \mathbb{R}^{(m+1)\times m}$, i.e.,

$$\bar{\Sigma}^{(m)} = \begin{pmatrix} \bar{\sigma}_1^{(m)} & & \\ & \ddots & \\ & & \bar{\sigma}_m^{(m)} \\ 0 & \cdots & 0 \end{pmatrix}.$$

Furthermore, when working with the full-dimensional problem, we use the following notation:

$$\begin{aligned} \hat{U}^{(m)} &= W_m U^{(m)} \in \mathbb{R}^{n\times m}; \\ \hat{V}^{(m)} &= W_m V^{(m)} \in \mathbb{R}^{n\times m}; \\ \hat{\bar{U}}^{(m)} &= W_{m+1} \bar{U}^{(m)} \in \mathbb{R}^{n\times(m+1)}; \\ \hat{\bar{V}}^{(m)} &= W_m \bar{V}^{(m)} \in \mathbb{R}^{n\times m}. \end{aligned}$$

The columns of every above listed matrix are denoted by the lowercase version of the letter used to indicate the matrix (together with the additional symbols and superscripts) and a subscript indicating the number of the column considered.

Finally, given two scalars $\tilde{a}$, $\tilde{b} \in \mathbb{R}$, we extensively adopt the notation $\tilde{a} = O(\tilde{b})$ meaning that there exists a constant $\tilde{c} \in \mathbb{R}$, $\tilde{c} \neq 0$, such that $\tilde{a} \leq \tilde{c}\tilde{b}$; indeed, when we use the "big O" notation in this setting we commit some abuse with respect to its standard analytic meaning (for instance, in our setting no proper functions are involved).

2.5.1 Convergence analysis for the Arnoldi algorithm

First of all, we present a core result that is used to derive many estimates in the remaining part of this section. This theorem is proved for the uncorrupted right-hand side and assuming the validity of the DPC (cf. Section 1.1.4); moreover, it holds just for severely ill-posed problems, according to the definition given in Section 1.1.1. Not to overload the notations, we simply denote the exact right-hand side by b, although this symbol is elsewhere used to indicate the available and noisy right-hand side.

Theorem 4. *Let us assume that the singular values of A are of the form*

$$\sigma_j = O(e^{-\alpha j}), \quad \alpha > 0;$$

let us moreover assume that the discrete Picard condition is satisfied. Let

$$\widetilde{V}_m = [\widetilde{v}_0, ..., \widetilde{v}_{m-1}] \in \mathbb{R}^{n\times m}, \quad \textit{where} \quad \widetilde{v}_k = {}^{A^k b}/_{\|A^k b\|} \in \mathbb{R}^n, \ k = 0, \ldots, m-1.$$

If $\widetilde{V}_m$ has full column rank, then there exist $C_m \in \mathbb{R}^{m\times m}$ nonsingular, and $E_m, F_m \in \mathbb{R}^{n\times m}$, such that

$$\widetilde{V}_m = U_m C_m + E_m, \quad \|E_m\| = O(\sigma_m), \tag{2.5.4}$$

$$U_m = \widetilde{V}_m C_m^{-1} + F_m, \quad \|F_m \Sigma_m\| = O(m\,\sigma_m). \tag{2.5.5}$$

Proof. Let $U_m^{\perp} = [u_{m+1}, ..., u_n] \in \mathbb{R}^{n\times(n-m)}$ be the orthogonal complement of U_m in $\mathbb{R}^n$. Since the orthonormal columns of $U = [U_m\, U_m^{\perp}]$ span the space $\mathbb{R}^n$, we can straightforwardly write

$$\widetilde{V}_m = U_m U_m^T \widetilde{V}_m + U_m^{\perp} (U_m^{\perp})^T \widetilde{V}_m.$$

In the above relation, defining $C_m := U_m^T \widetilde{V}_m \in \mathbb{R}^{m\times m}$ and

$$E_m := U_m^{\perp} \left(U_m^{\perp}\right)^T \widetilde{V}_m \in \mathbb{R}^{n\times m},$$

we formally obtain (2.5.4). Now we prove that

$$\left|u_j^T \widetilde{v}_k\right| = O(\sigma_j),\ j = 1, \ldots, n \tag{2.5.6}$$

for $0 \leq k \leq m-1$. The above relation is useful to estimate the elements of the matrices C_m, E_m and F_m. For $k = 0$, recalling that $\widetilde{v}_0 = {}^{b}/_{\|b\|}$, (2.5.6) follows directly from the DPC. For $k \geq 1$ we can write

$$\begin{aligned} u_j^T \widetilde{v}_k &= \underbrace{\frac{\|A^{k-1}b\|}{\|A^k b\|}}_{\gamma_k} u_j^T A \underbrace{\frac{A^{k-1}b}{\|A^{k-1}b\|}}_{\widetilde{v}_{k-1}} \\ &= \gamma_k e_j^T U^T A \widetilde{v}_{k-1} = \gamma_k e_j^T \Sigma V^T \widetilde{v}_{k-1} = \gamma_k \sigma_j v_j^T \widetilde{v}_k = O(\sigma_j). \end{aligned}$$

Equality (2.5.6) implies that

$$C_m = U_m^T \widetilde{V}_m = \begin{pmatrix} O(\sigma_1) & \cdots & O(\sigma_1) \\ \vdots & & \vdots \\ O(\sigma_m) & \cdots & O(\sigma_m) \end{pmatrix} \in \mathbb{R}^{m \times m} \tag{2.5.7}$$

and

$$\begin{aligned} (U_m^{\perp})^T \widetilde{V}_m &= \begin{pmatrix} O(\sigma_{m+1}) & \cdots & O(\sigma_{m+1}) \\ \vdots & & \vdots \\ O(\sigma_n) & \cdots & O(\sigma_n) \end{pmatrix} \\ &= \begin{pmatrix} k_{1,1}\sigma_{m+1} & \cdots & k_{1,m}\sigma_{m+1} \\ \vdots & & \vdots \\ k_{n-m,1}\sigma_n & \cdots & k_{n-m,m}\sigma_n \end{pmatrix} \in \mathbb{R}^{(n-m)\times m}. \end{aligned} \tag{2.5.8}$$

We show that the matrix C_m is nonsingular. Considering the QR factorization of the matrix $\widetilde{V}_m$, we can write

$$C_m = U_m^T W_m S_m, \tag{2.5.9}$$

where $W_m \in \mathbb{R}^{n \times m}$ has orthonormal columns and $S_m \in \mathbb{R}^{m \times m}$ is upper triangular and nonsingular, since $\widetilde{V}_m$ has full rank (we remark that the choice of denoting the orthogonal rectangular matrix of the QR decomposition by W_m is not casual since, thanks to the definition of $\widetilde{V}_m$, it coincides with the matrix appearing in (2.2.3)). Exploiting a well-known relation that is derived when estimating the distance between two subspaces (cf. [42, §2.6.3]), we can write

$$\left(\sigma_m(U_m^T W_m)\right)^2 = 1 - \left\| \left(U_m^{\perp}\right)^T W_m \right\|^2.$$

We can state that the term $\left\| \left(U_m^{\perp}\right)^T W_m \right\|$ (which expresses the distance between $\mathcal{R}(U_m)$ and $\mathcal{R}(W_m)$) is strictly less than 1 thanks to the discrete Picard condition: indeed, since w_i, $i = 1, \ldots, m$ is a linear combination of the vectors $\widetilde{v}_{k-1}$, $k = 1, \ldots, i$, each element of the matrix $(U_m^{\perp})^T W_m$ is at most $O(\sigma_{m+1})$ (recall relation (2.5.6)), and therefore infinitesimal (thanks to the rapid decrease of the singular values of A). We can conclude that the minimum singular value of $U_m^T W_m$ is greater than zero and the matrix C_m is nonsingular. To estimate $\|E_m\|$, we write

$$\|E_m\| = \max_{\|z\|=1} \left\| (U_m^{\perp})^T \widetilde{V}_m z \right\| = \left\| (U_m^{\perp})^T \widetilde{V}_m \hat{z} \right\| \leq \bar{k} \left\| (\sigma_{m+1}, \ldots, \sigma_n)^T \right\|,$$

where, referring to (2.5.8), we have set $\bar{k} = \max_{j=1,\dots,n-m} \sum_{i=1}^{m} k_{j,i}\hat{z}_i$. Using the hypothesis $\sigma_j = \gamma_j e^{-\alpha j}$ and the integral criterion for the convergence of a series, we get

$$\begin{aligned} \left\| (\sigma_{m+1}, \dots, \sigma_n)^T \right\| &\leq \bar{\gamma} \left(\sum_{j=m+1}^{n} e^{-2j\alpha} \right)^{1/2} \\ &\leq \bar{\gamma} \frac{1}{\sqrt{2\alpha}} e^{-m\alpha} = O(\sigma_m), \end{aligned} \tag{2.5.10}$$

where we have defined $\bar{\gamma} = \max_{j=m+1,\dots,n}\{\gamma_j\}$. This concludes the proof of the equality (2.5.4). We now focus on the second equality of the theorem statement. By simply inverting the matrix C_m in (2.5.4), we obtain

$$U_m = \widetilde{V}_m C_m^{-1} - \underbrace{E_m C_m^{-1}}_{-F_m},$$

which is formally relation (2.5.5). To conclude, we have to estimate $\|F_m \Sigma_m\|$. Recalling the definition of E_m and C_m, we have that

$$F_m \Sigma_m = -U_m^{\perp} (U_m^{\perp})^T \widetilde{V}_m (U_m^T \widetilde{V}_m)^{-1} \Sigma_m. \tag{2.5.11}$$

Using Cramer's rule to invert the matrix $(U_m^T \widetilde{V}_m)^{-1}$, we find that the elements of $(U_m^T \widetilde{V}_m)^{-1}\Sigma_m$ are of the form $O(1)$. Considering the norm of (2.5.11), recalling that the columns of $U_m^{\perp}$ are orthogonal, and using the relation (2.5.8), we can write

$$\|F_m \Sigma_m\| \leq m \left\| \begin{pmatrix} O(\sigma_{m+1}) & \cdots & O(\sigma_{m+1}) \\ \vdots & & \vdots \\ O(\sigma_n) & \cdots & O(\sigma_n) \end{pmatrix} \right\|.$$

Exploiting again the fact that $\sigma_j = O(e^{-\alpha j})$ and using the integral criterion to bound the second factor on the right-hand side of the above inequality, we get the desired estimate. □

The above theorem expresses a relation between the spaces $\mathcal{R}(U_m)$ and $\mathcal{K}_m(A, b)$: more precisely, it gives an estimate of the distance between them. We can explain the meaning of this theorem by saying that, neglecting a small quantity of order σ_m, the first m left singular values span the Krylov space $\mathcal{K}_m(A, b)$ (and viceversa); therefore, relations (2.5.4) and (2.5.5) express an approximated change of basis for the subspace $\mathcal{R}(U_m)$ (or, equivalently, $\mathcal{K}_m(A, b)$).

Remark 5. The hypothesis $\sigma_j = O(e^{-\alpha j})$ apparently limits the above results to severely ill-conditioned problems. Actually, this assumption is just employed when deriving a bound for $\|E_m\|$ and $\|F_m \Sigma_m\|$ by the integral criterion. Therefore, the results can be extended to moderately ill-conditioned problems having $\sigma_j = O(j^{-\alpha})$ (cf. the classification given in Section 1.1.1). In this situation, instead of (2.5.10), we would obtain

$$\left(\sum_{j=m+1}^{n} \sigma_j^2 \right)^{1/2} = O(m^{1-\alpha}),$$

so that, provided that we have replaced σ_m with $O(m^{1-\alpha})$, (2.5.4) and (2.5.5) are still valid. Indeed, for α sufficiently large, $m^{1-\alpha} \simeq m^{-\alpha}$ and therefore the relations of Theorem 4 are approximately unchanged.

Directly from the above theorem we have the following

Theorem 6. *Assume that A has full rank with singular values of the form $\sigma_j = O(e^{-\alpha j})$ $(\alpha > 0)$ and that b satisfies the discrete Picard condition, that is, $\left|u_j^T b\right| = O(\sigma_j)$, where u_j is the j-th left singular vectors. Then, if b is the starting vector of the Arnoldi process, we obtain*

$$h_{m+1,m} = O\left(m\,\sigma_m\right). \tag{2.5.12}$$

Proof. First of all we recall that the best rank-m approximation A_m of the matrix A is obtained by taking $A_m = U_m \Sigma_m V_m^T$ and that $\|A - A_m\| = \|\Delta_m\| = \sigma_{m+1}$ (cf. relation (1.1.11) and [42, §2.5.5]). By relation (2.2.3), we get

$$\begin{aligned} h_{m+1,m} &= w_{m+1}^T A w_m = w_{m+1}^T \Delta_m w_m + w_{m+1}^T A_m w_m \\ &\leq \sigma_{m+1} + w_{m+1}^T U_m \Sigma_m V_m^T w_m, \end{aligned}$$

where we have used the Cauchy−Schwarz inequality and the definition of A_m. Using (2.5.5) we obtain

$$h_{m+1,m} \leq \sigma_{m+1} + w_{m+1}^T (\widetilde{V}_m C_m^{-1} + F_m) \Sigma_m V_m^T w_m.$$

In the second term of the above relation, $w_{m+1}^T \widetilde{V}_m = 0$ (recall the properties of the QR decomposition (2.5.9)) and $\|F_m \Sigma_m\| = O(m\sigma_m)$. The thesis follows thanks to the fast decrease of the singular values of A. □

In general, it is well-known that the convergence of the Arnoldi-based iterative methods depends on the decay of the sequence $\{h_{m+1,m}\}_{m\geq 1}$. Thanks to the study performed in this section, the behavior of the scalars $h_{m+1,m}$'s is related to the singular values σ_{m+1}'s and therefore, for ill-posed problems, it is natural to expect a very fast convergence. Figure 2.5.1 displays the behavior of the sequences $\{\sigma_m\}_{m\geq 1}$ and $\{h_{m+1,m}\}_{m\geq 1}$ for four test problems. Just to have a

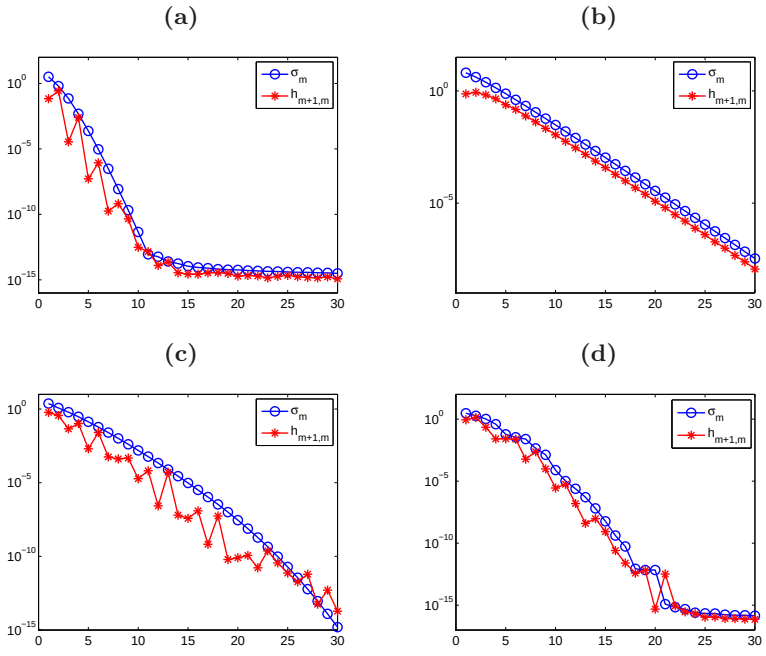

Figure 2.5.1: Values of the quantities σ_m's and $h_{m+1,m}$'s versus m, $1 \leq m \leq 30$. The size of each problem is 100 and the results relative to the test problems `baart` **(a)**, `gravity` **(b)**, `i_laplace` **(c)**, `shaw` **(d)** are displayed.

further explanation of the link between the quantities $h_{m+1,m}$'s and the convergence of Krylov subspace methods, we report some results that are collected in [95]. A theorem derived in [93, Theorem 5.8.10] states that, if we assume n to be arbitrarily large and the singular values to

be p summable (i.e., $\sum_{j\geq 1}\sigma_j^p < \infty$, $0 < p \leq 1$), then

$$\|p_m(A)\| \leq \left(\frac{\eta e}{m}\right)^{m/p}, \tag{2.5.13}$$

where $\eta \leq (1+p)\sum_{j\geq 1}\sigma_j^p$ and $p_m(A)$ is the characteristic polynomial of A. The link between the above equation and the quantities $h_{m+1,m}$'s is given by the following inequality, derived in [84]:

$$\prod_{i=1}^{m} h_{i+1,i} \leq \|p_m(A)b\|. \tag{2.5.14}$$

Assuming to deal with a severely ill-posed problem ($\sigma_j = O(e^{-\alpha j})$, $\alpha > 0$) and combining the estimates (2.5.13) and (2.5.14), the following bound is derived:

$$\left(\prod_{i=1}^{m} h_{i+1,i}\right)^{1/m} \leq \hat{k} e^{-\frac{m\alpha}{e^2}+\frac{\alpha+2}{2}+O\left(\frac{1}{m}\right)}, \tag{2.5.15}$$

where $\hat{k}$ is a constant independent of m. However, comparing the above estimate and the one proved in Theorem 6, we realize that the latter is more precise and describes a faster decay of the $h_{m+1,m}$'s.

2.5.2 Convergence analysis for the GMRES method

In this section we specifically focus on analyzing the convergence of the residual of the GMRES method. Thanks to the well-known optimality property of the GMRES method (recalled in Section 2.3.2), we can immediately state that

$$\|r_m\| \leq \|\rho_m\| = h_{m+1,m}\left|e_m^T H_m^{-1} c\right|, \tag{2.5.16}$$

where we used the relation (2.3.2). More precisely, the link between the GMRES and the FOM residuals is better understood looking at the following relation

$$\|r_m\|^2 = \frac{1}{1/\|\rho_m\|^2 + 1/\|r_{m-1}\|^2},$$

which expresses the so-called "peak-plateau" phenomenon. Three possible scenarios are described by the above formula:

- if the FOM residual converges to zero, then the GMRES residual converges to zero, too (this is also a trivial consequence of the optimality property of the GMRES);
- if the FOM residual converges to some quantity $\hat{c}$, then the GMRES residual is reduced of a factor $\hat{c}^2/\hat{c}^2+\|r_{m-1}\|^2$;
- if the FOM residual diverges, then the GMRES residual stagnates.

Looking for some bounds for the GMRES residual, first of all we analyze systems having an exact right-hand side and then we extend our considerations to systems with a corrupted right-hand side. As a matter of notation, we denote the quantities related to the uncorrupted problem by the superscript "ex".

GMRES for the unperturbed problem

In this case, the behavior of the norm of the GMRES residual can be fully described by the following estimate, which follows directly from (2.5.16) and Theorem 6.

Corollary 7. *Under the hypothesis of Theorem 6, the GMRES residuals are of the form*

$$\|r_m^{ex}\| = O\left(m\,\sigma_m\right). \tag{2.5.17}$$

Proof. Let $M > 0$ be a constant such that, $\forall\, m \leq n$,

$$\left|e_m^T H_m^{-1} c\right| \leq M. \tag{2.5.18}$$

Such a constant M exists since, if we express the quantity $H_m^{-1}c$ by means of the SVD of H_m, we obtain

$$H_m^{-1}c = V_m^{(m)}(\Sigma_m^{(m)})^{-1}(U_m^{(m)})^T c = \sum_{i=1}^{m} \frac{(u_i^{(m)})^T c}{\sigma_i^{(m)}} v_i^{(m)}. \tag{2.5.19}$$

The sum (2.5.19) is finite thanks to the interlacing property of the singular values. Indeed, considering the singular values $\bar{\sigma}_i^{(m)}$'s and $\bar{\sigma}_i^{(m+1)}$'s of $\bar{H}_m$ and $\bar{H}_{m+1}$, respectively, we have (cf.[19] and [42, §8.6])

$$\sigma_1 \geq \bar{\sigma}_1^{(m+1)} \geq \bar{\sigma}_1^{(m)} \geq \bar{\sigma}_2^{(m+1)} \geq \cdots \geq \bar{\sigma}_m^{(m)} \geq \bar{\sigma}_{m+1}^{(m+1)} \geq \sigma_n > 0.$$

After just a few steps of the Arnoldi algorithm, the singular values $\sigma_i^{(m)}$'s and $\sigma_i^{(m+1)}$'s of the matrices H_m and H_{m+1} approximate the singular values of $\bar{H}_m$ and $\bar{H}_{m+1}$, thanks to the fast decay of the quantities $h_{m+1,m}$'s (Theorem 6). This allows us to conclude that none of the denominators in (2.5.19) is zero. The thesis follows exploiting the bound (2.5.18) and the estimate (2.5.12) in (2.5.16). □

Remark 8. Of course, the constant M in (2.5.18) could be uselessly huge, although finite (analogously to what happens for the full-dimensional problem, cf. Section 1.1.4). The only way to guarantee a small M is to assume that the discrete Picard condition is inherited by the projected system $H_m y = c$. Since we assumed the DPC for the original system, the behavior of (2.5.19) depends on how the SVD of the projected matrix H_m approximate the SVD of the full-dimensional matrix A: this issue is analyzed in Section 2.5.3. The tests performed so far seem to suggest that the DPC is inherited by the projected problem; we experimentally prove this in Figures 2.5.2 and 2.5.3.

GMRES for the perturbed problem

When the right-hand side of $Ax = b$ is affected by noise, we can give the following preliminary estimate for the norm of the GMRES residual. Of course, in this setting, we assume that the noise-free right-hand side b^{ex} satisfies the DPC.

Proposition 9. *Let $b = b^{ex} + e$ and let $r_m^{ex} = p_m^{ex}(A)b^{ex}$ be the residual of the GMRES applied to the system $Ax = b^{ex}$. Assume that for $m \geq m^*$, $\|p_m^{ex}(A)\| \leq \eta^*$. Then the m-th residual of the GMRES applied to $Ax = b$ satisfies*

$$\|r_m\| \leq \hat{\eta}\|e\|, \tag{2.5.20}$$

where

$$\hat{\eta} = \frac{\|r_{m^*}^{ex}\|}{\|e\|} + \eta^*.$$

Proof. Exploiting the optimality property of the GMRES method (2.3.7) we can write

$$\|r_m\| = \min_{p_m(0)=1} \|p_m(A)b\| \leq \|p_m^{ex}(A)b\|\,;$$

splitting the right-hand side as $b^{ex} + e$ and using the assumption, we obtain

$$\|r_m\| \leq \|p_m^{ex}(A)b^{ex}\| + \|p_m^{ex}(A)\| \cdot \|e\| \leq \|r_m^{ex}\| + \eta^*\|e\|.$$

Since $\|r_m^{ex}\| \leq \|r_{m^*}^{ex}\|$ for $m \geq m^*$, the above inequality implies

$$\|r_m\| \leq \|r_m^{ex}\| + \eta^*\|e\| = \hat{\eta}\|e\|, \quad \forall m \geq m^*.$$

□

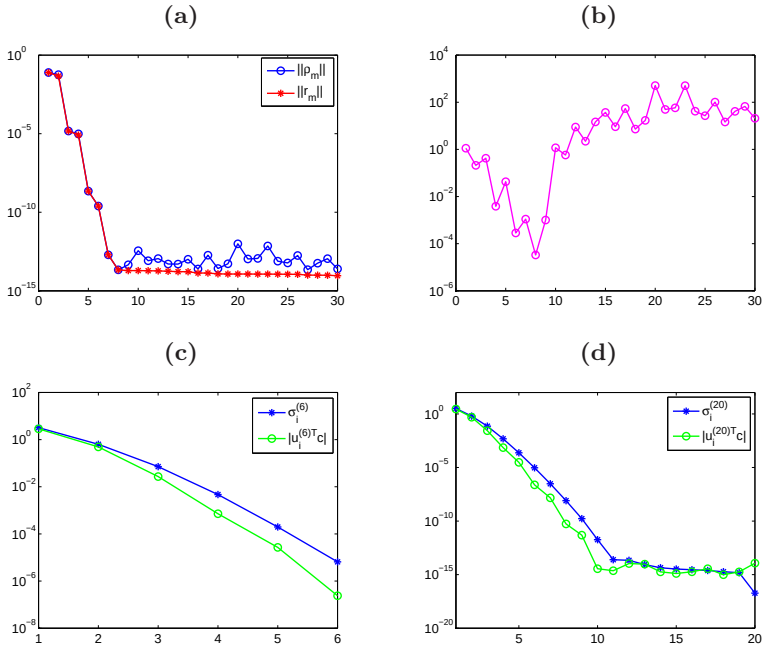

Figure 2.5.2: Test problem `baart` of size 100. Frame **(a)**: norm of the FOM and GMRES residuals versus the number of iterations. Frame **(b)**: logarithmic plot of the quantities $\left|e_m^T H_m^{-1} c\right|$ versus the number of iterations m. Frame **(c)**: plot of the Fourier coefficients $\left|(u_i^{(6)})^T c\right|$ and $\sigma_i^{(6)}$ versus the index i for the projected problem of size 6. Frame **(d)**: plot of the Fourier coefficients and of the singular values relative to the projected problem of size 20, versus the index i.

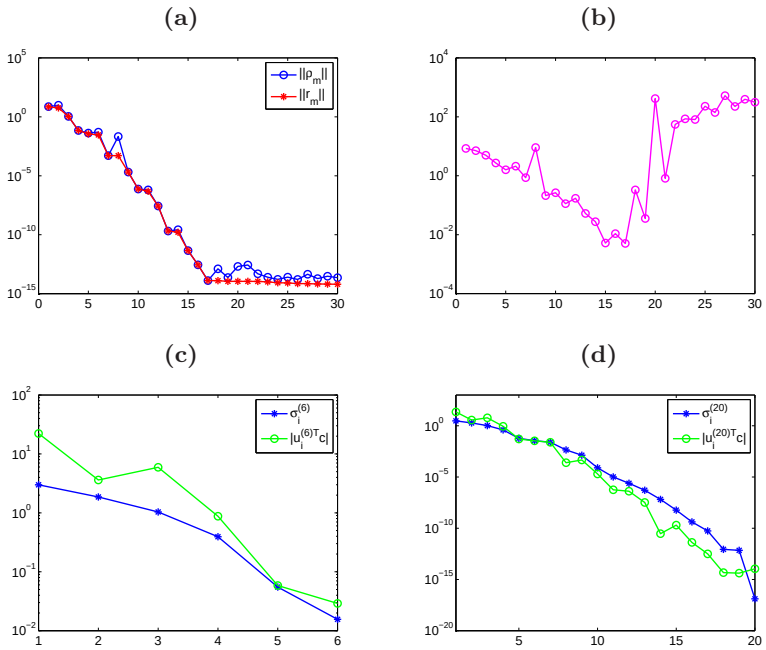

Figure 2.5.3: Test problem `shaw` of size 100; the description of each frame is analogous to the one of Figure 2.5.2.

We remark that, even if the assumption $\|p_m^{ex}(A)\| \leq \eta^*$ could seem limiting, it is commonly used when describing the behavior of the GMRES method (cf. [18]).

In the remaining part of this section, we try to give some additional information about the value of the constant $\hat{\eta}$ of Proposition 9. Looking at the above proof, it is clear that $\hat{\eta}$ depends on the scalar m, i.e., on the number of the performed iterations. In the following proposition, instead of looking at the polynomials associated with the GMRES, we consider some matrices whose columns span the involved Krylov subspaces. In particular, analogously to Theorem 4, we define

$$\widetilde{V}_m := \left[\frac{b}{\|b\|}, \frac{Ab}{\|Ab\|}, \ldots, \frac{A^{m-1}b}{\|A^{m-1}b\|}\right], \; \widetilde{V}_m^{ex} := \left[\frac{b^{ex}}{\|b^{ex}\|}, \frac{Ab^{ex}}{\|Ab^{ex}\|}, \ldots, \frac{A^{m-1}b^{ex}}{\|A^{m-1}b^{ex}\|}\right].$$

According to the above notations we can write

$$\|r_m\| = \min_{s \in \mathbb{R}^{m+1},\, s_1 = 0} \left\| b - \widetilde{V}_{m+1} s \right\|,$$

where s_1 is the first component of vector s. We prove the following

Proposition 10. *At the m-th step of the Arnoldi algorithm, the GMRES residual satisfies*

$$\|r_m\| \leq \hat{\eta}(m)\|e\|,$$

where

$$\hat{\eta}(m) = 1 + \frac{\|r_m^{ex}\| + \left\| \left(\widetilde{V}_{m+1} - \widetilde{V}_{m+1}^{ex} \right) s^{ex} \right\|}{\|e\|}, \tag{2.5.21}$$

where s^{ex} ($s_1^{ex} = 0$) is such that $\|r_m^{ex}\| = \left\| b^{ex} - \widetilde{V}_{m+1}^{ex} s^{ex} \right\|$.

Proof. Using the above definitions, we can quite straightforwardly derive the following inequalities

$$\begin{aligned} \|r_m\| &= \min_{s \in \mathbb{R}^{m+1}, s_1 = 0} \left\| b - \widetilde{V}_{m+1} s \right\| \leq \left\| b - \widetilde{V}_{m+1} s^{ex} \right\| \\ &= \left\| b^{ex} + e - \widetilde{V}_{m+1} s^{ex} + \widetilde{V}_{m+1}^{ex} s^{ex} - \widetilde{V}_{m+1}^{ex} s^{ex} \right\| \\ &\leq \|r_m^{ex}\| + \|e\| + \left\| \left(\widetilde{V}_{m+1}^{ex} - \widetilde{V}_{m+1} \right) s^{ex} \right\|, \end{aligned}$$

which prove our thesis. □

Remark 11. The fast decay of the singular values of A ensures that, for $k \geq 1$,

$$\left\| \frac{A^k b}{\|A^k b\|} - \frac{A^k b^{ex}}{\|A^k b^{ex}\|} \right\| \ll \|e\|; \tag{2.5.22}$$

indeed, the above condition is also the basic assumption behind the so-called range-restricted approach for Krylov type methods (see [16, 30, 79]). We also note that relation (2.5.22) can be interpreted as the discrete analogous of the Riemann-Lebesgue Lemma (Section 1.1.1), as far as the noise e does not involve low frequencies. We give an example of this behavior in Figure 2.5.4. For $k = 0$, there are no guarantees that the difference (2.5.22) is infinitesimal, but this does not affect the term $\|(\widetilde{V}_{m+1} - \widetilde{V}_{m+1}^{ex}) s^{ex}\| / \|e\|$ of (2.5.21), since $s_1^{ex} = 0$. This remark allows us to state that, whenever $\|r_m^{ex}\| \approx 0$, we have $\hat{\eta}(m) \approx 1$ in (2.5.21). In Figure 2.5.5 we prove experimentally the estimate that we have just derived, i.e., $\|r_m\| \approx \|e\|$.

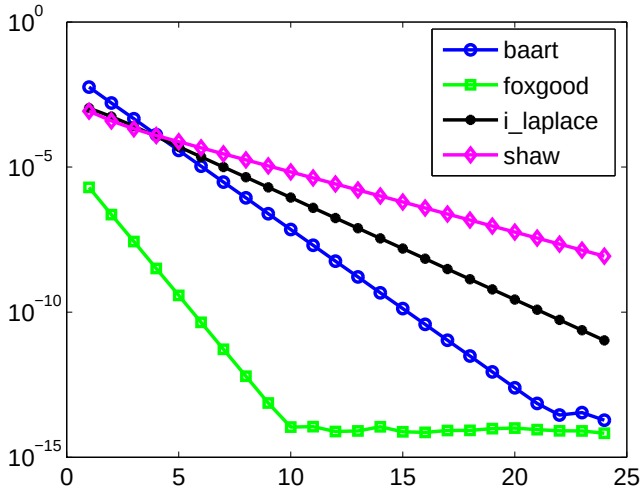

Figure 2.5.4: Decay of the quantities (2.5.22) versus the value of the exponent k for some test problems taken from [54]; the right-hand side b is affected by 1% white Gaussian noise.

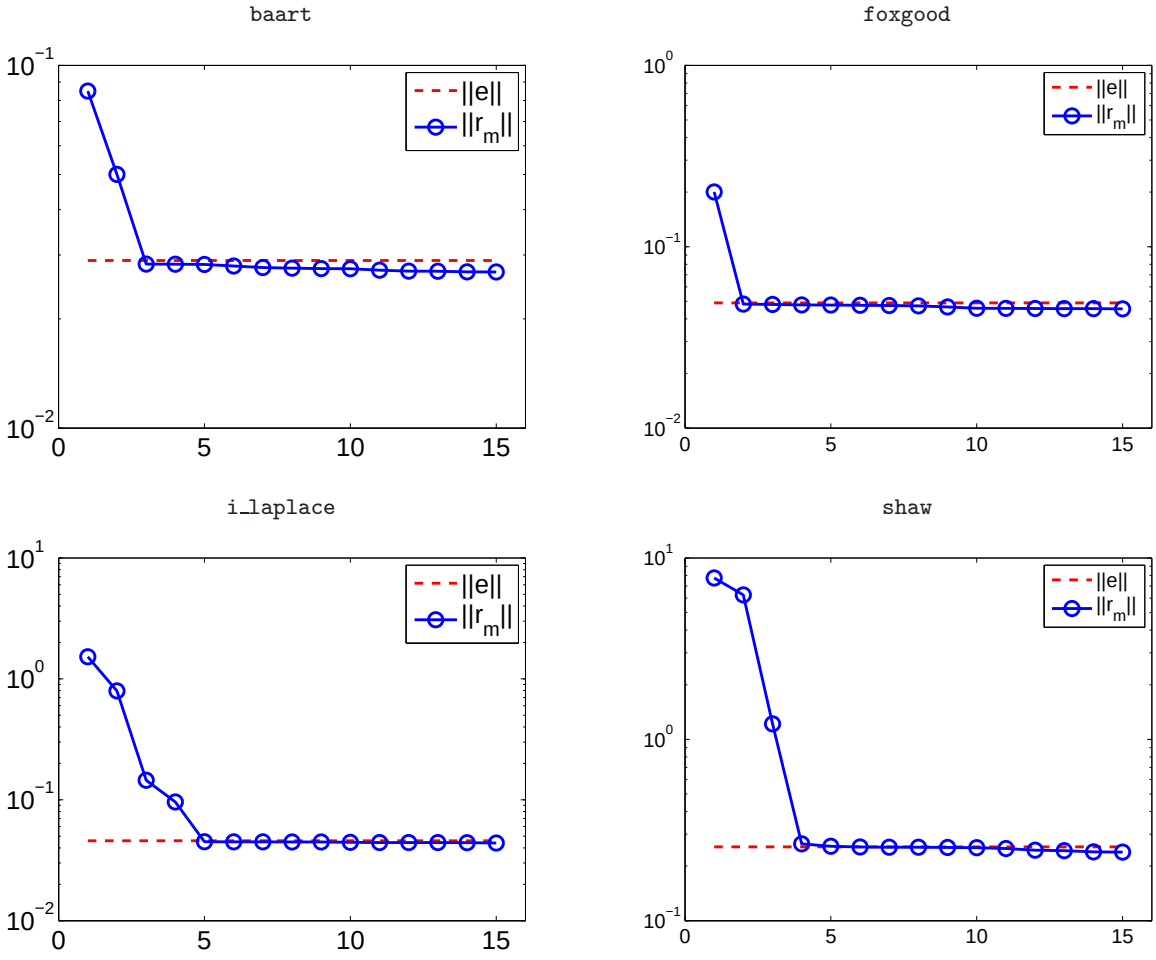

Figure 2.5.5: GMRES residual history when the right-hand side is affected by 1% noise.

2.5.3 Approximation of the SVD

The main methods studied in this thesis perform very well when the projected problem can quickly reproduce the basic features of the original, full-dimensional one (cf. Chapter 3). Since, as extensively seen in the previous chapter, a complete description of the ill-posedness of the discrete problem can be achieved by considering the SVD of the matrix A, it is natural to investigate how the SVDs of the matrices H_m in (2.2.2) and $\bar{H}_m$ in (2.2.3) approximate the SVD of the matrix A. Basically, we are looking for an approximation of the SVD of A belonging to the involved Krylov subspaces.

The use of the Arnoldi algorithm as a method to approximate the marginal eigenvalues of the spectrum of a matrix is widely known in literature. We may refer to [113, Chapter 6] for an exhaustive background. Using similar arguments, in this section we analyze the convergence of the singular values of the matrices $\bar{H}_m$ to the largest singular values of A. For the Lanczos bidiagonalization method [10, 96], the analysis can be done by exploiting the connection between this method and the symmetric Lanczos process (see, e.g., [43]). The results here proposed are based on the analysis performed in [95]. They are organized in this way: first of all we consider the matrix $\bar{H}_m$ and we derive a couple of relations involving the matrix A and its transpose; then we consider two equalities linking the approximate singular values and vectors; finally we generalize some of the previous relations considering the matrix H_m.

Proposition 12. *After performing m steps of the Arnoldi algorithm, the following equality holds:*

$$\left\| A - \hat{\bar{U}}^{(m)} \bar{\Sigma}^{(m)} (\hat{\bar{V}}^{(m)})^T \right\| = \left\| A(I - W_m W_m^T) \right\|. \tag{2.5.23}$$

Proof. The result follows directly by decomposition (2.2.3), which expresses the Arnoldi algorithm in matrix form:

$$\begin{aligned} A - \hat{\bar{U}}^{(m)} \bar{\Sigma}^{(m)} (\hat{\bar{V}}^{(m)})^T &= A - W_{m+1} \bar{U}^{(m)} \bar{\Sigma}^{(m)} (\bar{V}^{(m)})^T W_m^T \\ &= A - W_{m+1} \bar{H}_m W_m^T \\ &= A - A W_m W_m^T = A(I - W_m W_m^T). \end{aligned}$$

□

An immediate consequence of the above relation is that the triplet

$$\left(W_{m+1} \bar{U}^{(m)}_{[:,1:m]}, \bar{\Sigma}^{(m)}_{[1:m,:]}, W_m \bar{V}^{(m)} \right),$$

where the rank-m matrices are detailed using a MATLAB-like notation, defines an approximation of the (rank-m) TSVD of A, which cannot be too bad since

$$\left\| A(I - W_m W_m^T) \right\| \leq \|A\|.$$

Moreover, if the Arnoldi algorithm does not terminate before n iterations, then $\hat{\bar{U}}^{(n)} \bar{\Sigma}^{(n)} (\hat{\bar{V}}^{(n)})^T$ coincides with the SVD of A.

In the following proposition we give an alternative estimate for the quantities in (2.5.23).

Proposition 13. *After performing m steps of the Arnoldi algorithm with starting vector b^{ex}, the following inequality holds:*

$$\left\| A - \hat{\bar{U}}^{(m)} \bar{\Sigma}^{(m)} (\hat{\bar{V}}^{(m)})^T \right\| \leq \left\| W_{m+1}^T A W_m^{\perp} \right\| + \left\| (W_{m+1}^{\perp})^T A W_m^{\perp} \right\|. \tag{2.5.24}$$

Proof. Also in this case, the proof is straightforward. Starting from relation (2.5.23) and using the decomposition $I = W_{m+1}W_{m+1}^T + W_{m+1}^{\perp}(W_{m+1}^{\perp})^T$, we can write the following relations

$$\begin{aligned}\left\|A - \hat{\bar{U}}^{(m)}\bar{\Sigma}^{(m)}(\hat{\bar{V}}^{(m)})^T\right\| &= \left\|AW_m^{\perp}(W_m^{\perp})^T\right\| \\ &\leq \left\|W_{m+1}W_{m+1}^T AW_m^{\perp}\right\| + \left\|W_{m+1}^{\perp}(W_{m+1}^{\perp})^T AW_m^{\perp}\right\| \\ &= \left\|W_{m+1}^T AW_m^{\perp}\right\| + \left\|(W_{m+1}^{\perp})^T AW_m^{\perp}\right\|.\end{aligned}$$

□

Considering both A and A^T, we can state the following

Proposition 14. *After performing m steps of the Arnoldi algorithm with starting vector b^{ex}, the following relations hold:*

$$\left\|A\hat{\bar{V}}^{(m)} - \hat{\bar{U}}^{(m)}\bar{\Sigma}^{(m)}\right\| = 0\,; \tag{2.5.25}$$

$$\left\|A^T\hat{\bar{U}}^{(m)} - \hat{\bar{V}}^{(m)}(\bar{\Sigma}^{(m)})^T\right\| \leq \left\|W_{m+1}^T AW_m^{\perp}\right\|. \tag{2.5.26}$$

Proof. Equality (2.5.25) follows directly from (2.2.3) and (2.5.3):

$$\begin{aligned}\left\|A\hat{\bar{V}}^{(m)} - \hat{\bar{U}}^{(m)}\bar{\Sigma}^{(m)}\right\| &= \left\|W_{m+1}\bar{H}_m\bar{V}^{(m)} - W_{m+1}\bar{U}^{(m)}\bar{\Sigma}^{(m)}\right\| \\ &= \left\|\bar{H}_m\bar{V}^{(m)} - \bar{U}^{(m)}\bar{\Sigma}^{(m)}\right\| = 0\,.\end{aligned}$$

Inequality (2.5.26) can be proved exploiting some relations linked to (2.2.3) and the equality $\hat{\bar{U}}^{(m)}(\hat{\bar{U}}^{(m)})^T = W_{m+1}W_{m+1}^T$ (following directly from the definition of $\hat{\bar{U}}^{(m)}$):

$$\begin{aligned}\left\|A^T\hat{\bar{U}}^{(m)} - \hat{\bar{V}}^{(m)}(\bar{\Sigma}^{(m)})^T\right\| &= \left\|\hat{\bar{U}}^{(m)}(\hat{\bar{U}}^{(m)})^T A - \hat{\bar{U}}^{(m)}\bar{\Sigma}_m^{(m)}(\hat{\bar{V}}^{(m)})^T\right\| \\ &= \left\|W_{m+1}W_{m+1}^T A - W_{m+1}\bar{H}_m W_m^T\right\| \\ &= \left\|W_{m+1}^T A(W_m W_m^T + W_m^{\perp}(W_m^{\perp})^T) - \bar{H}_m W_m^T\right\| \\ &\leq \left\|W_{m+1}^T AW_m - \bar{H}_m\right\| + \left\|W_{m+1}^T AW_m^{\perp}\right\| \\ &= \left\|W_{m+1}^T AW_m^{\perp}\right\|.\end{aligned}$$

□

In practice, when considering some of the most common test problems, we observe that the Arnoldi algorithm seems to be very efficient for approximating the singular values of A, especially the largest ones (see Figure 2.5.6). Looking at both inequalities (2.5.24) and (2.5.26), we can see that these a-posteriori estimates involve the term $\left\|W_{m+1}^T AW_m^{\perp}\right\|$: directly from the Arnoldi decomposition (2.2.3), we obtain that $h_{i,j} = w_i^T Aw_j$, and therefore

$$W_{m+1}^T AW_m^{\perp} = \begin{bmatrix} h_{1,m+1} & \cdots & h_{1,n} \\ \vdots & & \vdots \\ h_{m+1,m+1} & \cdots & h_{m+1,n} \end{bmatrix}.$$

Since, in many cases, the elements of the projected matrix $\bar{H}_m$ tend to annihilate departing from the diagonal (this is the fundamental assumption underlying the methods based on incomplete orthogonalization, see e.g. [115, Chapter 6]), one may obtain useful approximations of the bounds (2.5.24) and (2.5.26) working with just a few columns of $W_{m+1}^T AW_m^{\perp}$, i.e., with a few

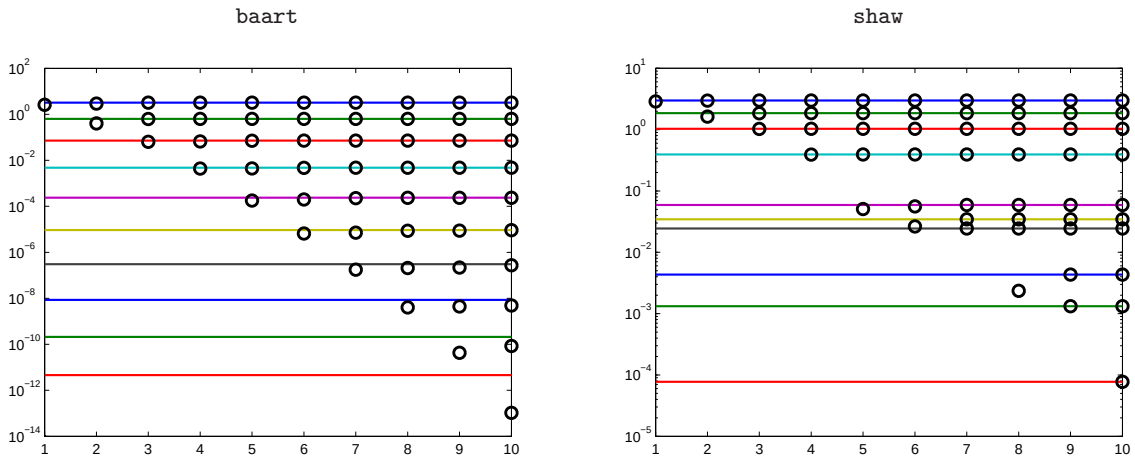

Figure 2.5.6: Approximation of the singular values by means of the Arnoldi algorithm (Householder version). The multi-color horizontal lines stand for the first 10 singular values of the original matrix A; the black dots corresponding to the i-th tick on the horizontal axis indicate the singular values of the matrix $\bar{H}_i$, $1 \leq i \leq 10$.

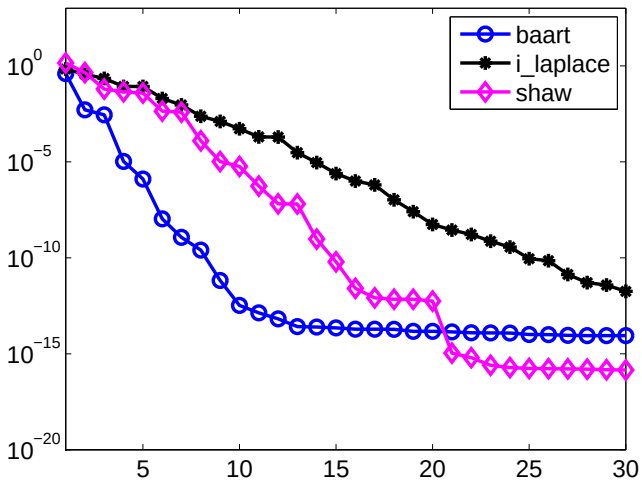

Figure 2.5.7: Decay of the quantities $\left\|(W_{m+1}^{\perp})^T A W_m^{\perp}\right\|$ versus the number of iterations m.

columns of $W_m^\perp$; in particular, regarding (2.5.24), this implies that the term $\left\|(W_{m+1}^\perp)^T A W_m^\perp\right\|$ is negligible (cf. Figure 2.5.7). In this way we can still derive two a-posteriori estimates for the quality of the SVD approximation in a much cheaper way (computationally, we need to perform just a few more steps of the Arnoldi algorithm). For instance, a quite accurate bound is obtained by just considering the first column of $W_m^\perp$. In Figure 2.5.8 we give an experimental confirmation of this fact. Since the previous results basically link A and its approximation obtained taking the decomposition $W_{m+1}\bar{H}_m W_m^T$, in the following we derive some vector-wise relations and bounds linking the the singular vectors of A and the singular vectors of $\bar{H}_m$.

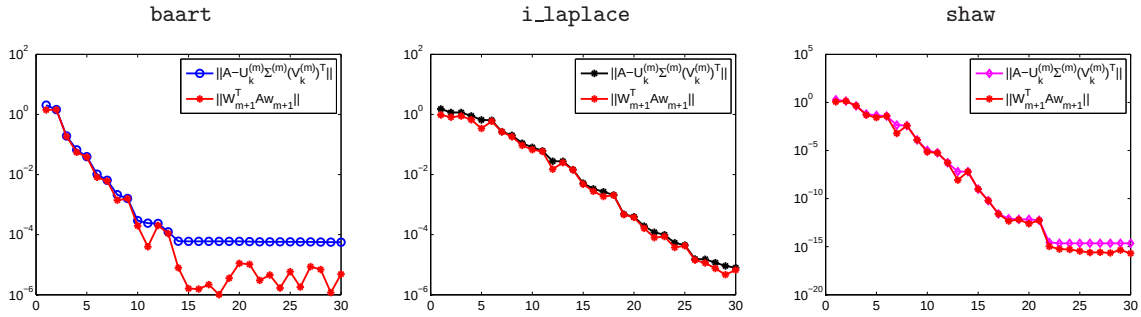

Figure 2.5.8: Examples of some a-posteriori bounds for the estimate (2.5.24). The quantities $\left\|A - \hat{\bar{U}}^{(m)}\bar{\Sigma}^{(m)}(\hat{\bar{V}}^{(m)})^T\right\|$ and $\left\|W_{m+1}^T A w_{m+1}\right\|$ versus the number of iterations m are displayed.

Proposition 15. *Let $\bar{u}_k^{(m)} \in \mathbb{R}^{m+1}$ and $\bar{v}_k^{(m)} \in \mathbb{R}^m$ be respectively the right and left singular vectors relative to the singular value $\bar{\sigma}_k^{(m)}$ of $\bar{H}_m$, i.e.,*

$$\begin{aligned} \bar{H}_m \bar{v}_k^{(m)} &= \bar{\sigma}_k^{(m)} \bar{u}_k^{(m)}, \\ \bar{H}_m^T \bar{u}_k^{(m)} &= \bar{\sigma}_k^{(m)} \bar{v}_k^{(m)}, \end{aligned}$$

with $1 \le k \le m$. Then

$$A\hat{\bar{v}}_k^{(m)} - \bar{\sigma}_k^{(m)}\hat{\bar{u}}_k^{(m)} = 0, \tag{2.5.27}$$

$$W_m^T\left(A^T\hat{\bar{u}}_k^{(m)} - \bar{\sigma}_k^{(m)}\hat{\bar{v}}_k^{(m)}\right) = 0. \tag{2.5.28}$$

Proof. Equality (2.5.27) is the vector-wise reformulation of equality (2.5.25). About equality (2.5.28), thanks to the assumptions and recalling that $\bar{H}_m^T = W_m^T A^T W_{m+1}$ (directly from (2.2.3)), we get

$$W_m^T\left(A^T\hat{\bar{u}}_k^{(m)} - \bar{\sigma}_k^{(m)}\hat{\bar{v}}_k^{(m)}\right) = \bar{H}_m^T\bar{u}_k^{(m)} - \bar{\sigma}_k^{(m)}\bar{v}_k^{(m)} = 0.$$

□

Of course, equality (2.5.27) implies that $A\hat{\bar{V}}^{(m)} - \hat{\bar{U}}^{(m)}\bar{\Sigma}^{(m)} = 0$. The Galerkin condition (2.5.28) is a consequence of the fact that the Arnoldi algorithm does not work with the transpose. Obviously, if $A = A^T$, the algorithm reduces to the symmetric Lanczos process and, under the hypothesis of Proposition 15, we easily obtain $A^T\hat{\bar{u}}_k^{(m)} - \bar{\sigma}_k^{(m)}\hat{\bar{v}}_k^{(m)} = 0$. In the general case of $A \neq A^T$, equality (2.5.28) ensures that, since $\hat{\bar{v}}_k^{(m)} = W_m\bar{v}_k^{(m)} \in \mathcal{K}_m(A, b)$, the vector $\bar{\sigma}_k^{(m)}\hat{\bar{v}}_k^{(m)}$ is just the orthogonal projection of $A^T\hat{\bar{u}}_k^{(m)}$ onto $\mathcal{K}_m(A, b)$, i.e., $\bar{\sigma}_k^{(m)}\hat{\bar{v}}_k^{(m)} = W_m W_m^T A^T\hat{\bar{u}}_k^{(m)}$, which implies

$$\left\|A^T\hat{\bar{u}}_k^{(m)} - \bar{\sigma}_k^{(m)}\hat{\bar{v}}_k^{(m)}\right\| \le \left\|(I - W_m W_m^T)A^T W_{m+1}\right\|. \tag{2.5.29}$$

This means that the approximation of the singular values and vectors is good if $A^T \hat{\hat{u}}_k^{(m)}$ is close to $\mathcal{K}_m(A, b)$. We remark that inequality (2.5.29) agrees with its matrix form version (2.5.26): this is evident since

$$\left\|(I - W_m W_m^T) A^T W_{m+1}\right\| = \left\|W_{m+1}^T A W_m^{\perp}\right\|.$$

Remark 16. So far, we have taken into account just the matrix $\bar{H}_m$ to approximate the SVD of A. We now consider the matrix H_m. In this case, defining $\hat{u}_k^{(m)} = W_m u_k^{(m)}$ and $\hat{v}_k^{(m)} = W_m v_k^{(m)}$ (where $H_m v_k^{(m)} = \sigma^{(m)} u_k^{(m)}$), equality (2.5.27) becomes

$$\left\|A\hat{v}_k^{(m)} - \sigma_k^{(m)} \hat{u}_k^{(m)}\right\| \leq h_{m+1,m}. \tag{2.5.30}$$

The above relation follows directly from (2.2.2), since

$$\begin{aligned}
\left\|A\hat{v}_k^{(m)} - \sigma_k^{(m)} \hat{u}_k^{(m)}\right\| &= \left\|AW_m v_k^{(m)} - \sigma_k^{(m)} W_m u_k^{(m)}\right\| \\
&= \left\|W_m (H_m v_k^{(m)} - \sigma_k^{(m)} u_k^{(m)}) + h_{m+1,m} w_{m+1} e_m^T\right\| \\
&\leq h_{m+1,m} \cdot 1.
\end{aligned}$$

Inequality (2.5.30) is very similar to the one which arises when using the eigenvalues of H_m (i.e., the Ritz values of A) to approximate the eigenvalues of A (cf. [113, §6.2]). Note moreover that, whenever $h_{m+1,m} \approx 0$ (and hence very quickly for linear ill-posed problems, cf. Section 2.5.1), the SVD of $\bar{H}_m$ or the SVD of H_m almost equivalently approximate the largest singular values of A; this agrees with the fact that adding a zero row to a given matrix does not change the singular values (when $h_{m+1,m} \approx 0$, $\bar{H}_m$ is essentially obtained by appending a zero row to H_m). Considering A^T and noting that, directly from relation (2.2.4), $W_m^T A^T W_m = H_m^T$, we obtain the following equality

$$W_m^T (A^T \hat{u}_k^{(m)} - \sigma_k^{(m)} \hat{v}_k^{(m)}) = 0, \tag{2.5.31}$$

which is equivalent to (2.5.28).

2.6 Regularizing properties of Krylov subspace methods

Many works in literature are focussed on proving that iteratively solving a linear discrete ill-posed problem by means of some Krylov subspace method has a regularizing effect. As addressed in Section 1.3.2, the number of iterations of an iterative method acts as a regularization parameter because of the semiconvergent behavior of the sequence of the errors (cf. Figure 2.6.1). From the discussion in the previous chapter, we realize that the optimal basis for analyzing and remedying the ill-posedness of a linear system is provided by the SVD of the matrix A (or by the GSVD of some matrix pair (A, L)). In general, when applying a Krylov subspace method, the basis vectors are of the form $A^i v$, $i \geq 0$, and $v = b$, $v = A^T b$, $v = b - Ax_0$, or $v = A^T(b - Ax_0)$ (depending on the chosen method). Therefore, the available right-hand side b (or some vector linked to it) enters the definition of the solution basis: in this way the Krylov subspace basis is somewhat "adapted" to the specific problem we wish to solve (cf. [47]). In some cases, this property could be very advantageous; for instance, let us consider a problem whose matrix is symmetric and whose exact solution is known to be symmetric: in this case the basis vectors are intrinsically symmetric and this property allows an efficient reconstruction of the solution.

Historically, the CGLS and the LSQR methods have been the first Krylov subspace methods to be employed with regularizing purposes and to be theoretically studied in the framework of ill-posed problems.

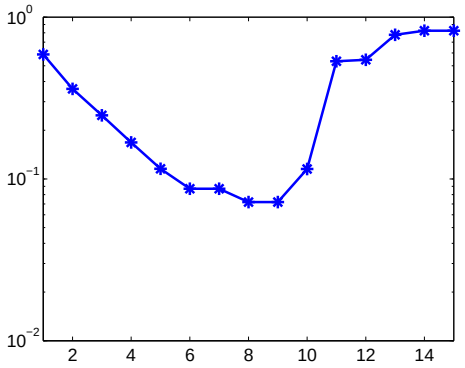

Figure 2.6.1: History of the relative errors for the `shaw` test problem regularized by the CGLS algorithm (Algorithm 7). The noise level on the right-hand side vector is $\widetilde{\varepsilon} = 10^{-2}$ and the semiconvergent behavior of the method is evident: we have convergence to the exact solution till the 9-th iteration, then the noise starts to deteriorate the reconstruction.

When considering the LSQR method, directly from (2.4.9) we recover the following expression for the regularized inverse

$$A_m^\sharp = W_m \bar{B}_m^\dagger V_{m+1}^T. \tag{2.6.1}$$

The singular values of the matrix $\bar{B}_m$ obtained by the Lanczos bidiagonalization algorithm provide some approximation of the singular values of A, even during the early iterations. In particular, it has been proved that the larger the relative gap between two consecutive singular values of A, the better the approximation (cf. [11, 43, 112]). Since, when treating moderately to severely ill-posed problems, the singular values of the involved matrices decay as $O(j^{-\alpha})$ or $\sigma_j = O(e^{-\alpha j})$, $\alpha > 0$ (cf. Section 1.1.1), the relative gap between the first singular values is much bigger than the relative gap between the last singular values: for this reason, the singular values of the associated matrices $\bar{B}_m$'s quickly converge to the largest singular values of A. We can conclude that the LSQR method has an inherent regularizing effect, in that just some approximations of the largest singular values of the original system matrix A enter the reconstruction process.

When dealing with the CGLS method, the considerations just made in the LSQR case still hold (remember that CGLS and LSQR are mathematically equivalent, cf. Section 2.4). Many different investigations of the regularization properties of the CGLS method have already been performed: in this section we just underline some remarkable aspects and we refer to [55] and the references therein for more details. Recalling that, in the CGLS case, the Krylov subspace is defined taking as initial vector $A^T b$, and recalling the smoothing properties of the matrix A (cf. Section 1.1.3), we can state that the noise (high-frequencies) components in the available right-hand side b are already partially damped in the starting vector $A^T b$. As a consequence, the original perturbations that corrupt the problems (2.1.1) and (2.4.1) are not fully included into the solution subspace $\mathcal{K}_m(A^T A, A^T b)$. Finally, we report a desirable property that can be proved for the solution and the residuals computed by the CGLS method: the norm of the solutions increases with the number of iterations, while the norm of the residual decreases with the number of iterations, i.e. $\|r_{m+1}\| \leq \|r_m\|$ and $\|x_{m+1}\| \geq \|x_m\|$ for $1 \leq m \leq n$. This property allows to efficiently employ some of the parameter selection strategies described in Chapter 4.

Analyzing the regularizing properties of the GMRES is a difficult task, mainly because of the undesired mixing of the SVD components, i.e., the GMRES cannot be expressed as a spectral filtering method [70]. The regularized inverse associated to the GMRES method is given by

$$A_m^\sharp = W_m \bar{H}_m^\dagger W_{m+1}^T. \tag{2.6.2}$$

Similarly to what has been done in [46] for CG-like methods, in [18] the authors prove that the

GMRES method equipped with a stopping rule based on the discrepancy principle (Section 4.1) is a regularization method, i.e.,

$$\lim_{\varepsilon \to 0} \sup_{\|b^{ex} - b^{\varepsilon}\| \leq \varepsilon} \left\| x^{ex} - x^{\varepsilon}_{m_\varepsilon} \right\| = 0,$$

where, as usual, $\varepsilon \in \mathbb{R}$ is the norm of the noise that affects the corrupted right-hand side; we decided to extensively use the sub/superscript ε to better underline the dependency of the m-th approximate solution $x^{\varepsilon}_{m_\varepsilon}$ as well as the stopping iteration m_ε on the amount of noise that affects the data. Moreover, we believe that the properties derived in Sections 2.5.2 and 2.5.3 help to understand the regularizing properties of the GMRES method. We again remark that, since the available vector b enters the definition of the solution subspace associated to the GMRES method, the original noise on b is spread on the basis vectors of $\mathcal{K}_m(A, b)$.

Modifying the Krylov subspaces:
range-restricted, augmented and preconditioning approaches

If the exact solution of the available system (2.1.1) is known to be smooth, the reconstruction obtained by the usual GMRES could be improved by employing the so-called Range-Restricted GMRES (RRGMRES) method [15, 16, 30]. The solution subspace associated to RRGMRES is the Krylov subspace $\mathcal{K}_m(A, Ab)$: since the starting vector of the Krylov subspace has been multiplied by A we can expect some damping of the noise components and an overall better quality of the reconstructed solution (similarly to the CGLS case). An advantage of the range-restricted methods with respect to the Lanczos-based iterative methods (implemented with reorthogonalization) is that less matrix-vector products are involved in building a basis for the solution subspace; moreover, just matrix-vector products with A are required: this is a clear advantage when dealing with problems for which A^T is not explicitly available, or when matrix-vector products with A can be inexpensively computed but the same is not true for A^T. Recently, a further extension of the concept of range-restricted method has been proposed in [30], where different options for a solution subspaces of the form $\mathcal{K}_m(A, A^\ell b)$, $\ell \geq 0$ are efficiently evaluated and a strategy to decide which space could deliver the most accurate reconstructions is outlined.

On the contrary, for some kind of problems, when dealing with Lanczos-based iterative methods the reconstructed solution would benefit from the inclusion of the available right-hand side b into the definition of the solution subspace. For instance, if the exact solution has some intrinsic irregularities (such as jumps) that are progressively annihilated when a multiplication by A occurs, then the choice of the space $\mathcal{K}_m(A^TA, A^Tb)$ is not very convenient: a solution subspace containing b would probably be more suitable. The possibility of including the vector b into the space $\mathcal{K}_m(A^TA, A^Tb)$ (or, more generally, the possibility of including a particular vector reproducing some known features of the exact solution into a given Krylov subspace) has been investigated in [2, 3, 21] and gives rise to the so-called enriched or augmented Krylov subspace methods.

As already done for general form Tikhonov regularization and the TGSVD method (cf. Section 1.3.1), also in the iterative regularization setting one can provide more accurate reconstructions by involving a problem-dependent regularization matrix L. The basic underlying idea is to formally consider a standard form transformation: in this way the Krylov subspaces involved in the iterative process are computed with respect to the matrix $AL^\dagger_A$ ($L^\dagger_A$ is the A-weighted generalized pseudoinverse defined in (1.3.13)). The matrix $L^\dagger_A$ acts therefore as a right preconditioner and the preconditioned version of all the Krylov subspace methods listed in Sections 2.3 and 2.4 have to be considered. However, we would like to underline that, in this setting, the term "preconditioner" referred to the matrix $L^\dagger_A$ is not fully correct: indeed, in a classical sense, a preconditioner should accelerate the convergence of an iterative method (cf. Section 1.3.2); $L^\dagger_A$ has not such an effect on the iterative process (often it rather slows

the convergence). Therefore it would be more proper to refer to $L_A^\dagger$ as a "regularizer", since its action is to force additional regularization into the reconstructed solution. Efficient ways to incorporate the matrix $L_A^\dagger$ or other matrices related to it in the setting of Krylov subspace method has been carefully analyzed in [22, 61, 85]. In some circumstances, the filtering effect of iterative methods can be enhanced by applying the so-called filtering preconditioners, originally introduced in [50]: the basic idea is to define suitable preconditioners that opportunely act on the singular values of the matrix A. Such preconditioners should be defined in order to cast the largest singular values of A near 1 and keep the smallest singular values of A almost unaltered, creating in this way a well-defined gap between the desired and undesired SVD components of the reconstructed solution. The task of defining a filtering preconditioner is quite easy if the SVD of A is available; however, this is not typically the case when applying an iterative regularization method designed to deal with large-scale problems. When the SVD of A is not available, one should first choose, a priori, two orthogonal matrices that could mimic the singular vectors of A (for instance, one can use the matrices whose columns are the discrete Fourier basis vectors, cf. Section 1.1.3) and then define a suitable filtering with respect to the chosen basis. For more details on this approach we refer to [49, 71, 87].

We refer to [70] for some insight and some comparisons of the performances of the minimum-residual methods so far taken into account (along with some other related ones).

Hybrid methods

Several schemes, called hybrid methods, have been proposed to remedy the semiconvergence behavior that affects iterative methods [10, 25, 47, 96]. Hybrid methods are defined by combining an iterative and a TSVD-like or Tikhonov-like approach to regularization. For instance, when applying the LSQR method, at the m-th iteration one has to solve a projected problem of the form (2.4.9), where the reduced-dimension matrix $\bar{B}_m$ is defined by the Lanczos bidiagonalization algorithm (Algorithm 8). Lanczos-hybrid methods consist in including some extra regularization at each iteration of the LSQR method; for instance, the problem

$$\min_{y\in\mathbb{R}^m} \left\{ \left\| c - \bar{B}_m y \right\|^2 + \lambda_m \left\| y \right\|^2 \right\}, \tag{2.6.3}$$

instead of (2.4.9), is solved at the m-th iteration of the Lanczos algorithm. Denoting by y_{m,λ_m} the solution of (2.6.3), the full-dimensional regularized solution x_{m,λ_m} is recovered by taking $x_{m,\lambda_m} = W_m y_{m,\lambda_m}$. Therefore, the expression of the regularized inverse for Lanczos-hybrid method is

$$A^\sharp_{m,\lambda_m} = W_m(\bar{B}_m^T\bar{B}_m + \lambda_m I_m)^{-1}\bar{B}_m^T V_{m+1}^T. \tag{2.6.4}$$

We note that, since m is typically very small with respect to n, solving the regularized problem (2.6.3) and setting the value of the parameter λ_m (using, for instance, some classical parameter choice technique, cf. [73]) is computationally quite cheap. Since, as just recalled, the SVD of $\bar{B}_k$ approximates the SVD of A, the conditioning of the matrix $\bar{B}_k$ worsens as the Lanczos iterations proceed: usually, after just a few steps of Lanczos algorithm, small and often spurious singular values are approximated together with the leading ones that would allow a good reconstruction and, as a consequence, the approximate solution rapidly deteriorates. Incorporating extra inner regularization at each iteration has a double benefit: from one side, it filters out the small singular values and therefore their bad influence on the reconstructed solution is diminished; from the other side, it allows the solution subspace to further grow and, in this way, some more components that could improve the solution are included. Moreover, the quality of the solution reconstructed by a hybrid method does not strictly depend on the stopping criterion, as for purely iterative methods: this is a clear advantage when an appropriate stopping criterion is not easily determined.

Chapter 3

The Arnoldi-Tikhonov methods

The goal of this chapter is to introduce the class of the Arnoldi-Tikhonov methods. Basically, each variant of an Arnoldi-Tikhonov method aims at computing an approximation of a fixed Tikhonov regularized problem (1.3.4) belonging to a certain Krylov subspace of small but increasing dimension. Different Arnoldi-Tikhonov methods are derived by changing the formulation of the original regularized problem one wants to solve and the Krylov subspaces containing the solution (for instance, one could consider the standard form or the general form Tikhonov regularization and generalized, preconditioned or range-restricted Krylov subspaces). From a theoretical point of view, the Arnoldi-Tikhonov approach is different from the hybrid methods described in the previous chapter (Section 2.6), despite both of them merge a variational and an iterative regularizing strategy. In Section 3.1 we formulate the first and most basic version of the Arnoldi-Tikhonov method, along with its range-restricted modification; in Section 3.2 we review some extension of the standard Arnoldi-Tikhonov method and we propose an original one. This is intended to be a mainly theoretical chapter: numerical tests and comparisons relative to the methods derived in the next sections are postponed to the next chapter where, while explaining different parameter choice strategies that could be adopted when dealing with the AT methods, we see them in action.

3.1 Basic formulation of the Arnoldi-Tikhonov (AT) method

The AT method was first proposed in [19], where it was applied to the Tikhonov regularized problem in standard form (1.3.9). It was derived considering the Galerkin equations associated to the normal equations (1.3.10)

$$W_m^T(A^TA + \lambda I_n)W_m y = W_m^T A^T b, \quad y \in \mathbb{R}^m, \tag{3.1.1}$$

i.e., by orthogonally projecting the normal equations onto the Krylov subspace $K_m(A, b)$ (for a fixed m), whose orthonormal basis is computed by the Arnoldi algorithm (Algorithm 2), and by determining an approximate solution of (1.3.9) belonging to the same Krylov subspace. Exploiting the equality (2.2.3), its transpose version

$$W_m^T A^T = \bar{H}_m^T W_{m+1}^T,$$

and some relations steaming from the Arnoldi algorithm, the system (3.1.1) can be rewritten as

$$(\bar{H}_m^T\bar{H}_m + \lambda I_m)y = \bar{H}_m^T c, \tag{3.1.2}$$

where $c \in \mathbb{R}^{m+1}$ is defined by (2.2.5). The above relations are the normal equations associated to the least squares problem

$$\min_{y\in\mathbb{R}^m} \left\| \begin{pmatrix} \bar{H}_m \\ \sqrt{\lambda} I_m \end{pmatrix} y - \begin{pmatrix} c \\ 0 \end{pmatrix} \right\|. \tag{3.1.3}$$

We note that, using an approach very similar to the second one adopted to derive the GMRES method (cf. Section 2.3.2), the AT method can be equivalently recovered starting from the basic formulation of the Tikhonov method as a penalized least squares problem (1.3.9): one should just impose the approximate solution to belong to the Krylov subspace $\mathcal{K}_m(A, b)$ by taking $x = W_m y$, $y \in \mathbb{R}^m$

$$\min_{y \in \mathbb{R}^m} \left\{ \|b - AW_m y\|^2 + \lambda \|W_m y\|^2 \right\},$$

exploit the Arnoldi decomposition (2.2.3)

$$\min_{y \in \mathbb{R}^m} \left\{ \left\| \|b\| W_{m+1} e_1 - W_{m+1} \bar{H}_m y \right\|^2 + \lambda \|W_m y\|^2 \right\},$$

and the orthonormality of the columns of W_{m+1} (and W_m) to obtain the reduced dimensional penalized minimization problem

$$\min_{y \in \mathbb{R}^m} \left\{ \left\| c - \bar{H}_m y \right\|^2 + \lambda \|y\|^2 \right\}. \tag{3.1.4}$$

The normal equations associated to the above problem are still given by (3.1.2) and the regularized least squares problem is still (3.1.3). This alternative derivation makes the extension to generalized Tikhonov regularization more natural.

Given the three equivalent formulations (3.1.2), (3.1.3) and (3.1.4), one typically chooses to solve (3.1.3), because of its better numerical properties (exactly as done for the classical Tikhonov regularization, cf. Section 1.3.1). However, looking at the normal equation formulation (3.1.2), we can easily determine the following expression for the regularized inverse of the AT method

$$A^{\sharp}_{m,\lambda} = W_m \left(\bar{H}_m^T \bar{H}_m + \lambda I_m \right)^{-1} \bar{H}_m^T W_{m+1}^T. \tag{3.1.5}$$

In order to make the AT method effective, a suitable value for the regularization parameter λ should be determined at each iteration: to underline the dependance of λ on the m-th iteration, we often use the notation λ_m. Moreover, a good value for the parameter m has to be set: this means that we should choose a suitable dimension of the Krylov subspace $\mathcal{K}_m(A, b)$ where we are looking for an approximation of the solution. We focus on these issues in the next chapter; here we just say that, since the most classical parameter selection strategies require the regularized problem to be solved for many different values of the regularization parameter, one could find convenient to perform some sort of preprocessing on the projected regularized problem at each iteration. For instance, one could reduce the matrix $\bar{H}_m$ into bidiagonal form by applying a sequence of Givens rotations from the left-hand and right-hand sides, as suggested in [19]; in this way, the cost to solve the m-th projected problem is $O(m)$. In the following we denote by $x_{\lambda,m}$ the approximate solution of the original regularized problem (1.3.9) obtained by the Arnoldi-Tikhonov method, i.e., $x_{\lambda,m} = W_m y_{\lambda,m}$, where $y_{\lambda,m}$ minimizes (3.1.3). The derivations made so far are summarized in Algorithm 9.

Regarding Algorithm 9, we remark that an approximation of the solution of the full-dimensional original linear system is just computed at the end of the iterative process: this contributes to keep the computational cost low. However, this means that at step 2 of Algorithm 9, the parameter choice strategy adopted should work exclusively with the projected problem, and perform all the required operations in reduced dimension. Still in [19], the authors propose to project the problem (1.3.9) using the decompositions (2.4.7), (2.4.6) provided by the Lanczos bidiagonalization algorithm (Algorithm 8); in the following we refer to this strategy as Lanczos-Tikhonov method. One of the basic features of the Arnoldi-Tikhonov method is that, if compared to the Lanczos-based methods (including the Lanczos-hybrid methods), only matrix-vector products with A are required; however, the Arnoldi-Tikhonov method can be applied only if the coefficient matrix A is square (or after some manipulations have been executed in order to transform A into a square matrix).

Algorithm 9: Arnoldi-Tikhonov (AT) method

Input: A, b

For $m = 1, 2, \ldots,$ until some stopping criterion is satisfied

1. Perform one step of the Arnoldi algorithm (Algorithm 2) with input A, b, and update the decomposition (2.2.3).
2. Apply a parameter selection strategy to determine a suitable value of the regularization parameter λ_m.
3. Compute the solution $y_{\lambda,m}$ of the projected problem (3.1.3), with $\lambda = \lambda_m$.

Compute the approximation $x_{\lambda,m}$ of the solution of the original problem (1.3.9) by $x_{\lambda,m} = W_m y_{\lambda,m}$.

The numerical experiments reported in Chapter 4 and in many works devoted to the AT method [19, 79, 104] show that this regularization method can often deliver accurately reconstructed solutions in very few iterations. Of course, recalling the remarks in Section 2.6, this mainly depends on the quality of the approximation of the SVD of A achieved by means of the Arnoldi algorithm. The analysis and the tests performed in Section 2.5.3 show that, in many circumstances, the singular values of the projected matrix $\bar{H}_m$ accurately approximate the largest singular values of A in just a few iterations: this assures that the most meaningful components of the solution of the original full-dimensional problem can be quickly recovered by solving the regularized projected problem.

Looking again at Algorithm 9 and recalling the general scheme exposed in Section 2.3, we understand that the Arnoldi-Tikhonov method can be legitimately regarded as a Krylov subspace method: indeed, for a fixed m and a fixed λ, the approximate solution of the regularized problem (1.3.9) is a vector belonging to $\mathcal{K}_m(A, b)$ and the m degrees of freedom are fixed by solving the problem (3.1.3). Considering (3.1.4) we also state that the AT method can be regarded as a regularized version of the GMRES method; in this sense, as far as just standard form Tikhonov regularization is considered and if we temporarily ignore the process used to derive the AT method, the final formulation (3.1.4) can formally be regarded as an Arnoldi-hybrid method (in analogy to Lanczos-hybrid methods (2.6.4)). However, the approaches used to derive the class of the hybrid methods and the class of the Arnoldi-Tikhonov and Lanczos-Tikhonov methods are theoretically very different: this is even more evident when one wants to include a regularization matrix different from the identity (cf. Section 3.2) and when some specific parameter selection strategies are adopted (cf. Section 4.3). We can summarize the differences between the two classes in the following sentence:

- hybrid methods aim at regularizing a projection;
- Arnoldi-Tikhonov and Lanczos-Tikhonov methods aim at projecting a regularization.

Indeed, as extensively explained in the previous chapter (Section 2.6), the starting point of an hybrid method is a regularizing iterative projection method: additional inner regularization is imposed at each iteration in order to avoid the semiconvergent behavior of a purely iterative regularization method. When dealing with Arnoldi-Tikhonov methods, one solves a given Tikhonov regularized problem by iteratively projecting it onto a Krylov subspace.

We conclude this section by briefly describing a modification of the Arnoldi Tikhonov method just derived: the so-called Range Restricted Arnoldi-Tikhonov (RRAT) method introduced in [79]. The RRAT method is obtained by projecting the full-dimensional regularized problem (1.3.9) into the Krylov subspace $\mathcal{K}_m(A, Ab)$, whose basis is generated by the Arnoldi algorithm (Algorithm 2) with starting vector Ab. Recalling the discussion in Section 2.6, it is immediate to state that one of the advantages of considering $\mathcal{K}_m(A, Ab)$ as solution subspace instead of

$\mathcal{K}_m(A, b)$ is that some noise components of the basis vector are already filtered out in Ab. To derive the RRAT method we basically act as in the AT method case, except that now the involved subspaces are different. Taking into account the factorization $AW'_m = W'_{m+1}\bar{H}'_m$, which stems from the Arnoldi algorithm (Algorithm 2) applied to A and Ab, substituting $x = W'_m y$ in (1.3.9), and after having performed some algebraic manipulations (due to the fact that b does not belong to the solution subspace anymore), we obtain the following problem

$$\min_{y \in \mathbb{R}^m} \left\{ \left\| W_{m+1}'^T b - \bar{H}'_m y \right\|^2 + \lambda \|y\|^2 \right\}. \tag{3.1.6}$$

After having set a good value $\lambda = \lambda_m$ for the regularization parameter at each iteration and after having computed the solution $y_{\lambda,m}$ of problem (3.1.6), we obtain an approximation of the solution of the full-dimensional problem (1.3.9) by taking $x_{\lambda,m} = W'_{m+1} y_{\lambda,m}$.

3.2 Incorporating a generic regularization matrix

As emphasized in Section 1.3.1, when one wants to apply Tikhonov regularization and has a good intuition of the behavior of the solution x^{ex} of the problem (1.1.8), a regularization matrix different from the identity can considerably improve the quality of the reconstruction. Many generalizations of the original formulation of the AT method (3.1.4) have recently been proposed with the purpose of efficiently including a generic regularization term into the Arnoldi-Tikhonov process. Of course, one can assume that, thanks to the well-known standard form transformation procedures (cf. Section 1.3.1), it is always possible to work with Tikhonov regularization in standard form; indeed, this assumption is made in the early works on Lanczos-Tikhonov and Arnoldi-Tikhonov methods (cf. [19, 20, 79]); however, although efficient strategies to perform transformation into standard form have been developed for many classes of regularization operators (cf. [60, 61, 85]), in general this approach could be very expensive, especially for large-scale problems. Moreover, when one wants to apply the Arnoldi algorithm to the transformed problem, the regularization matrix has to be square: for this reason, some procedures to transform rectangular underdetermined matrices into square ones that force some kind of boundary conditions have been derived (cf. again Section 1.3.1 and the references therein).

When adopting an hybrid approach, we believe that the most effective way of acting on the reconstructed solution with a generic regularization matrix is to apply inner regularization to the preconditioned system whose coefficient matrix is $AL_A^\dagger$, i.e. to act on the system transformed into standard form (cf. Section 2.6). Indeed, at the m-th step of an hybrid method, employing a $q_m \times m$ regularization matrix like the ones defined in (1.3.17) or (1.3.18) could be misleading, even if the corresponding full-dimensional regularization matrices are good choices for the full-dimensional problem: this is due to the fact that there are no guarantees that the same regularization operator is appropriate for enhancing the features of the reduced dimensional solution y_m. For this reason, hybrid methods are usually formulated by exclusively including additional Tikhonov regularization in standard form at each iteration.

In this section, we propose a couple of original strategies that can work directly with Tikhonov regularization method in its most general form (1.3.4), with initial guess $x^* \neq 0$ and regularization matrix $L \in \mathbb{R}^{q \times n}$ with either $q \leq n$ or $q > n$.

To the best of our knowledge, the very first strategy to solve the regularized problem (1.3.4) by means a projection method was derived in [72]: the main idea is to adopt a a procedure that simultaneously and efficiently bidiagonalizes both A and L and that allows to recover an approximate solution of the form $x_{\lambda,m} = R^{-1} y_{\lambda,m}$, where $y_{\lambda,m} \in \mathcal{K}_m(Q_A^T Q_A, Q_A^T b)$ and

$$\begin{pmatrix} A \\ L \end{pmatrix} = QR = \begin{pmatrix} Q_A \\ Q_L \end{pmatrix} R$$

is the QR factorization of the matrix $[A^T L^T]^T$. The basic idea behind this algorithm is that the generalized singular value decomposition of the matrix pair (Q_A, Q_L) provides an increasingly better approximation of the GSVD of the original matrix pair (A, L); recalling the well-known approximation properties of Lanczos bidiagonalization algorithm (cf. Sections 2.5.3 and 2.6), we can regard this procedure as a sort of "generalized" Lanczos bidiagonalization algorithm.

Another approach, which has been proposed very recently in [104], consists in simultaneously reducing both the system matrix and the regularization operator $L \in \mathbb{R}^{n \times n}$ by employing a generalized Arnoldi process originally introduced in [80]: thanks to this procedure, both A and L are transformed into reduced-dimensional upper generalized Hessenberg matrices and the approximate solution at the m-th step of the projection scheme belongs to the generalized Krylov subspace $g\mathcal{K}_m(\{A, L\}, w_1)$, where $w_1 = {}^{b}/_{\|b\|}$. The definition of the subspace $g\mathcal{K}_m(\{A, L\}, w_1)$ is quite cumbersome: it can be shown that, if $\dim(g\mathcal{K}_m(\{A, L\}, w_1)) = k$, then $\text{span}\{w_1, \ldots, w_k\} = g\mathcal{K}_m(\{A, L\}, w_1)$, where $\text{span}\{w_1, \ldots, w_k\}$ is generated by taking the first k vectors (from top to bottom, from left to right) of the following sequence

$$\begin{array}{c} w_1, \\ Aw_1,\ Lw_1, \\ A^2w_1,\ LAw_1,\ ALw_1,\ L^2w_1, \\ A^3w_1,\ LA^2w_1,\ ALAw_1,\ L^2Aw_1,\ A^2Lw_1,\ LALw_1,\ AL^2w_1,\ L^3w_1, \\ \vdots \end{array}.$$

To conclude this introductory survey, we briefly address a further strategy that is derived in [76] and that it is used to incorporate a generic regularization matrix $L \in \mathbb{R}^{q \times n}$ into the AT formulation: at the m-th iteration, the upcoming solution subspace W_{m+1} is obtained by updating the current subspace W_m with a vector of the form $(A^T A + \lambda_m L^T L) W_m y_{\lambda_m, m} - A^T b$, where $y_{\lambda_m, m} \in \mathbb{R}^m$ solves the system

$$(W_m^T (A^T A + \lambda_m L^T L) W_m) y = W_m^T A^T b;$$

an orthogonal basis for this subspace is computed adopting a generalized-Arnoldi-algorithm-like procedure similar to the one described in [104] and an efficient strategy to set the regularization parameter λ_m is also described.

The approach adopted here is different from the ones so far described in that the solution subspace is a Krylov subspace defined only with respect to the matrix A; our approach was first introduced in [94] and it is indeed very similar to the one derived in [67] in the Lanczos-Tikhonov method case. In the following, to keep our dissertation general, we assume that an initial guess $x^* \neq 0$ for the exact solution x^{ex} is available and we want to include this information in the projection process. Let us consider the full-dimensional regularized problem (1.3.4). Strictly following the second derivation proposed in Section 3.1 for the standard-form case, we search for approximations of the type

$$x = x^* + W_m y, \tag{3.2.1}$$

where $W_m \in \mathbb{R}^{n \times m}$ is the orthogonal basis of the Krylov subspace $\mathcal{K}_m(A, r^*)$ (as usual, $r^* = b - Ax^*$) computed by the Arnoldi algorithm (Algorithm 2). Substituting the expression (3.2.1) into (1.3.4) and exploiting the decomposition (2.2.3), we obtain the reduced minimization problem

$$\begin{aligned} & \min_{y \in \mathbb{R}^m} \left\{ \|r^* - AW_m y\|^2 + \lambda \|LW_m y\|^2 \right\} \\ = \quad & \min_{y \in \mathbb{R}^m} \left\{ \left\| \|r^*\| e_1 - \bar{H}_m y \right\|^2 + \lambda \|LW_m y\|^2 \right\}, \end{aligned}$$

whose minimizer $y_{\lambda,m}$ solves the the following regularized least square problem

$$y_{\lambda,m} = \arg \min_{y \in \mathbb{R}^m} \left\| \begin{pmatrix} \bar{H}_m \\ \sqrt{\lambda} L W_m \end{pmatrix} y - \begin{pmatrix} \|r^*\| e_1 \\ 0 \end{pmatrix} \right\|^2. \tag{3.2.2}$$

As already underlined in Sections 1.3.1 and 3.1, formulation (3.2.2) is the one to be solved in practice when dealing with Tikhonov regularized problems. Once the solution $y_{\lambda,m}$ of (3.2.2) is computed taking a suitable $\lambda = \lambda_m$, the solution of the original regularized problem (1.3.4) is recovered by taking $x_{\lambda,m} = x^* + W_m y_{\lambda,m}$. Sometimes, in the following, we refer to problem (3.2.2) as Generalized Arnoldi-Tikhonov (GAT) method, just to underline that a regularization operator different from the identity has been considered.

We note that the problem (3.2.2) has a coefficient matrix of dimension $(m+q+1) \times m$, since, in general, $L \in \mathbb{R}^{q\times n}$: therefore, the problem is not completely reduced as in the standard-form regularization case (recall that, when dealing with standard-form problems, the matrix in (3.1.3) is of size $(2m+1) \times m$). At a first glance, this formulation could seem computationally disadvantageous; however, we remark that the strategy (3.2.2) can deal with arbitrary regularization matrices so that this drawback is usually balanced by the positive effect that a suitable L can have on noisy problems. Furthermore, it is very important to observe that the AT method and, in general, each Krylov solver based on the construction of the Krylov subspaces $\mathcal{K}_m(A, r^*)$ (cf. Sections 2.6 and 3.1), is generally very fast for discrete ill-posed problems, and hence the number of columns of the matrix in (3.2.2) is very small; therefore this computational disadvantage is actually negligible, as revealed by many numerical experiments (cf. Chapter 4).

For theoretical purposes we can also recover the normal equations formulation of the Arnoldi-Tikhonov method for the case $L \neq I$. The normal equations associated to the regularized least squares problem (3.2.2) are

$$(\bar{H}_m^T \bar{H}_m + \lambda W_m^T L^T L W_m) y = \bar{H}_m^T (\|r^*\| e_1). \tag{3.2.3}$$

Of course, when $x^* = 0$, $x_{\lambda,m} \in \mathcal{K}_m(A, b)$ and in the formulations (3.2.2) and (3.2.3) it suffices to replace r^* by b. Equation (3.2.3) allows us to express the regularized inverse for the AT case with $L \neq I$ (and, for simplicity, $x^* = 0$) as

$$A^{\sharp}_{\lambda,m} = W_m \left(\bar{H}_m^T \bar{H}_m + \lambda W_m^T L^T L W_m\right)^{-1} \bar{H}_m^T W_{m+1}^T. \tag{3.2.4}$$

Some alternative formulations of the problem (3.2.2) have been derived in order to replace the tall regularization matrix $LW_m \in \mathbb{R}^{q\times m}$ with an analogous completely reduced one. A first possibility, also suggested in [67], could be to compute the "skinny" QR factorization of LW_m, i.e.

$$LW_m = Q_m^L R_m^L, \quad \text{where} \quad Q_m^L \in \mathbb{R}^{q\times m},\ (Q_m^L)^T Q_m^L = I_m \text{ and } R_m^L \in \mathbb{R}^{m\times m},$$

and to equivalently rewrite the problem (3.2.2) as

$$\begin{aligned} & \min_{y\in\mathbb{R}^m} \left\| \begin{pmatrix} I_{m+1} & \\ & Q_m^L \end{pmatrix} \left(\begin{pmatrix} \bar{H}_m \\ \sqrt{\lambda} R_m^L \end{pmatrix} y - \begin{pmatrix} \|r^*\| e_1 \\ 0 \end{pmatrix} \right) \right\|^2 \\ = & \min_{y\in\mathbb{R}^m} \left\| \begin{pmatrix} \bar{H}_m \\ \sqrt{\lambda} R_m^L \end{pmatrix} y - \begin{pmatrix} \|r^*\| e_1 \\ 0 \end{pmatrix} \right\|^2, \end{aligned}$$

where we have exploited the fact that the columns of the matrix

$$\operatorname{diag}(I_{m+1}, Q_m^L) \in \mathbb{R}^{(m+q+1)\times(2m+1)}$$

are orthonormal. In this way, the matrix associated to the regularized projected least squares problem is

$$[(\bar{H}_m)^T \ \sqrt{\lambda}(R_m^L)^T]^T \in \mathbb{R}^{(2m+1)\times m}.$$

Another possibility, that leads to a problem related but not anymore equivalent to (3.2.2), is to essentially project also the regularization matrix L onto the Krylov subspace $\mathcal{K}_m(A, r^*)$. However we remark that, to do this, L must be square. So, if $L \in \mathbb{R}^{q\times n}$ with $q < n$, it is just

sufficient to turn it into a square matrix by adopting one of the already mentioned strategies derived in [28, 29, 106]: the most immediate one is the so-called "zero-padding", i.e. $n-q$ zero rows are appended to the rectangular matrix L; moreover, applying "zero-padding" does not alter the null space of the original regularization matrix. If $L \in \mathbb{R}^{q\times n}$ with $q > n$, then we can perform an initial "skinny" QR factorization of L

$$L = Q^L R^L, \quad \text{where} \quad Q^L \in \mathbb{R}^{q\times n},\ (Q^L)^T Q^L = I_n \text{ and } R^L \in \mathbb{R}^{n\times n},$$

and consider the equivalent problem

$$\begin{aligned} & \min_{x\in\mathbb{R}^n} \left\| \begin{pmatrix} I_n & \\ & Q^L \end{pmatrix} \left(\begin{pmatrix} A \\ \sqrt{\lambda} R^L \end{pmatrix} x - \begin{pmatrix} b \\ 0 \end{pmatrix} \right) \right\|^2 \\ = & \min_{x\in\mathbb{R}^n} \left\| \begin{pmatrix} A \\ \sqrt{\lambda} R^L \end{pmatrix} x - \begin{pmatrix} b \\ 0 \end{pmatrix} \right\|^2, \end{aligned}$$

where we have exploited the fact that the columns of the matrix

$$\operatorname{diag}(I_n, Q^L) \in \mathbb{R}^{(n+q)\times(2n)}$$

are orthonormal. In the following we denote by $\hat{L}$ the square version of the original regularization matrix $L \in \mathbb{R}^{q\times n}$; summarizing the above considerations: if $q < n$ then $\hat{L}$ is some square version of the matrix L, if $q = n$ then $\hat{L} = L$, if $q > n$ then $\hat{L} = R^L$.

Once $\hat{L}$ has been computed, we replace the regularization matrix LW_m in the formulation (3.2.2) by the following regularization matrix

$$L_m = W_m^T \hat{L} W_m \in \mathbb{R}^{m\times m}, \tag{3.2.5}$$

obtaining the regularized least square problem

$$\min_{y\in\mathbb{R}^m} \left\| \begin{pmatrix} \bar{H}_m \\ \sqrt{\lambda} L_m \end{pmatrix} y - \begin{pmatrix} \|r^*\| e_1 \\ 0 \end{pmatrix} \right\|^2, \tag{3.2.6}$$

whose coefficient matrix is of size $(2m+1)\times m$. The normal equations associated to (3.2.6) are

$$(\bar{H}_m^T \bar{H}_m + \lambda L_m^T L_m) y = \bar{H}_m^T (\|r^*\| e_1). \tag{3.2.7}$$

and

$$\begin{aligned} A_{\lambda,m}^{\sharp} &= W_m \left(\bar{H}_m^T \bar{H}_m + \lambda L_m^T L_m \right)^{-1} \bar{H}_m^T W_{m+1}^T \\ &= W_m \left(\bar{H}_m^T \bar{H}_m + \lambda W_m^T L^T (W_m W_m^T) L W_m \right)^{-1} \bar{H}_m^T W_{m+1}^T. \end{aligned} \tag{3.2.8}$$

However, it is important to underline that considering formulation (3.2.6) instead of (3.2.2) is not always computationally advantageous. For instance, computing the matrix R^L in the case $L \in \mathbb{R}^{q\times n}$, $q > n$, could be very expensive, since the QR factorization of a full-dimensional matrix has to be computed. Moreover, computing the QR factorization of the reduced-dimensional matrix LW_m or computing the projected matrix $\hat{L}$ could be as expensive as solving directly (3.2.2), especially if the considered parameter selection strategy to set $\lambda = \lambda_m$ at each step of the Arnoldi algorithm does not require the problem (3.2.2) to be solved for many different values of the regularization parameter.

We conclude this section by making some remarks about the approximation of the GSVD of (A, L) by means of the GSVD of $(\bar{H}_m, L_m)$. As explained in Section 3.1, we can claim that the efficiency of the Arnoldi-Tikhonov method is mainly due to the good approximation of the SVD of A achieved by considering the SVD of $\bar{H}_m$; therefore, in the framework of general form Tikhonov regularization method, it is natural to investigate the links between the GSVD of

(A, L) and the GSVD of $(\bar{H}_m, L_m)$. We adopt an approach similar to the one described in Section 2.5.3 for the SVD. We remark that the use of Lanczos bidiagonalization (Algorithm 8) to iteratively approximate the GSVD of (A, L) has been studied in [72].

Fixing some notations that are a simplified version of the ones already employed in Sections 1.1.3 and 2.5.3, we consider the GSVD of the matrix pair (A, L) given by

$$A = U\Sigma X^{-1} \quad \text{and} \quad L = VMX^{-1};$$

the GSVD of the matrix pair $(\bar{H}_m, L_m)$, where L_m is defined in (3.2.5) is given by

$$\bar{H}_m = U^{(m)}\Sigma^{(m)}(X^{(m)})^{-1} \quad \text{and} \quad L_m = V^{(m)}M^{(m)}(X^{(m)})^{-1}, \tag{3.2.9}$$

where $U^{(m)} \in \mathbb{R}^{(m+1)\times m}$, $X^{(m)} \in \mathbb{R}^{m\times m}$, $\Sigma^{(m)} = \text{diag}(\sigma_1^{(m)}, \ldots, \sigma_m^{(m)})$, and $M^{(m)} = \text{diag}(\mu_1^{(m)}, \ldots, \mu_m^{(m)})$. We also define the matrices

$$\hat{U}^{(m)} = W_{m+1}U^{(m)}, \quad \hat{V}^{(m)} = W_m V^{(m)}, \quad \hat{X}^{(m)} = W_m X^{(m)}.$$

We state the following proposition, which is the GSVD analogous of Proposition 15 stated in Section 2.5.3.

Proposition 17. *Let $u_k^{(m)}$, $v_k^{(m)}$ and $x_k^{(m)}$ be the k-th columns of the matrices $U^{(m)}$, $V^{(m)}$ and $X^{(m)}$ respectively. Then, denoting the k-th columns of the matrices $\hat{U}^{(m)}$, $\hat{V}^{(m)}$ and $\hat{X}^{(m)}$ as $\hat{u}_k^{(m)} = W_{m+1}u_k^{(m)}$, $\hat{v}_k^{(m)} = W_m v_k^{(m)}$ and $\hat{x}_k^{(m)} = W_m x_k^{(m)}$, respectively, we have*

$$A\hat{x}_k^{(m)} - \sigma_k^{(m)}\hat{u}_k^{(m)} = 0, \tag{3.2.10}$$

$$W_m^T(L\hat{x}_k^{(m)} - \mu_k^{(m)}\hat{v}_k^{(m)}) = 0. \tag{3.2.11}$$

Proof. Both relations (3.2.10) and (3.2.11) are immediate consequences of the relations (2.2.3) and (3.2.9). About (3.2.10) it suffices to write:

$$AW_m x_k^{(m)} - \sigma_k^{(m)}W_{m+1}u_k^{(m)} = W_{m+1}(\bar{H}_m x_k^{(m)} - \sigma_k^{(m)}u_k^{(m)}) = 0.$$

About equality (3.2.11), it suffices to write

$$W_m^T(LW_m x_k^{(m)} - \mu_k^{(m)}W_m v_k^{(m)}) = L_m x_k^{(m)} - \mu_k^{(m)}v_k^{(m)} = 0.$$

□

As in the SVD case, Proposition 17 ensures that if the matrix A has full rank, then the Arnoldi algorithm allows to construct the GSVD of (A, L); step by step, the quality of the approximation depends on the distance between the subspaces $\text{span}\{Lw_1, ..., Lw_m\}$ and $\mathcal{K}_m(A, b)$. We can also write the following equivalent estimates

$$\left\|L\hat{x}_k^{(m)} - \mu_k^{(m)}\hat{v}_k^{(m)}\right\| \leq \left\|(I - W_m W_m^T)LW_m\right\| \cdot \|x_k^{(m)}\|$$

and

$$\left\|L\hat{x}_k^{(m)} - \mu_k^{(m)}\hat{v}_k^{(m)}\right\| \leq \left\|\left(W_m^{\perp}\right)^T LW_m\right\| \cdot \|x_k^{(m)}\|,$$

which are similar to the ones derived in (2.5.26) and (2.5.29) in the SVD case; we remark that in the GSVD case we also have the factor $\|x_k^{(m)}\|$ because the columns of $X^{(m)}$ are not orthogonal anymore.

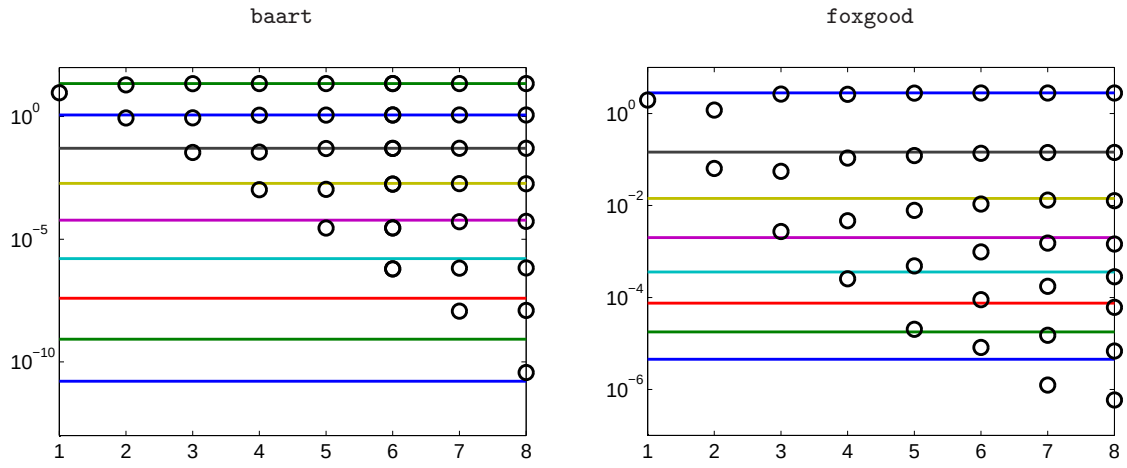

Figure 3.2.1: Approximation of the generalized singular values by means of the Arnoldi algorithm (Householder version). The multi-color horizontal lines stand for the first 8 singular values of the full-dimensional matrix pair (A, D_2^{ZP}); the black dots corresponding to the i-th tick on the horizontal axis indicate the generalized singular values of the projected matrix pair $(\bar{H}_i, L_i)$, $1 \leq i \leq 8$.

In Figure 3.2.1 we show a couple of tests about the convergence of the generalized singular values of the matrix pair $(\bar{H}_m, L_m)$, where L_m is computed as in (3.2.5) and by using the zero-padded version D_2^{ZP} of the matrix D_2 defined in (1.3.18), i.e.,

$$D_2^{ZP} = \begin{pmatrix} 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ 0 & \cdots & \cdots & \cdots & 0 \\ 0 & \cdots & \cdots & \cdots & 0 \end{pmatrix} \in \mathbb{R}^{n \times n}. \tag{3.2.12}$$

Chapter 4

Parameter Choice Strategies

As already underlined in the previous chapters, the regularization parameter specifies the amount of regularization to be imposed; therefore, no regularization method is complete without defining a strategy to set the regularization parameter. As carefully explained in [55, 59], choosing the regularization parameter is like setting a tradeoff between perturbation and regularization errors. The regularization error is caused by the regularization process: since the original problem has been substituted by a nearby one, the regularization error measures the "distance" between the unregularized and the regularized problems; the perturbation error is linked to the inversion of the noise components in the data. If a regularization method can be expressed as a spectral filtering method (1.3.2), we can recover an explicit expression for both the regularization and the perturbation errors: in this case the former is associated to the introduction of the filtering, the latter is associated to the inversion and the filtering of the noise. In general, the regularization error increases with the amount of imposed regularization, while the perturbation error decreases: if the reconstructed solution is over-smoothed (i.e., when λ is too big in the Tikhonov case, when the truncation parameter is too small in the TSVD case, when too few iterations have been carried out in the iterative case) the the regularization error is high and the perturbation error is low; on the contrary, if the reconstructed solution is under-smoothed the regularization error is low and the perturbation error is high.

Because of the importance of setting a proper regularization parameter when performing any kind of regularization, a lot of research has been devoted to the study of reliable methods to set the regularization parameters (we cite the surveys [55, 73, 102]): some of them can work just in connection with special regularization schemes, but most of them are quite general and have been adapted and improved to match different form of regularization. A common feature of all the parameter choice methods is that they usually deal with the residual norm and quantities linked to it; some of them require also the norm of the approximate solution or the norm of the perturbation that affects the data to be available.

In this chapter we present an overview of the most well-established parameter choice techniques along with some more recent ones, which have successfully been employed when performing Tikhonov, TSVD, iterative and Lanczos-hybrid methods. However, we believe that a deep investigation of the parameter choice strategies to be employed in connection with the Arnoldi-Tikhonov method has not been performed, yet. To the best of our knowledge, only a sort of projected L-curve and discrepancy principle have been adopted, so far. Therefore, after reviewing the basic features of each parameter choice scheme, we turn to describe how they can be adapted in the AT framework; in the discrepancy principle case we are even able to derive some schemes that are specifical for the AT method. This chapter is organized as follows: in Section 4.1 we review the basic features of the discrepancy principle and we propose three new discrepancy-related strategies that can be successfully and efficiently implemented in connection with the class of the Arnoldi-Tikhonov methods; in Section 4.2 we review the the basic L-curve criterion along with a version to be adopted in connection with the Arnoldi-Tikhonov method

described in [19]; in Section 4.3 we explain the basic features of the Generalized Cross Validation (GCV) method and we review a related version recently derived in [95] to be used in connection with the Arnoldi-Tikhonov method. The last section of this chapter (Section 4.4) is devoted to the so-called multi-parameter Tikhonov regularization method: the method is considered in conjunction with an Arnoldi-Tikhonov scheme and an efficient and original strategy to set the regularization parameters is described.

4.1 Discrepancy Principle

The discrepancy principle was originally introduced in [86] and it is perhaps the most successful and most popular parameter choice strategy that could be employed when a fairly accurate approximation of $\varepsilon = \|e\|$ is available. Denoting, as usual, by x_α the regularized solution associated to the parameter α, the discrepancy vector $\hat{\phi}(\alpha)$ associated to x_α is defined by

$$\hat{\phi}(\alpha) = b - Ax_\alpha = b - AA_\alpha^\sharp b, \tag{4.1.1}$$

where, as usual, b denotes the available and corrupted right-hand side and $A_\alpha^\sharp$ denotes the regularized inverse of A introduced in Section 1.3.1. We can regard the discrepancy as a sort of residual computed with respect to the regularized solution. The discrepancy principle prescribes to take, as regularization parameter α, the solution of the following equation

$$\phi(\alpha) = \eta\varepsilon, \quad \eta > 1, \tag{4.1.2}$$

where

$$\phi(\alpha) = \|\hat{\phi}(\alpha)\| = \|b - Ax_\alpha\|$$

is the norm of the discrepancy vector and η is a safety factor that is usually very close to 1 (its value reflects the uncertainty on the available estimate of ε: the more accurate the estimate of ε, the closer η to 1). Sometimes we refer also to $\phi(\alpha)$ as discrepancy, even if this is an abuse; however the discrepancy vector is always denoted by the "hat" symbol. If we rather know the noise level $\widetilde{\varepsilon} = {\|e\|}/{\|b^{ex}\|}$, then the discrepancy principle reads

$$\phi(\alpha) = \eta\widetilde{\varepsilon}\|b\|\,.$$

We remark that, for most methods, the function $\phi(\alpha)$ increases with the regularization parameter α and the discrepancy principle is very sensitive to the value of ε; therefore, underestimating ε could lead to severely under-regularized or over-regularized solutions (depending on the considered method). When, hypothetically, $x_\alpha = x^{ex}$, the discrepancy principle is naturally satisfied since

$$\phi(\alpha) = \|b - b^{ex}\| = \|e\| = \varepsilon.$$

We briefly review how the discrepancy principle applies to the different classes of regularization methods introduced so far.

- **Tikhonov regularization**. Assuming to work with standard form Tikhonov regularization, the discrepancy principle consists in solving, with respect to the regularization parameter λ, the following nonlinear equation

  $$\phi(\lambda) := \|b - Ax_\lambda\| = \eta\varepsilon, \quad \eta > 1,$$

  where we can substitute the explicit expression for x_λ given in (1.3.11). The quantity $\phi(\lambda)$ is monotonically increasing with respect to λ and, usually, Newton-like zero-finders are employed to solve the above equation. In [105] the authors derive a very efficient cubically convergent zero-finder, which employs quantities obtained by solving in a computationally convenient way the regularized least squares problem (1.3.10) along with other least

squares problems associated to it; in order to guarantee the convexity of the involved functions, this strategy applies to a Tikhonov regularized problem whose parameter is $\lambda = 1/\beta$, i.e. the proposed zero-finder determines the regularization parameter associated to the problem

$$\min_{x \in \mathbb{R}^n} \left\{ \|b - Ax\|^2 + \frac{1}{\beta} \|x\|^2 \right\}.$$

- **Truncated Singular Value Decomposition**. In this case the regularization parameter m is the truncation parameter (a discrete quantity). We say that the discrepancy principle is satisfied when

$$\phi(m) := \|b - Ax_m\| \leq \eta\varepsilon, \quad \eta > 1,$$

where x_m is explicitly defined in (1.3.26). Starting from a big value of m, we take as regularization parameter the first m that satisfies the above inequality.

- **Iterative Regularization**. In this case, at the m-th iteration of an iterative method, the discrepancy vector $\hat{\phi}(m)$ coincides with the residual vector; therefore, in this setting, we simply denote the discrepancy vector by $\hat{r}_m$ and the norm of the discrepancy vector by r_m (this notation is slightly different from the one adopted in Chapter 2, but it is coherent with the one adopted in this section for the discrepancy vector and its norm). Taking into account just the GMRES method (Section 2.3.2) and the CGLS/LSQR methods (Section 2.4), it is well-known that the residual r_m decreases as the iterations proceed. Therefore we can simply use the discrepancy principle as a stopping criterion and we stop the iterations at the first value of m such that

$$r_m \leq \eta\varepsilon.$$

We remark that, in the GMRES and LSQR cases, we can evaluate the discrepancy by simply taking the projected quantities, i.e.,

$$r_m = \|c - \bar{H}_m y_m\| \quad \text{and} \quad r_m = \|c - \bar{B}_m y_m\|$$

respectively.

- **Lanczos-based hybrid methods**. As already remarked in Section 2.6, in the hybrid case both an iteration-dependent regularization parameter λ_m and a stopping criterion should be specified. In this case we denote the discrepancy vector by $\hat{\phi}_m(\lambda)$, to underline that the discrepancy is a function of both m and λ; of course we have that $\hat{\phi}_m(0)$ is the residual $\hat{r}_m$ associated to the iterative method underlying the hybrid method. In [73] the authors suggest to employ the discrepancy principle just to set the value of the parameter λ at each iteration: indeed, at the m-th iteration, a set of trial values for λ is considered and the discrepancy

$$\phi_m(\lambda) = \left\| c - \bar{B}_m y_{m,\lambda} \right\|, \quad y_{m,\lambda} = (\bar{B}_m^T \bar{B}_m + \lambda I_m)^{-1} \bar{B}_m^T c$$

is evaluated for each parameter by using the projected quantities. The current regularization parameter λ_m is set by taking the largest trial value such that

$$\phi_m(\lambda) \leq \eta\varepsilon.$$

- **Arnoldi-Tikhonov methods**. As remarked in Section 3.1 and exactly as in the hybrid case, good values for both m and the regularization parameter $\lambda = \lambda_m$ should be determined; in this setting we again denote the discrepancy vector by $\hat{\phi}_m(\lambda)$. Also in the Arnoldi-Tikhonov (and Lanczos-Tikhonov) cases the discrepancy principle is typically employed to set an appropriate value for the parameter λ at each iteration. In the standard

Tikhonov regularization case, the most popular approach consists in taking $\lambda = 1/\beta$ and in applying the cubically convergent zero-finder described in [105] to solve the nonlinear equations

$$\left\|c - \bar{H}_m y_{\beta,m}\right\| = \eta\varepsilon, \quad y_{\beta,m} = (\bar{H}_m^T \bar{H}_m + \frac{1}{\beta} I_m)^{-1} \bar{H}_m^T c$$

in the Arnoldi-Tikhonov case, or

$$\left\|c - \bar{B}_m y_{\beta,m}\right\| = \eta\varepsilon, \quad y_{\beta,m} = (\bar{B}_m^T \bar{B}_m + \frac{1}{\beta} I_m)^{-1} \bar{B}_m^T c$$

in the Lanczos-Tikhonov case. A stopping criterion is derived by monitoring the value of some quantities obtained as a by-product of the implementation of the zero-finder. This strategy has also been adapted to work in connection with the RRAT method in [79] and the more generalized projection schemes derived in [76, 104].
We cite also the strategy proposed in [35], even if it cannot be properly regarded as an Arnoldi-Tikhonov method (although the original regularized problem is projected into Krylov subspaces of increasing dimension): the authors propose to efficiently solve the normal equations (1.3.10) by performing some iterations of the CG method (Section 2.3.3) for different values of λ and to employ the discrepancy principle to determine which value of λ is the most suitable one.

We now turn to describe an original discrepancy-principle-like strategy, called secant update method, which was first proposed in [38]. The secant update approach can be efficiently and successfully adopted in connection with the class of the Arnoldi-Tikhonov methods described in Chapter 3.

4.1.1 The secant update method

Let us consider the Arnoldi-Tikhonov method formulations (3.2.2) or (3.2.6) that are derived in the previous chapter by projecting the full-dimensional available problem (1.3.4) into the Krylov subspaces $\mathcal{K}_m(A, r^*)$, $m \geq 1$. As just explained, in the Arnoldi-Tikhonov setting one can equivalently consider the projected discrepancy principle instead of the full-dimensional one. Indeed, thanks to the decomposition (2.2.3) and the properties of the matrices therein involved, we can define the discrepancy at the m-th iteration as

$$\phi_m(\lambda_m) = \|b - Ax_{\lambda_m,m}\| = \|r^* - AW_m y_{\lambda_m,m}\| = \|c - \bar{H}_m y_{\lambda_m,m}\|, \tag{4.1.3}$$

where c is defined as in (2.2.5) and $y_{\lambda_m,m}$ solves (3.2.2) or (3.2.6) with $\lambda = \lambda_m$. As already explained, in this framework the value of the discrepancy depends on both m and λ_m and we say that the discrepancy principle is satisfied at the m-th iteration if

$$\phi_m(\lambda_m) = \|c - \bar{H}_m y_{\lambda_m,m}\| \leq \eta\varepsilon, \quad \eta > 1. \tag{4.1.4}$$

Since $y_{\lambda_m,m}$ solves the normal equations (3.2.3) or (3.2.7), the discrepancy can be rewritten as

$$\phi_m(\lambda_m) = \left\|c - \bar{H}_m(\bar{H}_m^T \bar{H}_{m+1} + \lambda_m W_m^T L^T L W_m)^{-1} \bar{H}_m^T c\right\|. \tag{4.1.5}$$

or

$$\phi_m(\lambda_m) = \left\|c - \bar{H}_m(\bar{H}_m^T \bar{H}_{m+1} + \lambda_m L_m^T L_m)^{-1} \bar{H}_m^T c\right\|, \tag{4.1.6}$$

respectively. Our basic idea is to consider, at each iteration m, the following linear approximation with respect to λ_m of the discrepancy function $\phi_m(\lambda)$:

$$\phi_m(\lambda) \simeq \phi_m(0) + \lambda d_m. \tag{4.1.7}$$

Looking at the expression (4.1.5) and (4.1.6) we immediately realize that $\phi_m(0)$ is indeed the norm of the m-th residual of the GMRES method applied to the unregularized available linear system; this also agrees with the description of the Arnoldi-Tikhonov methods proposed in Section 3.1. Therefore, as usual, in the following we simply use the notation $r_m = \phi_m(0)$. Since the matrix $\bar{H}_m \in \mathbb{R}^{(m+1)\times m}$ is already computed by the Arnoldi-Tikhonov method, to obtain r_m we just have to solve a projected least squares problem of the form (2.3.3); since m is typically much smaller than n, the computational overload to solve this system and to get r_m is negligible.

To provide a simple and computationally convenient expression for the scalar d_m we assume that, at the m-th iteration of the Arnoldi algorithm, a value $\lambda = \lambda_m$ for the current regularization parameter has been fixed: when $m = 1$, the value λ_1 is given in input by the user; when $m > 1$, the parameter λ_m has been computed at the end of the previous iteration. Once the solution $y_{\lambda_m,m}$ of the m-th projected problem (3.2.2) or (3.2.6) has been determined, we can easily compute the associated discrepancy $\phi_m(\lambda_m)$ by the expression (4.1.3). Imposing the linear approximation (4.1.7) to exactly hold for $\lambda = \lambda_m$, we obtain

$$\phi_m(\lambda_m) = r_m + \lambda_m d_m,$$

and therefore

$$d_m = \frac{\phi_m(\lambda_m) - r_m}{\lambda_m}. \tag{4.1.8}$$

At this point, at the m-th iteration, we have recovered an expression for both the scalars r_m and d_m: before the next iteration of the Arnoldi algorithm has been performed, we set the value of the regularization parameter λ_{m+1} to be used at the $(m+1)$-th iteration of the Arnoldi-Tikhonov method. To do this we force the linear approximation (4.1.7) to satisfy the discrepancy principle, i.e., we impose

$$\phi_m(\lambda_{m+1}) = r_m + \lambda_{m+1} d_m = \eta\varepsilon. \tag{4.1.9}$$

By substituting the value of d_m derived in (4.1.8) and by solving the above equation with respect to λ_{m+1} we recover

$$\lambda_{m+1} = \frac{\eta\varepsilon - r_m}{\phi_m(\lambda_m) - r_m}\lambda_m. \tag{4.1.10}$$

In this way, starting from an initial guess for λ_1, a sequence of parameters $\{\lambda_m\}_{m\geq 1}$ to be employed at the m-th iteration of the Arnoldi-Tikhonov method is generated by using the update formula (4.1.10); the iterations are stopped as soon as the discrepancy principle (4.1.4) is satisfied. We remark that, during the first iterations, care should be needed when applying the update formula (4.1.10). Indeed, being r_m the norm of the GMRES residual, it is natural to expect that, during the first iterations, $r_m > \eta\varepsilon$: in this situation the update formula (4.1.10) produces a negative value and therefore we should consider

$$\lambda_{m+1} = \left|\frac{\eta\varepsilon - r_m}{\phi_m(\lambda_m) - r_m}\right| \lambda_m. \tag{4.1.11}$$

Anyway, we know that, independently of the definition of λ_m, after some steps the GMRES method satisfies the discrepancy principle, i.e. $r_m < \eta\varepsilon$, and indeed stabilizes slightly under the level $\eta\varepsilon$ (cf. the arguments in Section 2.5.2).

Remark 18. The method (4.1.10) has a simple geometric interpretation that allows us to regard it as a zero finder. Indeed, recalling that $\phi_m(\lambda)$ is a monotonically increasing function such that $\phi_m(0) = r_m$, we can state that the linear function

$$f(\lambda) = r_m + \lambda\left(\frac{\phi_m(\lambda_m) - r_m}{\lambda_m}\right),$$

interpolates $\phi_m(\lambda)$ at 0 and λ_m, and the new parameter λ_{m+1} is obtained by solving $f(\lambda) = \eta\varepsilon$. Therefore, applying the update formula (4.1.10) at each iteration of the Arnoldi-Tikhonov method is like performing one step of a secant method in which the leftmost point is always $(0, r_m)$. For this reason this strategy is called "secant update method". This geometrical interpretation does not hold just during the first iterations (until $r_m > \eta\varepsilon$), since we are taking as λ_{m+1} the modulus of the negative solution of $f(\lambda) = \eta\varepsilon$. In Figure 4.1.1 we display what typically happens at the m-th iteration of the Arnoldi-Tikhonov method, when the condition $r_m < \eta\varepsilon$ is satisfied.

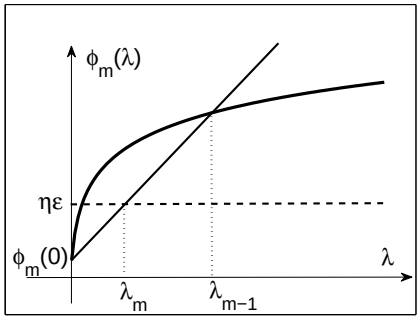

Figure 4.1.1: Zero finder interpretation of formula (4.1.10).

In Algorithm 10 we summarize the strategy just derived.

Algorithm 10: AT method with secant-update

Input: A, b, L, x^*, λ_1, η, ε.
Take $r^* = b - Ax^*$
For $m = 1, 2, \ldots,$ until $\|c - \bar{H}_m y_{\lambda_m,m}\| \leq \eta\varepsilon$

1. Perform one step of the Arnoldi algorithm (Algorithm 2) with A and r^* as inputs and update the decomposition (2.2.3).
2. Compute the solution $y_{\lambda_m,m}$ of the problem (3.2.2) or (3.2.6), with $\lambda = \lambda_m$.
3. Compute the GMRES solution (2.3.3) and evaluate the corresponding residual r_m.
4. Compute $\phi_m(\lambda_m)$ by (4.1.3).
5. Compute the regularization parameter λ_{m+1} by (4.1.11).

Compute the current approximation $x_{\lambda_m,m}$ of the solution of the original problem (1.3.9) by $x_{\lambda_m,m} = W_m y_{\lambda_m,m}$.

Remark 19. Numerically, formula (4.1.11) is very stable, in the sense that after the discrepancy principle is satisfied, $\lambda_m \approx$ const. This is due to the fact that both $\phi_m(\lambda_m)$ and r_m stagnate (cf. the estimates in Section 2.5.2). Indeed, since the approximations are computed minimizing the residual within a Krylov subspace, whenever $\mathcal{K}_{m+1}(A, b) \approx \mathcal{K}_m(A, b)$, the values of $\phi_m(\lambda_m)$ and r_m tend to remain almost constant because, in any case, at the m-th iteration, $b - AW_m y_{\lambda_m,m} \in \mathcal{K}_{m+1}(A, b)$. It is also natural to expect that the value around which the sequence $\{\lambda_m\}_{m\geq 1}$ stabilizes is a suitable value for the regularization parameter of the original full-dimensional regularized problem (1.3.4). Moreover, the secant update method is very stable with respect to the choice of the initial value λ_1 (cf. Figure 4.1.5): except for the first iterations (when we also have $r_m > \eta\varepsilon$), the values computed by formula (4.1.11) tend to be the same as the iterations proceed and therefore the quality of the reconstructed solution is independent of the initial

choice of λ_1. Let us recall the geometrical interpretation proposed in Remark 18: the fact that, after some iterations, $\phi_m(\lambda) \approx \phi_{m+1}(\lambda)$, implies that the discrepancy curves are overlapping and, going on with the iterations, the former secant update zero-finder turns into a proper secant zero finder, except that the first point of the linear approximation is always $(0, r_m)$ (cf., for instance, [101, §6.2]).

Remark 20. The secant update method acts simultaneously as a parameter choice strategy and as a stopping criterion. This is basically due to the stable behavior of the sequence $\{\lambda_m\}_{m\geq 1}$ after some iterations have been performed (cf. Remark 19). Indeed, if we recall the derivation of the secant update strategy and if we look at Algorithm 10, we understand that, at the current step of the Arnoldi-Tikhonov method, we use the regularization parameter that would have satisfied the linearly approximated discrepancy principle (4.1.9) at the previous step. Therefore, the discrepancy principle is satisfied at the end of the current iteration only if some stability in the sequence of the regularization parameters has occurred; since, when this happens, $\mathcal{K}_{m+1}(A, b) \approx \mathcal{K}_m(A, b)$ (cf. again Remark 19) it means that the solution subspace almost stagnates and the iterations can be properly stopped.

Remark 21. If compared to the other parameter choice strategies so far used in connection with the AT method, we realize that the secant update one is intrinsically simpler and cheaper. In fact, if at each iteration we want to apply the L-curve criterion (cf. Section 4.2) to the standard form reduced problem (3.1.4), as proposed in [19], where this algorithm is referred to as L_m-curve method, we have to compute the SVD of $\bar{H}_m$ in order to solve the system (3.1.3) for many values of λ. Then we need to employ a reliable algorithm to choose the point of maximum curvature, that sometimes may even provide an unsatisfactory value for λ. On the other side, using the methods proposed in [76, 79, 104], once determined a suitable m, we have to apply a convergent zero-finder to solve the nonlinear equation $\phi_m(\beta) = \eta\varepsilon$, $\lambda = 1/\beta$. The latter requires, at each step, the value of the first and the second derivative of ϕ_m computed for the β determined at the previous step of the zero-finder and to do this we have to solve two related linear systems of dimension m. Actually, since in both cases all the extra computations involve the reduced matrices, we also stress that the computational overload can still be considered negligible. However, the generalization of these strategies to the case of the generalized AT method is not straightforward. On the contrary, when applying the secant update method, just the residual of the GMRES method at each iteration should be computed. Moreover we believe that, with minor changes, the secant update strategy can successfully be applied to the other variations of AT described in Section 3.2.

To assess the quality of the reconstructions obtained applying the secant update method and to compare it with the already existing strategies commonly employed to set the regularization parameter when performing the Arnoldi-Tikhonov methods, we present the results of some meaningful numerical experiments.

Example 1. The first test considers the problem `shaw`: the coefficient matrix is of size 200×200 and the noise level that affects the available vector b is $\widetilde{\varepsilon} = 10^{-3}$; we choose $x^* = 0$ as initial guess for the solution and we take as safety factor in (4.1.4) $\eta = 1.001$. In order to present a straightforward comparison with the parameter choice strategies commonly used in connection with the AT method, in this first example we take $L = I_{200}$. In this context, when we employ the L_m-curve criterion, we stop at iteration $\hat{m}$ if the norm of the discrepancy associated to the parameter computed using the $L_{\hat{m}}$-curve is below the known threshold $\eta\varepsilon$.

In Figure 4.1.2 we display the results obtained performing 30 tests (for each test we define a new perturbed right-hand side to lessen the dependence of the results on the random components of e). Both the L_m-curve and the secant update method determine a regularized solution which always belongs to the Krylov space $\mathcal{K}_8(A, b)$. However the new method is in every situation the more stable one, since the relative error norms and the values of λ determined during

the last iteration are always comparable. The same does not hold for the L_m-curve method, that on average produces approximated solutions of slightly worse quality. We also show the norms of the relative error and the values of the regularization parameter determined solving the nonlinear equation $\phi_8(\lambda) = \eta\varepsilon$ by Newton's method.

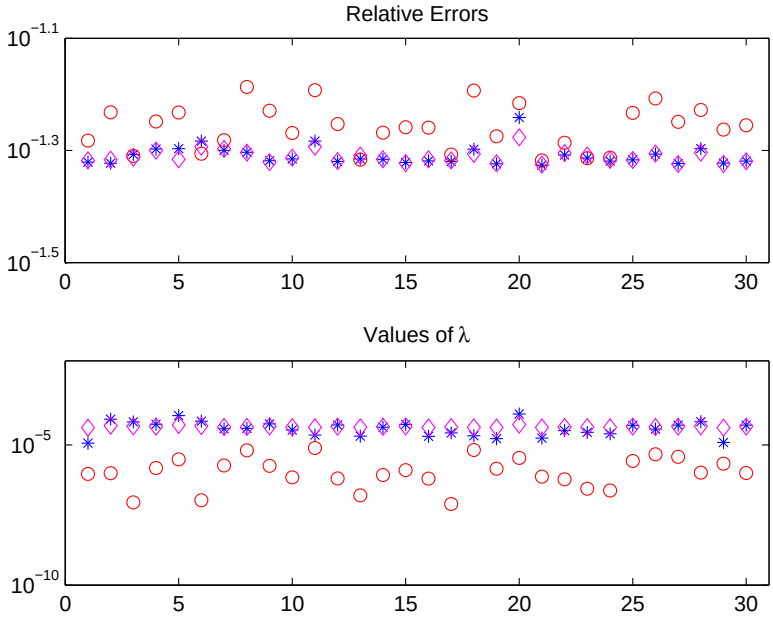

Figure 4.1.2: Norms of the relative error (upper frame) and the values of λ corresponding to the last iteration (lower frame): we plot one marker for each of the 30 tests performed. The asterisk denotes the secant update method, the circle denotes the L_m-curve method (Section 4.2) and the diamond denotes the values obtained solving $\phi_8(\lambda) = \eta\varepsilon$.

In Figure 4.1.3 we plot the solution, corresponding to the 8-th test reported in Figure 4.1.2, computed using the L_m-curve criterion and our secant approach. We can see that, employing Tikhonov regularization method in standard form, the quality of the solutions obtained using both methods is comparable, but the one computed by the L_m-curve method shows instability around the exact solution. This is due to the fact that this criterion allows a slight under-smoothing since it typically selects values of λ smaller than the secant update method (cf. Figure 4.1.2).

In Figure 4.1.4 we compare the behavior of the L_m-curve method and the one of our secant update at each iteration. The results correspond to the 22-th test reported in Figure 4.1.2. Since, in this example, the discrepancy principle is satisfied after only 8 iterations, we decide to compute some extra iterations to evaluate the behavior of both methods after the stopping criterion is fulfilled. In particular, we can note that the secant approach exhibits a very stable progress since, once the threshold is reached, the norm of the discrepancy stagnates and the values of the regularization parameter λ remain almost constant.

In Figure 4.1.5 we display the values of the regularization parameter, at each iteration, obtained varying the initial value λ_1 given in input. We choose $\lambda_1 = 0.1,\ 0.5,\ 1,\ 10,\ 50$. It is quite evident that the strategy is able to determine suitable values for λ and m independently on the choice of the initial guess.

Example 2. We now consider the behavior of the GAT method with respect to the one based on the generalized Krylov subspaces as described in [104], here denoted by AT-GKS. To determine the regularization parameter when performing the AT-GKS method we employ, at

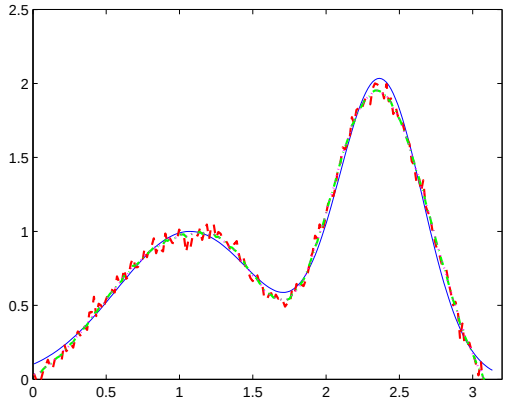

Figure 4.1.3: Computed solutions of the test problem `shaw`. Exact solution (solid line), regularized with the secant update (dash-dot line), regularized with the L_m-curve method (dashed line).

each iteration, Newton's method to solve the discrepancy equation $\phi_m(\beta) = \eta\varepsilon$. We take the test problem `baart`: the coefficient matrix is of size 500×500 and the noise level that affects the available vector b is $\widetilde{\varepsilon} = 10^{-2}$; we choose $x^* = 0$ as initial guess for the solution and we take as safety factor in (4.1.4) and $\eta = 1.1$. In order to apply the AT-GKS method we must take a square regularization matrix, so we use the matrix D_2^0 defined in (1.3.20). In Figure 4.1.6 we show the results obtained performing 25 iterations of the Arnoldi algorithm and of the generalized Arnoldi process when applying the GAT and the AT-GKS methods, respectively.

To present further comparisons between the two methods we summarize the results of some further test problems in Table 4.1. For each problem we generate a matrix of dimension 500 and, as before, we take $\widetilde{\varepsilon} = 10^{-2}$, $\eta = 1.1$ and $L = D_2^0$.

Test	Method	Time (sec)	Minimum Relative error
`baart`	AT-GKS	0.12	$1.4202 \cdot 10^{-2}$ (11)
	GAT	0.01	$9.0670 \cdot 10^{-3}$ (7)
`gravity`	AT-GKS	0.29	$8.4911 \cdot 10^{-3}$ (17)
	GAT	0.03	$6.2079 \cdot 10^{-3}$ (16)
`phillips`	AT-GKS	0.35	$3.8747 \cdot 10^{-2}$ (18)
	GAT	0.02	$3.0353 \cdot 10^{-2}$ (11)
`shaw`	AT-GKS	0.26	$7.9675 \cdot 10^{-2}$ (15)
	GAT	0.01	$6.9368 \cdot 10^{-2}$ (8)

Table 4.1: Comparisons of the results obtained performing the GAT method and the AT-GKS method. We list the number of iterations performed to attain the minimum relative error between brackets.

Example 3. We want to briefly show the advantages of using the generalized version of the Arnoldi-Tikhonov method with respect to the standard one. We exclusively employ the secant update method to set both the regularization parameter λ and the number of iterations m. We consider the test problem `gravity`: the coefficient matrix is of size 400×400 and the noise level that affects the available vector b is $\widetilde{\varepsilon} = 10^{-2}$; we choose $x^* = 0$ as initial guess for the solution and we take as safety factor in (4.1.4) and $\eta = 1.01$. As regularization matrix we take the matrix D_2 defined in (1.3.18). To evaluate the progress of the relative error we decide to run both the AT and the GAT methods for 20 iterations. We display the obtained results in Figure 4.1.7. Performing all the iterations takes 0.18 seconds using the GAT method and 0.04 seconds using the AT method; this agrees with the fact that, employing a regularization operator different from the identity, the computational demand is heavier. However, looking

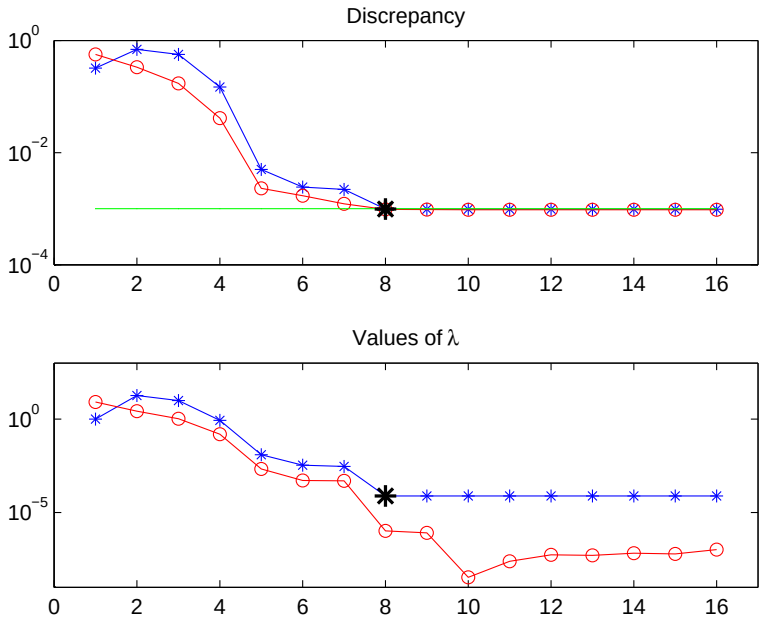

Figure 4.1.4: Comparison between the values of the norm of the discrepancy and the values of the regularization parameter computed, at each iteration, by the L_m-curve method (circles) and by the secant update (asterisks). The horizontal line in the upper graphic represents the threshold $\eta\varepsilon$. The values corresponding to the 8-th iteration, the one at which both methods would stop, are marked with a thicker asterisk.

at both the plots in Figure 4.1.7 we can see an undeniable improvement in the quality of the approximations obtained. In particular, looking at the right plot, we can clearly see that, using the GAT method, we can overcome all the spurious oscillations that affect the AT regularized solution.

Example 4. We now consider some examples coming from two-dimensional image restoration problems. To build the blurring matrix A we use the function `blur` from [56] and we consider a noise level $\widetilde{\varepsilon} = 10^{-2}$. In this context, we will exclusively apply Tikhonov regularization in general form and we consider, as regularization matrices: D_1 defined in (1.3.17), D_1^{hv} defined in (1.3.23), $D_1^{v+h} = D_1^v + D_1^h$ (where D_1^v and D_1^h are defined as in (1.3.21)), D_2 defined in (1.3.18) and $D_2^{v+h} = D_2^v + D_2^h$ (where D_2^v and D_2^h are defined as in (1.3.22)).

Our first task is to compare the performance of the L_m-curve criterion and of the secant update method. We consider the popular test image `peppers.png`, in its original size 256×256 pixels. The parameters of the function `blur` are `s=2.5` and `q=6`, and the associated matrix A has size 65536; we consider $\widetilde{\varepsilon} = 10^{-2}$ and we take D_2 as regularization matrix. In order to employ the function `l_curve` from [54] we have slightly modified the function `cgsvd`, belonging to the same package, with the purpose of working with the matrix $D_2 W_m$ that has more rows than columns. In Figure 4.1.8 we can examine the quality of the reconstruction obtained using both the L_m-curve criterion (frame **(c)**) and the secant approach (frame **(d)**). In Figure 4.1.9 we report the history of the norm of the relative error and of the discrepancy for both methods. We can see that the restored images are almost identical, even if the norm of the relative error is equal to $7.88 \cdot 10^{-2}$ if we use the L_m-curve method and $8.34 \cdot 10^{-2}$ if we use our secant approach; the number of iterations is 9 in the first case, 8 in the second case. However, considering the running time, the difference between the two approaches is more pronounced: using the L_m-curve criterion we need 3.05 seconds to compute the solution, while the new method restores the available image in 0.49 seconds. This gap is mainly due to the fact that, using the L_m-

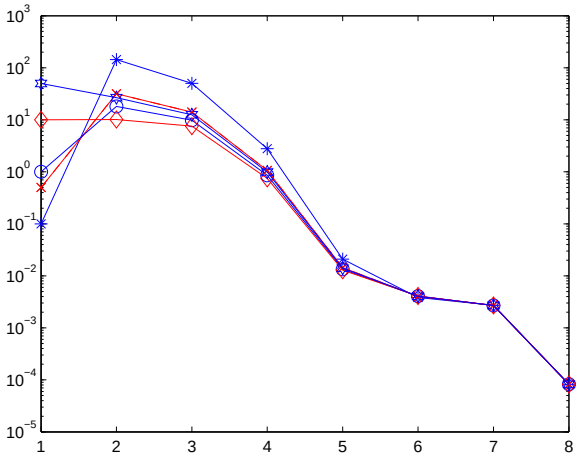

Figure 4.1.5: Values of the regularization parameter computed at each iteration by the secant update method, varying the initial guess λ_1.

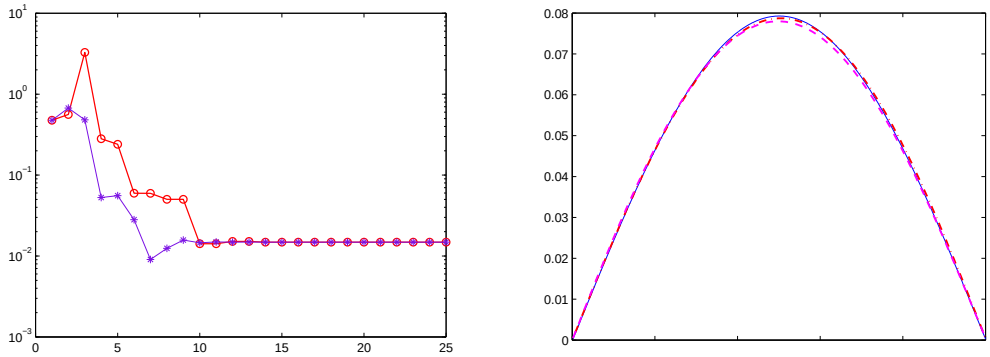

Figure 4.1.6: Solution of `baart`. Left box: relative errors versus the dimension of the Krylov subspaces considered for the new GAT method (asterisks) and for the AT-GKS method (circles). Right box: best regularized solution computed by the two methods. More precisely we display the exact solution (solid line) and the solutions regularized by the GAT (dash-dot line) and the AT-GKS (dashed line) methods.

curve method, at each step we have to evaluate the GSVD of the matrix pair $(\bar{H}_m, D_2 W_m)$ and the matrix $D_2 W_m$ has the same number of rows as the regularization matrix of the unreduced problem.

Example 5. We now focus exclusively on our method. We want to test the behavior of the GAT method varying the regularization matrix; we still consider the test image `peppers.png`. As a matter of fact, the corrupted image is always restored in less than a second and the results obtained by choosing different regularization matrices among the ones above listed are very similar. In particular, since the test image is smooth and lacks of highly definite edges, the best result is obtained applying the second derivative operator D_2. To improve the quality of the restoration, once the stopping criterion is fulfilled at a certain step with λ as regularization parameter, we try to carry on some extra iterations with λ fixed. We will denote this approach by $(L)_{\text{extra}}$, where L is one of the matrices listed in the previous example and the value of the variable "extra" is the number of extra performed iterations. In Table 4.2 we record the results obtained considering the function `blur` with parameters `s=1.5` and `q=6`; the results displayed in Table 4.3 are obtained considering the function `blur` with parameters `s=2.5` and `q=6`. The

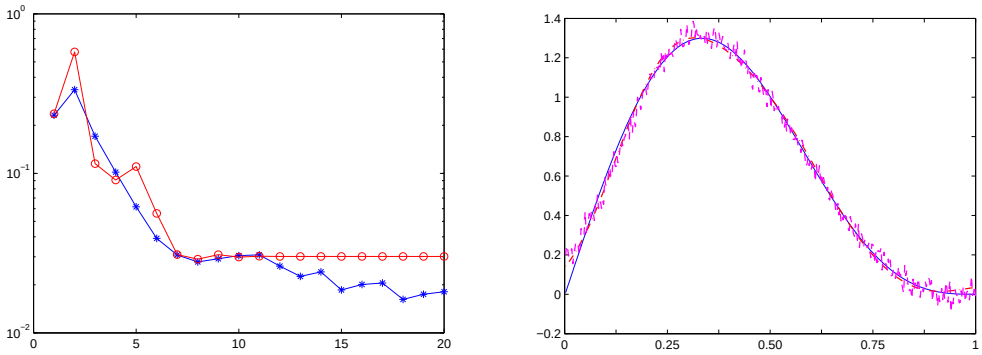

Figure 4.1.7: Solution of `gravity`. Left frame: relative errors obtained at each iteration running the GAT method (asterisks) and the standard AT method (circles). Right frame: the regularized solutions computed when the stopping criterion is satisfied by the two methods. More precisely we display the exact solution (solid line), and the solutions regularized by the GAT (dash-dot line) and the AT (dashed line, obtained at iteration) methods.

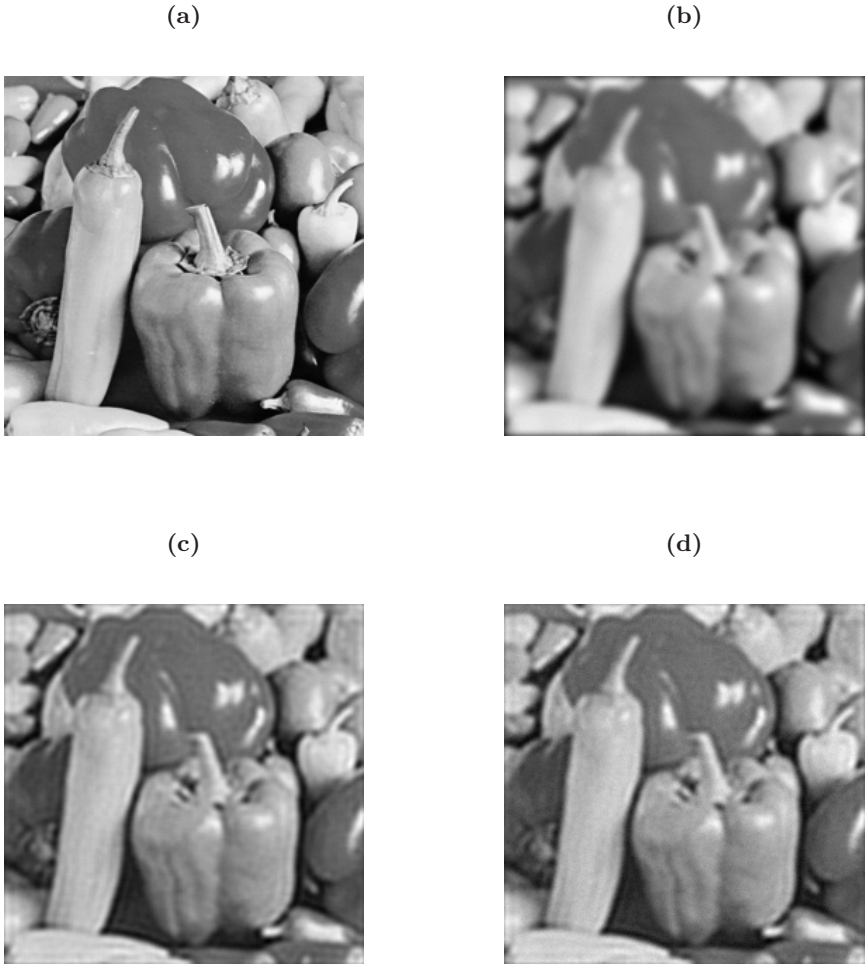

Figure 4.1.8: **(a)**: original image `peppers.png`; **(b)**: blurred and noisy image; **(c)**: image restored with the L_m-curve criterion; **(d)**: image restored with the secant update method.

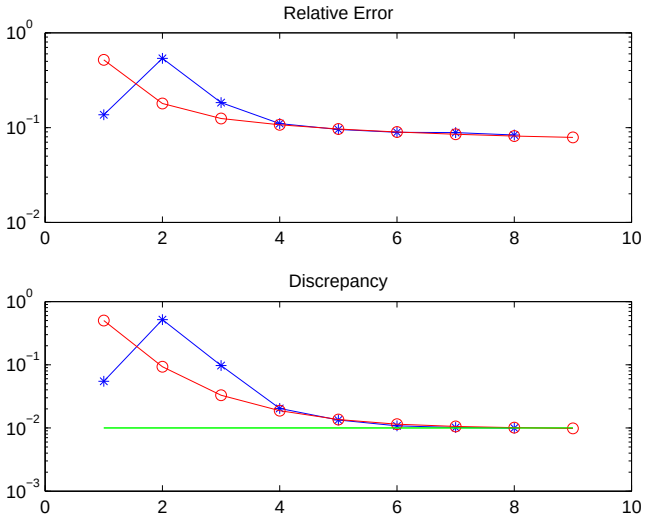

Figure 4.1.9: Restoration of `peppers.png`. Comparison between the norm of the relative error (upper frame) and the norm of the discrepancy (lower frame) obtained using the L_m-curve criterion (circles) and the secant update (asterisks). The horizontal continuous line marks the threshold $\eta\varepsilon$.

noise level $\widetilde{\varepsilon}$ is equal to 10^{-2} in both cases.

Reg. Matr.	Relative Error	Iterations	Running time (sec)
I_n	$7.3930 \cdot 10^{-2}$	4	0.21
D_1	$5.5585 \cdot 10^{-2}$	6	0.34
D_1^{hv}	$5.5594 \cdot 10^{-2}$	6	0.39
D_1^{v+h}	$5.5665 \cdot 10^{-2}$	6	0.31
$(D_1^{v+h})_4$	$5.0402 \cdot 10^{-2}$	10	0.62
D_2	$5.2268 \cdot 10^{-2}$	7	0.39
D_2^{v+h}	$5.2423 \cdot 10^{-2}$	7	0.40
$(D_2^{v+h})_4$	$5.0298 \cdot 10^{-2}$	11	0.79

Table 4.2: Results of the restoration of the image `peppers.png` affected by a Gaussian blur with $\sigma = 1.5$ and a noise level equal to 10^{-2}. In the first column we list the considered regularization matrices.

Finally we consider the performance of the new method applied to the restoration of corrupted medical images. We take the test image `mri.tif` from Matlab, of size 128×128 pixels, which represents a magnetic resonance image of a section of the human brain. Contrary to the previous test image, the present one is characterized by well marked edges. We again consider the function `blur` with parameters `s=1.5` and `q=6`. The noise level is equal to 10^{-2}. In Table 4.4 we report the results obtained changing the regularization operators and in Figure 4.1.10 we show the restored image obtained employing the regularization matrix D_1^{v+h} and running the GAT algorithm for 5 extra iterations.

4.1.2 $\|e\|_2$-free discrepancy principle: overestimation of the noise level

The secant-update method just presented can also be successfully employed as a simple procedure to detect the noise level whenever it is just overestimated. This technique was first proposed in [38]. As already remarked at the beginning of Section 4.1, when the norm of the noise that affects the data is not accurately known, then the discrepancy principle usually leads to poor results; in these situations one typically employs other techniques, such as the L-curve criterion (Section 4.2) or the GCV method (Section 4.3). Basically, our approach consists in

Reg. Matr.	Relative Error	Iterations	Running time (sec)
I_n	$1.1268 \cdot 10^{-1}$	6	0.23
D_1	$8.4488 \cdot 10^{-2}$	8	0.48
D_1^{hv}	$8.4487 \cdot 10^{-2}$	8	0.53
D_1^{v+h}	$8.4446 \cdot 10^{-2}$	8	0.47
$(D_1^{v+h})_3$	$7.6920 \cdot 10^{-2}$	11	0.73
D_2	$8.3142 \cdot 10^{-2}$	8	0.47
D_2^{v+h}	$8.3742 \cdot 10^{-2}$	8	0.47
$(D_2^{v+h})_3$	$7.6927 \cdot 10^{-2}$	11	0.80

Table 4.3: Results of the restoration of the image `peppers.png` affected by a Gaussian blur with $\sigma = 2.5$ and a noise level equal to 10^{-2}. In the first column we list the considered regularization matrices.

Reg. Matr.	Relative Error	Iterations	Running time (sec)
I_N	$1.8615 \cdot 10^{-1}$	6	0.07
D_1	$1.8459 \cdot 10^{-1}$	6	0.10
D_1^{hv}	$1.8471 \cdot 10^{-1}$	6	0.10
D_1^{v+h}	$1.8434 \cdot 10^{-1}$	6	0.08
$(D_1^{v+h})_5$	$1.7078 \cdot 10^{-1}$	11	0.17
D_2	$1.7704 \cdot 10^{-1}$	8	0.11
D_2^{v+h}	$1.7700 \cdot 10^{-1}$	8	0.11
$(D_2)_5$	$1.6854 \cdot 10^{-1}$	13	0.27

Table 4.4: Results of the restoration of the image `mri.tif` affected by a Gaussian blur with $\sigma = 1.5$ and a noise level equal to 10^{-2}. In the first column we list the considered regularization matrices.

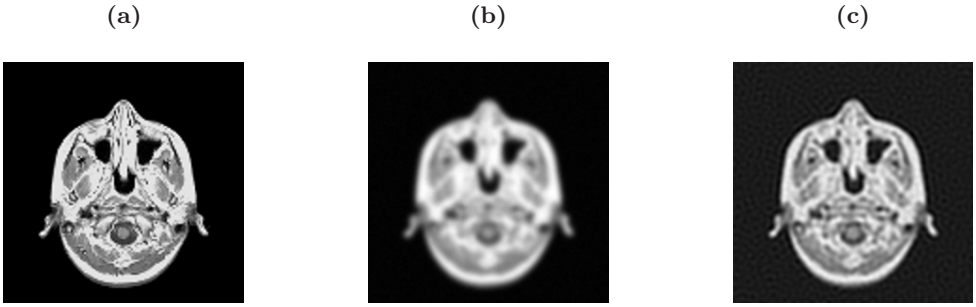

Figure 4.1.10: Restoration of `mri.tif` using $(L_1^m)_5$. **(a)**: original image; **(b)**: blurred and noisy image; **(c)**: restored image.

restarting the AT method, and in using the values of the discrepancy function computed at the end of each set of iterations to progressively improve the approximation of ε. The examples so far considered have shown that this strategy is really effective, and the additional expenses due to the restarts of the AT method do not heavily affect the total amount of work. This is due to the fact that the AT method is extremely fast whenever a suitable initial approximation x^* is available.

Let the true norm ε of the noise be overestimated by a quantity $\bar{\varepsilon}$, i.e. $\bar{\varepsilon} > \varepsilon$. In this situation, we can naturally expect that, applying the AT method, we can fully satisfy the discrepancy principle even taking $\eta = 1$, i.e.,

$$\phi_m(\lambda_m) < \overline{\varepsilon} \tag{4.1.12}$$

at some iteration m of the Arnoldi-Tikhonov method (more precisely, let m be the first iteration for which condition (4.1.12) holds). Applying the secant update method (4.1.10) for the definition of the parameter λ in the projected Tikhonov-regularized problems, the discrepancy would then stabilize around $\overline{\varepsilon}$, if the method is not terminated (recall the discussion in Remark 19). Our idea is to restart it immediately after (4.1.12) is fulfilled, taking as initial guess $x^* = x_{\lambda_m,m}$ and working with the Krylov subspaces generated by A and $b - Ax_{\lambda_m,m}$, where $x_{\lambda_m,m}$ is the last obtained approximation. At the same time we define a new approximation of the noise by taking $\overline{\varepsilon} = \phi_m(\lambda_m)$. We proceed until the discrepancy is almost constant (recalling again what is said in Remark 19, this condition is meaningful) and we introduce a threshold parameter $\tau > 0$ to check this situation step-by-step. This idea has been implemented following Algorithm 11.

Algorithm 11: AT method with noise-level detection

Input: A, b, L, $\lambda^{(0)}$, η, τ.
Take $x^{(0)} = 0$ and $\varepsilon^{(0)} = \overline{\varepsilon} > \varepsilon$.
For $k = 1, 2, \ldots,$ until

$$\frac{\|\varepsilon^{(k)} - \varepsilon^{(k-1)}\|}{\|\varepsilon^{(k-1)}\|} \leq \tau \tag{4.1.13}$$

1. Run Algorithm 10 with $x^* = x^{(k-1)}$, $\varepsilon = \varepsilon^{(k-1)}$, $\lambda_1 = \lambda^{(k-1)}$.
2. Let $x^{(k)}$ be the last approximation achieved, $\phi^{(k)}$ the corresponding discrepancy norm, and $\lambda^{(k)}$ the last parameter value when (4.1.12) is satisfied.
3. Define $\varepsilon^{(k)} = \phi^{(k)}$.
4. Define $\lambda^{(k)} = \frac{\phi^{(k)}}{\phi^{(k-1)}}\lambda^{(k)}$

We remark that Step 4 of Algorithm 11 is rather heuristic. Indeed, as explained in Remark 19, Algorithm 10 does not provide a further update for λ whenever the discrepancy principle is satisfied. Anyway, in this situation we may expect that, improving the quality of the noise estimate, at the k-th restart we have

$$\varepsilon^{(k)} = \phi^{(k)} \leq \phi^{(k-1)} = \varepsilon^{(k-1)}$$

because, by assumption, the noise is overestimated; therefore, the corresponding estimated optimal value for λ should decrease accordingly, since with less noise we need less amount of regularization. The definition in Step 4 also ensures that $\lambda^{(k)} \approx$ const whenever $\phi^{(k)} \approx \phi^{(k-1)}$ and therefore the same convergent behavior of the discrepancy and of the regularization parameter sequences is preserved.

Example. We test the method described in Algorithm 11 (with and without Step 4) considering again the test image `mri.tif` and, as before, we build the blurring matrix A with parameters

`s=1.5` and `q=6`; the available blurred image is corrupted in order to obtain $\widetilde{\varepsilon} = 10^{-3}$. At this point we assume to know only an overestimate $\bar{\varepsilon}$ of ε such that $\bar{\varepsilon}/\|b^{ex}\| = 10^{-2}$. We employ the regularization matrix D_1^{v+h} defined in Section 4.1.1 and we choose $\tau = 0.01$. In Figure 4.1.11 we display the results. We can clearly observe that, after 4 iterations, i.e., after the first call of the AT method, the norm of the discrepancy lays below the overestimated threshold $\overline{\varepsilon}$. At this point, we allow the AT method to restart immediately and to go on until the approximation is satisfactory, i.e., until condition (4.1.13) is fulfilled. The implementation with step 4 of Algorithm 11 satisfies the stopping criterion (4.1.13) after just 24 restarts, with an estimate $\varepsilon^{(24)}/\|b^{ex}\| = 1.03 \cdot 10^{-3}$ and after running for 0.48 seconds. The implementation without step 4 of Algorithm 11 requires 56 restarts to deliver the estimate $\varepsilon^{(56)}/\|b^{ex}\| = 1.05 \cdot 10^{-3}$ and runs for 1.45 seconds. For this example, after the first call of the Arnoldi-Tikhonov method, the discrepancy principle is always satisfied after just one iteration. If we do not include the update described at Step 4, the value of λ considered at each restart is equal to the one computed at the end of the first call (looking at Algorithm 10, we can clearly see that, if just one iteration of the Arnoldi algorithm is performed, then λ is not updated). If, coherently with the fact that we are progressively approaching a lower noise level, we force the value of λ to decrease at each restart, then also the discrepancy function (which is increasing with respect to λ and which, for a fixed λ, stabilizes as the iterations proceed) is forced to decrease.

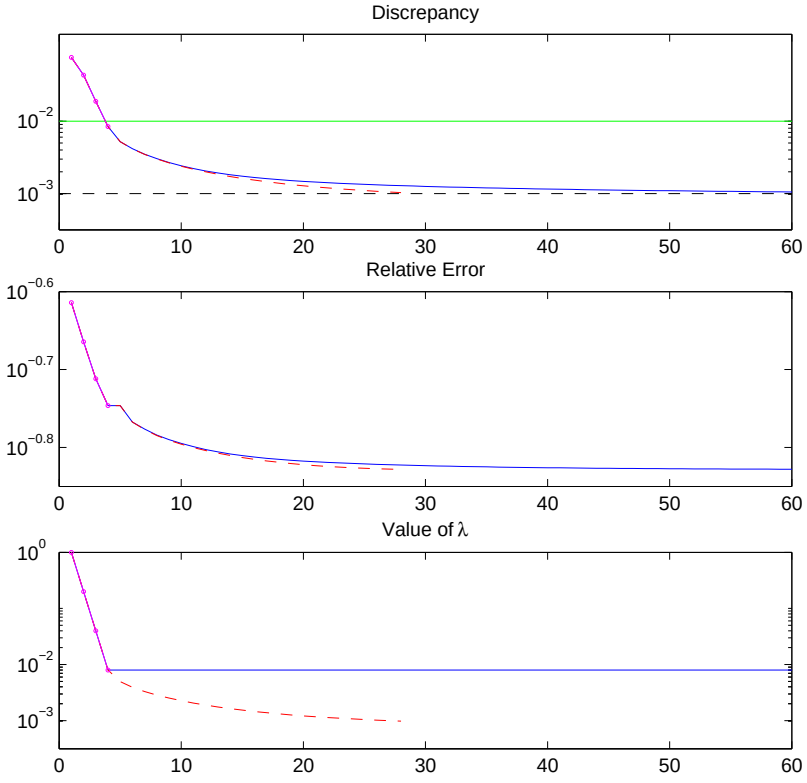

Figure 4.1.11: Results of Algorithm 11 applied to the restoration of `mri.tif` with overestimated noise level. In the upper frame we also plot two horizontal lines, which represent the overestimated and the true noise level. In each frame, the initial 4 iterations, corresponding to the first call of Algorithm 10, are highlighted using a small circle. The continuous line represents the slower version of Algorithm 11 (without Step 4), the dashed line represents its quicker version (with Step 4).

In Figure 4.1.12 and 4.1.13 we compare the quality of the reconstruction after the first call of Algorithm 10 (when the overestimated discrepancy principle is satisfied) and at the end of the scheme described in Algorithm 11 (with Step 4).

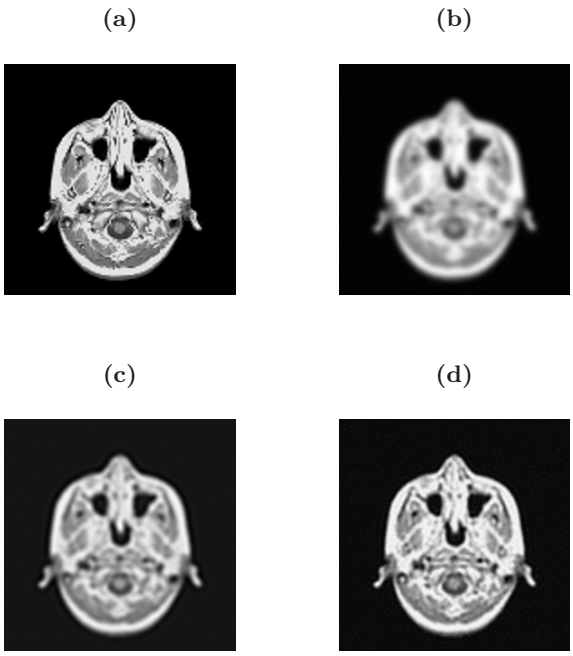

Figure 4.1.12: Restoration of the test image `mri.tif`. **(a)**: exact image; **(b)**: blurred and noisy image; **(c)**: restoration after the first 4 steps of Algorithm 10; **(d)**: restoration after 24 iterations of Algorithm 11.

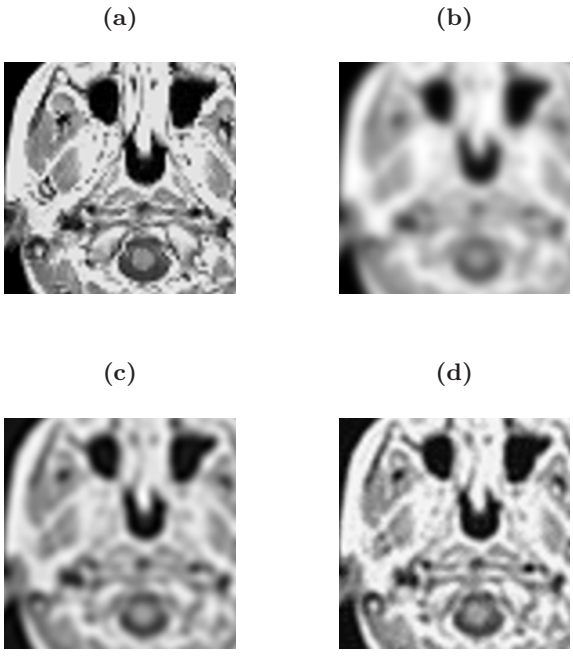

Figure 4.1.13: Restoration of the test image `mri.tif`: blow-up of the corresponding images shown in Figure 4.1.12.

The numerical experiments performed on various kind of discrete ill-posed problems have proved that this approach is really robust and fairly accurate; so far, we have not found examples in which the procedure failed.

4.1.3 $\|e\|_2$-free discrepancy principle: embedded approach

To derive this alternative formulation (first proposed in [40]) of the secant-update method that does not require any estimate of $\varepsilon = \|e\|$ to be available, we should assume that, after just a few iterations of the Arnoldi algorithm underlying the AT method, the norm of the residual associated to the GMRES method lies around the threshold $\|e\|$ and, despite being slightly decreasing, stabilizes during the following iterations. Recalling the discussion in Section 2.5.2, we understand that this hypothesis usually holds when considering severely ill-posed problems and that the test problems described in Section 1.2.1 fulfill this requirement (cf. Figure 2.5.5). This motivates the use of the following update strategy to choose the regularization parameter at the m-th iteration of the Arnoldi-Tikhonov method

$$\begin{aligned} \lambda_{m+1} &= \frac{\hat{\eta}\phi_{m-1}(0) - \phi_m(0)}{\phi_m(\lambda_m) - \phi_m(0)}\lambda_m \\ &= \frac{\hat{\eta} r_{m-1} - r_m}{\phi_m(\lambda_m) - r_m}\lambda_m \,, \end{aligned} \tag{4.1.14}$$

where $\hat{\eta} > 1$. Relation (4.1.14) is simply obtained by replacing the quantity $\eta\varepsilon$ in (4.1.11) by $\hat{\eta} r_{m-1}$, where $r_{m-1} = \|c - \bar{H}_{m-1} y_{m-1}\|$ is the $(m-1)$-th residual of the GMRES algorithm (i.e., y_{m-1} solves the problem (2.3.3)). With respect to (4.1.11), we have also discarded the modulus: indeed, formula (4.1.14) cannot produce negative values because $\phi_m(0) = r_m \leq r_{m-1} = \phi_{m-1}(0)$ (thanks to the optimality property of the GMRES residual (2.3.6)) and because $\phi_m(\lambda)$ is an increasing function with respect to $\lambda > 0$. In (4.1.14) we use the notation $\hat{\eta}$ instead of simply η as in (4.1.11) to underline that, in general, $\hat{\eta} \neq \eta$: the value of $\hat{\eta}$ depends on how fast the GMRES stabilizes and on how much it moves from the threshold ε (cf. Proposition 10). Of course, when, hypothetically, $\hat{\eta} = {}^{\eta\varepsilon}/_{r_{m-1}}$, we would obtain again formula (4.1.10).

We name this approach "embedded" for the following reason: when projecting the regularized problem into Krylov subspaces of increasing dimension, with very little computational overhead we can determine the residuals associated to the corresponding purely iterative projection method applied to the unregularized problem (cf. Remark 21); therefore the term "embedded" should underline the fact that, at each iteration, we can use the information provided when solving the projected linear system to set a good value for the parameter in the projected regularized problem. This approach is very general and could be potentially extended to any projection method.

While the original scheme (4.1.11) can simultaneously determine the value of the regularization parameter at each iteration and the number of iterations to be performed (cf. Remark 20), this is no more possible considering the rule (4.1.14). In order to determine when to stop the iterations of the Arnoldi algorithm, we have to consider a separate stopping criterion. Since both $\phi_m(\lambda_m)$ and r_m exhibit a stable behavior going on with the iterations, a way to set m is to monitor when such stability occurs, i.e. to evaluate the relative difference between the norm of the residuals and the relative difference between the discrepancy functions. Therefore, once two thresholds τ_{discr} and τ_{res} have been set, we decide to stop the iterations as soon as

$$\frac{r_m - r_{m-1}}{r_{m-1}} < \tau_{\text{res}} \tag{4.1.15}$$

and

$$\frac{\phi_m(\lambda_m) - \phi_{m-1}(\lambda_{m-1})}{\phi_{m-1}(\lambda_{m-1})} < \tau_{\text{discr}}. \tag{4.1.16}$$

This approach is very similar to the one adopted in [25] for the GCV method in a hybrid setting (cf. Section 4.3); also in [19] the authors decide to terminate the Arnoldi process when the corners of two consecutive projected L-curves are pretty close (cf. Section 4.2). As in the secant-update case, we can expect the value of λ_m obtained at the end of the iterations to be suitable for the original problem (1.3.4).

We summarize the method so far described in Algorithm 12.

Algorithm 12: AT method with an embedded approach

Inputs: A, b, L, x^*, λ_1, η, τ_{res}, τ_{discr}

For $m = 1, 2, \ldots,$ until both (4.1.15) and (4.1.16) are fulfilled

1. Update W_m and $\bar{H}_m$ by the Arnoldi algorithm (Algorithm 2).
2. Compute the reduced-dimension GMRES solution y_m by solving (2.3.3) and compute the corresponding residual norm r_m.
3. Compute the solution $y_{\lambda,m}$ of (3.2.2) or (3.2.6), taking

$$\begin{cases} \lambda = \lambda_1 & \text{if } m = 1, 2 \\ \lambda = \lambda_m & \text{otherwise} \end{cases}.$$

4. Compute the discrepancy $\phi_m(\lambda_m) = \|c - \bar{H}_m y_m\|$.
5. if $m \geq 2$ update λ_{m+1} by formula (4.1.14).

Compute $x_{\lambda,m} = W_m y_{\lambda,m}$.

To illustrate the behavior of the new parameter choice strategy (4.1.14), we report the results of some meaningful test problems. For all the experiments, Algorithm 12 is implemented with $\lambda_1 = 1$, $\eta = 1.02$, and $\tau_{\text{res}} = \tau_{\text{discr}} = 5 \cdot 10^{-2}$.

Example 1. First of all we consider some of the classical test problems described in Section 1.2.1. In particular, in Figure 4.1.14, we show some results for the problems `baart`, `foxgood`, `i_laplace`, `shaw`, whose coefficient matrices are of size 120×120 and whose available right-hand sides b's are affected by additive 0.1% additive Gaussian white noise e (i.e., $\widetilde{\varepsilon} = 10^{-3}$). The dimension of each problem is $n = 120$. For the test problems `i_laplace` and `shaw` we use the zero-padded version of the matrix D_1, denoted by D_1^{ZP} and obtained by appending one zero row to the matrix D_1 defined in (1.3.17); for the test problems `baart` and `foxgood`, we use the zero-padded version of the matrix D_2 (1.3.18), denoted by D_2^{ZP} and defined in (3.2.12).

As explained in the caption of Figure 4.1.14, for each experiment we show: the approximate solution, the relative residual and error history, and the value of the regularization parameter computed at each iteration by the secant update method (denoted by λ_{sec} and given by formula (4.1.11)), the embedded method (denoted by λ_{emb} and given by formula (4.1.14)), the one arising from the L-curve criterion (denoted by $\lambda_{L-\text{curve}}$ and defined in Section 4.2), and the optimal one (λ_{opt}) for the original, full-dimensional regularized problem (1.3.4) obtained by the minimization of the distance between the Tikhonov-regularized and the exact solution

$$\min_{\lambda} \|x_\lambda - x^{ex}\|^2 = \min_{\lambda} \left\| \sum_{i=1}^{q} \frac{\lambda^2}{(\gamma_i^2 + \lambda^2)} \frac{\widetilde{u}_i^T b}{\widetilde{\sigma}_i} x_i - \sum_{i=1}^{q} \frac{\widetilde{u}_i^T b^{ex}}{\widetilde{\sigma}_i} x_i \right\|,$$

where γ_i, $\widetilde{u}_i$, $i = 1, \ldots, q$ and x_i, $i = 1, \ldots, n$, are the generalized singular values/vectors of a generic matrix pair (A, L), as defined in (1.1.15). From the results displayed in the third column of Figure 4.1.14 we can clearly see that the sequence of the regularization parameters computed by both the secant update method (4.1.11) and the embedded method (4.1.14) exhibits a quite

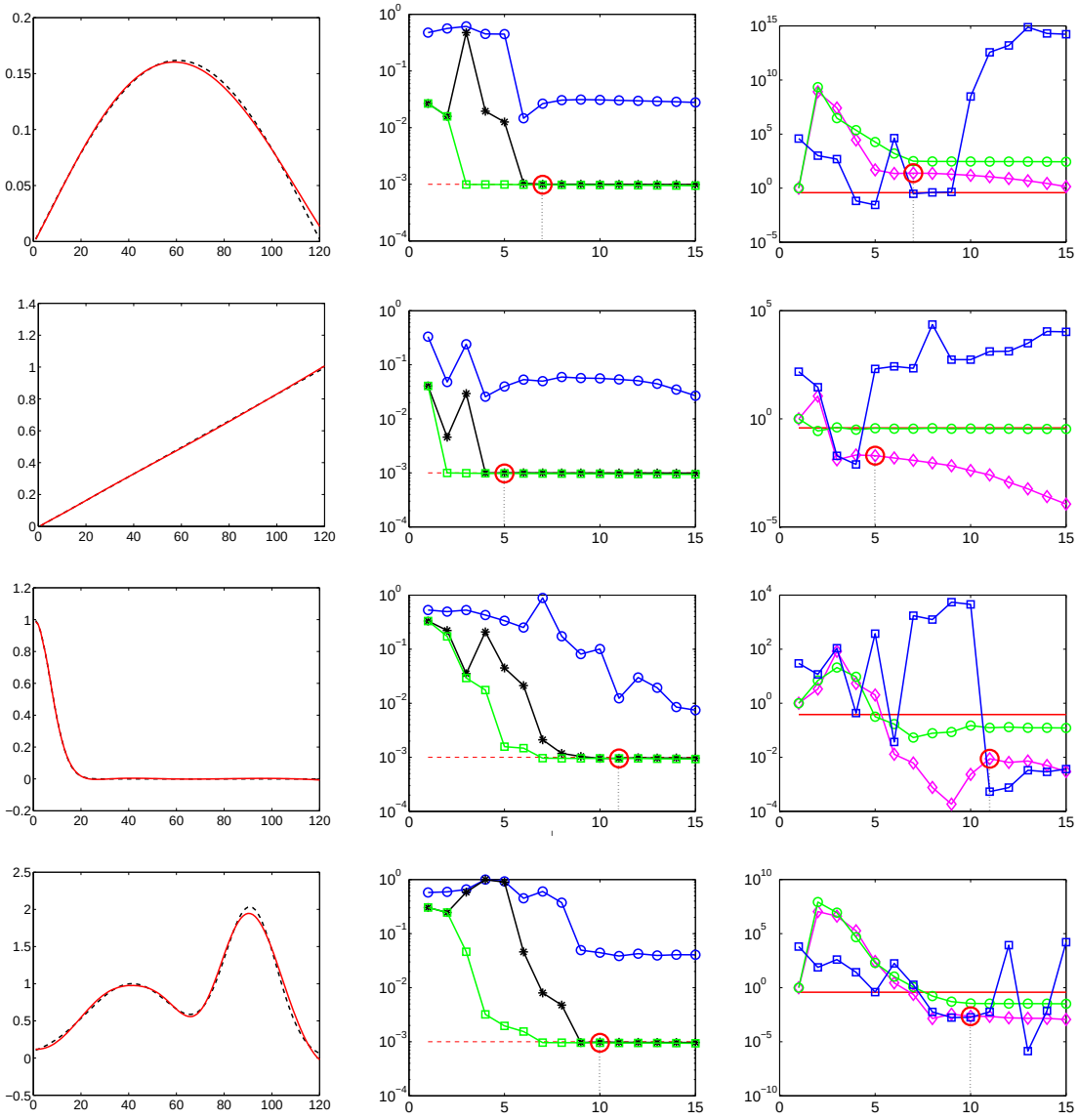

Figure 4.1.14: From top to bottom: results for `baart`, `foxgood`, `i_laplace`, `shaw`. On the left column we display the computed approximate solution (the dashed line marks the exact solution, the solid line the regularized one); in the middle column we show the convergence behavior of the new method (the relative error is marked by a circle, the relative discrepancy is marked by an asterisk, the relative GMRES residual is marked by a square, and the noise level is highlighted by a horizontal dashed line); on the right column we compare different parameter choice strategies (λ_{opt} is the horizontal solid line, λ_{emb} is marked by a diamond, λ_{sec} is marked by a circle, and $\lambda_{L-\text{curve}}$ is marked by a square). The big tick circle displayed in all the frames of the middle and the leftmost columns marks the iteration at which we would stop, according to the rule (4.1.15), (4.1.16).

stable behavior; the same is not true for the L-curve criterion (based on the algorithm described in [65]).

Example 2. We test the performance of our algorithm in the image restoration framework (Figures 4.1.15 and 4.1.16); in both the experiments the blur is Gaussian and it is defined using the function `blur` from [56]. In Figure 4.1.15 we take the test image `cameraman.tif`, which is a 256×256 pixels, 8-bit grayscale image, commonly used in image deblurring experiments. The parameters of the function `blur` are `s=2` and `q=7`; the noise level on the available right-hand side is $\widetilde{\varepsilon} = 10^{-3}$. As regularization operator we take the matrix $D_1^{v+h} = D_1^v + D_1^h$ already considered in Section 4.1.1, where D_1^v and D_1^h are given in (1.3.21). The restored image shown in Figure 4.1.15 has been obtained after 5 iterations of the Arnoldi algorithm, and the CPU-time required for this experiment is around 0.7 seconds. Many other experiments on image deblurring problems have shown similar performances.

Figure 4.1.15: Upper row, from left to right: original image, blurred and noisy image, reconstructed image obtained employing Algorithm 12 and stopping at the 5-th iteration of the Arnoldi algorithm. Lower row: blow-ups of the corresponding images displayed in the upper row.

In Figure 4.1.16 we reconstruct a medical test image of size 256×256 pixels, obtained by Magnetic Resonance Imaging (MRI): the available image is affected by Gaussian blur (the parameters of the function `blur` are `s=2.5` and `q=9`) and by Gaussian white noise whose level is $\widetilde{\varepsilon} = 10^{-1}$. The restored image is obtained at the 8-th iteration of the Arnoldi algorithm, adopting the regularization matrix D_1^{v+h} just described and choosing the parameter at each iteration according to the embedded strategy (4.1.14). The running time is around 1.2 seconds.

4.2 L-curve

The L-curve strategy was first proposed in [78], and later popularized and further developed in [65]: this method can be employed when a good estimate of $\|e\|$ is not available, and requires the computation of the norm of the discrepancy vector (4.1.1) and the norm of the regularized solution for many values of the regularization parameter. Basically, the L-curve is a logarithmic plot of the norm of the regularized solution versus the norm of the corresponding discrepancies,

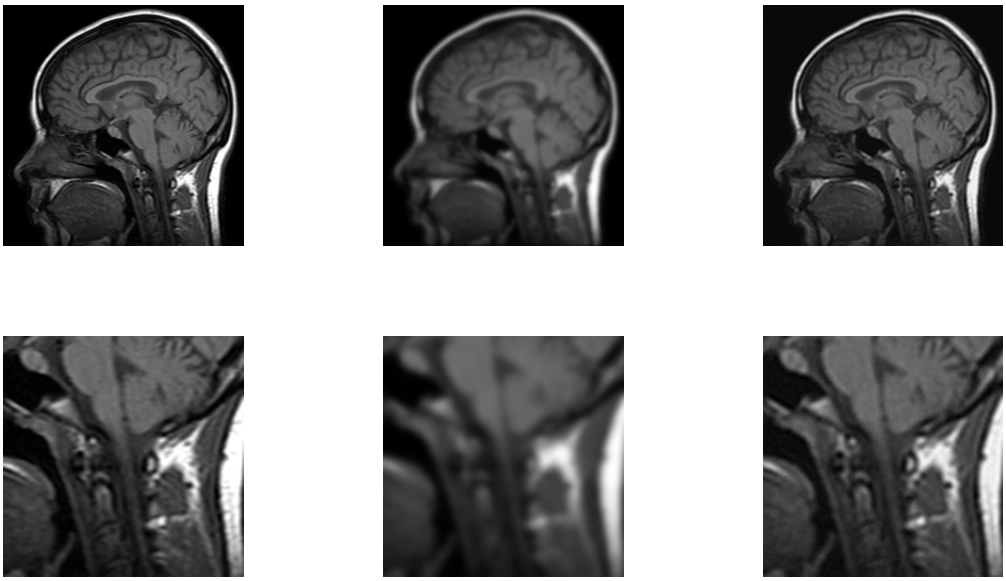

Figure 4.1.16: Upper row, from left to right: original image, blurred and noisy image, reconstructed image obtained employing Algorithm 12 and stopping at the 8-th iteration of the Arnoldi algorithm. Lower row: blow-ups of the corresponding images displayed in the upper row.

i.e., the points of the L-curve are of the form

$$(\log \|b - Ax_\alpha\|, \log \|x_\alpha\|)\,, \tag{4.2.1}$$

where α is, at least in principle, any admissible regularization parameter. In Figure 4.2.1 we show two examples of L-curves in the TSVD and in the Tikhonov case. The L-curve criterion prescribes to take, as suitable regularization parameter, the value of α corresponding to the corner of the L-curve.

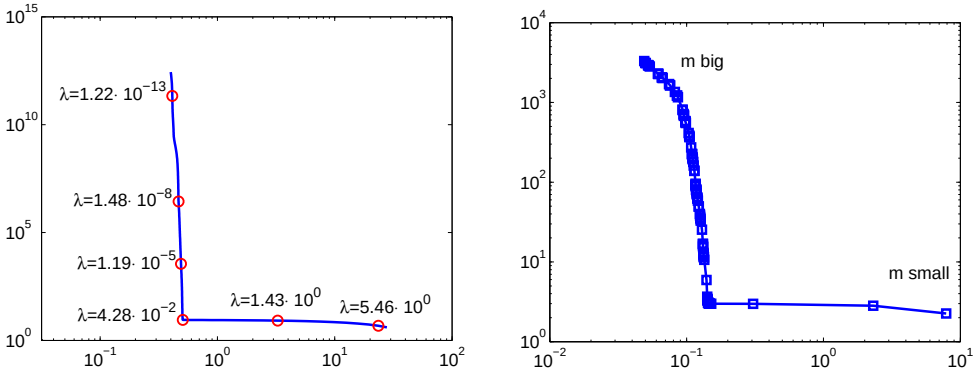

Figure 4.2.1: Leftmost frame: L-curve for Tikhonov regularization in standard form applied to the test problem `gravity`. Rightmost frame: L-curve for TSVD regularization applied to the test problem `phillips`. On the horizontal axis we consider the values of the norm of the discrepancy ($\log \|b - Ax_\alpha\|$) and on the vertical axis we consider values of the norm of the regularized solution ($\log \|x_\alpha\|$); in the Tikhonov case $\alpha = \lambda \in \mathbb{R}^+$ (giving rise to a continuous L-curve), in the TSVD case $\alpha = m \in \mathbb{N}$ (giving rise to a discrete L-curve).

It is clear that, if the reconstructed solution is over-regularized, then the norm of the solution is tiny and the norm of the discrepancy is huge; on the contrary, if the reconstructed solution is under-regularized, then the norm of the solution is huge and the norm of the discrepancy is tiny. Parameters leading to over-regularized solutions originate points laying on the bottom

right flat part of the L-curve plot; on the contrary, parameters leading to under-regularized solutions originate points laying on the top left vertical part of the L-curve plot. Therefore, the basic reasoning behind the L-curve is that a good value for the regularization parameter (i.e., a value that provides some balance between the norm of the discrepancy and the norm of the solution) should correspond to the points on the L-curve where the transition between the flat and the vertical part begins, i.e. on the corner of the L-curve. The corner of the L-curve corresponds to the point on the graph of the L-curve having the maximum "curvature"; once a reliable optimization routine is provided, a value of the regularization parameter relative to the corner can be determined. We underline that the norm of the discrepancies and the norm of the regularized solutions can be efficiently retrieved for many values of the regularization parameter if the the SVD of the matrix A (or, depending on the regularization method considered, the GSVD of some matrix pair (A, L)) is known and the regularization method is a spectral filtering method (1.3.2). Moreover, many techniques exist to efficiently estimate the corner without having to compute many points on the L-curve (cf. [65]). Of course, the L-curve criterion is quite heuristic and there are no guarantees that it could always properly work: indeed, for some problems, the transition between under-regularization and over-regularization is not very pronounced and the algorithms employed to find the corner of the L-curve could fail; moreover, more local corners (i.e. more points in which the curvature has a local maximum) could exist. In particular, when the norm of the discrepancy or the norm of the solution does not have a monotonic behavior (increasing for the discrepancy, decreasing for the solution) with respect to the amount of regularization, the L-curve (as a function of $\|x_\alpha\|$ with respect to $\|b - Ax_\alpha\|$) is not guaranteed to be monotonically decreasing and convex.

We briefly review how the L-curve criterion applies to the different classes of regularization methods introduced so far.

- **Tikhonov regularization**. In this case the L-curve is a continuous graph (cf. Figure 4.2.1, leftmost frame) and it can be proved that the norm of the discrepancy is an increasing function of λ and the norm of the regularized solution is a decreasing function of λ. If Tikhonov regularization in general form is involved, then we should plot the seminorm $\|Lx_\lambda\|$ of the regularized solution versus the norm of the discrepancy.

- **Truncated Singular Value Decomposition**. In this case the L-curve consists in a finite set of points that define the so-called discrete L-curve: of course, in this case, the "curvature" cannot be considered in a classical sense. An adaptive way to successfully locate the corner of the discrete L-curve has been derived in [62]. In the TSVD framework also some alterative L-curves have been proposed. We cite the so-called residual L-curve [103], which consists of a plot of the quantities

  $$(\log m, \log \|b - Ax_m\|) \text{ or } (m, \log \|b - Ax_m\|), \ m = 1, \ldots, n,$$

  and the so-called condition L-curve: since $\text{cond}(A_m) \leq \text{cond}(A_{m+1})$, this variation of the classical L-curve prescribes to plot the quantities

  $$(\log(\text{cond}(A_m)), \log \|b - Ax_m\|), \ m = 1 \ldots, n.$$

- **Iterative regularization**. As usual, we focus on the GMRES and the CGLS/LSQR methods. When regularizing by GMRES, the norm of the residual decreases (cf. Section 2.3.2), but the norm of the solution is not guaranteed to increase. Therefore, the classical L-curve could badly perform; for this reason, in [17] the authors have derived the condition L-curve, which is a plot of the quantities

  $$\left(\log(\text{cond}(\bar{H}_m)), \log \|b - Ax_m\|\right), \ m \geq 1;$$

recalling the arguments used in the proof of Corollary 7, we understand that $\text{cond}(\bar{H}_m)$ is an increasing function of m. When regularizing by CGLS/LSQR we know that the norm of the residual decreases and the norm of the solution increases as the iterations proceed [55, §6.3]. Therefore the discrete L-curve criterion could be successfully adopted to determine when the CG/LSQR iterations should be stopped. The residual L-curve just described for the TSVD regularization can be still used in connection with the GMRES, CGLS/LSQR methods.

- **Lanczos-based hybrid methods**. As in the discrepancy principle case, in [73] the authors suggest to employ the continuous L-curve just to set the value of the parameter λ at each iteration. Thanks to the orthogonality of the matrices involved in the Lanczos decomposition (2.4.7), at the m-th iteration of the Lanczos algorithm (Algorithm 8) one can equivalently consider the L-curve computed with respect to the projected quantities, i.e.,
$$\left(\log \|y_{m,\lambda}\|, \log \|c - \bar{B}_m y_{m,\lambda}\|\right).$$
- **Lanczos-Tikhonov and Arnoldi-Tikhonov methods**. When considering the Lanczos-Tikhonov methods, the so-called L-ribbon approach has been proposed in [13] and [19]: basically, at each iteration of the Lanczos algorithm (Algorithm 8), a lower and an upper bounds are derived for some fixed points laying on the original L-curve (4.2.1) and corresponding to candidate values of the regularization parameters for the full-dimensional problem. The bounds are obtained exploiting the connections between matrix functionals and Gaussian quadrature rules (cf. [44]) and they get tighter as the iterations proceed. The set of all the "boxes" defined by taking upper and lower bounds of the points on the original L-curve is called L-ribbon. When the upper and lower bounds are sufficiently close, the corner of the L-ribbon and the solution of the projected problem can be efficiently computed: in this way a suitable value for the regularization parameter and a suitable reconstructed solution are determined. An L-curve-like criterion was the first one to be considered in connection with the Arnoldi-Tikhonov method (indeed, it was the method proposed in [19] to set both the parameters λ and m when the Arnoldi-Tikhonov was first derived). At the m-th iteration of the Arnoldi algorithm, an L_m-curve
$$\left(\log \|y_{\lambda,m}\|, \log \|c - \bar{H}_m y_{\lambda,m}\|\right)$$
relative to the m-th projected problem (3.1.3) is considered for different values of the parameter λ. Because of the convergent behavior of the quantities $\|y_{\lambda,m}\|$ and $\|c - \bar{H}_m y_{\lambda,m}\|$ (cf. the arguments in Remark 19) as the iterations proceed, the L_m-curves tend to overlap after some step. In this way the corners located at each iteration tend to coincide and a stopping criterion can be derived by monitoring the distance between two consecutive corners. We can expect that the last parameter computed when the iterations are terminated is a good regularization parameter for the full-dimensional problem, too.

4.3 Generalized Cross Validation

The Generalized Cross Validation (GCV) is a well-known $\|e\|_2$-free parameter choice strategy that has been employed to set the regularization parameter when performing Tikhonov, TSVD, iterative or hybrid regularization [25, 55, 73]. The basic idea behind GCV is that a proper value of the regularization parameter should be able to accurately predict missing data. Applying some statistical reasoning, the Generalized Cross Validation prescribes to choose the parameter α that minimizes the following function
$$G(\alpha) = \frac{\left\|(I - AA_\alpha^\sharp)b\right\|_2^2}{\left(\text{trace}(I - AA_\alpha^\sharp)\right)^2}. \tag{4.3.1}$$

Following the analysis in [59], we explain how the function $G(\alpha)$ is derived when standard Tikhonov regularization ($\alpha = \lambda$ in formula (4.3.1))is employed to regularize a system whose coefficient matrix is $A \in \mathbb{R}^{k\times m}$, $k \geq n$. First of all, we can state that minimizing the quantity $\|b^{ex} - Ax_\lambda\|$, sometimes referred to as prediction error, delivers a suitable value for the regularization parameter. However, since in practice the vector b^{ex} is not available, we adopt a classical statistical technique called cross validation, which consists in separating the given data in two sets: one set is used to compute the solution that is then used to predict the elements in the other set. In particular, leaving out the i-th element of b to obtain the $(k-1)$-dimensional vector $b^{(i)}$ and leaving out the i-th row of A to obtain the $(k-1)\times n$ matrix $A^{(i)}$, we can compute the reduced-problem Tikhonov solution

$$x_\lambda^{(i)} = \left((A^{(i)})^T A^{(i)} + \lambda I_n\right)^{-1} (A^{(i)})^T b^{(i)} \in \mathbb{R}^n.$$

The element $b^{(i)}$ can then be estimated through the expression $A(i,:)x_\lambda^{(i)}$ and a good choice of λ should minimize the prediction errors for all data elements, i.e. we should consider

$$\min_\lambda \frac{1}{k}\sum_{i=1}^{k}\left(A(i,:)x_\lambda^{(i)} - b^{(i)}\right)^2.$$

The above problem can be proved to be equivalent to

$$\min_\lambda \frac{1}{k}\sum_{i=1}^{k}\left(\frac{A(i,:)x_\lambda - b^{(i)}}{1-t_{i,i}}\right)^2, \tag{4.3.2}$$

where the $t_{i,i}$'s are the diagonal elements of the matrix $A(A^TA+\lambda I_n)^{-1}A^T$. Furthermore, to make the latter problem invariant under permutations of the rows of A, we resort to generalized cross validation, which basically consists in replacing each diagonal element $t_{i,i}$ by their average. In this way, instead of (4.3.2), we consider

$$\min_\lambda \frac{1}{k}\sum_{i=1}^{k}\left(\frac{A(i,:)x_\lambda - b^{(i)}}{1-\text{trace}(A(A^TA+\lambda I)^{-1}A^T)/k}\right)^2,$$

which can be rewritten as

$$\min_\lambda \frac{k\|b - Ax_\lambda\|_2^2}{\left(k - \text{trace}(A(A^TA+\lambda I)^{-1}A^T)\right)^2}. \tag{4.3.3}$$

We note that the above objective function is indeed the one in (4.3.1) specified for the standard Tikhonov regularization case (recall the definition of $A_\lambda^\sharp$ given in (1.3.8)). The SVD of A should be available in order to efficiently compute the GCV function. Discarding the factor k, substituting $A = U\Sigma V^T$ and exploiting the orthogonality of the involved matrices, problem (4.3.3) can be rewritten as

$$\min_\lambda \frac{\sum_{i=1}^{n}\left(\frac{\lambda}{\sigma_i^2+\lambda}u_i^Tb\right)^2 + \sum_{i=n+1}^{k}(u_i^Tb)^2}{\left((k-n) + \sum_{i=1}^{n}\frac{\lambda}{\sigma_i^2+\lambda}\right)^2}. \tag{4.3.4}$$

When dealing with Tikhonov regularization, one drawback of this strategy is that the function $G(\lambda)$ may be pretty flat near the minimum; therefore a precise determination of a proper regularization parameter could be difficult (cf. [59] and Figure 4.3.1).
In some frameworks, for instance in [75], a modified version of the classical GCV criterion, called weighted GCV (w-GCV) has been considered. Introducing a weighting parameter $\omega > 0$,

which basically controls the amount of data that we are leaving out when deriving the usual GCV function, the new function to be minimized becomes

$$G(\omega, \lambda) = \frac{k\|b - Ax_\lambda\|_2^2}{(k - \omega \text{trace}(A(A^TA + \lambda I)^{-1}A^T))^2}. \tag{4.3.5}$$

Setting $\omega > 1$ results in smoother solutions, and therefore this is the right choice when the classical GCV determines under-regularized solutions; choosing $\omega < 1$ we obtain less smooth solutions and this can be useful when the GCV tends to over-regularize the solution; for $\omega = 1$ we obtain the usual GCV function (4.3.3).
Before deriving a GCV parameter choice strategy for the AT method, we review the expressions of the GCV functions for the most well-known regularization techniques.

- **Tikhonov Regularization (general form)**. Let us assume that $L \in \mathbb{R}^{q\times n}$, $q \leq n$. Recalling, once again, the definition of $A_\lambda^\sharp$ given in (1.3.8), an efficient expression for $G(\lambda)$ can be derived considering the GSVD (1.1.15) of the matrix pair (A, L). After some algebraic manipulations we obtain

$$G(\lambda) = \frac{\sum_{i=1}^q \left(\frac{\lambda}{\gamma_i^2+\lambda} u_i^T b\right)^2 + \sum_{i=n+1}^k (u_i^T b)^2}{\left((k-n) + \sum_{i=1}^q \frac{\lambda}{\gamma_i^2+\lambda}\right)^2}. \tag{4.3.6}$$

- **Truncated Singular Value Decomposition**. Recalling the definition of $A_m^\sharp$ given in (1.3.29) we can derive the expression

$$G(m) = \frac{\sum_{i=m+1}^k (u_i^T b)^2}{(k-m)^2} \tag{4.3.7}$$

  by directly substituting $A_m^\sharp$ into (4.3.1)

- **Iterative Regularization**. As usual, we exclusively consider regularization by projection and, in particular, we focus on the CGLS/LSQR and the GMRES methods. Recalling the expressions (2.6.1) and (2.6.2) for the regularized inverse associated to the LSQR and GMRES methods, respectively, and exploiting the properties of the matrices involved in the decompositions (2.4.7) and (2.2.3), we can write

$$G(m) = \frac{\|c - \bar{B}_m y_m\|^2}{(k-m)^2} \tag{4.3.8}$$

  in the LSQR case, and

$$G(m) = \frac{\|c - \bar{H}_m y_m\|^2}{(k-m)^2} \tag{4.3.9}$$

  in the GMRES case.

- **Lanczos-based hybrid methods**. According to [73], in this setting the GCV is just employed to choose the parameter $\lambda = \lambda_m$ at the m-th iteration of the Arnoldi algorithm, and therefore the function $G(m, \lambda) = G_m(\lambda)$ is just minimized with respect to λ. Recalling the expression (2.6.4) for the regularized inverse associated to the Lanczos-hybrid method, we obtain

$$G_m(\lambda) = \frac{\|c - \bar{B}_m y_{m,\lambda}\|_2^2}{\left(m + 1 - \sum_{i=1}^m \frac{\left(\sigma_i^{(m)}\right)^2}{\left(\sigma_i^{(m)}\right)^2+\lambda}\right)^2}, \tag{4.3.10}$$

where $\sigma_i^{(m)}$, $i = 1, \ldots, m$ are the singular values of the matrix $\bar{B}_m$. A very successful version of the w-GCV criterion has been considered in [25] in connection with Lanczos-hybrid methods: the authors fully exploit the iterative setting of the Lanczos algorithm to adaptively define the weight $\omega = \omega_m$ in order to obtain a value for λ_m close to the optimal one.

We now determine a formulation of the generalized cross validation to be used in connection with the Arnoldi-Tikhonov method; our considerations are mainly based on the recent paper [95]. As already underlined in many occasions, when performing AT we should set two regularization parameters: the continuous parameter λ, which appears in (3.2.6) and that has to be updated at each step of the Arnoldi algorithm, and the number of iterations m. As in the Lanczos-hybrid case, we use the GCV strategy just to compute a proper value of $\lambda = \lambda_m$ at each step of the Arnoldi algorithm and we denote the GCV function by $G_m(\lambda)$; in order to stop the iterations we consider a standard stopping criterion. We work directly on the general case, assuming $L \in \mathbb{R}^{q \times n}$ to be the regularization matrix for the full-dimensional problem and defining its projections $L_m \in \mathbb{R}^{m \times m}$ (3.2.5) at each step of the Arnoldi algorithm. In this case, the expression for the regularized inverse of A is given in (3.2.8). For a fixed m, let us consider the GSVD of the matrix pair $(\bar{H}_m, L_m)$ that, adopting the notations of Section 3.2, can be denoted by

$$\bar{H}_m = U^{(m)} \Sigma^{(m)} (X^{(m)})^{-1} \quad \text{and} \quad L_m = V^{(m)} M^{(m)} (X^{(m)})^{-1}, \tag{4.3.11}$$

where $U^{(m)} \in \mathbb{R}^{(m+1) \times m}$ and $V^{(m)} \in \mathbb{R}^{m \times m}$ have orthonormal columns,

$$\Sigma^{(m)} = \text{diag}(\sigma_1^{(m)}, \ldots, \sigma_m^{(m)}) \in \mathbb{R}^{m \times m}, \quad M^{(m)} = \text{diag}(\mu_1^{(m)}, \ldots, \mu_m^{(m)}) \in \mathbb{R}^{m \times m},$$

and $\gamma_i^{(m)} = \sigma_i^{(m)} / \mu_i^{(m)}$, $i = 1, \ldots, m$, are the generalized singular values of $(\bar{H}_m, L_m)$. Substituting (3.2.8) and (4.3.11) into (4.3.1) we can easily see that, in the AT case, the numerator of $G_m(\lambda)$ is given by the following expression

$$\begin{aligned} \left\| (I - AA^\sharp) b \right\|_2^2 &= \left\| c - \bar{H}_m y_{\lambda,m} \right\|_2^2 \\ &= \sum_{i=1}^{m} \left(\frac{\lambda}{\gamma_i^{(m)2} + \lambda} u_i^{(m)T} c \right)^2 + \left(u_{m+1}^{(m)T} c \right)^2. \end{aligned} \tag{4.3.12}$$

Similarly, using the compact notation $\bar{H}_{\lambda,m}^\dagger = (\bar{H}_m^T \bar{H}_m + \lambda L_m^T L_m)^{-1} \bar{H}_m^T$ and considering that

$$\left(I_n - AA^\sharp \right) = \left(I_n - W_{m+1} \bar{H}_m \bar{H}_{\lambda,m}^\dagger W_{m+1}^T \right),$$

we obtain the following expression for the denominator of $G_m(\lambda)$:

$$\begin{aligned} & \text{trace} \left(I_n - W_{m+1} \bar{H}_m \bar{H}_{m,\lambda}^\dagger W_{m+1}^T \right) \\ = \; & \text{trace}(I_n) - \text{trace}(\bar{H}_m \bar{H}_{m,\lambda}^\dagger W_{m+1}^T W_{m+1}) \\ = \; & n - \sum_{i=1}^{m} \frac{\gamma_i^{(m)2}}{\gamma_i^{(m)2} + \lambda} = (n - m) + \sum_{i=1}^{m} \frac{\lambda}{\gamma_i^{(m)2} + \lambda}. \end{aligned} \tag{4.3.13}$$

Therefore, at m-th iteration of the Arnoldi algorithm, the regularization parameter $\lambda = \lambda_m$ is determined by solving the problem

$$\min_\lambda G_m(\lambda), \quad G_m(\lambda) = \frac{\sum_{i=1}^{m} \left(\frac{\lambda}{\gamma_i^{(m)2} + \lambda} u_i^{(m)T} c \right)^2 + \left(u_{m+1}^{(m)T} c \right)^2}{\left((n - m) + \sum_{i=1}^{m} \frac{\lambda}{\gamma_i^{(m)2} + \lambda} \right)^2}. \tag{4.3.14}$$

When, in particular, $L = I_n$, we just take the singular values $\sigma_i^{(m)}$'s of $\bar{H}_m$ instead of the $\gamma_i^{(m)}$'s. As suggested in [95], the expression of $G_m(\lambda)$ can also be recovered directly from (4.3.6) and coherently with the approximation properties of the Arnoldi algorithm stated in Section 3.2 and in Proposition 17: it just suffices to replace the first m generalized singular values of (A, L) by the singular values of $(\bar{H}_m, L_m)$ and set the remaining ones to zero.

Remark 22. Looking at the denominator of (4.3.14) we can appreciate the contribution of two terms: the first one, $(n-m)$, is typical of the iterative regularization (4.3.9), while the second one (the sum) is connected with Tikhonov regularization (4.3.6). Of course, if just a few iterations m are performed, the term $(n-m)$ dominates. This fact gives us further insight into the nature of the AT method: both an iterative and a direct approach contribute to define its regularizing properties, supporting the fact that AT method essentially aims at projecting a regularized problem. Comparing (4.3.14) to the GCV function for Lanczos-hybrid regularization (4.3.10) we note that, in the latter, the sum is still present while the term $n - m$ is simply substituted by $m + 1$; this fact confirms that hybrid methods exclusively aim at regularizing each iteration, without any explicit reference to the original full-dimensional regularized problem (1.3.10).

We note that this strategy is well-suited for large scale problems, since the only GSVD required is the one of the projected matrices: being typically $m \ll n$, the computational overload to set the parameter at each iteration is negligible. We can be confident that a good value for $\lambda = \lambda_m$ is quickly determined if, after just a few iterations, the GSVD of $(\bar{H}_m, L_m)$ is a good approximation of the GSVD of (A, L) (cf. the arguments at the end of Section 3.2). To decide when to stop the Arnoldi iterations we monitor the progression of the norm of the discrepancies: since they are supposed to stabilize when a good value for λ has been found and when the Krylov subspace has grown enough, we fix a threshold $\tau > 0$ and stop as soon as

$$\frac{\|\phi_m(\lambda_m)\| - \|\phi_{m-1}(\lambda_{m-1})\|}{\|\phi_m(\lambda_m)\|} < \tau. \tag{4.3.15}$$

The default value for τ is 10^{-2}. We summarize the previous derivation in the Algorithm 13.

Algorithm 13: GCV-AT algorithm

Input: A, b, L, $\tau > 0$
For $i = 1, 2, \ldots,$ until (4.3.15) is satisfied

1. Update W_m, $\bar{H}_m$ and L_m by performing a step of the Arnoldi algorithm (Algorithm 2).
2. Compute the GSVD of $(\bar{H}_m, L_m)$.
3. Compute $\lambda_m = \arg\min_\lambda G_m(\lambda)$.
4. Compute the solution $y_{\lambda_m,m}$ of

$$\min_{y \in \mathbb{R}^m} \left\| \begin{pmatrix} \bar{H}_m \\ \sqrt{\lambda_m} L_m \end{pmatrix} y - \begin{pmatrix} c \\ 0 \end{pmatrix} \right\|^2$$

and compute the corresponding discrepancy $\phi_m(\lambda_m)$.

Compute $x_{\lambda_m,m} = W_m y_{\lambda_m,m}$.

In order to assess the performance of the GCV-AT algorithm and to compare it with the previously described ones, we report the results of some numerical experiments.

Example 1. We consider the test problems `gravity` and `foxgood`; the size of the matrix A is 200×200 and the right-hand-side vector is perturbed by adding gaussian white noise, whose level is 10^{-2}. In particular, for `gravity` we consider regularization in standard form,

so that we can easily apply GCV to all the main regularization methods listed above. In Figure 4.3.1 we summarize the results obtained in the Tikhonov, TSVD, LSQR, Lanczos-hybrid (implemented without reorthogonalization) and Arnoldi-Tikhonov cases: we show the computed reconstructions, we plot the GCV functions in logarithmic scale and, for the iterative methods, we display the history of some meaningful quantities, such as the relative errors and the norm of the discrepancies at each iteration (in the the GCV-AT case we display these values also after the stopping criterion has been satisfied, the iteration at which this happens being marked by a circle). The results are also summarized in Table 4.5. Looking at the graphs in Figure

Method	Relative Error	λ	m
Tikhonov	$4.2130 \cdot 10^{-2}$	$1.2090 \cdot 10^{-2}$	-
TSVD	$4.6217 \cdot 10^{-2}$	-	6
LSQR	$4.0413 \cdot 10^{-2}$	-	6
Lanczos-hybrid	$5.3091 \cdot 10^{-2}$	$3.8097 \cdot 10^{-1}$	10
AT	$4.3344 \cdot 10^{-2}$	$9.3325 \cdot 10^{-2}$	9

Table 4.5: Performance of some regularization methods equipped with GCV when applied to the problem `gravity` affected by 1% noise.

4.3.1, frame **(k)**, we can state that the GCV function $G_m(\lambda)$ is quite flat near the minimum, especially during the first iterations. We can explain this feature recalling that, performing the AT method, regularization occurs also projecting the problem into a Krylov subspace and this is indeed more apparent during the first iterations, i.e. when the projected matrix is quite well-conditioned. However we can appreciate that the projected GCV function quickly approximates the full-dimensional one and, as a consequence, the selected parameters λ_m's stabilize around a common value that is suitable for the original regularized problem, too. This behavior is not recovered for Lanczos-hybrid methods (frame **(f)**): this supports our statement that the AT method is more deeply linked to the original regularized problem than Lanczos-hybrid method. Anyway, the sequence of the relative errors and the sequence of the regularization parameters exhibit a convergent behavior for both the AT and the Lanczos-hybrid methods, even if it happens more rapidly in the AT case. Regrettably, the values of the relative errors are not very stable at the beginning of the AT iterations, but this does not affect the final solution. For the problem `foxgood` we consider, as regularization matrix, the zero-padded matrix D_2^{ZP} defined in (3.2.12). This is the best possible choice, since in this case the exact solution belongs to the null space of the regularization matrix (recall the discussion in Section 1.3.1). As a consequence, we also expect the value of the regularization parameter to be quite high: indeed, in the Generalized Tikhonov regularization case, we have $\lambda = 11.2481$ and for the AT method we have $\lambda_3 = 34.6737$. The performances of the Generalized Tikhonov and of the GCV-AT methods are shown in Figure 4.3.2. The relative error associated to Tikhonov regularization is $2.0821 \cdot 10^{-2}$, while at the 3-rd iteration of the AT method we get $3.098 \cdot 10^{-2}$. Remarkably, for this test problem, the sequence of the relative errors associated to the AT method keeps decreasing and this allow a very accurate reconstruction at the end of the process; probably this behavior is due to the optimal choice of the regularization matrix.

Example 2. We apply the newly derived Algorithm 13 to an image restoration problem (Figure 4.3.3). The ideally exact image is showed in frame **(a)**. We corrupt it applying a symmetric Gaussian blur, using the function `blur` with parameters `s=3` and `q=8` and adding some Gaussian white noise whose level is 10^{-3}: in this way we obtain the blurred and noisy image displayed in frame **(b)**. To restore it we apply the GCV-AT method, considering, as regularization matrix, $D_1^{v+h} = D_1^v + D_1^h$, where D_1^h and D_1^v are defined in (1.3.21). In frame **(c)** we display the result obtained at the 30-th iteration; the corresponding regularization parameter is $5.1286 \cdot 10^{-5}$ and the relative error is

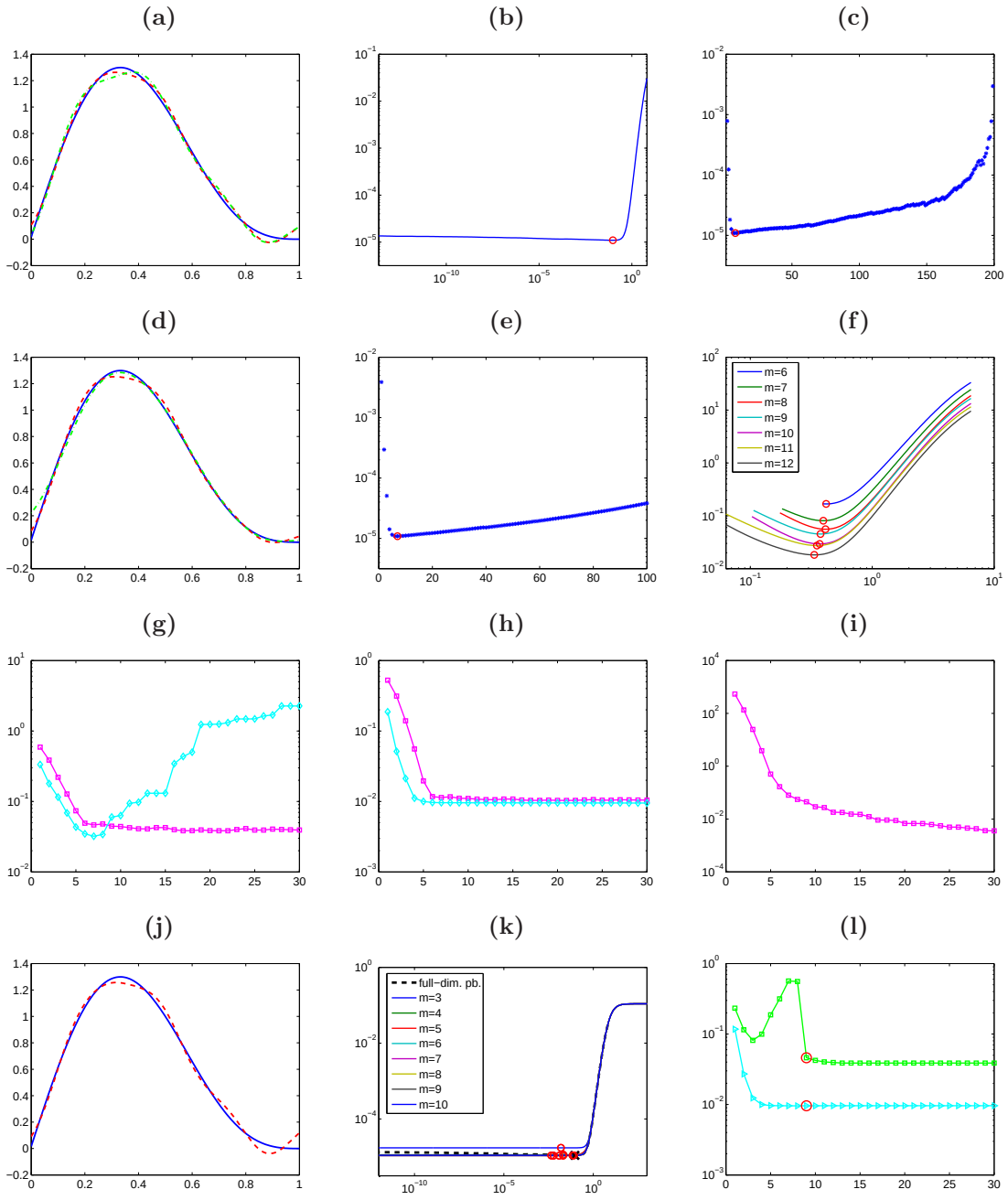

Figure 4.3.1: Test problem `gravity`. **(a)** exact (solid line), GCV-Tikhonov (dashed line) and GCV-TSVD (dash-dot line) reconstructed solutions; **(b)** GCV function for Tikhonov method; **(c)** GCV function for TSVD method; **(d)** exact (solid line), LSQR (dashed line) and Lanczos-hybrid (dash-dot line) reconstructed solutions; **(e)** GCV function for LSQR method; **(f)** GCV functions for some iterations of the Lanczos-hybrid method; **(g)** relative errors versus the number of iterations for LSQR (diamond markers) and Lanczos-hybrid (square markers) methods; **(h)** relative discrepancies versus the number of iterations for LSQR (diamond markers) and Lanczos-hybrid (square markers) methods; **(i)** values of λ_m versus the number of iterations for Lanczos-hybrid method; **(j)** exact (solid line) and GCV-AT reconstructed (dashed line) solutions; **(k)** GCV functions for some iterations of the AT method; **(l)** relative errors (square markers) and relative discrepancies (triangle markers) versus the number of iterations for the GCV-AT method.

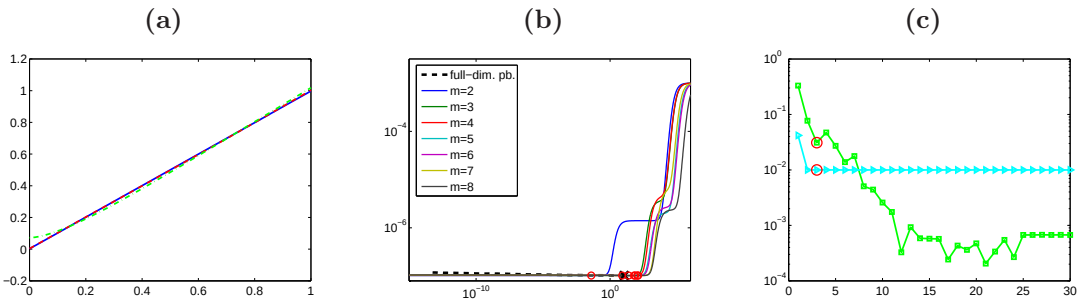

Figure 4.3.2: Test problem `foxgood`. **(a)** exact (solid line), GCV-Tikhonov (dashed line) and GCV-AT (dash-dot line) reconstructed solutions; **(b)** GCV functions for some iterations of the AT method; **(c)** relative errors (square markers) and relative discrepancies (triangle markers) versus the number of iterations for the GCV-AT method.

$1.6511 \cdot 10^{-1}$.

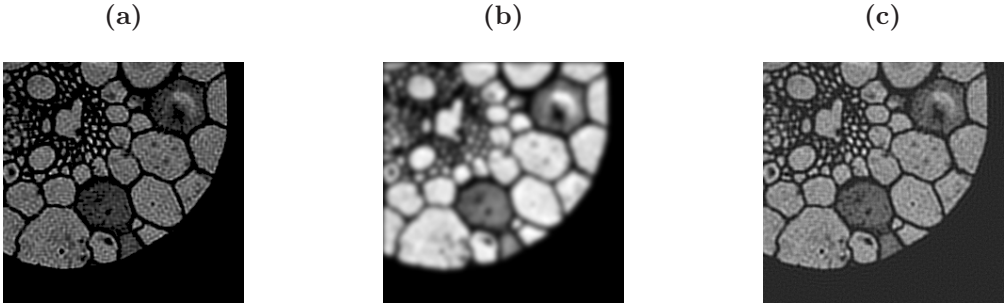

Figure 4.3.3: **(a)** exact test image; **(b)** blurred and noisy image (0.1% noise); **(c)** reconstruction obtained at the 30-th iteration of the GCV-AT method.

4.4 The multi-parameter case

In the framework of Tikhonov regularization method (1.3.4), the use of multi-parameter regularization (even called multiple penalty regularization) has been basically introduced with the aim of acting simultaneously on different frequency bands of the solution, in the hope of reproducing all the basic features of the unknown solution with a good accuracy. Due to the wide range of applications, there is a growing interest in this kind of regularization, and many numerical schemes have been recently presented in various contexts (we cite [82] and the references therein for an overview).

Multi-parameter Tikhonov regularization is characterized by a vector of regularization parameter $\Lambda = (\lambda_1, \ldots, \lambda_k)^T$ ($\lambda_i \geq 0$, $i = 1, ..., k$, $\Lambda \neq 0$) and by a k-tuple of regularization matrices $\mathcal{L} = \{L_1, \ldots, L_k\}$; the regularized solution x_Λ is defined by

$$x_\Lambda = \arg \min_{x \in \mathbb{R}^n} \mathcal{M}(x, \Lambda, \mathcal{L}), \tag{4.4.1}$$

where

$$\mathcal{M}(x, \Lambda, \mathcal{L}) = \|b - Ax\|^2 + \sum_{i=1}^{k} \lambda_i \|L_i x\|^2 .$$

We remark that, when treating multi-parameter methods, one usually requires that each component of the vector Λ is different from zero, in order to make sure that every regularization term affects the approximate solution. However, we prefer to present the multi-parameter Tikhonov

method like a generalization of the one-parameter one, and therefore the only condition is that at least one regularization parameter is different from 0. Note that, since the regularized solution also depends on the set of matrices $\mathcal{L}$, one should more rigorously denote it by $x_{\Lambda,\mathcal{L}}$. However, in the following, we just focus on setting the vector Λ and we assume $\mathcal{L}$ to be fixed; to keep the notations simple we omit the subscript $\mathcal{L}$.

While the multi-parameter regularization is theoretically superior to any single-parameter regularization that uses just one of the matrices L_i in (4.4.1), the main issue is that, in practice, it may be quite difficult to work simultaneously with more than one regularization matrix and to suitably define the parameters λ_i's. The existing methods for choosing the the regularization parameters are essentially based on the generalized L-curve criterion (cf. [7]) and on the generalization of the GCV criterion (cf. [12]). More recently, an algorithm based on the knowledge of the noise structure has been introduced in [5]. Assuming that a fairly accurate estimate of η is available, the authors of [82] introduce an algorithm for the definition of the regularization parameters based on the numerical solution with respect to Λ of the discrepancy equation

$$\|b - Ax_\Lambda\| = \eta\varepsilon, \quad \eta > 1.$$

Up to now, to the best of our knowledge, such technique seems to be the only existing one based on the discrepancy principle. In this section, we introduce a new algorithm that allow to extend the AT method (Section 3.2) to work with multi-parameter Tikhonov regularization (4.4.1) and that employs a parameter choice strategy based on the secant update method (Section 4.1.1): indeed, we may refer to it as vectorial secant update method.

4.4.1 Problem formulation

Repeating the steps explained in Sections 3.1 and 3.2, the multi-parameter Arnoldi-Tikhonov method is derived by looking for an approximation of the solution of the problem (4.4.1) belonging to the space $\mathcal{K}_m(A, b)$. More precisely, replacing $x = W_m y$ ($y \in \mathbb{R}^m$) into (4.4.1) and using (2.2.3), we obtain

$$\begin{aligned} & \min_{x \in \mathcal{K}_m(A,b)} \mathcal{M}(x, \Lambda, \mathcal{L}) \\ = & \min_{y \in \mathbb{R}^m} \left\{ \left\| \|b\| e_1 - \bar{H}_m y \right\|^2 + \sum_{i=1}^{k} \lambda_i \|L_i W_m y\|^2 \right\} \qquad (4.4.2) \\ = & \min_{y \in \mathbb{R}^m} \left\| \begin{pmatrix} \bar{H}_m \\ \sqrt{\lambda_1} L_1 W_m \\ \vdots \\ \sqrt{\lambda_k} L_k W_m \end{pmatrix} y - \begin{pmatrix} \|b\| e_1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \right\|^2 . \qquad (4.4.3) \end{aligned}$$

In the sequel we will refer to (4.4.2) and (4.4.3) as penalized and regularized least squares formulation of the multi-parameter Arnoldi-Tikhonov method, respectively. In this setting, let us denote by $x_\Lambda^{(m)}$ the m-th approximation arising from the Arnoldi-Tikhonov process, while the reduced-dimension minimizer of (4.4.3) is denoted by $y_\Lambda^{(m)}$. In the following, we adopt some notations that are related, but not identical, to the one-parameter ones; indeed, we try to keep the notations as immediate as possible, but sometimes they could result a bit overloaded because of the different parameters we have to take into account.

Exactly as in the one-parameter case, the projected problem (4.4.3) can be derived even when the regularization matrices are rectangular: in this case, the size of the matrix in (4.4.3) is $\left((m+1) + \sum_{i=1}^{k} q_i\right) \times m$, if $L_i \in \mathbb{R}^{q_i \times n}$, $i = 1, \ldots, k$. However, we could also adopt a strategy similar to the alternative described in Section 3.2, i.e., we can consider the projection of each matrix L_i, $i = 1, \ldots, k$, into the Krylov subspace $\mathcal{K}_m(A, b)$ and replace it by

$$L_i^{(m)} = W_{m+1}^T L_i W_m \quad i = 1, \ldots, k. \qquad (4.4.4)$$

In this way, the size of the matrix in (4.4.3) is further reduced to

$$((m+1)+k\cdot m)\times m.$$

We remark that, also in the multi-parameter setting, if an initial approximation x^* of the solution x^{ex} is available, then we can incorporate it into the multi-parameter Arnoldi-Tikhonov scheme by defining the initial residual $r^* = b - Ax^*$ and by considering the Krylov subspaces $\mathcal{K}_m(A, r^*)$. Consequently, the approximate solution of the problem (4.4.1) is of the form $x_\Lambda^{(m)} = x^* + W_m y_\Lambda^{(m)}$ (where the columns of W_m span the Krylov subspace $\mathcal{K}_m(A, r^*)$) and we simply have to substitute b with r^* in the expressions (4.4.2), (4.4.3).

4.4.2 Vectorial secant update method

As already addressed, the algorithm here proposed to define Λ at the m-th step of the Arnoldi algorithm (Algorithm 2) and to stop the underlying iterative procedure, is based on the solution of

$$\left\| b - Ax_\Lambda^{(m)} \right\| = \eta\varepsilon \tag{4.4.5}$$

by means of a linear approximation (with respect to each parameter λ_i, $i = 1, ..., k$) of the function

$$\phi^{(m)}(\Lambda) = \left\| b - Ax_\Lambda^{(m)} \right\|;$$

this strategy was first proposed in [39]. This method generates a sequence of regularization vectors $\Lambda^{(m)}$, $m \geq 1$, whose components $\lambda_i^{(m)}$ are automatically defined. The idea extends the one described in Section 4.1.1 for the single-parameter case: basically, at the m-th iteration of the Arnoldi algorithm, we simultaneously solve the problem (4.4.3) employing an approximation $\Lambda^{(m)}$ of the vectors of the regularization parameters and we sequentially define the new vector $\Lambda^{(m+1)}$ to be considered during the next iteration; as in the one-parameter case, the initial vector $\Lambda^{(1)}$ is provided by the user.

Also in the multi-parameter case, we immediately note that, since for the AT method the approximations are of the form $x_\Lambda^{(m)} = W_m y_\Lambda^{(m)} \in \mathcal{K}_m(A, b)$, the discrepancy can be rewritten as

$$\phi^{(m)}(\Lambda) = \|b - AW_m y_\Lambda^{(m)}\| = \|c - \bar{H}_m y_\Lambda^{(m)}\|, \tag{4.4.6}$$

i.e., we can equivalently consider a projected discrepancy.

As pointed out by many works in literature (cf. [12] and [82]), the most natural way to face a multi-parameter problem is to first solve some single-parameter problems, one for each regularization matrix, and then to find a connection between all the problems. In our case, at the m-th step of the Arnoldi-Tikhonov algorithm and for a given j, $1 \leq j \leq k$, we consider the problem

$$\min_{y\in\mathbb{R}^m} \left\| \begin{pmatrix} \bar{H}_m \\ \sqrt{\lambda_1^{(m+1)}} L_1 W_m \\ \vdots \\ \sqrt{\lambda_{j-1}^{(m+1)}} L_{j-1} W_m \\ \sqrt{\lambda} L_j W_m \end{pmatrix} y - \begin{pmatrix} c \\ 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix} \right\|^2, \tag{4.4.7}$$

which is a j-parameter Arnoldi-Tikhonov scheme; it can also be regarded as a reduced version of the system (4.4.3), where the corresponding regularization vector is

$$\Lambda_j = \left((\Lambda_{j-1}^{(m+1)})^T, \lambda, 0, \ldots, 0 \right)^T \in \mathbb{R}^k, \tag{4.4.8}$$

where

$$\Lambda_{j-1}^{(m+1)} = (\lambda_1^{(m+1)}, \ldots, \lambda_{j-1}^{(m+1)})^T \in \mathbb{R}^{j-1}.$$

This notation underlines the fact that we have already solved, in a sequential way, $(j-1)$ reduced problems obtained adding to the unregularized projected problem

$$\min_{y\in\mathbb{R}^m} \left\|c - \bar{H}_m y\right\| \tag{4.4.9}$$

a new regularization term and that, for the problems (4.4.7) so far considered, we have already updated the regularization parameters $\lambda_1^{(m+1)},\ldots,\lambda_{j-1}^{(m+1)}$ to be employed during the next iterations. Therefore, the present task is to determine the parameter $\lambda_j^{(m+1)}$; since we only have to update one parameter, we can resume the strategy employed for the single parameter AT method. We define the function

$$\phi_j^{(m)}(\lambda) = \phi^{(m)}(\Lambda_j) = \left\|c - \bar{H}_m y_{\Lambda_j}^{(m)}\right\|, \tag{4.4.10}$$

where Λ_j is defined as in (4.4.8) and $y_{\Lambda_j}^{(m)}$ is the solution of (4.4.7). In this framework, the normal equations associated to the problem (4.4.7) are

$$\left(\bar{H}_m^T\bar{H}_m + \sum_{i=1}^{j-1}\lambda_i^{(m+1)}W_m^TL_i^TL_iW_m + \lambda W_m^TL_j^TL_jW_m\right) y_{\Lambda_j}^{(m)} = \bar{H}_m^Tc.$$

As before, we are looking for a linear approximation, with respect to the parameter λ, of the discrepancy associated to the reduced multi-parameter problem so far considered, i.e.,

$$\phi_j^{(m)}(\lambda) \approx \phi_j^{(m)}(0) + \lambda d_j^{(m)}. \tag{4.4.11}$$

According to our notation, evaluating the function $\phi_j^{(m)}$ in zero in the multi-parameter case means taking

$$\phi_j^{(m)}(0) = \left\|c - \bar{H}_m\left(\bar{H}_m^T\bar{H}_m + \sum_{i=1}^{j-1}\lambda_i^{(m+1)}W_m^TL_i^TL_iW_m\right)^{-1}\bar{H}_m^Tc\right\|. \tag{4.4.12}$$

Observing the above expression we see that now, in general, we have to deal with the discrepancy associated to the $(j-1)$-parameter method with vector of the regularization parameters given by $\Lambda_{j-1}^{(m+1)}$. Using the definition (4.4.10) we also have

$$\phi_j^{(m)}(0) = \phi^{(m)}(\Lambda_{j-1}^{(m+1)}). \tag{4.4.13}$$

We emphasize that, to obtain the quantity $\phi_j^{(m)}(0)$, we have to solve again the $(j-1)$-parameter problem with the regularization vector given by $\Lambda_{j-1}^{(m+1)}$. Of course, when $j=1$, the determination of $\lambda_1^{(m+1)}$ again requires the solution of the problem (4.4.9) as in the one-parameter case, i.e., $\phi_1^{(m)}(0)$ is still the norm of the residual of the GMRES method.

Concerning the quantity $d_j^{(m)}$, once we have solved (4.4.7) for $\lambda = \lambda_j^{(m)}$, we obtain

$$\phi_j^{(m)}(\lambda_j^{(m)}) = \left\|c - \bar{H}_m y_{\Lambda_j}^{(m)}\right\|, \quad \Lambda_j = \left((\Lambda_{j-1}^{(m+1)})^T, \lambda_j^{(m)}, 0, \ldots, 0\right)^T, \tag{4.4.14}$$

and consequently, using the approximation (4.4.11), we get

$$d_j^{(m)} = \frac{\phi_j^{(m)}(\lambda_j^{(m)}) - \phi_j^{(m)}(0)}{\lambda_j^{(m)}}.$$

Finally, imposing $\phi_j^{(m)}(\lambda_j^{(m+1)}) = \eta\varepsilon$ and forcing again (4.4.11), we compute the new j-th component of the regularization vector as

$$\lambda_j^{(m+1)} = \frac{\eta\varepsilon - \phi_j^{(m)}(0)}{\phi_j^{(m)}(\lambda_j^{(m)}) - \phi_j^{(m)}(0)} \lambda_j^{(m-1)}.$$

As in the one-parameter case, the computation of each $\lambda_j^{(m+1)}$, $j = 1, \ldots, k$ can be meaningless for the first few iterations, since $\eta\varepsilon$ is above $\phi_j^{(m)}(0)$ and the values of $\lambda_j^{(m+1)}$ are therefore negative. For this reason we actually consider

$$\lambda_j^{(m+1)} = \left| \frac{\eta\varepsilon - \phi_j^{(m)}(0)}{\phi_j^{(m)}(\lambda_j^{(m)}) - \phi_j^{(m)}(0)} \right| \lambda_j^{(m)}. \tag{4.4.15}$$

At this point, if $j < k$ we add a regularization term and we repeat the previous computations considering $(j + 1)$ instead of j; otherwise, if $j = k$, the solution $y_{\Lambda_k}^{(m)}$ of (4.4.7) is indeed the solution of the complete multi-parameter problem (4.4.3). We stop the iterations as soon as $\phi^{(m)}(\Lambda) \leq \eta\varepsilon$.

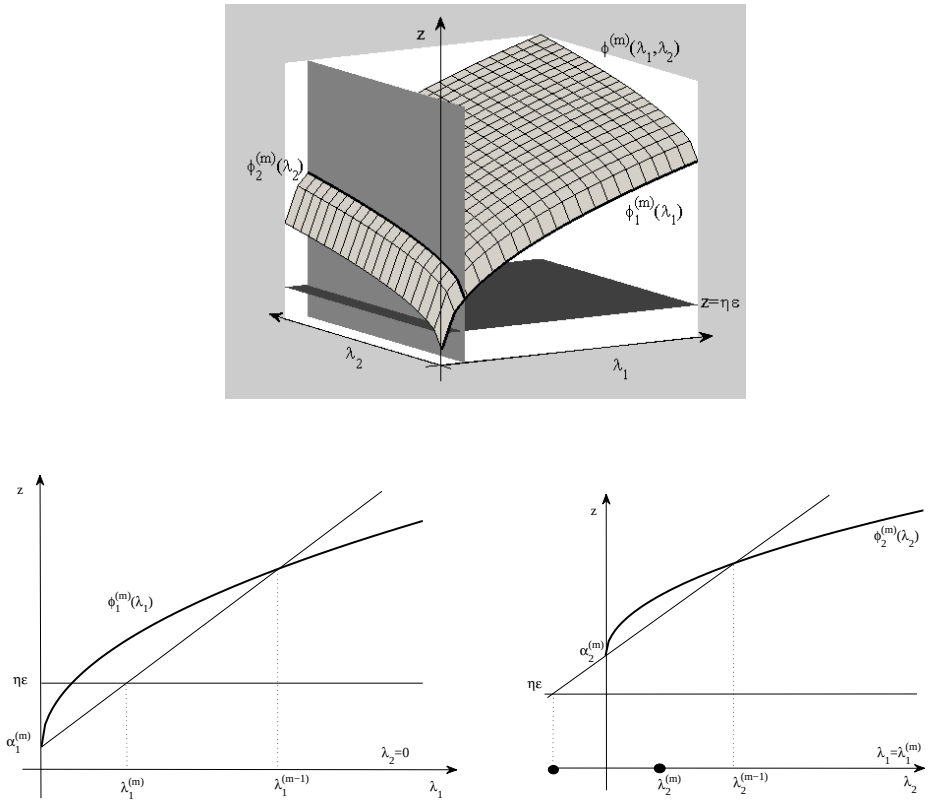

Figure 4.4.1: Geometrical interpretation of the strategy proposed to find the values of the regularization parameters when performing Arnoldi-Tikhonov multi-parameter method in the case $k = 2$ and for a fixed m. Upper frame: plot of the surface $z = \phi^{(m)}(\lambda_1, \lambda_2)$ along with the planes $z = \eta\varepsilon$ and $\lambda_1 = \lambda_1^{(m+1)}$. Lower leftmost frame: plot of the curve $\phi_1^{(m)}(\lambda_1) = \phi^{(m)}(\lambda_1, 0)$ on the plane $\lambda_2 = 0$. Lower rightmost frame: plot of the curve $\phi_2^{(m)}(\lambda_2) = \phi^{(m)}(\lambda_1^{(m+1)}, \lambda_2)$ on the plane $\lambda_1 = \lambda_1^{(m+1)}$.

This multi-parameter scheme has a clear geometrical interpretation, which generalizes the single-parameter secant one described in Section 4.1.1. For simplicity we treat the case $k = 2$, but the exposed ideas can be generalized to an arbitrary number of regularization terms.

Once an index m and a Cartesian coordinate system $(\lambda_1, \lambda_2, z)$ have been fixed, the equation $z = \phi^{(m)}(\lambda_1, \lambda_2)$ characterizes a differentiable surface in $\mathbb{R}^3$; solving (4.4.5) means finding the intersections between the just mentioned surface and the plane $z = \eta\varepsilon$ (see Figure 4.4.1, upper frame). The strategy above described prescribes to initially take $\lambda_2 = 0$; in this way we actually work on the plane (λ_1, z) and the approximate solution $\lambda_1^{(m+1)}$ of $\phi^{(m)}(\lambda_1, 0) = \phi_1^{(m)}(\lambda_1) = \eta\varepsilon$ is the intersection between $z = \phi_1^{(m)}(0) + \lambda_1 d_1^{(m)}$ and $z = \eta\varepsilon$ if this scalar is positive, otherwise its absolute value (see Figure 4.4.1, lower leftmost frame). At this point we take $\lambda_1 = \lambda_1^{(m+1)}$, i.e., we work on the plane $(\lambda_1^{(m+1)}, \lambda_2, z)$; the new value $\lambda_2^{(m+1)}$ is the approximate solution of $\phi^{(m)}(\lambda_1^{(m+1)}, \lambda_2) = \phi_2^{(m)}(\lambda_2) = \eta\varepsilon$, which is the intersection between $z = \phi_2^{(m)}(0) + \lambda_2 d_2^{(m)}$ and $z = \eta\varepsilon$ if this scalar is positive, otherwise its absolute value (see Figure 4.4.1, lower rightmost frame; in this case we display what happens when the quantity $\phi_2^{(m)}(0)$ is above the noise level).

We summarize the method described so far in Algorithm 14: we underline that this strategy requires to solve twice each reduced systems (i.e., for each $j = 1, \ldots, k$), in order to sequentially update the values of the components of the regularization vector Λ.

Algorithm 14: Multi-parameter Arnoldi-Tikhonov method

Input: A, b, $\mathcal{L} = \{L_1, \ldots, L_k\}$, $\Lambda = (\lambda_1^{(1)}, \ldots, \lambda_k^{(1)})$, x^*, η, ε.

For $m = 1, 2, \ldots,$ until $\|c - \bar{H}_m y_\Lambda^{(m)}\| \leq \eta\varepsilon$

1. Update W_m, $\bar{H}_m$ by the Arnoldi algorithm (Algorithm 2).
2. For $j = 1, \ldots, k-1$
   (a) Solve (4.4.7) with the parameters $((\Lambda_{j-1}^{(m+1)})^T, \lambda_j^{(m)})^T$ and evaluate $\phi_j^{(m)}(\lambda_j^{(m)})$ by (4.4.14).
   (b) Solve (4.4.7) with parameters $((\Lambda_{j-1}^{(m+1)})^T, 0)^T$ and evaluate $\phi_j^{(m)}(0)$ by (4.4.12).
   (c) Compute the new parameter $\lambda_j^{(m+1)}$ by (4.4.15) and then $\Lambda_j^{(m+1)}$ (4.4.8).
3. Compute the vector $y_\Lambda^{(m)} := y_{\Lambda_k}^{(m)}$ by solving the complete problem (4.4.3), with $\Lambda = ((\Lambda_{k-1}^{(m+1)})^T, \lambda_k^{(m)})^T$.
4. Compute the new parameter $\lambda_k^{(m+1)}$ by (4.4.15) and then update Λ.

Compute the approximate solution $x_\Lambda^{(m)} = W_m y_\Lambda^{(m)}$.

There is however a cheaper alternative, described in Algorithm 15, which consists in not using the updated values of the parameters at step 2. In other words, for $j = 1, \ldots k-1$, we do not need to refresh $\lambda_j^{(m)}$ with $\lambda_j^{(m+1)}$, but we can work with the regularization vector $((\Lambda_{j-1}^{(m)})^T, \lambda_j^{(m)})^T = (\lambda_1^{(m)}, \ldots, \lambda_{j-1}^{(m)}, \lambda_j^{(m)})^T$ at Step 2a of Algorithm 14. The new expression of $\phi_j^{(m)}(0)$ is now (cf. (4.4.13))

$$\phi_j^{(m)}(0) = \phi^{(m)}(\Lambda_{j-1}^{(m)}). \tag{4.4.16}$$

This alternative approach, described by Algorithm 15, needs only one solution of (4.4.7), for $j = 1, \ldots, k$, at each step.

Many numerical tests show that Algorithm 15 can compute regularized solutions whose relative error is still comparable to the one obtained running Algorithm 14. However, the number of iterations required by Algorithm 15 to return the solution is, on average, higher than the one associated to Algorithm 14.

Remark 23. In our computations both Algorithm 14 and Algorithm 15 have been implemented with some minor changes regarding the stopping criterion. Indeed, we have employed a sort of

Algorithm 15: Multi-parameter Arnoldi-Tikhonov method without updates

Input: A, b, $\mathcal{L} = (L_1, \dots, L_k)$, $\Lambda = (\lambda_1^{(1)}, \dots, \lambda_k^{(1)})$, x^*, η, ε.
For $m = 1, 2, \dots$ until $\|c - \bar{H}_m y_\Lambda^{(m)}\| \leq \eta\varepsilon$

1. Update W_m, $\bar{H}_m$ by the Arnoldi algorithm (Algorithm 2).
2. For $j = 1, \dots, k$
   (a) Solve (4.4.7) with the parameters $(\Lambda_{j-1}^{(m)})^T$ and evaluate $\phi_j^{(m)}(\lambda_j^{(m)})$ by
   $$\phi_j^{(m)}(\lambda_j^{(m)}) = \left\| c - \bar{H}_m y_{\Lambda_j}^{(m)} \right\|, \quad \Lambda = \left((\Lambda_{j-1}^{(m)})^T, \lambda_j^{(m)}, 0, \dots, 0 \right)^T,$$
   (b) Take $\phi_j^{(m)}(0)$ as in (4.4.16).
   (c) Compute the new parameter $\lambda_j^{(m+1)}$ by (4.4.15).
3. Update the vector $\Lambda = (\lambda_1^{(m+1)}, \dots, \lambda_k^{(m+1)})$.

Compute the approximate solution $x_\Lambda^{(m)} = W_m y_\Lambda^{(m)}$.

weakened discrepancy principle, i.e., we stop the iterations as soon as

$$\phi^{(m)}(\lambda) - \eta\widetilde{\varepsilon}\|b\| < 10^\theta, \tag{4.4.17}$$

where $\theta < 0$ is automatically determined as the sum of the order of the noise level $\widetilde{\varepsilon}$ and the order of the last significant digit of η. In this way, when applying the discrepancy principle, we neglect any quantity coming after the last significant digit of the product $\eta\widetilde{\varepsilon}$. For instance, if $\widetilde{\varepsilon} = 10^{-2}$ and $\eta = 1.01$ then $\theta = -4$ and we stop the iterations as soon as

$$\frac{\phi^{(m)}(\lambda)}{\|b\|} - 1.01 \cdot 10^{-2} < \frac{10^{-4}}{\|b\|}.$$

We remark that, if the "classical" discrepancy principle (4.4.5) is fulfilled, then also (4.4.17) is satisfied. We introduced this weakened version of the discrepancy principle because, while executing the numerical experiments, we noted that very often the discrepancy stagnates slightly above the prescribed threshold without crossing it and, performing too many iterations, the quality of the approximate solution slowly deteriorates.

At the same time we decide to enforce the stopping criterion in order to assure that all the solutions of the intermediate reduced regularization problems (4.4.7) together with the solution $y_\Lambda^{(m)}$ of the complete problem (4.4.3) satisfy the weakened discrepancy principle (4.4.17), i.e.,

$$\phi_j^{(m)}(\lambda_j^{(m)}) - \eta\widetilde{\varepsilon}\|b\| < 10^\theta \quad \forall j = 1, \dots, k-1.$$

This is a quite natural choice, since the solution of the multi-parameter problem is built taking into account the k solutions of the associated one-parameter problems.

4.4.3 Numerical Experiments

In this section we test the behavior of Algorithm 14 to solve the multi-parameter problem. We believe that the best way to validate the AT multi-parameter method just described is to make suitable comparisons with what happens in the AT one-parameter case; in the sequel we will explain the details and the goal of each experiment. We will exclusively focus on the two-parameter and the three-parameter cases. Moreover, analogously to what we have done in Section 4.1.1, we always consider the initial guess $x^* = 0$, we set $\eta = 1.01$ and $\Lambda = (1, \dots, 1)^T \in \mathbb{R}^k$.

Example 1: results obtained considering particular solutions.

The aim of the first set of performed experiments is to show that, when applying the multi-parameter method to a problem whose exact solution x^{ex} lies in the null space of a regularization operator L_i, the parameter selection strategy correctly weights the i-th component of the regularization vector Λ by assigning to λ_i a value dominating the other components. Indeed, in this situation, the regularization operator L_i is the most suitable one, since the important features of the solution are not damped (cf. the discussion in Section 1.3.1). Therefore we start to consider two particular exact solutions: the constant one, i.e., $x^{ex} = \bar{x}_c := (1, 1, \ldots, 1)^T \in \mathbb{R}^n$, and the linear one, i.e., $x^{ex} = \bar{x}_l := (1, 2, \ldots, n)^T \in \mathbb{R}^n$; we recall that $\bar{x}_c \in \mathcal{N}(D_1) \cap \mathcal{N}(D_2)$, while $\bar{x}_l \in \mathcal{N}(D_2)$. For this reason we employ both the two and three-parameter methods with different combinations of the regularization matrices I_n, D_1 and D_2. Indeed, in this case we are considering a sort of discrete weighted Sobolev norm approach.

First of all we take the solution $\bar{x}_c$ and we consider the matrix of size $n = 200$ associated to the problem `i_laplace`. The involved noise level is $\widetilde{\varepsilon} = 10^{-2}$ and we determine a regularized solution by using the (I_{200}, D_1) two-parameter method; each test problem is generated 100 times to reduce the dependence of the results on the random components of the noise vector e. To

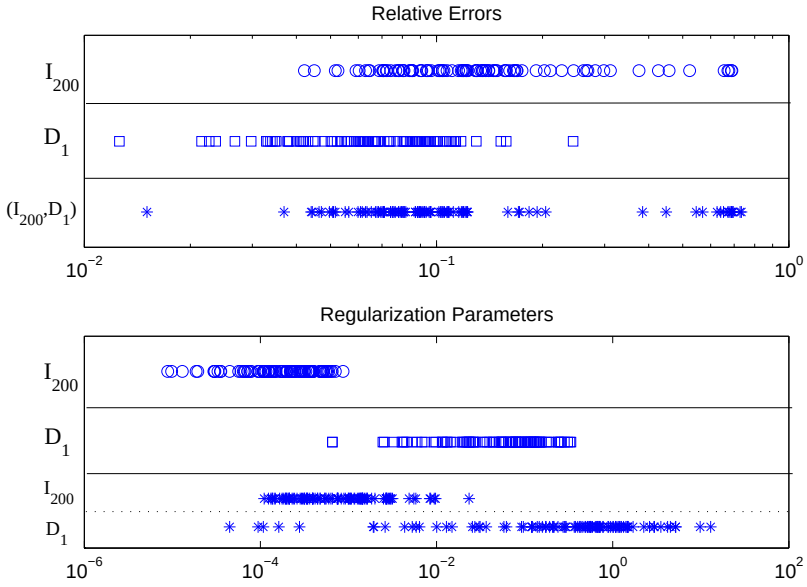

Figure 4.4.2: Results obtained running 100 times the test problem `i_laplace` with the particular solution $\bar{x}_c$ (we plot one single marker for each performed test). Upper frame: we report the values of the relative errors in logarithmic scale on the horizontal axis and, at each vertical level, we mark the values corresponding to the I_{200} one-parameter (circle), the D_1 one-parameter (square) and the (I_{200}, D_1) two-parameter (asterisk) methods. Lower frame: we report the values of the regularization parameters, using the layout just described; concerning the multi-parameter method, the first line (labeled by I_{200}) refers to the parameter that weights the term $\|x\|^2$, while the second line (labeled by D_1) refers to the parameter that weights the term $\|D_1 x\|^2$.

be aware of what happens using the single parameter Tikhonov method, for each test we also report the results obtained considering exclusively $L = I_{200}$ and $L = D_1$. We display the values of the relative errors and regularization parameter obtained in Figure 4.4.2. We can clearly see that, with very few exceptions, the components of the regularization vector associated to I_{200} and D_1 replicate the behavior of the parameter of the Tikhonov method with $L = I_{200}$ and $L = D_1$, respectively. This means that, in the regularization process, the most appropriate regularization operator (in this case D_1) weights more than the others. In almost all cases, the solutions of the I_{200} and D_1 one-parameter methods belong to Krylov subspaces of dimension

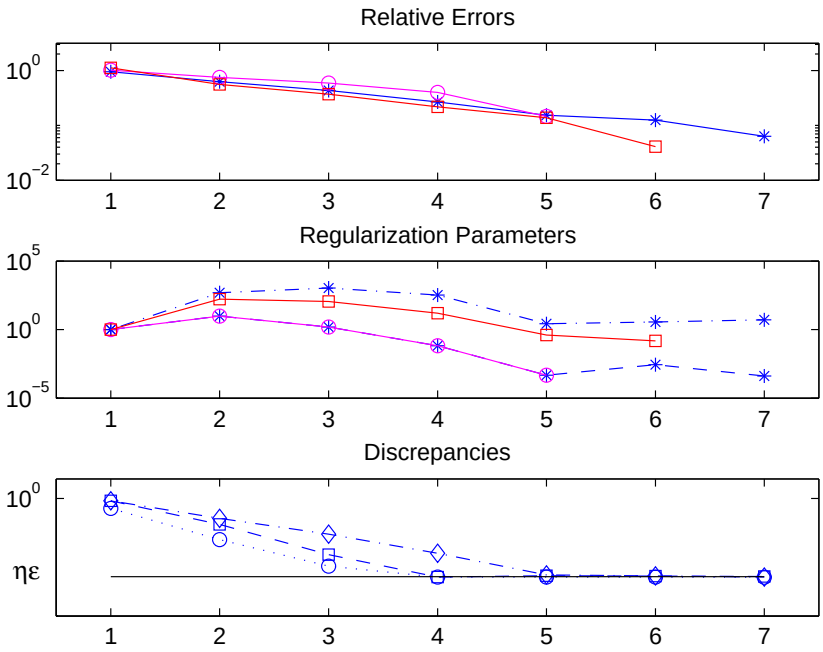

Figure 4.4.3: Behavior of the relative errors, regularization parameters and discrepancies versus the number of iterations for the test problem `i_laplace` with solution $\bar{x}_c$. Upper frame: we consider the multi-parameter method (asterisk), the I_{200} one-parameter method (circle) and the D_1 one-parameter method (square); middle frame: we display the values of the parameters λ_1 associated to I_{200} (asterisk with dashed line) and λ_2 associated to D_1 (asterisk with dash-dot line) and the values of the parameters of the two one-parameter methods considered above (with the same markers as listed above); lower frame: the norm of the residual of the GMRES over $\|b\|$ (circle) and the discrepancies $\phi_1^{(m)}/\|b\|$ associated to the first regularization term (square), $\phi_2^{(m)}/\|b\|$ associated to the second regularization term (diamond).

5 and 6, respectively, while most of the solutions associated to the two-parameter methods belong to Krylov subspaces of dimension 6 or 7. In Figure 4.4.3 we focus on a single test and we display the course of the relative error, the regularization parameters and the discrepancies of the examined methods at each step of the Arnoldi algorithm. Looking at both figures we can see that the quality of the solutions computed by the multi-parameter method does not improve with respect to the results associated to the D_1 one-parameter method. However this is quite reasonable since, as just explained, the task of the multi-parameter methods is to preserve many different features of the solution; when, as in this case, the solution belongs to the null space of one of the considered operator, the one-parameter method with that regularization operator is the one that works better. Now we consider the matrix associated to the problem `phillips` with $n = 200$ and we take, as exact solution, the linear one $\bar{x}_l$; the noise level is again $\widetilde{\varepsilon} = 10^{-2}$. We compute the regularized solution employing the three-parameter method with regularization matrices $L_1 = I_{200}$, $L_2 = D_1$ and $L_3 = D_2$. We display the results in Figure 4.4.4, together with what we have obtained solving the same problem with the I_{200}, D_1, D_2 one-parameter methods. Even in this case the parameter selection strategy can automatically weight the regularization matrices, assigning the highest parameter to the matrix whose null space contains the exact solution (in this case, D_2). Regarding the number of iterations required to satisfy the weakened discrepancy principle, the three-parameter method needs in most of the cases 8, 11 or 13 iterations, the I_{200} one-parameter method needs 7 or 8 iterations while both the D_1 and D_2 one-parameter methods require 8 or 9 iterations. In Figure 4.4.5 we show the values of the relative errors, of the regularization parameters and of the discrepancies versus the number of iterations for the problem `shaw` of size 200; we take again the linear vector $\bar{x}_l$ as exact solution.

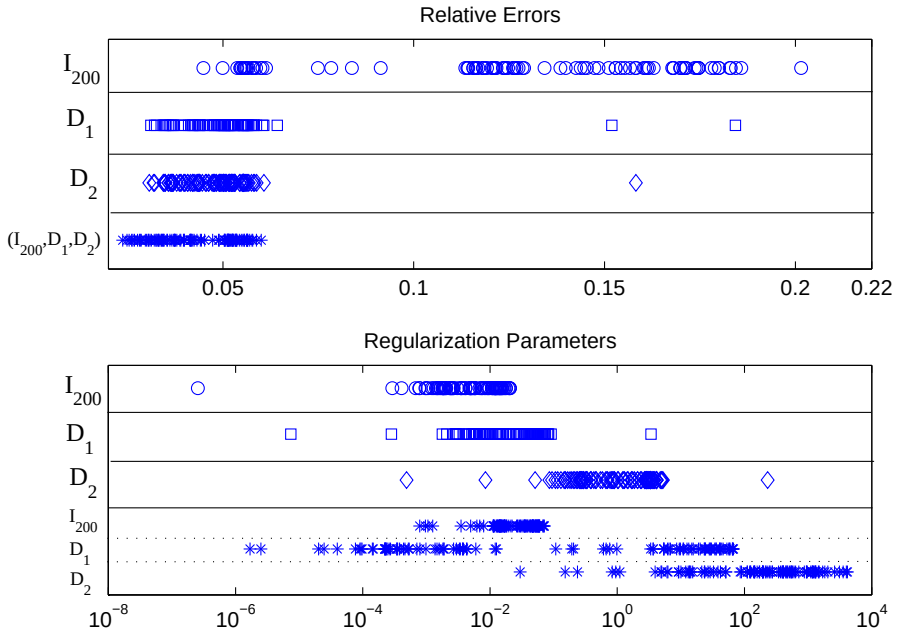

Figure 4.4.4: Results obtained running 100 times the test problem `phillips` with the particular solution $\bar{x}_l$ (we plot one single marker for each performed test). Upper frame: we report the values of the relative errors on the horizontal axis and, at each vertical level, we mark the values corresponding to the I_{200} one-parameter (circle), the D_1 one-parameter (square), the D_2 one-parameter (diamond) and the (I_{200}, D_1, D_2) three-parameter (asterisk) methods. Lower frame: we report the values of the regularization parameters, using the layout just described; concerning the multi-parameter method, the first line (labeled by I_{200}) refers to the parameter that weights the term $\|x\|^2$, the second line (labeled by D_1) refers to the parameter that weights the term $\|D_1 x\|^2$, and the third line (labeled by D_2) refers to the parameter that weights the term $\|D_2 x\|^2$.

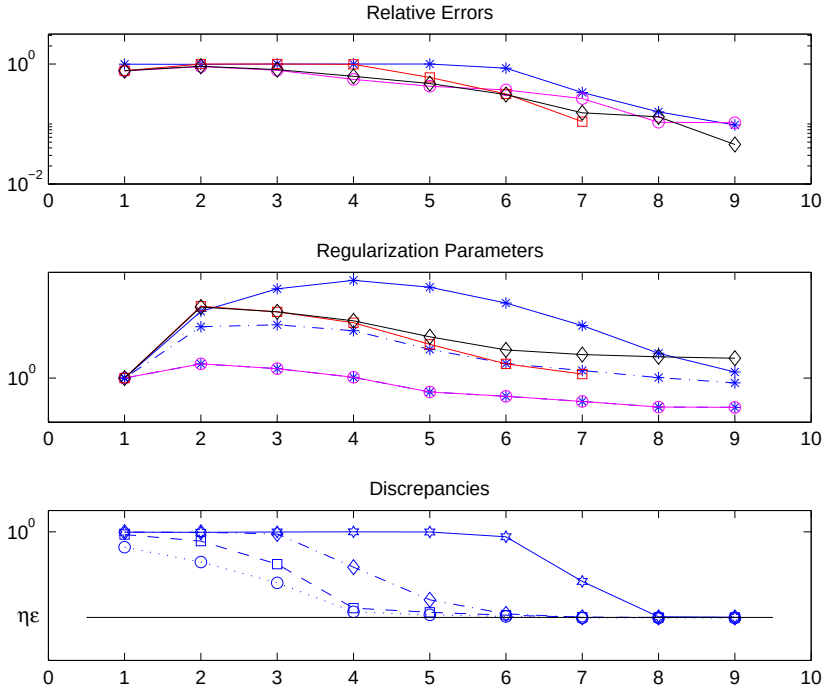

Figure 4.4.5: Behavior of the relative errors, regularization parameters and discrepancies versus the number of iterations for the test problem `shaw` with solution $\bar{x}_l$. The displayed quantities are the same as in Figure 4.4.3 and are denoted by the same markers. In addition: in the upper frame we visualize the performance of the D_2 one-parameter method (diamond); in the middle frame we visualize the history of the parameter λ_3 (asterisk with solid line) that weights the term $\|D_2 x\|^2$ of the multi-parameter method along with the regularization parameter associated to the D_2 one-parameter method (diamond); in the lower frame we visualize the history of the discrepancy $\phi_3^{(m)}/\|b\|$ (hexagram) associated to the third regularization term.

The method has been experimented on the test problems described in Section 1.2.1, all of dimension $n = 200$, using the two particular solutions $\bar{x}_c$ and $\bar{x}_l$; we have also considered two different noise levels ($\widetilde{\varepsilon} = 10^{-2}$ and $\widetilde{\varepsilon} = 5 \cdot 10^{-2}$) and several combinations of regularization operators. We report some results in Table 4.6.

	Relative Errors	λ_1	λ_2	λ_3	**Iterations**
`baart`	4.7271e-002	1.8289e-002	-	-	3.03
`baart`	4.6467e-002	-	2.6946e+002	-	3.00
`baart`	4.8727e-002	-	-	3.1295e+001	3.00
`baart`	**2.8299e-002**	3.5002e-002	2.8047e+003	-	3.81
`baart` (Alg. 15)	**4.5396e-002**	1.8319e-002	3.1644e+002	-	3.01
`baart`	5.6287e-002	3.5177e-002	-	6.1848e+004	3.81
`baart` (Alg. 15)	**4.5595e-002**	1.8319e-002	-	2.0673e+004	3.01
`baart`	**4.1186e-002**	-	2.6891e+003	3.5107e+006	3.12
`baart` (Alg. 15)	**4.2843e-002**	-	1.2127e+003	4.1811e+006	3.09
`baart`	**2.9684e-002**	3.4540e-002	2.8129e+003	7.0676e+006	3.95
`baart` (Alg. 15)	**4.5433e-002**	1.8319e-002	3.1644e+002	1.4420e+005	3.01
`gravity`	1.4412e-001	6.2068e-002	-	-	3.00
`gravity`	7.3863e-002	-	1.0178e+003	-	3.38
`gravity`	7.6596e-002	-	-	5.8340e+002	3.30
`gravity`	7.5657e-002	8.9338e-002	2.6968e+001	-	3.52
`gravity` (Alg. 15)	**5.9147e-002**	1.7299e-001	1.4920e+003	-	4.61
`gravity`	7.6178e-002	9.4794e-002	-	7.9399e+002	3.41
`gravity` (Alg. 15)	7.7175e-002	6.9617e-002	-	1.6570e+003	3.23
`gravity`	**5.6443e-002**	-	3.4291e+003	1.0032e+005	5.13
`gravity` (Alg. 15)	**5.7096e-002**	-	2.1291e+003	1.7057e+005	5.14
`gravity`	7.5426e-002	1.1257e-001	3.4710e+001	1.6360e+004	3.90
`gravity` (Alg. 15)	**5.5631e-002**	3.2129e-001	7.2494e+002	5.6887e+004	10.39
`shaw`	3.8658e-001	1.1241e-002	-	-	4.73
`shaw`	3.7087e-001	-	1.0679e+001	-	4.30
`shaw`	3.7499e-001	-	-	1.1396e+002	4.08
`shaw`	**3.4765e-001**	4.0968e-002	6.2112e+000	-	5.77
`shaw` (Alg. 15)	**3.2295e-001**	2.8325e-002	8.9987e+000	-	6.71
`shaw`	**3.6824e-001**	9.7160e-002	-	5.0404e+002	4.85
`shaw` (Alg. 15)	**3.5303e-001**	1.8491e-002	-	1.3922e+003	5.06
`shaw`	**2.2610e-001**	-	8.0840e+001	1.0614e+003	6.59
`shaw` (Alg. 15)	**2.8593e-001**	-	2.4070e+001	2.7850e+003	6.02
`shaw`	**3.4812e-001**	3.0250e-002	6.1392e+000	5.6965e+002	7.06
`shaw` (Alg. 15)	**3.2119e-001**	3.3386e-002	3.6780e+000	1.0717e+003	9.23

Table 4.6: Averages of the values of the displayed quantities obtained running 100 times the same test problem with different noise realizations. The considered solution is the constant one, $\bar{x}_c$, with noise level $\widetilde{\varepsilon} = 5 \cdot 10^{-2}$.

Finally we propose the results of a couple of experiments built considering the artificial solutions

$$\bar{x}_{\sin} = x^{(a)} + x^{(b)} := 10 \sin\left(\frac{x}{2}\right) + x \in \mathbb{R}^n, \tag{4.4.18}$$

$$\bar{x}_{\tan} = x^{(a)} + x^{(b)} := \frac{1}{10} \tan\left(\frac{x}{n+1}\frac{\pi}{2}\right) + x \in \mathbb{R}^n. \tag{4.4.19}$$

$\bar{x}_{\sin}$ is oscillating while $\bar{x}_{\tan}$ is quickly increasing. This test is motivated by the fact that the so far considered couple of matrices D_1 and D_2 indeed represents a particular situation, since $\mathcal{N}(D_1) \subset \mathcal{N}(D_2)$. Taking instead the solution (4.4.18) or (4.4.19), by (1.3.25) we can build two particular regularization matrices $L^{(a)}$ and $L^{(b)}$ such that $x^{(a)} \in \mathcal{N}(L^{(a)})$, $x^{(b)} \in \mathcal{N}(L^{(b)})$ and $\mathcal{N}(L^{(a)}) \cap \mathcal{N}(L^{(b)}) = \{0\}$. As consequence, both $\bar{x}_{\sin}$ and $\bar{x}_{\tan}$ do not belong to the null space of the matrices $L^{(a)}$ or $L^{(b)}$. In this way we can really appreciate the essence of the multi-parameter methods, i.e., to preserve many different features of the solution of the original problem that may be distorted employing only one regularization operator. For both solutions we consider the matrix $A \in \mathbb{R}^{200\times 200}$ associated to the test problem `foxgood`, a noise level $\widetilde{\varepsilon} = 10^{-2}$ and the regularization matrices $L_1 = L^{(a)}$, $L_2 = L^{(b)}$. We display the results relative to (4.4.18) and (4.4.19) in Figure 4.4.6.

Example 2: results obtained considering more general solutions.

In the second set of computed experiments we simply consider the most common test problems described in Section 1.2.1, with their appropriate solution. We are just going to display some graphs that compare the performances of the new multi-parameter method and the usual Arnoldi-Tikhonov method. We will only consider the regularization matrices I_n, D_1 and D_2.

In Figure 4.4.7 we display the behavior of the relative errors and the values of the regularization parameters obtained solving the test problem `i_laplace` of dimension $n = 200$ with a noise of level $\widetilde{\varepsilon} = 10^{-2}$ that affects the right-hand-side vector; we consider the I_{200} and D_1 one-parameter methods and the (D_1, I_{200}) two-parameter method. We remark that, when performing the multi-parameter method, the results can be affected by the order in which the regularization matrices appear. Indeed, looking at the parameters selection strategy described in Section 4.4.2, we can understand that the first regularization matrix (in this case, D_1) is weighted similarly to the one-parameter case, while the following ones work as corrections. This is a consequence of the fact that many reduced problems are solved sequentially and each one is based on the solutions and on the parameters associated to the previous ones; in this sense the first regularization operator is somehow advantaged with respect to the others. Therefore, if one has some intuition about the regularity of the solution, we suggest to put in the first place the most suitable regularization matrix. In Table 4.7 we collect some results obtained considering the one-parameter and the two and three-parameter methods applied to the usual test problems described in Section 1.2.1: every problem has dimension 200, the noise level is $\widetilde{\varepsilon} = 10^{-2}$ and the regularization matrices employed are I_{200}, D_1, D_2.

Further considerations

Finally, we highlight a couple of important features of the new multi-parameter AT method that we observed while performing the numerical experiments just described.

The first property is that the AT multi-parameter method is very robust with respect to the initial choice of the regularization vector Λ, that is, considering different values of the components of Λ, the accuracy of the results and the number of iterations are basically stable. In Figure 4.4.8 we display the values of the regularization parameters obtained by solving the test problem `shaw` of dimension $n = 200$ and considering the noise level $\widetilde{\varepsilon} = 10^{-2}$. We have employed the (I_{200}, D_1, D_2) three-parameter method and we have executed four tests considering the vector Λ whose three entries are all equal to 0.5, 1, 10 or 100. We can see that, except in

the very first iterations, the behavior of each λ_i, $i = 1, 2, 3$ is very similar independently on the value of $\lambda_i^{(1)}$. We have also tried to consider different components of the initial vector Λ and the results, even if not shown, are identical to the ones just described.

The second property is about the performance of the method when many extra iterations are executed after the stopping criterion is fulfilled. Despite we had to review the stopping criterion introducing the weakened discrepancy principle (cf. Section 4.4.2), we can appreciate that, in many cases, the behavior of the method is very stable even when we decide to go on with an arbitrary number of iterations. For instance, in Figure 4.4.9 we display what happens solving the problem `shaw` by the three-parameter method and considering, as before, $n = 200$, $\widetilde{\varepsilon} = 10^{-2}$, $L_1 = I_{200}$, $L_2 = D_1$ and $L_3 = D_2$. Similar results have been obtained also for `phillips` and `foxgood`.

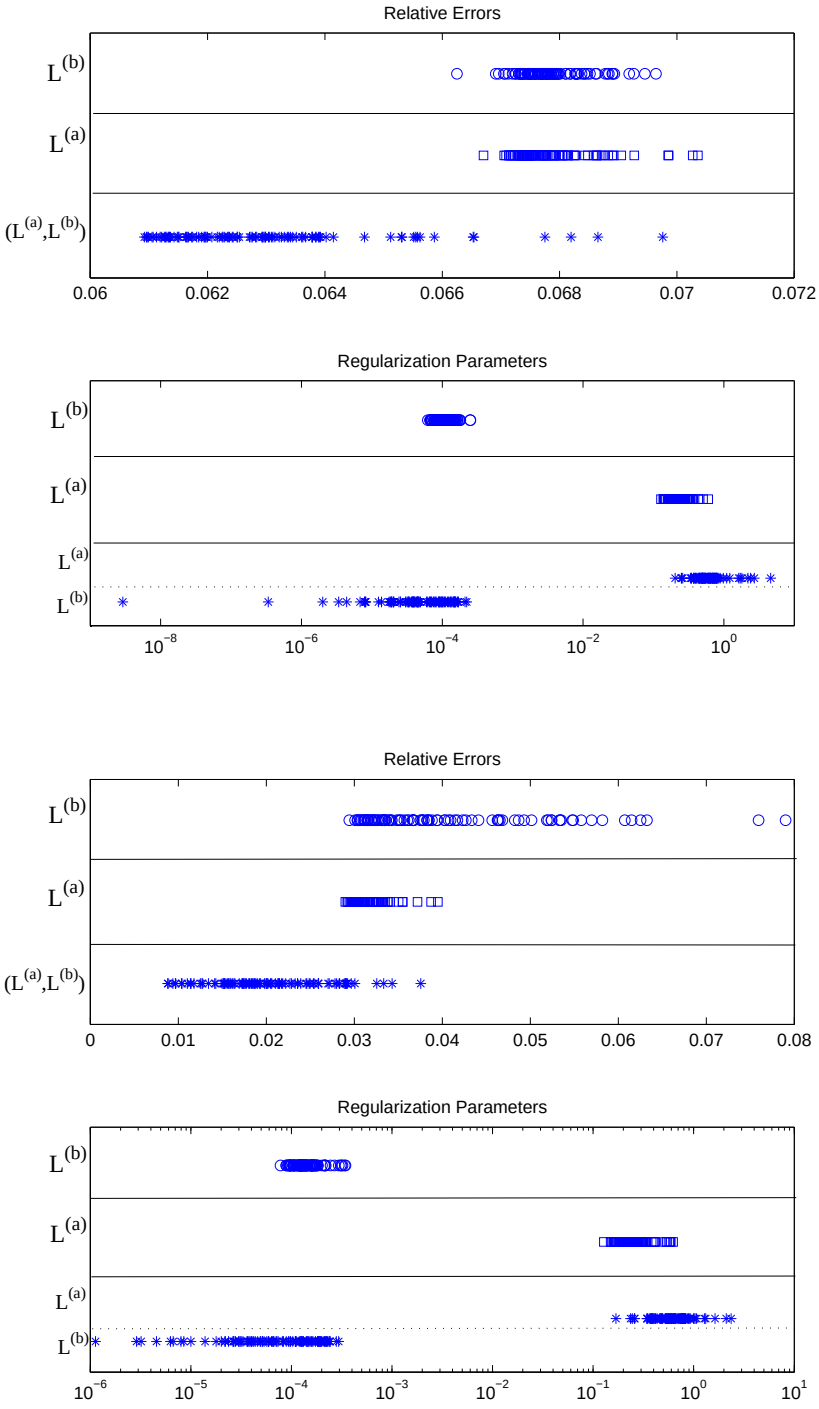

Figure 4.4.6: Results obtained running 100 times the test problem `foxgood` with the particular solutions $\bar{x}_{\sin}$ (first and second frames) and $\bar{x}_{\tan}$ (third and fourth frames); as before, we plot one single marker for each performed test. The regularization operators $L^{(a)}$ and $L^{(b)}$ are projection operators of the form (1.3.25). First and third frames: we report the values of the relative errors on the horizontal axis and, at each vertical level, we mark the values corresponding to the $L^{(b)}$ one-parameter (circle), the $L^{(a)}$ one-parameter (square) and the $(L^{(a)}, L^{(b)})$ two-parameter (asterisk) methods. Second and fourth frames: the values of the regularization parameters are reported using the layout just described.

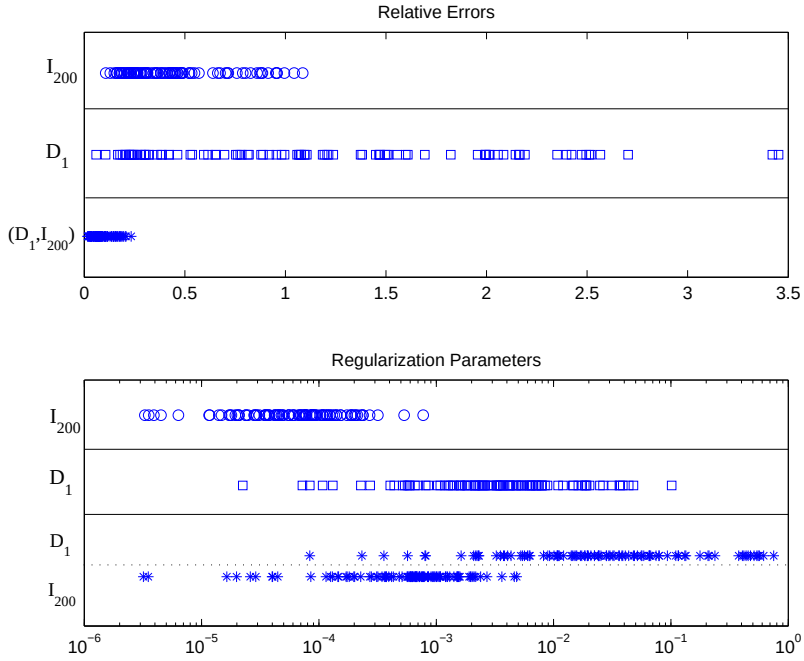

Figure 4.4.7: Results obtained running 100 times the test problem `i_laplace` (we plot one single marker for each performed test). Upper frame: we report the values of the relative errors on the horizontal axis and, at each vertical level, we mark the values corresponding to the I_{200} one-parameter (circle), the D_1 one-parameter (square) and the (D_1, I_{200}) two-parameter (asterisk) methods. Lower frame: we report the values of the regularization parameters in logarithmic scale on the horizontal axis and, at each vertical level, we mark the values corresponding to the I_{200} one-parameter (circle), the D_1 one-parameter (square) and the (D_1, I_{200}) two-parameter (asterisk) methods; concerning the multi-parameter method, the first line (labeled by D_1) refers to the parameter that weights the term $\|D_1 x\|^2$, while the second line (labeled by I_{200}) refers to the parameter that weights the term $\|x\|^2$.

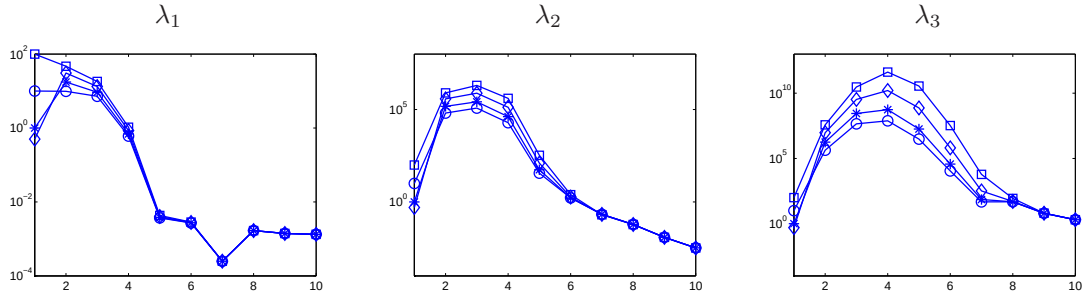

Figure 4.4.8: Values of the components of the regularization vector Λ versus the number of iterations (each frame corresponds to a different component). The test problem is `shaw` and we consider the (I_{200}, D_1, D_2) multi-parameter AT method. The initial values for the regularization vector are $\Lambda = (0.5, 0.5, 0.5)^T$ (diamond), $\Lambda = (1, 1, 1)^T$ (asterisk), $\Lambda = (10, 10, 10)^T$ (circle), $\Lambda = (100, 100, 100)^T$ (square).

	Relative Errors	λ_1	λ_2	λ_3	**Iterations**
`baart`	5.0485e-002	5.9453e-004	-	-	4.00
`baart`	9.6425e-002	-	4.2167e-001	-	6.00
`baart`	6.2569e-002	-	-	1.0876e+003	5.01
`baart`	1.5099e-001	1.0683e-003	6.3735e-002	-	5.50
`baart` (Alg. 15)	1.5135e-001	1.0854e-003	1.0809e-001	-	6.10
`baart`	8.8097e-002	8.3136e-004	-	1.3274e+002	4.38
`baart` (Alg. 15)	1.2243e-001	1.0936e-003	-	2.3528e+002	5.67
`baart`	1.2223e-001	-	8.5082e-001	1.6022e+002	7.57
`baart` (Alg. 15)	1.2907e-001	-	8.9299e-001	1.1968e+002	8.93
`baart`	1.4903e-001	1.1395e-003	1.5122e-002	9.7826e+001	6.63
`baart` (Alg. 15)	2.0029e-001	1.2088e-003	2.5714e-003	3.3557e+001	15.88
`gravity`	1.2013e-001	9.7765e-003	-	-	5.27
`gravity`	4.0751e-002	-	3.4584e+000	-	6.24
`gravity`	4.0657e-002	-	-	5.4844e+002	6.19
`gravity`	4.3901e-002	3.3339e-002	7.3607e-001	-	6.15
`gravity` (Alg. 15)	4.2829e-002	2.7101e-002	3.6701e+000	-	6.50
`gravity`	4.2992e-002	4.1944e-002	-	9.7444e+001	6.04
`gravity` (Alg. 15)	4.1431e-002	2.8425e-002	-	2.3548e+003	6.60
`gravity`	4.5887e-002	-	1.1104e+001	2.0749e+003	7.92
`gravity` (Alg. 15)	4.6282e-002	-	1.2389e+001	2.5341e+003	8.83
`gravity`	**3.7745e-002**	4.0109e-002	8.4321e-001	4.1857e+002	7.80
`gravity` (Alg. 15)	**3.5941e-002**	5.1580e-002	6.8753e-001	8.0771e+002	13.03
`phillips`	2.8920e-002	1.8711e-002	-	-	5.00
`phillips`	2.5621e-002	-	5.2041e+000	-	5.05
`phillips`	2.5663e-002	-	-	5.5949e+002	5.00
`phillips`	2.5654e-002	5.5102e-002	2.2946e+000	-	7.52
`phillips` (Alg. 15)	**2.5428e-002**	4.2635e-002	2.2588e+000	-	8.06
`phillips`	2.6108e-002	5.0990e-002	-	2.7694e+002	7.48
`phillips` (Alg. 15)	2.6021e-002	4.1527e-002	-	3.0252e+002	8.05
`phillips`	2.7134e-002	-	1.0548e+001	1.4744e+002	7.54
`phillips` (Alg. 15)	2.7043e-002	-	9.1030e+000	1.3533e+002	8.43
`phillips`	**2.5571e-002**	4.6571e-002	9.4471e-001	4.5558e+001	9.71
`phillips` (Alg. 15)	**2.5307e-002**	5.1642e-002	3.8008e-001	5.2265e+001	12.56
`shaw`	1.3445e-001	7.5858e-004	-	-	5.85
`shaw`	1.2074e-001	-	5.4351e-001	-	6.29
`shaw`	1.2074e-001	-	-	1.2207e+002	6.01
`shaw`	1.3477e-001	1.8739e-003	2.5149e-001	-	6.73
`shaw` (Alg. 15)	1.4452e-001	3.1749e-003	2.6832e-001	-	8.02
`shaw`	1.3466e-001	2.0832e-003	-	5.8343e+001	6.71
`shaw` (Alg. 15)	1.4767e-001	3.6720e-003	-	5.1928e+001	8.18
`shaw`	2.0162e-001	-	1.8871e-001	2.9227e+000	9.59
`shaw` (Alg. 15)	2.0445e-001	-	1.8076e-001	4.0254e+000	10.85
`shaw`	1.3631e-001	3.1890e-003	2.6252e-001	1.7495e+001	7.71
`shaw` (Alg. 15)	1.3297e-001	3.6163e-003	2.2794e-002	9.6222e+000	15.36

Table 4.7: Performance of the one-parameter and multi-parameter methods applied to the test problems described in Section 1.2.1, with $n = 200$ and noise level $\widetilde{\varepsilon} = 10^{-2}$. The reported numbers are the averages of the values of the displayed quantities and are obtained running 100 times the same test problem with different noise realizations.

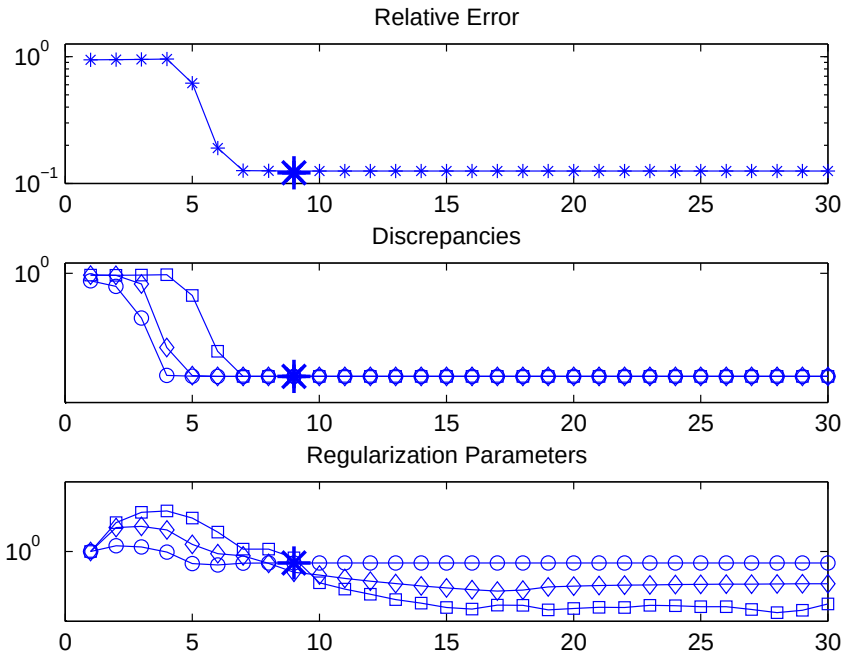

Figure 4.4.9: Values of the relative error, of the discrepancies and of the regularization parameters versus the number of iterations for the test problem `shaw` solved by the (I_{200}, D_1, D_2) multi-parameter method. In the second and third boxes, the circle denotes the quantities associated to the first regularization matrix (I_{200}), the diamond denotes the quantities associated to the second regularization matrix (D_1), and the square denotes the quantities associated to the third regularization matrix (D_2). This method would stop at the 9-th iteration (denoted by the big asterisk), but we decide to run it till the 30-th iteration.

Chapter 5

Beyond the 2-norm

Regularizing certain kinds of problems by employing 2-norm filtering schemes can be rather restrictive: for instance, such methods deliver poor results when one is concerned with enhancing the sparsity of the reconstructed solution (cf. [123, 125]). In these cases, in order to achieve an accurate approximation of the desired solution, one usually employs some Tikhonov-like methods, where the fit-to-data and the regularization terms are evaluated by p-norms or are defined by a nonlinear functional (cf. [24, 34, 59, 110]). Many efficient numerical methods have been developed to deal with these more general forms of regularization: the main difference with respect to the classical Tikhonov method introduced in Section 1.3.1 is that the regularized problem is not anymore equivalent to a linear system and some sophisticated optimization methods are used to find the minimum of the Tikhonov-like functional. These techniques are often applied in signal processing problems and image restoration problems.

This chapter introduces two original strategies to impose sparsity constraints in the usual Tikhonov regularization framework: the main idea is to consider suitable adaptively-defined regularization matrices that allow the 2-norm regularization term to approximate a more general regularization term. Therefore, the basic idea behind the newly proposed methods is to adapt an Iteratively Reweighted Least Squares (IRLS) approach. When applying the Arnoldi-Tikhonov method (3.2.2) described in Chapter 3 to solve the regularized problem (1.3.4), the regularization matrix can be updated both at each step and after some iterations have been performed, leading to two different schemes: the first one is based on Flexible Krylov Subspaces; the second one is based on restarting the Arnoldi algorithm.

In Section 5.1 we introduce the Tikhonov-like general formulations that are considered in this chapter and we survey some state-of-the-art methods that are commonly applied to solve the regularized problems. In Section 5.2 we illustrate, in a general framework, the IRLS method. In Section 5.3 we describe the first algorithm (called "Flexi-AT"), giving some details about Flexible Krylov subspaces. In Section 5.4 we describe the second algorithm (called "ReSt-AT"), explaining how to incorporate suitable restarts and how to impose some additional constraints in the Arnoldi-Tikhonov setting. Finally, in Section 5.5, we present some numerical results and we make comparisons with already existing methods for sparse reconstruction.

5.1 Problem formulation

In the previous chapters, many well-established regularization methods have been analyzed and some new algorithms have been proposed; all of them can be classified as 2-norm filtering schemes, meaning that they deal with classical least-squares problems expressed in the 2-norm and they regularize the 2-norm of the approximate solution. Every relation derived so far (starting from the expression of the regularized solution by means of a filtered SVD expansion, cf. Section 1.3) is valid as far as the 2-norm is concerned.

However, employing exclusively the 2-norm is rather restrictive for certain kinds of problems:

for instance, it is well-known that the 2-norm can deliver over-smooth reconstructions (even if the regularization parameter is suitably chosen) when we wish to reconstruct solutions that are themselves not very smooth. Indeed, dealing with a sparse solution (i.e., a vector x_{SP} having most of its entries equal to 0) or a solution that presents some jumps (i.e., a vector x_{J} whose first "derivative" x'_{J} is sparse), a 2-norm regularization term of the form $\|x_{\mathrm{SP}}\|_2^2$ or $\|x'_{\mathrm{J}}\|_2^2$ tends to penalize the inherent discontinuities of the exact solution (this phenomenon is clearly illustrated by some simple examples in [59, §8.6]). The sparse reconstruction issue often appears in the signal processing framework, for instance when one wants to acquire and recover a signal using Compressed Sensing (CS, cf. [26]); jumps are ubiquitous when dealing with images, since the edges of the items being portrayed are expressed by a steep change in the pixel values.

In general, better approximations can be computed by considering optimization problems of the form

$$\min_{x\in\mathbb{R}^n} \{\mathcal{J}(x) + \lambda\mathcal{R}(x)\} \tag{5.1.1}$$

where $\mathcal{J}(x)$ is a fit-to-data term, $\mathcal{R}(x)$ is a regularization term and, as usual, $\lambda > 0$. The functional $\mathcal{J}(x) + \lambda\mathcal{R}(x)$ is sometimes referred to as an objective function, consistently with the terminology commonly used in optimization methods. It is well-known that solving

$$\min_{x\in\mathbb{R}^n} \left\{\|b - Ax\|_2^2 + \lambda\|x\|_p^p\right\} \tag{5.1.2}$$

reconstructs a sparse approximation of x when $p = 1$. Recall that the p-norm (or ℓ^p norm) of a vector $x \in \mathbb{R}^n$ is defined as

$$\|x\|_p = \left(\sum_{i=1}^{n} |x_i|^p\right)^{1/p} \tag{5.1.3}$$

and, in particular, for $p = 1$ we have

$$\|x\|_1 = \sum_{i=1}^{n} |x_i|. \tag{5.1.4}$$

In this setting, the functional $\|\cdot\|_0$ is defined as

$$\|x\|_0 = \sum_{i=1}^{n} \mathbb{1}_{\{x_i\neq 0\}}, \tag{5.1.5}$$

i.e. $\|x\|_0$ counts the number of nonzero entries of the vector x. Looking at the definitions (5.1.4) and (5.1.5), we could argue that, in (5.1.1), a penalization term given by $\|x\|_0$ could be more appropriate in order to recover sparsity (the smaller $\|x\|_0$, the sparser x). However the choice of the 1-norm, is widely accepted in literature (cf. [26]), since the quality of the reconstruction is similar to the one we would obtain employing the functional $\|\cdot\|_0$; moreover, solving (5.1.2) for $p = 1$ is computationally cheaper than solving it for $p = 0$, since the objective function of the former is convex. Referring again to (5.1.1), one can also consider changing the norm on the fit-to-data term, and solve the general optimization problem

$$\min_{x\in\mathbb{R}^n} \left\{\|b - Ax\|_q^q + \lambda\|x\|_p^p\right\}, \tag{5.1.6}$$

where we assume $1 \le p < 2$, $1 \le q < 2$. If the goal is to preserve jumps in x, then we may prefer to solve

$$\min_{x\in\mathbb{R}^n} \left\{\|b - Ax\|_2^2 + \lambda\mathrm{TV}(x)\right\}, \tag{5.1.7}$$

where $\mathrm{TV}(x)$ is the *Total Variation* functional [110], defined in a continuous one-dimensional setting setting as

$$\mathrm{TV}(f) = \sup_{\Delta} \left(\sum_{\substack{i \\ x_{i-1},\, x_i \,\in\, \Delta}} |f(x_i) - f(x_{i-1})|\right),$$

where f is a function defined on the interval $[a, b]$ and Δ is the set of all the partitions $\mathfrak{p}$ of $[a, b]$, i.e., $\mathfrak{p} \in \Delta$ if and only if

$$\mathfrak{p} = \{x_0, \ldots, x_n,\ n \in \mathbb{N} : a = x_0 < x_1 < \cdots < x_n = b\}.$$

In particular, assuming that f is a smooth function, one can give the following simpler definition of the TV functional:

$$\mathrm{TV}(f) = \int_a^b \left| \frac{df}{dx} \right| dx.$$

Still assuming that $f : [a, b] \times [c, d] \to \mathbb{R}$ is a smooth function of two variables, we can define the TV functional in the two-dimensional setting as

$$TV(f) = \int_c^d \int_a^b \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}\, dx\, dy. \tag{5.1.8}$$

Nonlinear optimization methods are needed to solve problems (5.1.2), (5.1.6), (5.1.7): therefore, these regularization schemes are more computationally demanding than simply solving the linear least squares problem associated with the standard ℓ^2 formulation of Tikhonov regularization (1.3.10). A lot of work has been done to propose suitable and efficient algorithms that can deal with this class of nonlinear problems; see, for instance, [123] for problem (5.1.7) and [125] for problem (5.1.6), and the references therein. Just to give an idea of how these strategies work, we briefly describe the method SpaRSA (Sparse Reconstruction by Separable Approximation) derived in [125]. SpaRSA is an algorithmic framework that can be employed to solve problem (5.1.1), where $\mathcal{J}$ is a smooth convex function and $\mathcal{R}$ is a nonsmooth, possibly nonconvex, regularizer; SpaRSA is currently regarded as one of the state-of-the-art methods to solve a wide class of problems of the form (5.1.1). The solution of (5.1.1) is essentially recovered by solving a sequence of intermediate optimization subproblems of the form

$$x_{m+1} = \arg\min_z 2(z - x_m)^T \nabla \mathcal{J}(x_m) + \alpha_m \|z - x_m\|_2^2 + \lambda \mathcal{R}(z), \tag{5.1.9}$$

for some $\alpha_m \in \mathbb{R}^+$. The first two terms of the optimization function in (5.1.9) can be regarded as a quadratic separable approximation of $\mathcal{J}$ at x_m, which interpolates the first derivative and uses a diagonal approximation $\alpha_m I_m$ for the Hessian. The basic SpaRSA algorithm is an inner-outer iteration scheme that can be outlined as follows: once a factor $\beta > 1$ and two bounds $\alpha_{\min}$, $\alpha_{\max}$ for the values of the parameter α_m at each outer and inner iteration have been set, and an initial guess x_0 has been chosen, at the m-th outer iteration a suitable value for α_m is computed and the subproblem (5.1.9) is solved. Then some inner iterations, whose purpose is going on updating $\alpha_m = \beta\alpha_m$ and solving the subproblem (5.1.9) until x_{m+1} satisfies an acceptance criterion, are performed. At this point, the inner iterations are stopped and the next outer iteration is performed, i.e., α_{m+1} is again computed, intermediate problems of the form (5.1.9) are solved, and the values of α_{m+1} are updated. The outer iterations are stopped as soon as an approximate solution computed at the end of an inner iteration cycle satisfies some stopping criterion. The default method used to set α_m at the beginning of each inner cycle is the standard Barzilai-Borwein [4] formula: different variants of the SpaRSA approach are defined by varying the strategy to choose α_m, the acceptance criterion at the end of each inner iteration and the stopping criterion at the end of each outer iteration. Of course, SpaRSA method is particularly convenient when the solution of each subproblem (5.1.9) is computationally cheap: for instance, this happens when the regularizer $\mathcal{R}$ is separable (we can easily prove that $\mathcal{R}(x) = \|x\|_p^p$ is separable and that, for certain values of p, problem (5.1.9) has an unique closed form solution), when the regularizer $\mathcal{R}$ is group separable, or even when $\mathcal{R}(x) = \mathrm{TV}(x)$. Therefore, the SpaRSA approach is competitive with the state-of-the-art methods specifically designed to solve problems (5.1.2) and (5.1.7) and, at the same time, can efficiently handle

more general problems of the form (5.1.1). Finally we remark that, whenever a good starting point x_0 is available, including it into the SpaRSA algorithm (SpaRSA with warm restarts) can significantly accelerate the convergence; moreover, SpaRSA is slow when the parameter λ in (5.1.1) is small. Therefore, the value of the parameter λ in (5.1.1) is determined by employing the following adaptive continuation scheme: basically, at the beginning, one solves the problem (5.1.1) using a large value for λ and, once the stopping criterion is fulfilled, the value of λ is decreased and the SpaRSA scheme is restarted using the last computed solution as an initial guess; the restarting scheme is stopped as soon as a suitable value for λ has been recovered.

In the following we just consider problems of the form (5.1.1), where the first term of the objective function is the squared 2-norm residual $\|b - Ax\|_2^2$ (exactly as in the usual Tikhonov regularization (1.3.4)). However, the approach we describe is fairly general and we believe that, with minor changes, the methods described can be used to deal with a variety of combinations of fit-to-data and regularization terms. This strategy has been first proposed in [37].

5.2 Iteratively Reweighted Least Squares (IRLS) Method

The approach we extensively describe in Sections 5.3 and 5.4 consists in approximating the regularization term by means of a quadratic functional, very similarly to what is done for the Iteratively Reweighted Least Squares (IRLS) method (cf. [11, §4.5]). This method was originally introduced in a statistical framework to approximate the solution of

$$\min_x \|b - Ax\|_p^p, \quad 1 \leq p < 2, \tag{5.2.1}$$

where an ℓ^p norm estimator is employed instead of the usual ℓ^2 norm, since it can lead to a more robust solution when recovering certain parameters in a linear model. The basic idea behind the IRLS method is to reduce the minimization (5.2.1) to a sequence of least squares problems involving a weighted ℓ^2 norm, i.e., a sequence of problems of the form

$$\min_x \|L(b - Ax)\|_2^2, \tag{5.2.2}$$

where L is a diagonal weighting matrix that is updated at each step using the solution obtained at the previous iteration. In Algorithm 16 we summarize the details of the IRLS method.

Algorithm 16: IRLS method

Input: A, b, $x^{(0)}$, p

For $m = 1, 2, \ldots,$ until some stopping criterion is satisfied

1. Compute $[r^{(k)}]_i = [b - Ax^{(k)}]_i$, $i = 1, \ldots, n$.
2. Define $L^{(k)} = \operatorname{diag}\left((|[r^{(k)}]_i|^{(p-2)/2})_{i=1,\ldots,n}\right)$.
3. Compute
$$\delta^{(k)} = \arg\min_\delta \left\| L^{(k)} \left(r^{(k)} - A\delta \right) \right\|.$$
4. Consider $x^{(k+1)} = x^{(k)} + \delta^{(k)}$.

In Algorithm 16 and later in this chapter, we use the notation $[\cdot]_i$ to denote the i-th element of the vector inside the brackets; analogously $[W]_{ij}$ denotes the (i, j)-th entry of the matrix W. We introduce this notation in order to avoid confusion with the subscript of the vectors x_m, which is always used to denote the vector x at the m-th iteration of some iterative method. Looking at Algorithm 16, we note that we should give in input an initial guess for the solution, which is employed to compute the initial residual and, therefore, to obtain the first weighting matrix.

We remark that, in order to define the weighting matrix, the best choice would be to take $x^{(k)} = x^{ex}$ at step 2 of Algorithm 16: indeed, in this case, we would obtain

$$\|Lx^{ex}\|_2^2 = \|x^{ex}\|_p^p.$$

Of course, this is impossible in practice. The main idea behind IRLS is that, if a convergent iterative scheme is used to compute $x^{(k)}$, then the matrix $L^{(k)}$ is an increasingly better approximation of the optimal L and, as a consequence, the norm $\|x\|_p$ is increasingly better approximated.

In the following we adapt this strategy to work in connection with the Arnoldi-Tikhonov method introduced in Chapter 3: in the AT method case, we use the IRLS approach to adaptively define a suitable regularization matrix (or, using the IRLS terminology, weighting matrix) L that allows us to approximate

$$\|Lx\|_2^2 \simeq \mathcal{R}(x)\,. \tag{5.2.3}$$

In other words, we use the IRLS approach just to approximate the regularization term. In the following we just focus on the cases $\mathcal{R}(x) = \|x\|_1$ and $\mathcal{R}(x) = \mathrm{TV}(x)$. The matrix L is adaptively defined since it is automatically updated at each iteration or when a convenient number of iterations has been performed (we refer to Sections 5.3 and 5.4, respectively, for the details). In order to derive the regularization matrices that are used in the remaining part of this chapter, we just assume that some approximation x^* of the solution is available. We specify how to choose this approximation in the next sections.

p-norm case. We define the diagonal matrix

$$L^* = \mathrm{diag}\left(\left(|[x^*]_i|^{\frac{p-2}{2}}\right)_{i=1,\dots,n}\right), \tag{5.2.4}$$

where we use the notation L^* to indicate that the considered matrix depends on the approximation x^*. Typically, x^* is the intermediate solution computed at some previous iteration of the Arnoldi-Tikhonov method (Algorithm 9). We can immediately check that, if $x^* = x^{ex}$, then

$$\|L^{ex}x^{ex}\|_2^2 = \sum_{i=1}^{n}\left(|[x^{ex}]_i|^{\frac{p-2}{2}}\,x_i^{ex}\right)^2 = \sum_{i=1}^{n}|x_i^{ex}|^p = \|x^{ex}\|_p^p\,. \tag{5.2.5}$$

Since we are particularly interested in the $\|x\|_1$ case, we explicitly write the matrix (5.2.4) for $p = 1$

$$L^* = \mathrm{diag}\left(\left(\frac{1}{\sqrt{|[x^*]_i|}}\right)_{i=1,\dots,n}\right). \tag{5.2.6}$$

We remark that, when $p < 2$, care is needed when defining (5.2.4), since division by 0 may occur if $[x^*]_i = 0$ for some $i = 1,\dots,n$. Therefore it is safer to set a small threshold $\tau > 0$ and take

$$L^* = \mathrm{diag}\left(f_\tau([x^*]_i)_{i=1,\dots,n}\right)\,, \tag{5.2.7}$$

where

$$f_\tau(\chi) = \begin{cases} |\chi|^{(p-2)/2} & \text{if } \chi > \tau \\ \tau^{(p-2)/2} & \text{if } \chi \le \tau \end{cases}. \tag{5.2.8}$$

TV case. We now focus on approximating the TV regularization operator in (5.1.7), which is often used in image restoration problems. For this reason we consider the definition of the two-dimensional TV functional given in (5.1.8). Recalling that the matrices D_1^h and D_1^v defined in (1.3.21) are scaled finite-differences approximations of the first derivative in

the horizontal and vertical directions, it is natural to define the discrete version of the functional (5.1.8) by

$$\mathrm{TV}(x) = \sum_{i=1}^{N} \sqrt{[D_1^h x]_i^2 + [D_1^v x]_i^2} = \|\sqrt{d}\|_1 \,, \tag{5.2.9}$$

where, assuming to work with an image $X \in \mathbb{R}^{N\times N}$ expressed in vectorial form as $x = \mathrm{vec}(X) \in \mathbb{R}^n$ (cf. Section 1.2.2) and taking $\bar{n} = (N-1)N = n - N$,

$$d \in \mathbb{R}^{\bar{n}}, \quad [d]_i = [D_1^h x]_i^2 + [D_1^v x]_i^2, \quad i = 1, \ldots, \bar{n}$$

and the square root operation in (5.2.9) is done element-wise. In the following we prove that it is still possible to approximate this term in the IRLS framework, even if the regularization operator is defined in a more cumbersome way because we also have to incorporate the derivatives D_1^h and D_1^v; to do this we closely follow the derivations made in [108]. The first considerations are made assuming to work with the exact solution x (not to overload the notations we discard the superscript "ex"). Before approximating the ℓ^1 norm in (5.2.9), it is convenient to provide the following relation for the vector d: recalling the definitions of the matrices D_1^h, D_1^v and D_1^{hv} given in (1.3.21) and (1.3.23), we obtain

$$\|\sqrt{d}\|_2^2 = \sum_{i=1}^{\bar{n}} [D_1^h x]_i^2 + [D_1^v x]_i^2 = \|D_1^{hv} x\|_2^2 \,.$$

Let us consider the diagonal weighting matrix

$$W = \mathrm{diag}([\widetilde{W}, \widetilde{W}]) \in \mathbb{R}^{2\bar{n}\times 2\bar{n}},$$

where

$$\widetilde{W} = \mathrm{diag}\left(\left(\left([D_1^h x]_i^2 + [D_1^v x]_i^2\right)^{-1/4}\right)_{i=1,\ldots,\bar{n}}\right) \in \mathbb{R}^{\bar{n}\times\bar{n}} \,.$$

We can verify that the optimal regularization matrix to choose in (5.2.3) in order to recover the TV regularization operator is

$$L = W D_1^{hv} \in \mathbb{R}^{2\bar{n}\times n},$$

since

$$\begin{aligned}
\|Lx\|_2^2 &= \sum_{i=1}^{\bar{n}} \left([\widetilde{W}]_{ii}^2 \left(\left[D_1^h x\right]_i^2 + [D_1^v x]_i^2 \right) \right) \\
&= \sum_{i=1}^{\bar{n}} \left(\left([D_1^h x]_i^2 + [D_1^v x]_i^2 \right)^{-1/2} \left([D_1^h x]_i^2 + [D_1^v x]_i^2 \right) \right) \\
&= \sum_{i=1}^{\bar{n}} \left([D_1^h x]_i^2 + [D_1^v x]_i^2 \right)^{1/2} = \mathrm{TV}(x) \,.
\end{aligned}$$

When, in practice, the exact solution x is not available, one considers an approximation x^* of x and defines

$$L^* = W^* D_1^{hv}, \tag{5.2.10}$$

where

$$W^* = \mathrm{diag}([\widetilde{W}^*, \widetilde{W}^*])$$

and

$$\widetilde{W}^* = \mathrm{diag}\left(\left([D_1^h x^*]_i^2 + [D_1^v x^*]_i^2\right)^{-1/4}\right) \,.$$

As before, to avoid division by zero when both $[D_1^h x^*]_i$ and $[D_1^v x^*]_i$ are null for some $i = 1, \ldots, \bar{n}$, we set a small threshold $\nu > 0$ and we instead consider

$$\widetilde{W}^* = \operatorname{diag}\left(g_\nu\left([D_1^h x^*]_i^2 + [D_1^v x^*]_i^2\right)\right),$$

where

$$g_\nu(\chi) = \begin{cases} \chi^{-1/4} & \text{if } \chi > \nu \\ \nu^{-1/4} & \text{if } \chi \leq \nu \end{cases}. \tag{5.2.11}$$

We note that the IRLS approach has been previously used to solve problems (5.1.2) and (5.1.7). For example, in [124] the authors define a particular matrix W that can be used to approximate $\|\cdot\|_1$ as well as the TV operator. A very similar approach for the TV regularization is adopted in [97], where the authors show that the algorithm can be regarded as a majorization-minimization (MM) process [69]. However, to the best of our knowledge, the only published work that adopts the IRLS strategy to deal with problems of the form (5.1.6) is [109]. All these approaches solve, at each iteration, a weighted least squares problem using the conjugate gradient method (cf. Sections 2.3.3 and 2.4) applied to the normal equations. In particular, each IRLS iteration generates a new Krylov subspace from scratch or, at most, starting from the last computed solution at the end of the previous set of conjugate gradient iterations. Furthermore, to generate each Krylov subspace, it is necessary to compute matrix-vector multiplications with both A and A^T. The approach described in the next sections is designed to work in connection with some variant of the Arnoldi-Tikhonov method (3.2.2) and, specifically, we show that our strategy can be implemented very efficiently by interpreting the scheme as a flexibly preconditioned Krylov subspace method (mainly in the 1-norm case, cf. Section 5.3) or a restarted Krylov subspace method (cf. Section 5.4): this means that we generate only one Krylov subspace. Moreover, each iteration of our approach only requires one matrix-vector multiplication with A. Finally, since after some iterations the Arnoldi-Tikhonov method exhibits a stable behavior thanks to the approximation properties of the Krylov subspaces in which we are looking for a solution (recall the arguments in Sections 2.5.3, 3.1, 4.1.1), the use an IRLS strategy in connection with the AT method is fully justified. Basically, in the Arnoldi-Tikhonov case, the iterative setting of the generic IRLS method (described in Algorithm 16) is provided by the Arnoldi algorithm and the regularization (weighting) matrices are updated at every iteration (typically this is the case when p-norms are involved, cf. Section 5.3) or after a convenient number of iterations have been performed (cf. Section 5.4).

5.3 Flexible AT method

In this section we describe the first strategy to practically implement the variant of the Arnoldi-Tikhonov method defined in (3.2.2) in connection with the regularization matrices of the type (5.2.4). Thanks to the progressively stable behavior of the AT method and referring to the derivations made in Section 5.2, at the m-th iteration of the Arnoldi algorithm it is natural to consider, as approximate solution x^*, the vector $x_{m-1} = x_0 + W_{m-1} y_{m-1}$ (i.e., the solution obtained at the previous step of the Arnoldi-Tikhonov method) and define, as approximate regularization matrix L^*,

$$L_m = \operatorname{diag}\left(\left(|[x_{m-1}]_i|^{\frac{p-2}{2}}\right)_{i=1,\ldots,n}\right). \tag{5.3.1}$$

If no meaningful initial guess is available when the first iteration is performed (i.e. when $m = 1$), we can simply take $x_0 = 0$ and $L_1 = I$.

At this point we recall that, at m-th step of the AT method, the matrix associated with the regularized least squares problem given by equation (3.2.2) is made of two blocks: the first one,

$\bar{H}_m$, is a Hessenberg matrix of size $(m+1) \times m$ while the second one, LW_m, is typically a tall rectangular matrix of size $q \times m$, $q \simeq n$. As remarked in Section 3.2, from a computational point of view this implies that, as long as $m \ll n$, we can solve the regularized least squares problem (3.2.2) without much effort, for instance computing the so-called "skinny" QR factorization; however, as the number of iterations m increases, solving directly the projected problem (3.2.2) becomes computationally demanding. On the other hand, if we are able to cheaply transform the original problem (1.3.4) into standard form, the matrix associated to the least squares problem (3.1.3) is of dimension $(2m+1) \times m$ and therefore the computational effort to solve the direct problem is no longer very demanding, even when m increases. As addressed in Section 1.3.1, many sophisticated strategies have been elaborated to transform the problem (1.3.4) into standard form. However, in the case of sparse reconstruction ($p = 1$), the regularization matrix is a square diagonal matrix defined by (5.3.1); moreover, once a threshold has been set, the matrix is nonsingular. In this case (more generally, when a square nonsingular matrix L is involved), transformation into standard form is formally rather simple, since it suffices to consider a change of variables: one just sets $\hat{x} = Lx$, $\hat{x}^* = Lx^*$, $\hat{A} = AL^{-1}$, solves the problem

$$\min_{\hat{x} \in \mathbb{R}^n} \left\{ \|b - \hat{A}\hat{x}\|_2^2 + \lambda \|\hat{x} - \hat{x}^*\|_2^2 \right\}, \tag{5.3.2}$$

and comes back to the solution of the original problem by taking $x = L^{-1}\hat{x}$. Since the involved matrices are diagonal, the inverses can be computed inexpensively. One more positive side-effect of involving the matrix L^{-1} is that, when $1 \leq p < 2$ (and therefore also when we want to force sparsity), one should not worry about setting a threshold in (5.2.7), since the fractions do not appear anymore.

In general, when we want to apply the Arnoldi-Tikhonov method to the transformed system (5.3.2), we build the Krylov subspace $\mathcal{K}_m(AL^{-1}, \hat{r}^*)$, where $\hat{r}^* = b - AL^{-1}\hat{x}^*$ and therefore the inverse of the regularization matrix L can be formally regarded as a right-preconditioner for the original system: we again underline that L^{-1} is just employed to enforce some regularity (in this case, sparsity) into the reconstructed solution and it is not a proper preconditioner (i.e., recalling the discussion in Section 2.6, L^{-1} does not affect the speed of convergence of the iterative method, nor improves the conditioning of the matrix A). Since the Arnoldi-Tikhonov method approximate solution of the (standard form) Tikhonov regularized problem (1.3.9) can be considered as a regularized GMRES solution of the original linear system (cf. the discussion in Section 3.1), in this setting we can naturally regard the solution of (5.3.2) achieved by the AT method as a regularized right-preconditioned GMRES solution. More specifically, the approximate solution of (5.3.2) is given by $\hat{x}_m = \hat{x}^* + W_m y_m$, where the orthonormal columns of W_m span the space $\mathcal{K}_m(AL^{-1}, \hat{r}^*)$ and are such that

$$AL^{-1}W_m = W_{m+1}\bar{H}_m\,; \tag{5.3.3}$$

the approximate solution of the original (general form) Tikhonov regularized problem (1.3.4) is recovered by taking $x_m = x^* + L^{-1}W_m y_m$.

Since, in this setting, we want to adopt the IRLS approach to solve the regularized problem (5.1.2), the matrix L must be updated at each step of the AT method and, as a consequence, the preconditioner (5.3.1) in $\mathcal{K}_m(AL^{-1}, b)$ changes at each iteration. For this reason we must consider particular Krylov subspaces that allow variable preconditioning: the so-called Flexible Krylov subspaces.

5.3.1 Flexible Krylov Subspaces

Flexible Krylov subspaces [118] were introduced in various frameworks in order to incorporate an increasingly improved preconditioner into the original Krylov subspace: a typical situation is when the preconditioning matrix is obtained by iteratively solving a linear system (indeed,

often the system defining the preconditioning is itself solved by applying a Krylov subspace method); in this case an inner-outer iterative scheme is established since not only the solution of the main linear system, but also the system defining the preconditioning, are updated at each iteration (we cite [117] for a deep understanding of how the inner-outer iteration scheme works). Because of the analogies between the GMRES method the AT method (both of them are based on the Arnoldi algorithm, cf. Section 3.1), in Algorithm 17 we report the FGMRES (Flexible GMRES) method derived in [114].

Algorithm 17: FGMRES method

Input: A, b, x^*.
Compute $r^* = b - Ax^*$ and normalize r^*: $w_1 = {}^{r^*}/_{\|r^*\|}$.
For $j = 1, 2, \ldots, m$

1. Compute $z_j = L_j^{-1} w_j$.
2. Compute $v = Az_j$.
3. For $i = 1, \ldots, j$: compute $h_{i,j} = (v, w_j)$.
4. Compute $v = v - \sum_{i=1}^{j} h_{i,j} w_j$.
5. Take $h_{j+1,j} = \|v\|_2$.
6. If $h_{j+1,j} = 0$ take $m = j$ and stop; else take $w_{j+1} = {}^{v}/_{h_{j+1,j}}$.

Define

- $Z_m = [z_1, \ldots, z_m] \in \mathbb{R}^{n \times m}$;
- $W_{m+1} = [w_1, \ldots, w_{m+1}] \in \mathbb{R}^{n \times (m+1)}$;
- $\bar{H}_m = \{h_{i,j}\}_{1 \le i \le j+1\,;\, 1 \le j \le m}$.

Compute $y_m = \arg\min_{y \in \mathbb{R}^m} \|c - \bar{H}_m y\|$.
Take $x_m = x^* + Z_m y_m$.

Steps 1–6 of Algorithm 17 can be regarded as a flexible variant of the usual Arnoldi algorithm (Algorithm 2); we also note that the flexible Arnoldi algorithm is implemented using the Standard Gram-Schmidt orthogonalization procedure. We remark that the FGMRES method can be considered just when right preconditioning is involved since, in this case, the variable preconditioner directly enters the definition of the basis for the approximate solution.

Looking at Algorithm 17 we understand that the flexible Arnoldi algorithm leads to the decomposition

$$AZ_m = W_{m+1} \bar{H}_m \,, \tag{5.3.4}$$

where only the matrix $W_{m+1} \in \mathbb{R}^{n \times (m+1)}$, whose first vector is $r^*/\|r^*\|_2$, has orthonormal columns; $\bar{H}_m \in \mathbb{R}^{(m+1) \times m}$ is still upper Hessenberg. Anyway, relation (5.3.4) is formally very similar to (2.2.3) and (5.3.3), the main difference being that in (5.3.4) the action of AL_j^{-1} on a vector of the Krylov subspace z_j does not belong to the Krylov subspace anymore. Indeed, the basis vector for the approximate solution x_m now belong to to the subspace

$$\text{span}\left\{L_1^{-1} w_1,\, L_2^{-1} w_2, \ldots, L_m^{-1} w_m\right\},$$

and the preconditioning is implicitly defined into the columns of Z_m (which are not orthogonal anymore); we also underline that the approximate solution of a the FGMRES method takes the form

$$x_m = x^* + Z_m y_m, \tag{5.3.5}$$

as stated in the last line of Algorithm 17 (this is equivalent to saying that the vector $x_m - x^*$ is given by a linear combination of the columns of Z_m).

We again remark that, when considering the Arnoldi-Tikhonov method with a regularization matrix of the form (5.3.1), we are basically operating as described in Section 5.2, since at each iteration we update the regularization matrix exploiting the intermediate solutions computed at the previous steps: if the approximate solutions converge to the exact one (cf. the discussion on the convergence of the AT method in Section 3.1), we can increasingly better approximate the term (5.2.5). In the following we better explain how the Arnoldi-Tikhonov method can be used in connection with flexible Krylov subspaces: we call this strategy Flexible-Arnoldi-Tikhonov (Flexi-AT) method. Basically, at the m-th iteration of the Flexible AT method, we solve a problem of the form (5.3.2) with $L^{-1} = L_m^{-1}$, i.e.,

$$\min_{x \in \mathbb{R}^n} \left\{ \|b - AL_m^{-1}x\|_2^2 + \lambda \|x - x^*\|_2^2 \right\}, \tag{5.3.6}$$

where the matrix L_m is given in (5.3.1): to do this we perform a step of the flexible Arnoldi algorithm (lines 1–6) and we update the matrices W_{m+1}, Z_m and $\bar{H}_m$. Exploiting once more the link between GMRES (in this case, FGMRES) and standard AT, we project the problem (5.3.6) into flexible Krylov subspaces by substituting $x = x^* + Z_m y$ into (5.3.6) and we use the relation (5.3.4) and the properties of the matrices therein involved. In this way we obtain the following regularized projected problem that we have to solve at each iteration

$$\begin{aligned} y_{\lambda,m} &= \arg\min_{y \in \mathbb{R}^m} \left\{ \left\| \|r^*\|_2 e_1 - \bar{H}_m y \right\|_2^2 + \lambda \|y\|_2^2 \right\} \\ &= \arg\min_{y \in \mathbb{R}^m} \left\| \begin{pmatrix} \bar{H}_m \\ \sqrt{\lambda} I_m \end{pmatrix} y - \begin{pmatrix} \|r^*\|_2 e_1 \\ 0 \end{pmatrix} \right\|_2^2 . \end{aligned} \tag{5.3.7}$$

Once the vector $y_{\lambda,m}$ has been computed (by solving, as usual, the flexible Arnoldi-Tikhonov regularized least squares problem (5.3.7)), the approximate solution of the full-dimensional problem $x_{\lambda,m}$ is recovered by taking $x_{\lambda,m} = x^* + Z_m y_{\lambda,m}$.

5.3.2 Parameter choice strategy

To choose the the regularization parameter $\lambda = \lambda_m$ at each step of the Flexi-AT method we can potentially apply any of the schemes described in Chapter 4. In particular, assuming that a fairly accurate estimate for the norm of the noise e is available, we consider the secant update method (Section 4.1.1). Thanks to the analogies between the AT and the Flexi-AT method, we can expect that the secant update method can easily be extended to work in connection with the method (5.3.7). First of all, at the m-th iteration of the flexible Arnoldi algorithm, let us consider the norm of the discrepancy vector (4.1.1) that, in the Flexi-AT case, can be written as

$$\begin{aligned} \phi_m(\lambda) &= \|b - Ax_{\lambda,m}\| = \|b - A(x^* + Z_m y_{\lambda,m})\| \\ &= \left\| r^* - W_{m+1}\bar{H}_m y_{\lambda,m} \right\| = \left\| \|r^*\|_2 e_1 - \bar{H}_m y_{\lambda,m} \right\|. \end{aligned} \tag{5.3.8}$$

As usual, to derive the above equalities we have exploited the decomposition (5.3.4) and the properties of the matrices therein involved: as a result, in the following, we can consider the projected discrepancy (5.3.8). We also note that relation (5.3.8) is formally equivalent to (4.1.3), the only difference being that the matrix $\bar{H}_m$ appearing in (5.3.8) is defined by relation (5.3.4), while the matrix $\bar{H}_m$ appearing in (4.1.3) is defined by relation (2.2.3). Analogously, the quantity $\phi_m(0)$ is recovered by computing

$$y_m = \arg\min_{y \in \mathbb{R}^m} \|c - \bar{H}_m y\|, \tag{5.3.9}$$

and then taking

$$\phi_m(0) = \left\| \|r^*\|_2 e_1 - \bar{H}_m y_m \right\| ; \tag{5.3.10}$$

we can clearly regard $\phi_m(0)$ as the residual associated to the m-th iteration of the FGMRES method (cf. Algorithm 17). Considering the linear approximation

$$\phi_m(\lambda) \simeq \phi_m(0) + \lambda d_m,$$

which is formally equivalent to (4.1.7), and computing d_m as described in equation (4.1.8), we obtain the following update strategy for the regularization parameter at each iteration of the flexible Arnoldi algorithm

$$\lambda_{m+1} = \left| \frac{\eta\varepsilon - \phi_m(0)}{\phi_m(\lambda_m) - \phi_m(0)} \right| \lambda_m. \tag{5.3.11}$$

As explained in Section 4.1.1, the parameter λ_1 (employed at the first iteration of the Flexi-AT method) has to be specified by the user, and $\eta > 1$ is a safety parameter for the discrepancy principle. In Algorithm 18 we summarize the derivations of the previous and the present sections.

Algorithm 18: Flexible Arnoldi-Tikhonov (Flexi-AT) method

Input: A, b, x^*, p, λ_1, η, ε.
Initialize: $r^* = b - Ax^*$ and $w_1 = {}^{r^*}/_{\|r^*\|_2}$; take $L_1^{-1} = I_n$.
For $m = 1, 2, \ldots$ until $\|c - \bar{H}_m y_m\| \leq \eta\varepsilon$

1. Perform one step of the flexible Arnoldi algorithm (Algorithm 18) with A, L_m and r^* as inputs, and update the decomposition (5.3.4).
2. Compute the solution $y_{\lambda,m}$ of (5.3.7) with $\lambda = \lambda_m$ and the approximate solution $x_{\lambda,m} = x^* + Z_m y_{\lambda,m}$ of (5.1.2).
3. Compute the FGMRES solution (5.3.9) and evaluate the corresponding residual $\phi_m(0)$ by (5.3.10).
4. Compute $\phi_m(\lambda_m)$ by (5.3.8).
5. Compute d_m by the relation (4.1.8).
6. Compute the regularization parameter λ_{m+1} by (5.3.11).
7. Compute the matrix L_{m+1} by formula (3.2.5).

5.4 Restarted AT method

As said in the previous section, the approach based on the transformation of problem (1.3.4) into standard form is particularly convenient when the regularization matrix is cheaply invertible: this is not the case when, for instance, we want to approximate the Total Variation regularization (5.1.7) using the operator (5.2.10), as described in Section 5.2. In the following we derive an approach, again based on the Arnoldi-Tikhonov method and on the IRLS strategy, which is an alternative to Algorithm 18 and which, beside being still valid to solve problems like (5.1.2), can be also used in the case (5.1.7).

This strategy is essentially based on restarting the Arnoldi algorithm, establishing a sort of inner-outer iteration scheme: at each restart (outer iteration), the regularization matrix $L^{(k)}$ and the initial guess $x_0^{(k)}$ are updated employing the last values computed when a stopping test for the inner iterations is satisfied. Even for this method, all the parameter choice strategies described in Chapter 4 can be potentially employed to find a suitable regularization parameter at each inner iteration and to decide when to stop each inner cycle. As in the previous section, we focus our attention on the secant update method (Section 4.1.1); at the beginning of each outer cycle we set the initial parameter $\lambda_0^{(k)}$ equal to the last computed one at the end of the

previous set of inner iterations. As a matter of notations, the superscript (k) denotes that these quantities are employed during the k-th restart; the first set of inner iterations is performed by simply taking $L^{(0)} = I$, $x_0^{(0)} = 0$ and $\lambda_1^{(0)} = 1$. Waiting until the discrepancy principle is satisfied before restarting the Arnoldi algorithm guarantees that the solution employed to update $L^{(k)}$ and $x_0^{(k)}$ is quite accurate even if, especially during the first restarts, we are solving intermediate problems that roughly approximate the original ones described in (5.1.2) and (5.1.7).

About the number of iterations for each restart, it is well-known that the approximate solutions computed by the Arnoldi-Tikhonov method can very quickly fulfill the discrepancy principle and deliver a regularized solution belonging to a Krylov subspace of dimension $m \ll n$, even for problems of huge dimension (cf. [104] and the arguments Chapter 3). Therefore, considering both the standard form (3.1.3) and the general form (3.2.2) problems, the computational cost for each restart is kept low. Furthermore, as the number of restarts increases, the number of iterations required to satisfy the discrepancy principle decreases: this is due to the stable behavior of the AT method, which after some restarts can compute solutions of similar quality. As a consequence, the regularization matrices $L^{(k)}$ and the Krylov subspaces generated by A and $r_0^{(k)}$ tend to be the same and the discrepancy principle continues to be satisfied; eventually, the discrepancy principle continues to be satisfied after only one step of the Arnodi-Tikhonov method has been performed, resulting in very cheap computations. In this situation, although the quality of the reconstruction may not be substantially improved, performing additional restarts could still be useful in order to keep updating the regularization matrix with slightly better approximations of the solution and, as a consequence, obtain slightly more accurate reconstructions (cf. Figure 5.5.7, upper frame). In the following we will denote by $m_{\text{in}}^{(k)}$ the number of iterations required to fulfill the discrepancy principle at the k-th restart.

Determining when to exactly stop the restarts is not a crucial issue, mainly for two reasons: first of all, as the iterations proceed, the behavior of the solution is very stable (due to the convergent behavior of the AT method, cf. the arguments in Section 3.1) and, secondly, because the cost of each restart is lower and lower. However we can employ some heuristic to set a stopping criterion: looking at the performed tests (cf. Section 5.5) it can be noted that, when the discrepancy is satisfied at the end of each set of inner iterations, the values of the discrepancy function keep to slightly decrease and this can be regarded as a sign that we are computing an increasingly more accurate solution; therefore, we can decide to stop the iterations after reaching a pre-specified threshold for the discrepancy function. We can also choose to continue the iterations until a fixed maximum number of restarts has been carried out. In the following we will denote by m_{out} the total number of restarts.

A variant of the approach just described is to restart the Arnoldi algorithm taking always $x_0^{(k)} = 0$ and exclusively updating the regularization matrix. Although some improvements can be achieved going on with the restarts, the reconstructions are worse than the ones obtained when updating also $x_0^{(k)}$, and the behavior of the error is extremely non-monotone. Moreover, even if we perform many restarts, the number of steps required to fulfill the discrepancy principle at each restart is almost constant. We can conclude that taking into account an initial guess both in the formulation (1.3.4) and in the definition of the Krylov subspaces is beneficial in order to improve the quality of the solution and the efficiency of the method. We propose an example of this fact in Section 5.5.

The method described in this section is quite similar to the ones outlined in [124] and [97]; however, as mentioned in Section 5.1, these strategies deal with the normal equations corresponding to problem (1.3.4) and the resulting linear system (whose coefficient matrix is symmetric positive definite) is solved by performing a limited number of Conjugate Gradient iterations; moreover, the regularization parameter cannot be updated in an adaptive way during the CG iterations.

In Algorithm 19 we summarize the approach so far outlined.

Algorithm 19: Restarted AT (ReSt-AT) method

Input: A, b, η, ε.
Initialize: $\lambda_1^{(0)} = 1$, $L^{(0)} = I_n$ and $x_0^{(0)} = 0$;
For $k = 1, 2, \ldots, m_{\text{out}}$

1. Initialize $r_0^{(k-1)} = b - Ax_0^{(k-1)}$.
2. For $j = 1, \ldots, m_{\text{in}}^{(k-1)}$
   (a) Update the decomposition (2.2.3) by performing one step of the Arnoldi algorithm (Algorithm 2) with inputs A and $r_0^{(k-1)}$.
   (b) Compute $y_{\lambda,j}^{(k-1)}$ by solving (3.2.2) with $L = L^{(k-1)}$ and $\lambda = \lambda_j^{(k-1)}$.
   (c) Compute $\lambda_{j+1}^{(k-1)}$ by formula (4.1.11).
3. Compute $x_{\lambda,m_{\text{in}}^{(k-1)}}^{(k-1)} = x_0^{(k-1)} + W_{m_{\text{in}}^{(k-1)}} y_{\lambda,m_{\text{in}}^{(k-1)}}^{(k-1)}$.
4. Take $\lambda_1^{(k)} = \lambda_{m_{\text{in}}^{(k-1)}}$, $x_0^{(k)} = x_{\lambda,m_{\text{in}}^{(k-1)}}^{(k-1)}$.
5. Define $L^{(k)}$ as (5.2.10), employing $x_0^{(k)}$.

5.4.1 Imposing nonnegativity

In many applications, for instance in image restoration problems, the solution is known to be nonnegative; however, methods based on Krylov subspaces are not guaranteed to compute nonnegative solutions. Finding a way to force nonnegativity can greatly improve the quality of the approximate solution (cf. [14, 92]). In the framework of Algorithm 19 we can enforce nonnegativity at each restart, employing an approach very similar to one of those described in [14]. Referring to Algorithm 19, after a set of inner iterations has been completed, and before updating the solution at step 4 for the next restart, we can project $x_{\lambda,m_{\text{in}}^{(k-1)}}$ into the set

$$\mathfrak{P} = \{x \in \mathbb{R}^n \,:\, [x]_i \geq 0 \;\forall i = 1, \ldots, n\} \tag{5.4.1}$$

of nonnegative vectors. In this way, at each restart, we can consider an initial guess that is nonnegative even if, especially during the first restarts, this nonnegative vector is not guaranteed to satisfy the discrepancy principle. However, thanks to the stable behavior of the AT method, after some restarts we can obtain a solution that is nonnegative and that also fulfills the discrepancy principle. We emphasize that our approach does not properly solve a constrained minimization problem whose constraint set is $\mathfrak{P}$; it is a rather heuristic approach that forces nonnegative solutions by imposing proximity to a nonnegative initial guess at each iteration. The numerical results reported in the next section clearly show the improvements obtained applying the strategy just derived.

In Algorithm 20 we describe the nonnegative version of Algorithm 19.

5.4.2 Flexible AT revisited

Even if the matrix is easily invertible and the Flexible Arnoldi-Tikhonov method can be applied, one can also choose not to update the regularization matrix at each step and to rather employ the ReSt-AT method (Algorithm 19). Provided that we apply the restarting strategy to the standard form problem (5.3.2) and that, at the k-th restart, we build the right-preconditioned Krylov subspaces $\mathcal{K}_m(A(L^{(k)})^{-1}, r_0^{(k)})$, the results are similar to the ones obtained applying the Flexi-AT method (Algorithm 18). This alternative approach is halfway between the flexible AT and the restarted AT methods, since the solution belongs to some preconditioned Krylov

Algorithm 20: Nonnegative Restarted AT (NN-ReSt-AT) method

Input: A, b, η, ε.
Initialize: $\lambda_1^{(0)} = 1$, $L^{(0)} = I_n$ and $x_0^{(0)} = 0$;
For $k = 1, 2, \ldots, m_{\text{out}}$

1. Initialize $r_0^{(k-1)} = b - Ax_0^{(k-1)}$.
2. For $j = 1, \ldots, m_{\text{in}}^{(k-1)}$
   (a) Update the decomposition (2.2.3) by performing one step of the Arnoldi algorithm (Algorithm 2) with inputs A and $r_0^{(k-1)}$.
   (b) Compute $y_{\lambda,j}^{(k-1)}$ by solving (3.2.2) with $L = L^{(k-1)}$ and $\lambda = \lambda_j^{(k-1)}$.
   (c) Compute $\lambda_{j+1}^{(k-1)}$ by formula (4.1.11).
3. Compute $x_{\lambda,m_{\text{in}}^{(k-1)}}^{(k-1)} = x_0^{(k-1)} + W_{m_{\text{in}}^{(k-1)}} y_{\lambda,m_{\text{in}}^{(k-1)}}^{(k-1)}$.
4. Project $x_{\lambda,m_{\text{in}}^{(k-1)}}^{(k-1)}$ into $\mathfrak{P}$:
   for $i = 1, \ldots, n$, if $[x_{\lambda,m_{\text{in}}^{(k-1)}}^{(k-1)}]_i < 0$ then $[x_{\lambda,m_{\text{in}}^{(k-1)}}^{(k-1)}]_i = 0$.
5. Take $\lambda_1^{(k)} = \lambda_{m_{\text{in}}^{(k-1)}}$, $x_0^{(k)} = x_{\lambda,m_{\text{in}}^{(k-1)}}^{(k-1)}$.
6. Define $L^{(k)}$ as (5.2.10), employing $x_0^{(k)}$.

subspace and the preconditioner is updated during the iterations (in this case it is updated after a suitable number of iterations has been performed and not at each iteration). Sometimes we refer to this strategy as Restarted Preconditioned AT (ReSt-Prec-AT). An advantage of the ReSt-Prec-AT over the Flexi-AT, is that some additional constraints can be enforced (exactly as in Algorithm 20).

5.5 Numerical Experiments

In this section we show the results of some numerical tests that contribute to validate the strategies described in the previous sections. Along with the reconstruction obtained using the new algorithms, we present some comparisons with other well-known methods to recover sparse solutions or to perform Total Variation regularization.

Example 1. For the first test we focus on sparse reconstruction and we take, as a test image, a synthetic astronomical image of size 256×256 pixels, characterized by a very sparse pattern: only 0.7% of its elements corresponds to non-black pixels, i.e. can be considered different from zero (cf. Figure 5.5.3, upper left frame). We assume that the available image is corrupted by a spatially variant blur and it is divided into 25 different regions: the PSF is spatially invariant in each region (cf. Section 1.2.2). Gaussian white noise is added and we consider two successive noise levels: the first one is equal to 10^{-2} (cf. Figure 5.5.3, upper right frame), the second one is equal to 10^{-1}. We refer to [88] for a background on the solution of this kind of problem. Further information on this test problem, as well as the associated data, can be obtained from the MATLAB package Restore Tools [90]. In Figure 5.5.1 we plot the values of the relative error, the discrepancy function, and the regularization parameter (all displayed in logarithmic scale) versus the number of iterations. These results are obtained applying Algorithm 18; the noise level in the data is $\widetilde{\varepsilon} = 10^{-2}$. As said in Section 4.1.1, when applying the secant update

method, we choose as starting value for the regularization parameter $\lambda_1 = 1$ and, to define the discrepancy principle (4.1.4), we take the scalar $\eta = 1.01$. The value of the truncation parameter in (5.2.8) is set to $\tau = 10^{-8}$. In this case the stopping criterion determines an approximate solution that belongs to a Flexible Krylov subspace of dimension 23. We note that, for this problem, all the quantities in Figure 5.5.1 exhibit a quite stable behavior after the discrepancy principle (and the stopping criterion) is satisfied: in particular we observe that the relative errors do not deteriorate as the iterations proceed. This feature is typical of the AT methods applied to ill-posed problems since, after the first iterations, the largest singular values of the Hessenberg matrix $\bar{H}_m$ (5.3.4) approximate the largest singular values of the original matrix A and, after some iterations, the quality of the approximation stagnates. Solving the projected problem by means of Tikhonov regularization (which can also be regarded as a spectral filtering method [55]) the computed quantities essentially depend on the decay of the singular values of the Hessenberg matrices, and therefore they tend to have a similar behavior after a certain number of iterations have been performed.

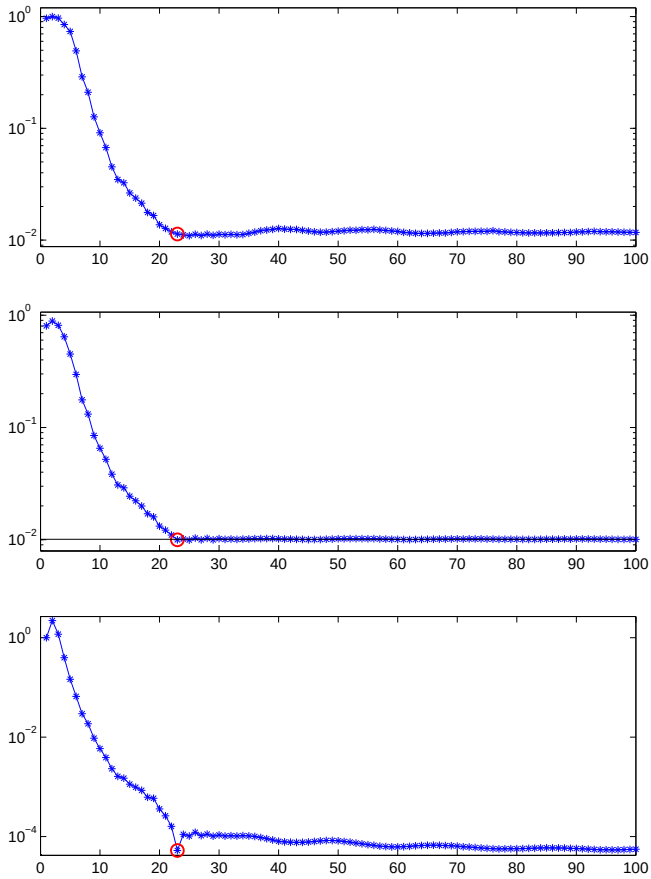

Figure 5.5.1: Values of the relative error (upper frame), the discrepancy function (middle frame) and the regularization parameter (lower frame) versus the number of iterations. These plots are obtained applying Algorithm 18 to the first test problem with $\widetilde{\varepsilon} = 10^{-2}$. The circle at the 23-th iteration highlights the quantities obtained when the discrepancy principle is satisfied; we continue till iteration 100 to illustrate the stable convergence behavior of our algorithms. The horizontal line in the second frame marks the threshold under which the discrepancy principle is satisfied.

In Figure 5.5.2 we display the history of the relative errors obtained when considering different versions of the method described in Section 5.4.2, including the nonnegative one described in Algorithm 20. The test problem and the parameters are as the ones above specified; in this example we consider 20 restarts. It is interesting to note that, when we restart the right-preconditioned Arnoldi scheme with an initial guess that has been projected into the space $\mathfrak{P}$

(see equation (5.4.1)), we typically need slightly more iterations than when taking as initial guess the last computed intermediate solution. In particular, in this experiment, the number of inner iterations during the very first restart is different. Indeed, when we restart for the first time modifying the initial guess, the newly generated Krylov subspace can be pretty different from the old one and therefore we need some additional iterations to satisfy the discrepancy principle. We can also remark that, when restarts are performed taking simply $x_0^{(k)} = 0$, the same number of iterations are needed at each restart and the quality of the computed solution does not significantly improve with each restart.

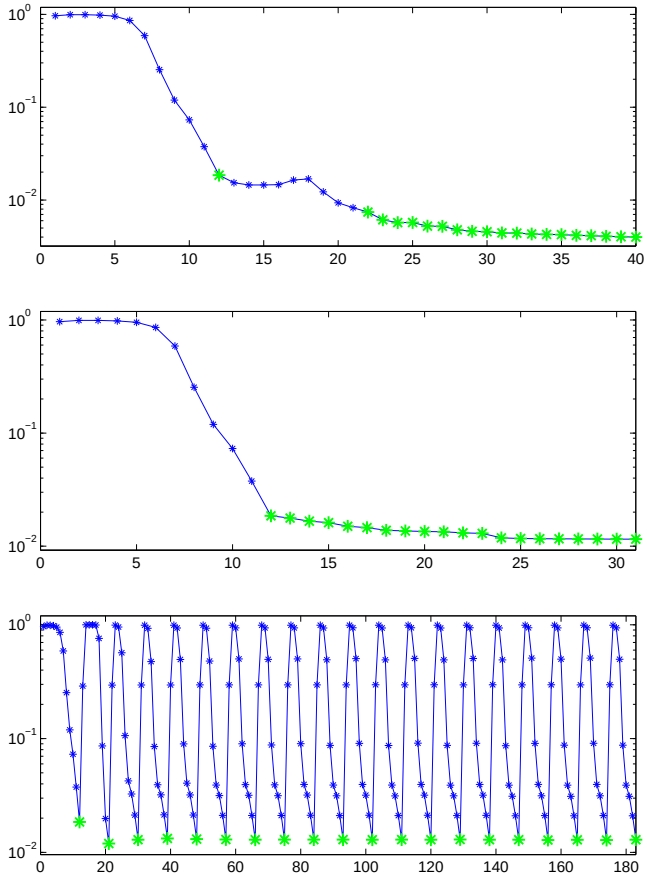

Figure 5.5.2: Behavior of the relative error versus the number of iterations obtained applying three variants of the restarting strategy (Section 5.4.2). In the upper frame we force nonnegativity at each restarts, in the middle frame we take as initial guess the last solution computed at the end of the previous restart, and in the lower frame we take as initial guess $x_0^{(k)} = 0$ at each restart. The bigger asterisks highlight the iteration at which a restart happens.

In Figure 5.5.3 we show the images obtained applying the methods just considered.

We now make some comparisons with other well-established and recently designed methods. In particular we focus on NESTA [6], SpaRSA [125], TwIST [9] and `l1_ls` [74]: these methods can efficiently handle a wide class of minimization problems whose objective function is the sum of a fit-to-data term and a regularization term. We also consider the method IRN-BPDN described in [109], which employs cyclically updated weighting matrices and therefore is quite close to the algorithms described in Section 5.4. Most of the methods just mentioned basically require the user to set a suitable value for the regularization parameter: looking at the graph displayed in the lower frame of Figure 5.5.1, we can assume that a good value for this parameter relative to the problem at hand should be $\lambda = 10^{-4}$ because, going on with the number of iterations m, the values λ_m's stabilize around this point. In addition, we also consider the performance of the standard Arnoldi-Tikhonov method (Section 3.1), and of its range-restricted

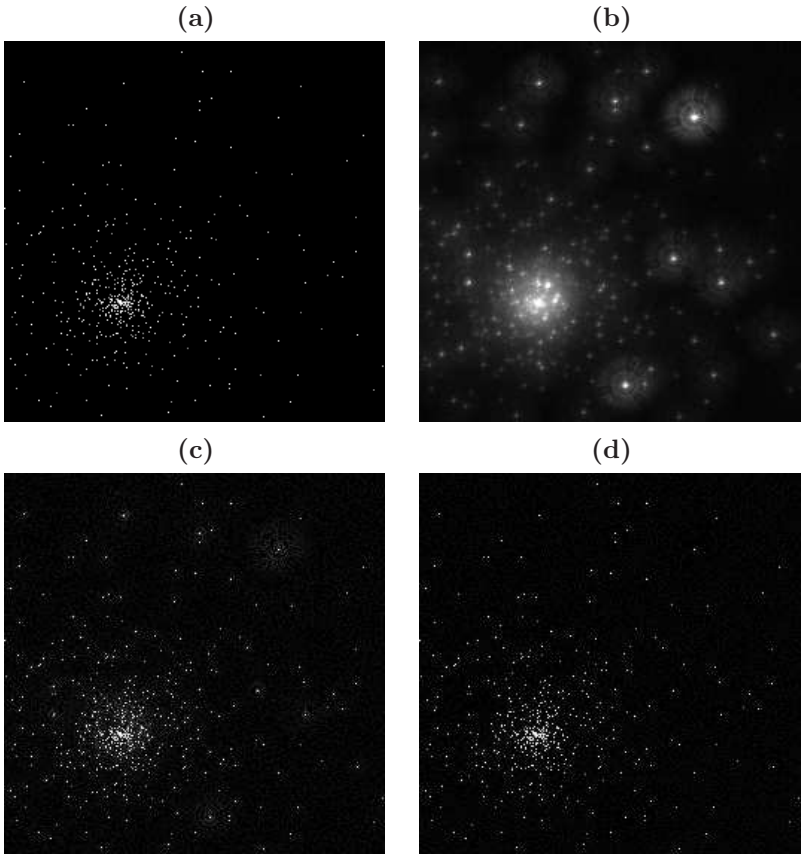

Figure 5.5.3: Images considered in the first example: **(a)** exact image; **(b)** blurred and noisy image, with $\widetilde{\varepsilon} = 10^{-2}$; **(c)** solution obtained at the 23-th iteration of Algorithm 18, i.e. when the discrepancy principle is satisfied; **(d)** solution obtained at the end of Algorithm 20, after 20 restarts have been performed.

version RR-AT [79]. In Figure 5.5.4 we show the behavior of the relative errors versus the number of iterations for many of the methods above cited; in Table 5.1 we report the value of the relative errors obtained when the stopping criterion of each method is fulfilled or when a maximum number of iterations has been performed, along with the total and average (per iteration) running time. In order to keep the comparisons fair, we decide to stop the iterations as soon as the relative change of the error drops below a certain threshold; moreover, we basically use the published version of each method along with the pre-specified parameters (except for λ): therefore, a more accurate tuning of all the parameters can possibly result in a better performance of some of the methods. Looking at the results displayed in Table 5.1 we can

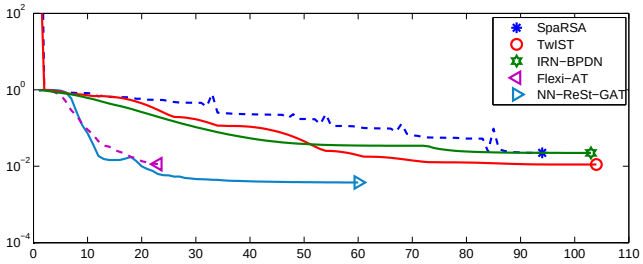

Figure 5.5.4: Behavior of the relative error versus the number of iterations for different optimization methods. The last computed value is highlighted with a different marker and corresponds to the one reported in Table 5.1.

Method	Relative Error	Iterations	Total Time	Average Time
SpaRSA	$2.2365 \cdot 10^{-2}$	94	24.76	0.26
NESTA	$1.7800 \cdot 10^{-2}$	248	306.17	1.23
TwIST	$1.1089 \cdot 10^{-2}$	104	28.02	0.27
`l1_ls`	$2.2257 \cdot 10^{-2}$	298	683.55	2.29
IRN-BPDN	$2.2294 \cdot 10^{-2}$	103	35.72	0.35
AT	$1.8512 \cdot 10^{-2}$	12	0.91	0.08
RR-AT	$1.9171 \cdot 10^{-2}$	18	3.77	0.21
Flexi-AT	$1.1345 \cdot 10^{-2}$	23	2.44	0.11
ReSt-AT	$1.1033 \cdot 10^{-2}$	51	5.95	0.12
NN-ReSt-AT	$3.7530 \cdot 10^{-3}$	60	6.25	0.10

Table 5.1: Comparisons of the performance of some algorithms developed to solve problem (5.1.2). The relative error reported is the one computed at the iteration displayed in the third column. The number of iterations is the minimum between the iterations required to fulfill the stopping criterion and the maximum number of allowed iterations. Both the total time and the average time are expressed in seconds.

state that, for this example, the newly proposed algorithms exhibit excellent performances both in terms of quality of the results and computational time. The primary reason is that, adopting a Krylov subspace approach, at each iteration we deal with projected quantities and therefore all the main computations are executed in reduced dimension. However some of the considered methods, such as SpaRSA, can deal with much more general minimization problems (for instance, problems whose objective function involves a nonlinear fit-to-data functional): in this situation the algorithms described in the present paper cannot be straightforwardly applied.

Finally, in Figure 5.5.5 we show the same quantities displayed in Figure 5.5.1, but this time the noise level is $\widetilde{\varepsilon} = 10^{-1}$.

Example 2. We now consider another example regarding an image restoration problem and we apply the restarted AT method in order to approximate the Total Variation regularization. As a test image we take a computer simulation of how a satellite can be detected by ground based telescopes; in this case the PSF is spatially invariant and models an atmospheric blur.

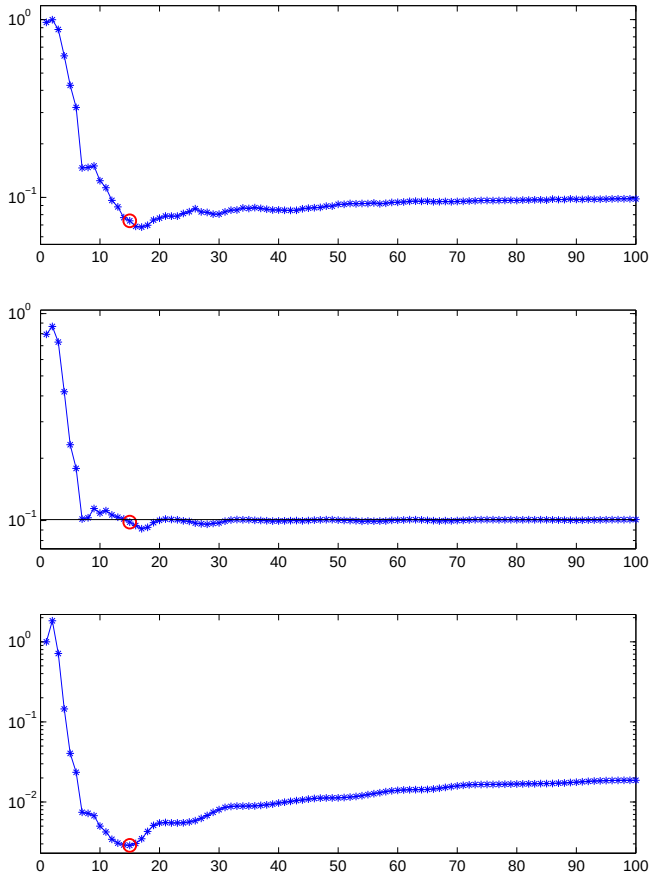

Figure 5.5.5: Values of the relative error, the discrepancy function and the regularization parameter versus the number of iterations. These plots are obtained applying Algorithm 18 to the first test problem with $\widetilde{\varepsilon} = 10^{-1}$. The circle at the 15-th iteration highlights the quantities obtained when the discrepancy principle is satisfied.

This image is of size 256×256 pixels and we further corrupt the blurred image adding white noise, in such a way that the noise level is $\widetilde{\varepsilon} = 10^{-2}$; as in the previous example, the safety factor for the discrepancy principle is $\eta = 1.01$ and the threshold considered in (5.2.11) is $\nu = 10^{-8}$. Both the exact image and the PSF are available in the MATLAB package Restore Tools [90].

In Figure 5.5.6 we show the exact image, the blurred and noisy one and the reconstruction obtained applying Algorithm 20 to enforce nonnegativity: at each restart the regularization matrix, defined by (5.2.10), is updated and the intermediate problems are solved using the Arnoldi Tikhonov method (3.2.2), since in this case the matrix (5.2.10) is not easily invertible. The fundamental difference between this test and the ones so far described is that, in this case, the solution belongs to the Krylov subspace $\mathcal{K}_m(A, b)$ defined taking into account exclusively the matrix A, while in the previous examples the solution belongs to the right-preconditioned Krylov subspaces $\mathcal{K}_m(A(L^{(k)})^{-1}, b)$, where the matrix $L^{(k)}$ is updated at each restart. We perform 200 restarts: this choice is supported by the fact that the discrepancy principle is satisfied after 9 iterations at the beginning, 3 iterations after the first restart and immediately, i.e. after just 1 iteration, in the following restarts. In this way the computational cost of each restart is very low. Moreover, evaluating the error, we see that it is always slightly decreasing.

(a) **(b)** **(c)**

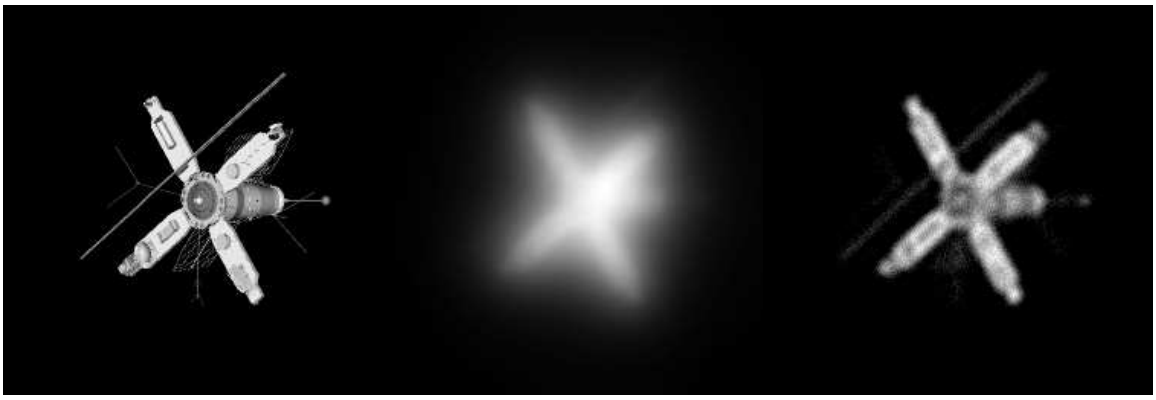

Figure 5.5.6: Images considered in the second example: **(a)** exact image; **(b)** blurred and noisy image, with $\widetilde{\varepsilon} = 10^{-2}$; **(c)** reconstruction obtained applying Algorithm 20 to approximate the Total Variation regularization, after 200 restarts are performed.

Finally we examine the performance of the NN-ReSt-AT Algorithm (Algorithm 20) with respect to some other regularization methods. The first set of comparisons involves the standard Arnoldi-Tikhonov method (3.1.3) and its generalized version (3.2.2) applied with the fixed regularization matrix (1.3.23). In Figure 5.5.7 we display the behavior of the relative errors, of the discrepancy function and of the regularization parameter versus the number of iterations: looking at the upper frame we can clearly see that the approach based on Total Variation regularization can deliver better results than the other ones (this is not unexpected for image restoration problems, cf. [97]). It is also interesting to remark that, for this particular test problem, the standard AT method (3.1.3) slightly outperforms the GAT method (3.2.2).

The second set of comparisons involves three methods that have been designed to iteratively deal with Total Variation regularization and that are closely related to the algorithms described in this Section 5.4, since they both adopt an IRLS procedure to linearly approximate the TV functional. The first one is the Adaptive Majorization-Minimization approach to Total Variation described in [97] (in the following we refer to it as aMM-TV): this method is adaptive in the sense that a parameter selection strategy based on Bayesian considerations is derived. The second one is the algorithm IRN-TV derived in [108]: although the authors in [81] propose a strategy, based on statistical considerations, to automatically set the regularization parameter, we decided to run the IRN-TV method assigning to λ a fixed value: looking at the plot in the

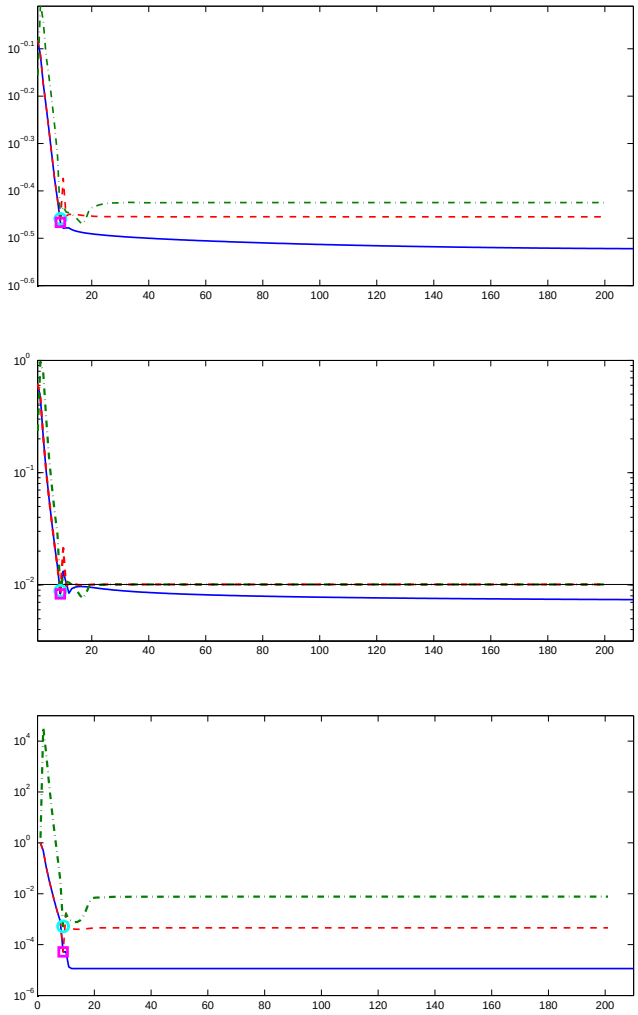

Figure 5.5.7: Comparison of the relative errors, the values of the discrepancy function and the values of the regularization parameter obtained when restoring the image of Example 2 with different methods. The solid line refers to Algorithm 19, the dashed line refers to the standard Arnoldi-Tikhonov method (3.1.3), and the dash-dot line refers to the Generalized Arnoldi-Tikhonov method (3.2.2) with the regularization matrix D_1^{hv} defined in (1.3.23). For both the AT and the GAT method the discrepancy principle is satisfied at the 9th iteration and in all the plots we mark it with a square and a circle, respectively.

lower frame of Figure 5.5.7, we choose $\lambda = 10^{-5}$. Moreover, even if the authors do not seem to suggest it, we take as initial guess at each restart the last computed approximation. The third one is NESTA [6] and we still consider, as regularization parameter, $\lambda = 10^{-5}$.

In Figure 5.5.8 we display the history of the relative errors for the aMM-TV, IRN-TV, NN-ReSt-AT, ReSt-AT and NESTA methods. Some comparisons regarding all the above listed

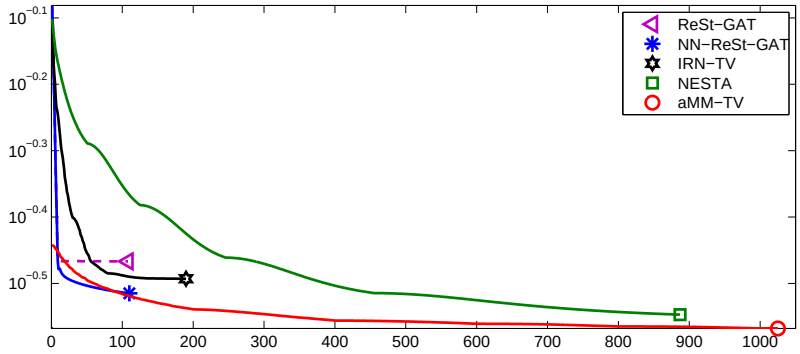

Figure 5.5.8: Behavior of the relative error versus the number of iterations for the NN-ReSt-AT, aMM-TV, IRN-TV(0) and IRN-TV(wr) algorithms. The last computed value is highlighted with a different marker and corresponds to the one reported in Table 5.2.

methods used to solve the present test problem are summarized in Table 5.2, whose layout is identical to that of Table 5.1. Looking at Table 5.2 we can state that the aMM-TV algorithm

Method	Relative Error	Iterations	Total Time	Average Time
aMM-TV	$2.7056 \cdot 10^{-1}$	1025	2159.35	2.10
IRN-TV	$3.2141 \cdot 10^{-1}$	190	14.67	0.08
NESTA	$2.8382 \cdot 10^{-1}$	887	69.57	0.08
ReSt-AT	$3.4138 \cdot 10^{-1}$	108	12.87	0.12
NN-ReSt-TV	$3.0556 \cdot 10^{-1}$	110	13.37	0.12
AT	$3.4176 \cdot 10^{-1}$	9	0.34	0.04
GAT	$3.4809 \cdot 10^{-1}$	9	0.70	0.08
RR-AT	$3.5321 \cdot 10^{-1}$	14	1.39	0.10

Table 5.2: Comparisons of the performances of some algorithms developed to solve the problem (5.1.7). The relative error reported is the one computed at the iteration displayed in the third column. The number of iterations is that required to fulfill the stopping criterion, or a fixed maximum number of allowed iterations. Both the total time and the average time are expressed in seconds.

surely is the best one in terms of quality of the reconstruction, but it is also the most expensive one: indeed, the parameter choice strategy proposed in [97] requires the method to perform a lot of iterations and a lot of restarts before determining a reasonable value for the regularization parameter, resulting in an overall slow convergence. We further remark that, among all the algorithms listed in the above Table 5.2, NN-ReSt-AT is the only one that produces nonnegative solutions.

Bibliography

[1] W. E. Arnoldi. The principle of minimized iteration in the solution of matrix eigenvalue problem. *Quart. Appl. Math.*, 9:17–29, 1951.

[2] J. Baglama and L. Reichel. Augmented GMRES-type methods. *Numer. Linear Algebra Appl.*, 14:337–350, 2007.

[3] J. Baglama and L. Reichel. Augmented LSQR method. *Numer. Algorithms*, 64:263–293, 2013.

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

[5] F. Bauer and S. V. Pereverzev. An utilization of a rough approximation of a noise covariance within the framework of multi-parameter regularization. *Int. J. Tomogr. Stat.*, 4:1–12, 2006.

[6] S. Becker, J. Bobin, and E. Candès. NESTA: A Fast and Accurate First-Order Method for Sparse Recovery. *SIAM J. Imaging Sciences*, 4(1):1–39, 2011.

[7] M. Belge, M. E. Kilmer, and E. L. Miller. Efficient determination of multiple regularization parameters in a generalized L-curve framework. *Inverse Problems*, 18:1161–1183, 2002.

[8] S. Berisha and J. G. Nagy. Iterative image restoration. In R. Chellappa and S. Theodoridis, editors, *Academic Press Library in Signal Processing*, volume 4, chapter 7, pages 193–243. Elsevier, 2014.

[9] J. Bioucas-Dias and M. Figueiredo. A new TwIST: two-step iterative shrinkage/thresholding algorithms for image restoration. *IEEE Trans. Image Proc.*, 16(12):2992–3004, 2007.

[10] Å. Björck. A bidiagonalization algorithm for solving large and sparse ill-posed systems of linear equations. *BIT*, 28(3):659–670, 1988.

[11] Å. Björck. *Numerical Methods for Least Squares Problems*. SIAM, Philadelphia, PA, 1996.

[12] C. Brezinski, M. Redivo-Zaglia, G. Rodriguez, and S. Seatzu. Multi-parameter regularization techniques for ill-conditioned linear systems. *Numer. Math.*, 94:203–228, 2003.

[13] D. Calvetti, G. H. Golub, and L. Reichel. Estimation of the L-curve via Lanczos bidiagonalization. *BIT*, 39(4):603–619, 1999.

[14] D. Calvetti, G. Landi, L. Reichel, and F. Sgallari. Non-negativity and iterative methods for ill-posed problems. *Inverse Problems*, 20:1747–1758, 2004.

[15] D. Calvetti, B. Lewis, and L. Reichel. GMRES-type methods for inconsistent systems. *Lin. Alg. Appl.*, 316:157–169, 2000.

[16] D. Calvetti, B. Lewis, and L. Reichel. On the choice of subspace for iterative methods for linear discrete ill-posed problems. *Int. J. Appl. Math. Comput. Sci.*, 11(5):1069–1092, 2001.

[17] D. Calvetti, B. Lewis, and L. Reichel. GMRES, L-curves, and discrete ill-posed problems. *BIT*, 42:44–65, 2002.

[18] D. Calvetti, B. Lewis, and L. Reichel. On the regularizing properties of the GMRES method. *Numer. Math.*, 91:605–625, 2002.

[19] D. Calvetti, S. Morigi, L. Reichel, and F. Sgallari. Tikhonov regularization and the L-curve for large discrete ill-posed problems. *J. Comput. Appl. Math.*, 123:423–446, 2000.

[20] D. Calvetti and L. Reichel. Tikhonov regularization of large linear problems. *BIT*, 43(2):263–283, 2003.

[21] D. Calvetti, L. Reichel, and A. Shuibi. Enriched Krylov subspace methods for ill-posed problems. *Linear Algebra Appl.*, 362:257–273, 2003.

[22] D. Calvetti, L. Reichel, and A. Shuibi. Invertible smoothing preconditioners for linear discrete ill-posed problems. *Appl. Numer. Math.*, 54:135–149, 2005.

[23] R. H. Chan, J. G. Nagy, and R. J. Plemmons. FFT-based preconditioners for Toeplitz-block least squares. *SIAM J. Numer. Anal.*, 30:1740–1768, 1993.

[24] T. F. Chan and J. Shen. *Image Processing and Analysis: Variational, PDE, Wavelet, and Stochastic Methods*. SIAM, Philadelphia, PA, 2005.

[25] J. Chung, J. G. Nagy, and D. P. O'Leary. A weighted GCV method for Lanczos hybrid regularization. *Elec. Trans. Numer. Anal.*, 28:149–167, 2008.

[26] M. A. Davenport, M. F. Duarte, Y. C. Eldar, and G. Kutyniok. Introduction to Compressed Sensing. In Y. C. Eldar and G. Kutyniok, editors, *Compressed Sensing: Theory and Applications*. Cambridge University Press, 2012.

[27] M. Donatelli, C. Estatico, A. Martinelli, and S. Serra-Capizzano. Improved image deblurring with anti-reflective boundary conditions and re-blurring. *Inverse Problems*, 22:2035–2053, 2006.

[28] M. Donatelli, A. Neuman, and L. Reichel. Square regularization matrices for large linear discrete ill-posed problems. *Numer. Linear Algebra Appl.*, 19:896–913, 2012.

[29] M. Donatelli and L. Reichel. Square smoothing regularization matrices with accurate boundary conditions. *J. Comput. Appl. Math.* To appear.

[30] L. Dykes and L. Reichel. A family of range restricted iterative methods for linear discrete ill-posed problems. *Dolomites Research Notes on Approximation*, 6:27–36, 2013. Proceedings of DWCAA12.

[31] L. Eldén. Algorithms for the regularization of ill-conditioned least squares problems. *BIT*, 17:134–145, 1977.

[32] L. Eldén. A weighted pseudoinverse, generalized singular values, and constrained least squares problems. *BIT*, 22:487–501, 1982.

[33] Y. W. Fan and J. G. Nagy. Synthetic boundary conditions for image deblurring. *Linear Alg. Appl.*, 434:2244–2268, 2010.

[34] M. A. T. Figueiredo, R. D. Nowak, and S. J. Wright. Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems. *IEEE J. Selected Topics in Signal Processing*, 1(4):586–597, 2007.

[35] A. Frommer and P. Maass. Fast CG-based methods for Tikhonov-Phillips regularization. *SIAM J. Sci. Comput.*, 20:1831–1850, 1999.

[36] M. Fuhry and L. Reichel. A new Tikhonov regularization method. *Numer. Algorithms*, 59:433–445, 2012.

[37] S. Gazzola and J. G. Nagy. Generalized Arnoldi-Tikhonov method for sparse reconstruction. Accepted for pubblication in SIAM J. Sci. Comput., 2014.

[38] S. Gazzola and P. Novati. Automatic parameter setting for Arnoldi-Tikhonov methods. *J. Comput. Appl. Math.*, 256:180–195, 2014. DOI:10.1016/j.cam.2013.07.023.

[39] S. Gazzola and P. Novati. Multi-parameter Arnoldi-Tikhonov methods. Accepted for publication in Electron. Trans. Numer. Anal., 2014.

[40] S. Gazzola, P. Novati, and M. R. Russo. Embedded techniques for choosing the parameter in Tikhonov regularization. Under Review, 2013.

[41] G. H. Golub. Numerical methods for solving linear least squares problems. *Numer. Math.*, 7:206–216, 1965.

[42] G. H. Golub and C. F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, Baltimore, MD, third edition, 1996.

[43] G. H. Golub, F. T. Luk, and M. L. Overton. Block Lanczos Method for Computing the Singular Values of Corresponding Singular Vectors of a Matrix. *ACM Trans. Math. Software*, 7(2):149–169, 1981.

[44] G. H. Golub and G. Meurant. *Matrices, Moments and Quadrature with Applications*. Princeton Series in Applied Mathematics, NJ, 1994.

[45] J. Hadamard. *Lectures on Cauchy's problem in linear partial differential equations*. Yale Univeristy Press, New Heaven, CT, 1923.

[46] M. Hanke. *Conjugate Gradient Type Methods for Ill-Posed Problems*. Longman, Essex, UK, 1995.

[47] M. Hanke. On Lanczos based methods for the regularization of discrete ill-posed problems. *BIT*, 41(5):1008–1018, 2001.

[48] M. Hanke and P. C. Hansen. Regularization methods for large-scale problems. *Surv. Math. Ind.*, 3:253–315, 1993.

[49] M. Hanke and J. G. Nagy. Restoration of atmospherically blurred images by symmetric indefinite conjugate gradient techniques. *Inverse Problems*, 12:157–173, 1996.

[50] M. Hanke, J. G. Nagy, and R. J. Plemmons. Preconditioned iterative regularization for ill-posed problems. In L. Reichel, A. Ruttan, and R. S. Varga, editors, *Numerical Linear Algebra*, pages 141–163. De Gruyter, Berlin, 1993.

[51] P. C. Hansen. Computation of the singular value expansion. *Computing*, 40:185–199, 1988.

[52] P. C. Hansen. Regularization, GSVD and truncated GSVD. *BIT*, 29:491–504, 1989.

[53] P. C. Hansen. The discrete Picard condition for discrete ill-posed problems. *BIT*, 30:658–672, 1990.

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

[55] P. C. Hansen. *Rank-deficient and discrete ill-posed problems.* SIAM, Philadelphia, PA, 1998.

[56] P. C. Hansen. Regularization Tools version 3.0 for Matlab 5.2. *Numer. Algorithms*, 20:195–196, 1999.

[57] P. C. Hansen. Regularization Tools version 4.0 for Matlab 7.3. *Numer. Algorithms*, 46:189–194, 2007.

[58] P. C. Hansen. Regularization Tools: A Matlab package for analysis and solution of discrete ill-posed problems, 2008.
http://www2.imm.dtu.dk/~pcha/Regutools/RTv4manual.pdf.

[59] P. C. Hansen. *Discrete Inverse Problems: Insight and Algorithms.* SIAM, Philadelphia, PA, 2010.

[60] P. C. Hansen. Oblique projections and standard-form transformation for discrete inverse problems. *Numer. Linear Algebra Appl.*, 20:250–258, 2013.

[61] P. C. Hansen and T. K. Jensen. Smoothing-norm preconditioning for regularizing minimum-residual methods. *SIAM J. Matrix Anal. Appl.*, 29:1–14, 2006.

[62] P. C. Hansen, T. K. Jensen, and G. Rodriguez. An adaptive pruning algorithm for discrete L-curve criterion. *J. Comp. Appl. Math.*, 198:483–492, 2007.

[63] P. C. Hansen, M. Kilmer, and R. H. Kjeldsen. Exploiting residual information in the parameter choice for discrete ill-posed problems. *BIT*, 46:41–59, 2006.

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

[65] P. C. Hansen and D. P. O'Leary. The use of the L-curve in the regularization of discrete ill-posed problems. *SIAM J. Sci. Comput.*, 14:1487–1503, 1993.

[66] I. Hnetynkova, M. Plesinger, and Z. Strakos. The regularizing effect of the Golub-Kahan iterative bidiagonalization and revealing the noise level in the data. *BIT*, 49:669–696, 2009.

[67] M. Hochstenbach and L. Reichel. An iterative method for Tikhonov regularization with a general linear regularization operator. *J. Integral Equations Appl.*, 22:463–480, 2010.

[68] B. Hofmann. *Regularization for Applied Inverse and Ill-Posed Problems.* Teubner, Stuttgart, Germany, 1986.

[69] D. Hunter and K. Lange. A tutorial on MM algorithms. *The American Statistician*, 58:30–37, 2004.

[70] T. K. Jensen and P. C. Hansen. Iterative regularization with minimum-residual methods. *BIT*, 47:103–120, 2007.

[71] J. Kamm and J. G. Nagy. Kronecker product and SVD approximations in image restoration. *Linear Algebra Appl.*, 284:177–192, 1998.

[72] M. E. Kilmer, P. C. Hansen, and M. I. Español. A projection-based approach to general-form Tikhonov regularization. *SIAM J. Sci. Comput.*, 29(1):315–330, 2007.

[73] M. E. Kilmer and D.P. O'Leary. Choosing regularization parameters in iterative methods for ill-posed problems. *SIAM J. Matrix Anal. Appl.*, 22(4):1204–1221, 2001.

[74] S. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinvesky. An interior-point method for large-scale ℓ_1-regularized least squares. *IEEE J. Selected Topics in Image Processing*, 1:606–617, 2007.

[75] Y. Kim and C. Gu. Smoothing spline Gaussian regression: more scalable computation via efficient approximation. *J. Roy. Stat. Soc.*, 66:73–93, 2004.

[76] J. Lampe, L. Reichel, and H. Voss. Large-scale Tikhonov regularization via reduction by orthogonal projection. *Linear Algebra Appl.*, 436:2845–2865, 2012.

[77] C. Lanczos. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. *Journal of Research of the National Bureau of Standards*, 45:255–282, 1950.

[78] C. L. Lawson and R. J. Hanson. *Solving Least Squares Problems*. Prentice-Hall Inc., Englewood Cliffs, NJ, 1974.

[79] B. Lewis and L. Reichel. Arnoldi-Tikhonov regularization methods. *J. Comput. Appl. Math.*, 226:92–102, 2009.

[80] R. C. Li and Q. Ye. A Krylov subspace method for quadratic matrix polynomials with application to constrained least squares problems. *SIAM J. Matrix Anal. Appl.*, 25:405–428, 2003.

[81] Y. Lin, B. Wohlberg, and H. Guo. UPRE method for total variation parameter selection. *Signal Processing*, 90:2546–2551, 2010.

[82] S. Lu and S. V. Pereverzer. Multi-parameter regularization and its numerical realization. *Numer. Math.*, 118:1–31, 2001.

[83] C. D. Meyer. *Matrix Analysis and Applied Linear Algebra*. SIAM, Philadelphia, PA, 2000.

[84] I. Moret and P. Novati. RD-Rational Approximations of the Matrix Exponential. *BIT*, 44(3), 2004.

[85] S. Morigi, L. Reichel, and F. Sgallari. Orthogonal projection regularization operators. *Numer. Algorithms*, 44:99–114, 2007.

[86] V. A. Morozov. On the solution of functional equations by the method of regularization. *Soviet Math. Dokl.*, 7:414–417, 1966.

[87] J. G. Nagy, M. K. Ng, and L. Perrone. Kronecker product approximation for image restoration with reflexive boundary conditions. *SIAM J. Matrix Anal. Appl.*, 25:829–841, 2004.

[88] J. G. Nagy and D. P. O'Leary. Restoring images degraded by spatially-variant blur. *SIAM J. Sci. Comput.*, 19:1063–1082, 1998.

[89] J. G. Nagy and K. M. Palmer. Steepest descent, CG, and iterative regularization of ill-posed problems. *BIT*, 43(5):1003–1017, 2005.

[90] J. G. Nagy, K. M. Palmer, and L. Perrone. Iterative methods for image deblurring: a Matlab object oriented approach. *Numer. Algorithms*, 36:73–93, 2004.
See also: http://www.mathcs.emory.edu/~nagy/RestoreTools.

[91] J. G. Nagy, R. J. Plemmons, and T. C. Torgersen. Iterative image restoration using approximate inverse preconditioning. *IEEE Trans. on Image Processing*, 15:1151–1162, 1996.

[92] J. G. Nagy and Z. Strakoš. Enforcing nonnegativity in image reconstruction algorithms. In D. C. Wilson, et. al., editor, *Mathematical Modeling, Estimation, and Imaging*, volume 3461, pages 182–190. SPIE, 2000.

[93] O. Nevanlinna. *Convergence of Iterations for Linear Equations*. Birkhäuser, Basel, CH, 1993.

[94] P. Novati and M. R. Russo. Adaptive Arnoldi-Tikhonov regularization for image restoration. *Numer. Algorithms*, 2013. DOI:10.1007/s11075-013-9712-0.

[95] P. Novati and M. R. Russo. A GCV based Arnoldi-Tikhonov regularization method, 2013. Accepted for pubblication in BIT.

[96] D. P. O'Leary and J. A. Simmons. A bidiagonalization-regularization procedure for large scale discretizations of ill-posed problems. *SIAM J. Sci. Stat. Comp.*, 2(4):474–489, 1981.

[97] J. P. Oliveira, J. M. Bioucas-Dias, and M. A. T. Figueiredo. Adaptive total variation image deblurring: a majorization-minimization approach. *Signal Processing*, 89:1683–1693, 2009.

[98] C. C. Paige and M. A. Saunders. Algorithm 583 LSQR: sparse linear equations and sparse least squares. *ACM Trans. Math. Software*, 8:195–209, 1982.

[99] C. C. Paige and M. A. Saunders. LSQR: an algorithm for sparse linear equations and and sparse least squares. *ACM Trans. Math. Software*, 8:43–71, 1982.

[100] D. L. Phillips. A technique for the numerical solution of certain integral equations of the first kind. *J. Assoc. Comput. Mach.*, 9:84–97, 1962.

[101] A. Quarteroni, R. Sacco, and F. Saleri. *Numerical Mathematics*. Springer-Verlag, New York, NY, 2000.

[102] L. Reichel and G. Rodriguez. Old and new parameter choice rules for discrete ill-posed problems. *Numer. Algor.*, 63:65–87, 2013.

[103] L. Reichel and H. Sadok. A new L-curve for ill-posed problems. *J. Comput. Appl. Math.*, 219:493–508, 2008.

[104] L. Reichel, F. Sgallari, and Q. Ye. Tikhonov regularization based on generalized Krylov subspace methods. *Appl. Numer. Math.*, 62:1215–1228, 2012.

[105] L. Reichel and A. Shyshkov. A new zero-finder for Tikhonov regularization. *BIT*, 48:627–643, 2008.

[106] L. Reichel and Q. Ye. Simple square smoothing regularization operators. *Electron. Trans. Numer. Anal.*, 33:63–83, 2009.

[107] J. D. Riley. Solving systems of linear equations with a positive definite, symmetric, but possibly ill-conditioned matrix. *Math. Tables Aids Comput.*, 9:96–101, 1955.

[108] P. Rodríguez and B. Wohlberg. An Iteratively Reweighted Norm algorithm for Total Variation regularization. In *Proceedings of the 40th Asilomar Conference on Signals, Systems and Computers (ACSSC)*, 2006.

[109] P. Rodríguez and B. Wohlberg. An efficient algorithm for sparse representations with ℓ^p data fidelity term. In *Proceedings of 4th IEEE Andean Technical Conference (ANDESCON)*, 2008.

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

[111] B. W. Rust. Truncating the Singular Values Decomposition for ill-posed problems. Report NISTIR 6131, Mathematical and Computational Sciences Division, NIST, 1998.

[112] Y. Saad. On the rates of convergence of the Lanczos and block-Lanczos methods. *SIAM J. Numer. Anal.*, 17(5):687–706, 1980.

[113] Y. Saad. *Numerical methods for large eigenvalue problems. Algorithms and Architectures for Advanced Scientific Computing.* Halsted Press, New York, NY, 1992.

[114] Y. Saad. A flexible inner-outer preconditioned GMRES algorithm. *SIAM J. Sci. Comput.*, 14(2):461–469, 1993.

[115] Y. Saad. *Iterative Methods for Sparse Linear Systems, 2nd Ed.* SIAM, Philadelphia, PA, 2003.

[116] Y. Saad and M. H. Schultz. GMRES: a generalized minimal residual method for solving nonsymmetric linear systems. *SIAM J. Sci. Comput.*, 7:856–869, 1986.

[117] V. Simoncini and D. B. Szyld. Flexible inner-outer Krylov subspace methods. *SIAM J. Numer. Anal.*, 40(6):2219–2239, 2006.

[118] V. Simoncini and D. B. Szyld. Recent computational developments in Krylov subspace methods for linear systems. *Numer. Linear Algebra Appl.*, 14:1–59, 2007.

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

[120] A. N. Tikhonov and V. Y. Arsenin. *Solutions of ill-posed problems.* Wiley, New York, NY, 1977.

[121] S. Twomey. On the numerical solution of Fredholm integral equations of the first kind by inversion of the linear system produced by quadrature. *J. Assoc. Comput. Mach.*, 10:97–101, 1963.

[122] C. F. Van Loan. Generalizing the singular value decomposition. *SIAM J. Numer. Anal.*, 13:76–83, 1976.

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

[124] B. Wohlberg and P. Rodríguez. An Iteratively Reweighted Norm algorithm for minimization of Total Variation functionals. *IEEE Signal Processing Letters*, 14:948–951, 2007.

[125] S. J. Wright, M. A. T. Figueiredo, and R. D. Nowak. Sparse Reconstruction by Separable Approximation. *IEEE Trans. Signal Processing*, 57(7):2479–2493, 2009.