FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University

MASTER THESIS

Andrea Kučerová

Iterative methods for Tichonov regularization with generalized regularization terms

Department of Numerical Mathematics

Supervisor of the master thesis: doc. RNDr. Iveta Hnětynková, Ph.D.
Study programme: Mathematics
Study branch: Computational Mathematics

Prague 2022

I declare that I carried out this master thesis independently, and only with the cited sources, literature and other professional sources. It has not been used to obtain another or the same degree.

I understand that my work relates to the rights and obligations under the Act No. 121/2000 Sb., the Copyright Act, as amended, in particular the fact that the Charles University has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 subsection 1 of the Copyright Act.

In .... date ....

Author's signature

I am extremely grateful to my supervisor doc. RNDr. Iveta Hnětynková, Ph.D. for her kindness and support. Without her patient guidance I could never have completed this thesis. Also I would like to thank my dear ones Miška, Lukáš and Karča for helping me see joy in days full of work.

Title: Iterative methods for Tichonov regularization with generalized regularization terms

Author: Andrea Kučerová

Department: Department of Numerical Mathematics

Supervisor: doc. RNDr. Iveta Hnětynková, Ph.D., Department of Numerical Mathematics

Abstract: The aim of this thesis is to study hybrid methods for solving ill-posed linear inverse problems corrupted by white noise. These approaches are based on the combination of iterative Krylov subspace methods and the Tichonov regularization with a general regularization term. We explain the basic properties of ill-posed problems, the idea of regularization, the role of the regularization term to enforce desirable properties to the solution and the theoretical background of Standard and General Tichonov minimization. Then we explain shift invariance of Krylov subspaces. This allows us to describe a hybrid approach where the full size problem is first projected onto a Krylov subspace of a smaller dimension and then the Tichonov minimization is applied to the small projected problem. We focus on the regularization based on the finite difference approximation of derivatives of the solution. The well known regularization terms constructed from forward differences for the first and the second derivative are summarized, then we use the Taylor expansion to construct finite differences of higher orders of precision. We incorporate different variants of boundary conditions. Then the impact of the order of precision of the finite difference schemes on the quality of the solution is studied. In the experiments we use the hybrid method combining the LSQR with the General Tichonov regularization.

Keywords: ill-posed problems, regularization, Krylov subspace methods, hybrid methods, generalized norms

Contents

Notation

$\mathbb{N}$	the set of natural numbers
$\mathbb{R}$	the set of real numbers
$\mathbb{R}^n$	the set of real n-dimensional vectors
$\mathbb{R}^{m\times n}$	the set of real matrices of the dimension $m \times n$
$\mathcal{C}^k$	the class of functions with k continuous derivatives on $\mathbb{R}$
μ_e	the noise level in b
x_{naive}	the naive solution of an inverse problem
λ	the regularization parameter
L	the regularization matrix
$dim(V)$	the dimension of a vector space V
$span\{v_1, \dots, v_n\}$	the linear span of vectors $v_1, \dots, v_n$
$\mathcal{K}_k(A, b)$	the k-th Krylov subspace for a matrix A and a vector b
$rank(A)$	the rank of a matrix A
$Im(A)$	the image of a matrix A, so-called column space
$Ker(A)$	the kernel of a matrix A, so-called null space
$det(A)$	the determinant of a matrix A
$trace(A)$	the trace of a matrix A
$sp(A)$	the spectrum of a matrix A
$\sigma_1, \dots, \sigma_r$	the singular values of a matrix with the rank r
A^T	the transpose of matrix A
A^{-1}	the inverse of a matrix A
I	the identity matrix
$diag(a_1, \dots, a_n)$	the diagonal matrix with diagonal elements $a_1, \dots, a_n$
$vec(A)$	the vectorization of a matrix A
$\|v\|$	the euclidean norm of a vector v
$\|v\|_1$	the 1-norm of a vector v
$\|v\|_p$	the p-norm of a vector v
$\|v\|_A$	the vector A-norm for A symmetric positive definite
$\|A\|_F$	the Frobenius norm of a matrix A
$A \otimes B$	the Cronecker product of matrices A and B
f', f'', f'''	the first, second and third derivative of f, respectively
α	the approximate noise level for the Discrepancy principle
ξ	the safety factor for the Discrepancy principle

Introduction

In this thesis we deal with methods for solving linear inverse problems, which arise naturally in many fields as geoscience, medicine, engineering, astronomy or image processing. We focus on problems in the form

$$Ax \approx b, A \in \mathbb{R}^{m \times n}, x \in \mathbb{R}^n, b \in \mathbb{R}^m,$$

where the task is to arrive at a reconstruction of the unknown data x from the measurement b corrupted by *noise*. We restrict ourselves with the assumption, that the noise is only included in the right hand side b and that the matrix A is known precisely. The noise is assumed to be white, since white noise makes a good model for random errors. There have been many papers and books written on the difficulties of solving inverse problems, see for example [1, Chapter 2 and Chapter 4], [2, Chapter 1], [3], [4, Chapter 2], [5, Chapter 1], [6], [7], [8]. The main factors that play a role here are the fact that the matrix A often has ill-determined rank, the problem is sensitive to (especially high-frequency) perturbations present in the vector b. Linear inverse problems described above are often called *ill-posed.* It is well known that they cannot be solved by classical least squares techniques, since such solutions are dominated by amplified noise.

One of the well known methods for solving inverse problems is the Tichonov regularization (see [6], [7], [1, Chapter 4.4]) that incorporates the so-called *regularization term* to enforce some desirable properties (such as smoothness) to the solution. But when using the Tichonov method, which minimizes a linear combination of the residual norm and the regularization term, we encounter a difficulty. Usually the matrix A is sparse and large and sometimes it is not available explicitly, but only as a function handle that allows matrix-vector multiplication. Therefore it is often beyond the computational possibilities to apply the Tichonov regularization to the full size problem. However, according to the result published by Per Christian Hansen in [1, Chapter 6.4] based on the shift invariance of Krylov subspaces, it is possible to first project the full size problem onto a Krylov subspace of a smaller dimension and then apply the Tichonov minimization to the small projected problem. The constructed iterations will converge to the solution of the original Tichonov minimization problem, see [1]. In this setting we do not need the matrix A explicitly since the projection onto a Krylov subspace only requires evaluation of matrix-vector multiplications with A and possibly A^T. We arrive at a procedure that incorporates dual regularization - the regularization by projection onto a Krylov subspace and Tichonov regularization. Such combination is usually called a *hybrid approach* [5, Chapter 3], [1, Chapter 6.4].

In the thesis we describe properties of inverse problems and the difficulties that occur when solving them. We select particular iterative Krylov subspace methods (see [5, Chapter 2.2], [1, Chapter 6.3]) and combine them with the Standard Tichonov minimization. We proof the interchangeability of the two approaches *first regularize, then project* and *first project, then regularize* and we describe a general way of constructing a hybrid method. *Regularization parameters*, which control the strength of the regularization, will be selected by the *Discrepancy principle* [1, Chapter 5.2], [2, Chapter 7.2].

We further describe the Tichonov regularization with a general regularization term and we mention two known ways of its transformation to the Standard

Tichonov formulation, see [9] and [1, Chapter 8.4]. It is well known that regularization terms can be constructed as a penalization by the finite difference for approximating derivatives. Well studied are especially the variants using forward difference for the first and sometimes the second derivative, see [1, Chapter 8]. We describe these terms and then we focus on the use of the Taylor expansion to construct regularization terms for approximating derivatives for 1D and 2D examples with higher orders of precision than the ones used traditionally. We also incorporate different variants of boundary conditions. Then we study whether the regularization terms that approximate derivatives with higher orders of precision can offer better approximations of the solution. We perform experiments to test the studied methods with different regularization terms on 1D and 2D examples with various levels of noise.

1. Inverse problems and regularization

1.1 Inverse problem

Consider inverse problem in the form

$$Ax \approx b, A \in \mathbb{R}^{m\times n}, x \in \mathbb{R}^n, b \in \mathbb{R}^m, \tag{1.1}$$

where A and b are given nontrivial data and x is an unknown vector. For simplicity we concentrate on the problem with noise included only in the right hand side vector b, and we therefore assume that A is given exactly.

There are many ways how noise can get into the problem, for example, errors can arise during physical measurements, discretization process, limited storage space, rounding errors and so on. For our simulations we consider only additive white noise. We will therefore denote

$$b = \tilde{b} + e, \tag{1.2}$$

where e, sometimes referred to as a *perturbation* or *error*, stands for the noise and $\tilde{b}$ is the unknown precise part of the right hand side b satisfying

$$A\tilde{x} = \tilde{b}, \tag{1.3}$$

where $\tilde{x}$ stands for the precise solution of the problem without noise. By saying that the noise is white we mean that all the components of the vector e come from the same normal distribution with zero mean. The reason for this choice is that white noise makes a good model for random errors.

Other way to describe white noise is to say, that all the frequencies occur with the same probability. It can be seen in Figure 1.1, where we compare the cosine Fourier coefficients of random white noise vector and a typical observation b coming from an inverse problem, both scaled to one. We can see that all the coefficients for the white noise vector are at similar level, unlike the coefficients for the vector b, which tend to be large at the left side of the spectrum, where the coefficients represent very low frequencies, and small or zero on the rest of the plot, where the coefficients correspond to higher frequencies.

There exist many other types of noise, such as signal-correlated noise and Poisson noise. Some of them are studied and explained in [1, Chapter 3.6].

For further needs we present the following definition of *the noise level in b.*

Definition 1 (The noise level in b)**.** *Given the right hand side vector b of an inverse problem* (1.1) *satisfying the equality* (1.2)*, we define*

$$\mu_e = \frac{\|e\|}{\|b\|}$$

the noise level in b.

Our goal when dealing with an inverse problem given by (1.1) is to arrive to some approximation x that is close enough to the vector $\tilde{x}$.

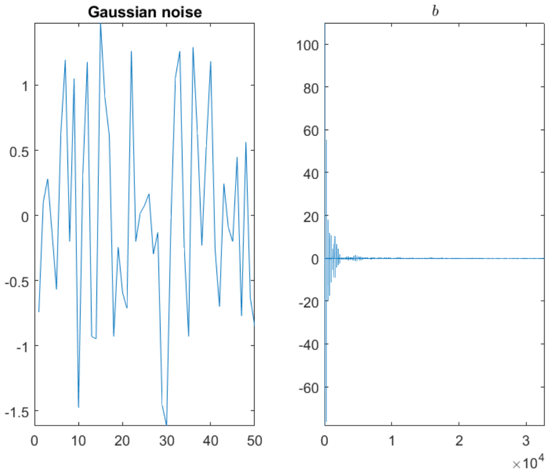

Figure 1.1: Comparison of cosine Fourier coefficients for two vectors scaled to one. Only the left half of the coefficients is shown, since the Fourier coefficients are symmetric. The random white noise vector coefficients are on the left, the coefficients of a typical observation b coming from an inverse problem are on the right. It can be seen that the sizes of the coefficients for white noise are comparable. On the contrary, in b we can see the large coefficients only in low frequencies. In general, frequency distributions in x and b are strongly problem dependent.

1.2 Introducing examples

For the purpose of demonstration of the results and ideas in this thesis we use the functions and scripts involved in the Matlab packages [10] (2D examples and methods) and [11] (1D examples). We introduce two 2D problems that are used in this thesis. I chose examples that represent image deblurring problems, that are generally more illustrative compared to other types of problems.

Before introducing concrete examples we will describe how we can transform an image deblurring problem into a problem in the form (1.1), more information regarding this topic can be found in [12, Chapter 1]. We will only consider images in the shades of grey. An image deblurring problem aims to reconstruct an image $X \in \mathbb{R}^{p\times q}$ from blured and noisy observation $B \in \mathbb{R}^{p\times q}$, where the values in the matrices represent the shape of grey of the corresponding pixel. In the setting of an image deblurring problem we also assume to know the blurring process, which can be for example represented by a point spread function, that determines the way how the information is spread from one pixel to its neighbourhood. In order to introduce an inverse problem in the form (1.1), we need to transform both the matrices X and B into vectors. This can be done by the so-called *vectorization of a matrix.*

Definition 2 (The vectorization of a matrix)**.** *Let us have a matrix*

$$B = (b_1, \dots, b_q) \in \mathbb{R}^{p\times q}.$$

Then we define the vectorization of a matrix B *as*

$$vec(B) = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_q \end{pmatrix} \in \mathbb{R}^{pq}.$$

We now denote $x = vec(X) \in \mathbb{R}^{pq}$ and $b = vec(B) \in \mathbb{R}^{pq}$. Note that we further identify the vector and matrix versions of the original image X and the observation B and we write only the vector versions x and b even when talking about the images.

It remains to introduce a matrix A in this problem. The matrix represents a linear model of the bluring process and it is constructed from the so-called *point spread function* that discribes how one pixel affects its neighbouhood in such a way, that it suits the vectorized nature of the picture. Therefore we can write the image deblurring problem as an inverse problem of the type (1.1).

In both test problems we need to add noise to the right hand side. To do so, we use the predefined function *PRnoise* from the toolbox [10]. We call the function with parameters v, which is any vector, and $\mu > 0$. It generates a vector e of random numbers from the same normal distribution with mean at zero computed using the Matlab function *randn*, so that it satisfies the condition

$$\|e\| = \mu * \|v\|.$$

Parameter μ is the desired level of noise. In the case of calling *PRnoise* without the second parameter, the μ is set to 0.01. The function then returns new vector computed as a sum of v and e. We can see that if we use this function to a precise right hand side $\tilde{b}$, we obtain new right hand side that has the predefined noise level.

Example 1 (Construction of the first 2D example problem in Matlab)*.* The first example is an image deblurring problem generated by a function *PRblur* from the Matlab toolbox [10]. In the following frame we show the related code used for generating the problem.

```
function [A, b_tilde, x, ProbInfo, b, NoiseInfo] = DT_PR1
p = 256;
optblur.trueImage = 'satellite';
optblur.PSF = 'shake';
optblur.BlurLevel = 'severe';
[A, b_tilde, x, ProbInfo] = PRblur(p, optblur);
[b,NoiseInfo] = PRnoise(b_tilde);
end
```

Here the vector x represents the vectorized version of a satellite test image which can be seen in Figure 1.2. The original image is squared of the size $p \times p$. The matrix $A \in \mathbb{R}^{p^2 \times p^2}$ is derived from a random, shaking motion point spread function. The level of blur is set to 'severe'. The vector *b_tilde* is a vector corresponding to $\tilde{b}$ in (1.3). The vector b is the right hand side of the problem, containing errors added using the function *PRnoise*, with noise added automatically at the level 0.01. We can see comparison of the true solution, precise,

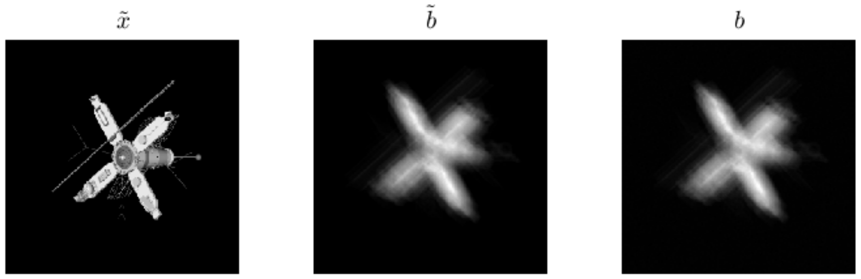

Figure 1.2: The first example problem from Example 1. The precise image $\tilde{x}$ on the left, precise right hand side $\tilde{b}$ in the middle and the unprecise right hand side b on the right. The images $\tilde{b}$ and b are both smooth compared to the original vector $\tilde{x}$.

but shaked right hand side and unprecise right hand side in Figure 1.2. The observations in real problems tend to be smooth, as we can see for b in the figure. This same fact can be also observed in Figure 1.1, which shows that b is dominated by low frequencies. More information regarding this test problem can be found directly in the comments of the source code of the function *PRblur* in [10].

Example 2 (Construction of the second 2D example problem in Matlab). Second example is of the same type as the first one, obtained using the same function *PRblur* from toolbox [10]. We get all the problem components using the function defined by the Matlab code below.

```
function [A, b_tilde, x, ProbInfo, b, NoiseInfo] = DT_PR2
p = 256;
optblur.trueImage = 'pattern1';
optblur.PSF = 'gauss';
optblur.BlurLevel = 'severe';
[A, b_tilde, x, ProbInfo] = PRblur(p, optblur);
noise_level = 1e-6;
[b,NoiseInfo] = PRnoise(b_tilde, noise_level);
end
```

The vector x is a vector representation of a simple image that is shown in Figure 1.3. The size of the image is $p \times p$. On the same figure we can see also the precise right hand side $\tilde{b}$ and noisy right hand side b. The matrix A in this case comes from a gaussian point spread function. The level of blur is set to 'severe'. The noise level (10^{-6}) is much lower than in the previous case. More information about this test problem can be also found in the comments of the source code of the function *PRblur* in [10].

1.3 Sensitivity of the solution

It is well known that many inverse problems have certain properties that make them difficult to be solved by classical methods without regularization, see [1], [4]. E.g., they are usually sensitive to small perturbations in data, the matrix

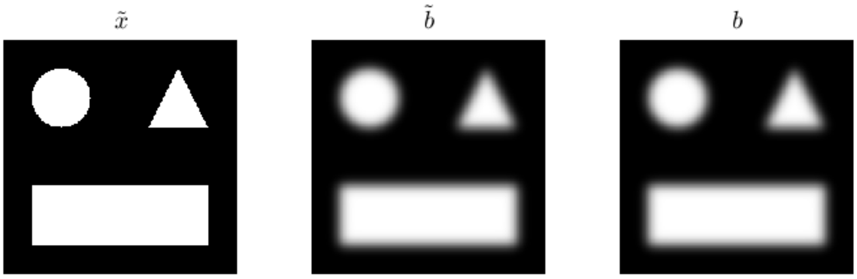

Figure 1.3: The second example problem from Example 2. The precise image $\tilde{x}$ on the left, precise right hand side $\tilde{b}$ in the middle and the unprecise right hand side b on the right. The images $\tilde{b}$ and b are both smooth compared to the original vector $\tilde{x}$.

A is ill-conditioned and has ill-determined rank, which means that its singular values decay without any significant gap.

Such properties complicate the process of finding a solution to an inverse problem. In this chapter we explain why, and to do so, we first need to take closer look on the problem itself and present few definitions and results.

One important result is the Singular Value Decomposition Theorem. Proof, detailed derivation and other details can be found in [13] and [14, Chapter 2.5].

Theorem 1 (The Singular Value Decomposition Theorem)**.** *Assume that $A \in \mathbb{R}^{m\times n}$, where $m, n \in \mathbb{N}$, is a matrix with the rank r. Then there exists a final sequence of numbers $\sigma_1, \dots, \sigma_r$ and orthogonal matrices $U = (u_1, \dots, u_m) \in \mathbb{R}^{m\times m}$ and $V = (v_1, \dots, v_n) \in \mathbb{R}^{n\times n}$ satisfying*

$$A = U\Sigma V^T = \sum_{i=1}^{r} \sigma_i u_i v_i^T, \tag{1.4}$$

where

$$\Sigma = \begin{pmatrix} \Sigma_r & 0 \\ 0 & 0 \end{pmatrix} \in \mathbb{R}^{m\times n}, \Sigma_r = diag(\sigma_1, \dots, \sigma_r) \in \mathbb{R}^{r\times r}.$$

Without loss of generality we can assume that the numbers $\sigma_1, \dots, \sigma_r$ are sorted in the descending order. Then the condition $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$ holds. We call the columns of U *left singular vectors of A*, the columns of V *right singular vectors of A* and the numbers $\sigma_1, \dots, \sigma_r$ *singular values of A*. The relation (1.4) is well known as the *singular value decomposition of A*.

Assuming that the matrix A in (1.1) is square and nonsingular, there is a simple way to express the solution to the problem using its inverse as

$$x_{naive} = A^{-1}b.$$

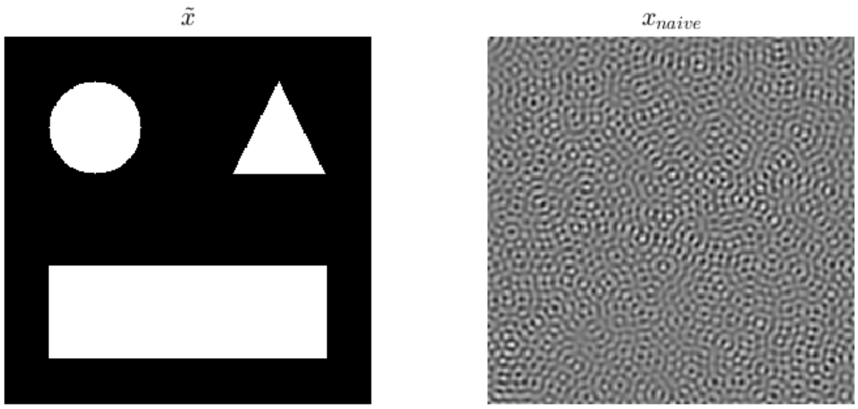

Figure 1.4: The comparison of the exact solution $\tilde{x}$ and the *naive* solution x_{naive} for the problem from Example 2. The inverted noise in the naive solution completely destroys the result. The euclidean norms for both vectors are 133.9104 and $6.8516 * 10^3$, respectively.

Here we assume that the inverse of A is available even though this assumption is usually not satisfied in real problems, either because the inverting process would be too expensive or because we do not have access to the whole matrix and its components.

In Figure 1.4 we can see the result of the computation of x_{naive} for the example from Examle 2 compared to the exact solution $\tilde{x}$. It is obvious that the result is not satisfactory and does not offer a good approximation to the exact solution x. Moreover, the comparison of the norms of the true solution and x_{naive}, as we can see in the same figure, is alarming. The norm of the naive solution is of different order than the norm of the exact solution.

Having the singular value decomposition defined, we can use it to rewrite the formula for x_{naive} and explain the reasons for such a bad results. Inverting the expression (1.4) and applying the inverse to b we get

$$x_{naive} = A^{-1}b = (U\Sigma V^T)^{-1}b = V\Sigma^{-1}U^Tb = \sum_{i=1}^{n} \frac{u_i^T b}{\sigma_i} v_i. \tag{1.5}$$

Using the relation (1.2) we can split the above expression, and therefore the vector x_{naive} itself, into two parts, the true solution x and a vector of *inverted noise*. We have

$$x_{naive} = \sum_{i=1}^{n} \frac{u_i^T \widetilde{b}}{\sigma_i} v_i + \sum_{i=1}^{n} \frac{u_i^T e}{\sigma_i} v_i. \tag{1.6}$$

In this expression for x_{naive}, we can see, that the first sum on the right hand side is the exact solution $\tilde{x}$, that we seek to find. The second part of the expression is the inverse of A applied to the vector e. We will now explain, why we observe such amplification of noise e in this process.

Let us have a closer look at the properties of the matrix A and especially on its singular value decomposition $U\Sigma V^T$. In Figure 1.5 we can see the first ten left singular vectors of the matrix A in our example. As the index gets higher, the

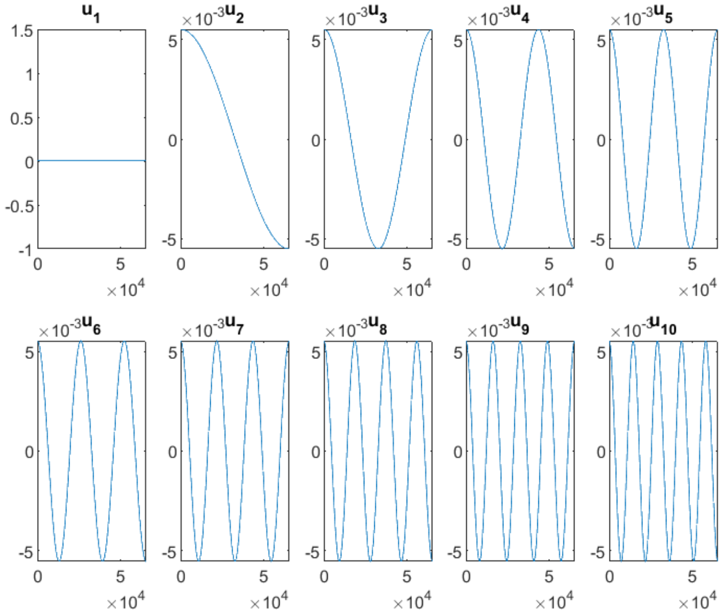

Figure 1.5: The first ten left singular vectors $u_1, \dots, u_{10}$ of the matrix A from Example 2. First vectors are very smooth. As we go further, the vectors tend to change the sign more often.

vectors tend to change the sign more often, therefore we call them high frequency vectors.

We can see in Figures 1.2 and 1.3, that the right hand sides $\tilde{b}$ of the inverse problems are smoother than the original vectors $\tilde{x}$. This phenomena is really common in real world problems, it is caused by the so called *smoothing property* of the matrix A. The explanation follows from the *Riemann-Lebesgue lemma* that states the same phenomena for kernel in the continuous version of our problem, the *integral equation*, see [1, Chapter 2.2]. In the process of multiplying a vector by the matrix A, the high frequencies in the vector are dumped, unlike the small frequencies.

Looking at the equation (1.6), this implies, that the coefficients $u_i^T\tilde{b}$ will become smaller in size with higher indices. On the contrary, the coefficients $u_i^T e$ do not satisfy the same condition, the vector e is random noise and its projections to the left singular vectors are likely to be similarly large throughout the whole spectrum. The problem comes, when we divide these two sets of coefficients with singular values σ_i while computing the naive solution x_{naive}.

As have been said, the projections of the exact right hand side $\tilde{b}$ are likely to be large only for the first left singular vectors and then they decay. Usually the decay of the coefficients is so fast that the sizes of projections with the corresponding indices are smaller than the singular values, as we can see in the figure. We say, that the exact right hand side $\tilde{b}$ satisfies *the Discrete Picard condition*. It follows, that the coefficients ${}^{u_i^T\tilde{b}}/_{\sigma_i}$ will decay as the index i rises.

Because of the nature of e, the projections $u_i^T e$ tend to be at the similar level throughout the whole spectrum of left singular values, e.g. the noise level in b. It is therefore obvious that, starting from some index, the singular values are smaller than the corresponding projections. The vector e, neither b satisfy the

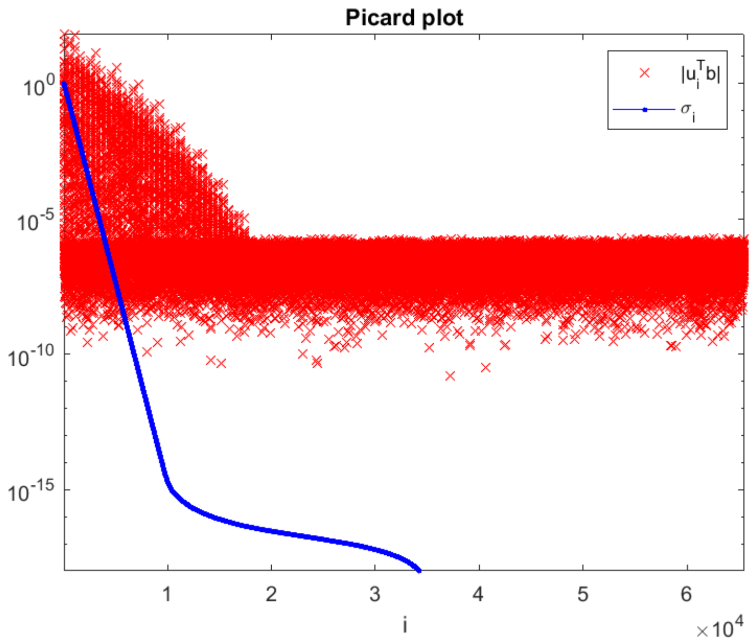

Figure 1.6: The Picard plot for the test problem from Example 2. The blue line represents the singular values σ_i, that decay quickly and without any significant gap to the machine precision. The numerical rank of the matrix A is not well defined. The red crosses represent sizes of the projections of the right hand side b to the left singular vectors. From the index approximately 10 000 onwards the projections are all larger than the corresponding singular values.

Discrete Picard condition. Therefore the sizes of coefficients ${}^{u_i^T e}/_{\sigma_i}$ will rise as the index i rises.

The result of these facts is, that as the index i grows, the coefficient ${}^{u_i^T b}/_{\sigma_i}$ becomes more and more dominated by the part ${}^{u_i^T e}/_{\sigma_i}$. The members of the sum defining x_{naive} are for higher indices dominated by the error part of the sum. We can see this behaviour in Figure 1.6.

Even small perturbation e added to the right hand side makes such a big difference in the naive solution, that the result is completely useless. The fact that small changes in data make a huge impact on the solution (so called *sensitivity*) is typical for a huge class of problems called *ill-posed problems*. Note that if A is rectangular, analogous derivations can be performed for

$$x_{naive} = \sum_{i=1}^{r} \frac{u_i^T b}{\sigma_i} v_i,$$

where $r = rank(A)$, i.e. for the well-known least squares solution to (1.1).

1.4 Methods dealing with ill-posedness

It is obvious that one has to incorporate knowledge about the properties of the particular problem in order to come up with a suitable approximate solution.

Different types of techniques and methods for dealing with ill-posed problems are covered by the term *regularization*. The goal of regularization is to overcome the sensitivity to small perturbations in data. Fore more general information, see [1, Chapter 4].

The basic idea of regularization is to replace the sensitive problem (1.1) by a different problem, that is more stable. Regularization methods can be divided into two groups: *variational* and *iterative*, see [5, Chapter 1.2].

- *Variational methods* modify the problem (1.1) into the following setting:

  $$\min_{x \in C} \{\mathcal{J}(Ax - b) + \lambda^2 \mathcal{R}(x)\},$$

  where C stands for a space in which we search for the solution, $\mathcal{J}$ is *a loss function*, and $\mathcal{R}$ is *a regularization operator*, which incorporates some constraints, which we want to put on the solution, λ is a parameter that sets the strength of the regularization.

  Well known example of a variational method is the Tichonov regularization, which is in the standard form formulated with $C = \mathbb{R}^n$, $\mathcal{J}(Ax - b) = \|Ax - b\|^2$ and $\mathcal{R}(x) = \|x\|^2$. As we saw in the previous section, in some cases it makes sense to restrict the norm of an approximate solution, since the naive solution tends to have a huge norm. Therefore we will focus on this particular method in the next chapter.

- *Iterative methods* consist of applying some iterative solver to a problem

  $$\min_{x \in \mathcal{C}} \mathcal{J}(Ax - b),$$

  where we take the number of computed iterations as a regularization parameter.

  One class within these methods is based on the singular value decomposition (1.4). One for all we mention the Truncated SVD (see [1, Chapter 4.2], [12, Chapter 6.1] and [2, Chapter 3.2]), which solves the problem of inverted noise by omitting the parts of the sum (1.5) defining the naive solution, that are dominated by inverted noise.

  Other huge class of iterative regularization methods are *Krylov subspace methods*, see [5, Chapter 2.2] and [1, Chapter 6.3], which incorporate the idea of searching the solution in a smooth subspace with smaller dimension, taking the growing dimension of a Krylov subspace as a regularization parameter. The Krylov space also carries some additional information about the problem itself, which makes the method usually even more effective.

- Special group of methods are *hybrid methods*, see [5] and [2, Chapter 6.6], that combine both approaches mentioned above. The methods usually introduce more than one regularization parameter, which makes it more difficult when trying to find optimal combination of the parameters. On the other hand, well built hybrid method brings the advantages of both variational and iterational approache, leading to better solution for some problems. The hybrid methods, namely methods combining Tichonov regularization and different iterative approaches, are the subject of the study in this thesis, therefore we will be dealing with hybrid methods in future chapters.

2. Tichonov regularization

As we could see in the end of the previous chapter, when solving the so called ill-posed problems, it is desirable, if not necessary, to incorporate some techniques to overcome the sensitivity. We have also noticed, that the norm of the naive solution for inverse problems tends to be large, therefore it might be reasonable to introduce a method, that penalizes large norms in the approximate solution. In the next subsection we concentrate on the formulation of the Standard Tichonov regularization method, which offers an alternative to the naive solution.

2.1 Standard Tichonov method

Standard Tichonov regularization (see [6], [7] and [1, Chapter 4.4]) is a variational method formulated as follows:

$$\min_{x \in \mathbb{R}^n} \{\|Ax - b\|^2 + \lambda^2 \|x\|^2\}, \tag{2.1}$$

where the first term is the resuidual norm, which, if standing alone, defines the least squares minimization, and the second term represents the penalization for the solution norm, while the parameter $\lambda > 0$ sets the balance between these two terms.

We get the original least squares formulation from the setting (2.1) by putting $\lambda = 0$. It is obvious that, as λ rises, the emphasis shifts to the penalization term $\|x\|^2$, we get smoother solution, that fits the data less. In other words λ plays the role of the regularization parameter. We can see the results for different values of λ in Figure 2.1. If the parameter is too small, we get a result close to the naive solution, in other words, the result is *undersmoothed.* On the contrary, when putting λ too large, the emphasis on the penalization term is too strong, it spoils the effect of the residual minimization, and the result is *oversmoothed.*

As we can see in the following table, λ determines the ratio between the residual norm and the norm of the solution, the last column corresponds to the naive solution.

Table 2.1: The residual norms $\|r_\lambda\|$ and the solution norms $\|x_\lambda\|$ for different values of λ for the problem from Example 2

λ	10	$1 * 10^{-2}$	$1 * 10^{-6}$	$1 * 10^{-8}$	0
$\|x_\lambda\|$	1.1616	131.5545	132.8067	612.7474	$6.8516 * 10^3$
$\|r_\lambda\|$	121.0206	0.0601	$1.2550 * 10^{-4}$	$1.1675 * 10^{-4}$	$1.2791 * 10^{-4}$

2.1.1 Analysis using SVD

Recall that the least squares problem in the form

$$\min_{x \in \mathbb{R}^n} \|Ax - b\|$$

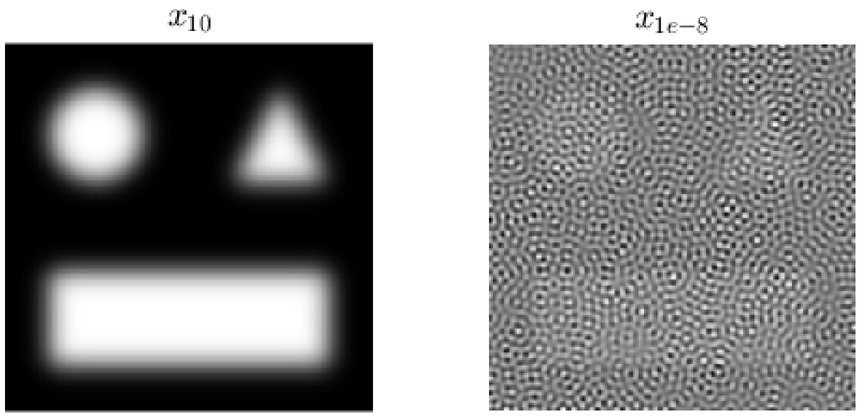

Figure 2.1: The solutions of the Standard Tichonov regularization (2.1) for two different values of λ for the problem from Example 2. On the left we can see the result for $\lambda = 10$. The resulting picture is oversmoothed, the regularization is too strong. On the right there is the result for $\lambda = 10^{-8}$. The regularization is weak, the solution is undersmoothed and very close to the naive solution.

is equivalent to solving the *normal equations* corresponding to (1.1),

$$A^T Ax = A^T b. \tag{2.2}$$

In this section we show that the Standard Tichonov formulation can be expressed as a least squares problem with a modified data and then we use the singular value decomposition of A and the corresponding normal equations to clarify the behaviour of this method. Note that this analysis is only suitable to express the general properties of the Standard Tichonov method in theory. By no means is the following equivalent expression appropriate to use for real computations, because it would be highly ineffective.

Let us reformulate the criterion function in (2.1) as follows:

$$\begin{aligned} \|Ax - b\|^2 + \lambda^2\|x\|^2 &= (Ax - b)^T(Ax - b) + (\lambda x)^T(\lambda x) \\ &= \begin{pmatrix} Ax - b \\ \lambda x \end{pmatrix}^T \begin{pmatrix} Ax - b \\ \lambda x \end{pmatrix} \\ &= \left\| \begin{pmatrix} Ax - b \\ \lambda x \end{pmatrix} \right\|^2 \\ &= \left\| \begin{pmatrix} A \\ \lambda I \end{pmatrix} x - \begin{pmatrix} b \\ 0 \end{pmatrix} \right\|^2 . \end{aligned} \tag{2.3}$$

We can substitute this to (2.1). Then the Standard Tichonov takes the form

$$\min_{x \in \mathbb{R}^n} \left\| \begin{pmatrix} A \\ \lambda I \end{pmatrix} x - \begin{pmatrix} b \\ 0 \end{pmatrix} \right\|^2 , \tag{2.4}$$

which is a least squares problem with modified data, that can be also written in the form of the normal equations

$$\begin{pmatrix} A \\ \lambda I \end{pmatrix}^T \begin{pmatrix} A \\ \lambda I \end{pmatrix} x = \begin{pmatrix} A \\ \lambda I \end{pmatrix}^T \begin{pmatrix} b \\ 0 \end{pmatrix}.$$

Equivalently,

$$(A^T A + \lambda^2 I)x = A^T b, \tag{2.5}$$

which has a solution, denoted for particular λ as x_λ, in the form

$$x_\lambda = (A^T A + \lambda^2 I)^{-1} A^T b.$$

Suppose now that we have a singular value decomposition (1.4) of the matrix A and recall that $VV^T = I$. We can write

$$\begin{aligned} x_\lambda &= (A^T A + \lambda^2 I)^{-1} A^T b \\ &= (V\Sigma U^T U \Sigma V^T + \lambda^2 V V^T)^{-1} V \Sigma U^T b \\ &= (V\Sigma^2 V^T + \lambda^2 V V^T)^{-1} V \Sigma U^T b \\ &= V(\Sigma^2 + \lambda^2 I)^{-1} V^T V \Sigma U^T b \\ &= V(\Sigma^2 + \lambda^2 I)^{-1} \Sigma U^T b, \end{aligned}$$

and, therefore,

$$\begin{aligned} x_\lambda &= \sum_{i=1}^{n} \frac{\sigma_i}{\sigma_i^2 + \lambda^2} u_i^T b \, v_i \\ &= \sum_{i=1}^{n} \frac{\sigma_i^2}{\sigma_i^2 + \lambda^2} \frac{u_i^T b}{\sigma_i} v_i, \end{aligned}$$

assuming for simplicity of notation that $rank(A) = n$.

If we denote

$$\varphi_i^{[\lambda]} := \frac{\sigma_i^2}{\sigma_i^2 + \lambda^2}$$

the so-called *filter factors*, we can express x_λ as

$$x_\lambda = \sum_{i=1}^{n} \varphi_i^{[\lambda]} \frac{u_i^T b}{\sigma_i} v_i$$

which is very similar to the expression (1.5) of the naive solution, but with added filter factors.

The filter factors are always smaller then 1 and they satisfy

$$\varphi_i^{[\lambda]} \approx \begin{cases} \sigma_i^2/\lambda^2, & \sigma_i \ll \lambda, \\ 1, & \sigma_i \gg \lambda. \end{cases}$$

It follows, and we can also see it in Figure 2.2, that for the lower indices, where the singular values are large, the factors are almost one and therefore the low frequency vectors v_i are not damped in x_λ. On the contrary, for very high indices, for which the singular values are much lower than lambda, the factors are falling to zero quickly enough to compensate the increasing factors $u_i^T b/\sigma_i$, so that the undesired high frequency vectors v_i are damped or completely filtered out.

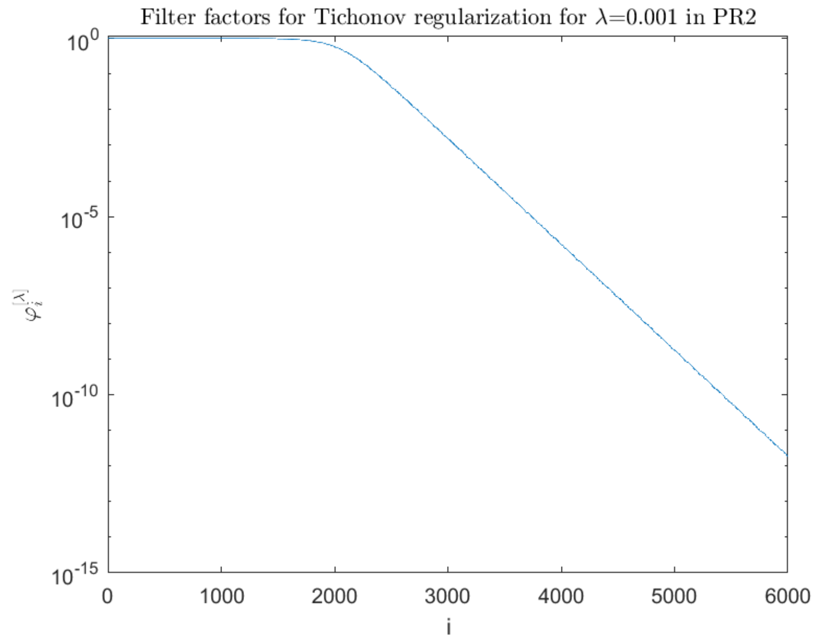

Figure 2.2: The first 6000 filter factors for the Standard Tichonov method for the problem from Example 2. The horizontal axis represents the indices of the filter factors. We can see, that the first few filter factors are almost one, but as the index rises, the factors fastly decay.

It might also be interesting to take a look at the spectrum of the matrices in the normal equations. The normal equations of the original problem without regularization take the form (2.2). It can be seen that the spectrum of the matrix $A^T A$ is the following:

$$sp(A^T A) = \{\sigma_1^2, \ldots, \sigma_n^2\}.$$

The singular values of the matrix A, as we could see in (1.6), decay quickly to zero, and cause that the problem in this form is difficult to be solved and very sensitive to perturbations. On the contrary, when using Tichonov regularization, we replace the expression (2.2) by (2.5). Using the singular value decomposition and the fact $I = VV^T = U^T U$, we can derive the statement

$$\begin{aligned} A^T A + \lambda^2 I &= V\Sigma^T U^T U \Sigma V^T + \lambda^2 V V^2 \\ &= V(\Sigma^T \Sigma + \lambda^2) V^T, \end{aligned}$$

therefore the spectrum of the matrix from the problem (2.5) is

$$sp(A^T A + \lambda^2) = \{(\sigma_1^2 + \lambda^2), \ldots, (\sigma_n^2 + \lambda^2)\}.$$

Obviously, when we add λ to all the eigenvalues in the matrix, we do not have the problem of dividing with very small numbers (similar or smaller than computer accuracy), as we experienced in the previous chapter, and we obtain a formulation that is less sensitive.

2.1.2 Selecting the regularization parameter

In previous sections we mentioned the importance of an appropriate choice of a regularization parameter in different types of regularization methods. We now offer a brief discussion on some known parameter selection methods and then we take a closer look on the Discrepancy principle, which we use further in the computations. For more detailed analysis, see [1, Chapter 5], [2, Chapter 7], [5, Chapter 3.3].

There are two groups of parameter choice methods. The first one contains approaches that do not require any a-priori knowledge about the noise contained in the right hand side b [5, Chapter 3.3.2]. One very popular method from this group is *the L-curve* [3], which is based on the idea of balancing the residual error norm and the solution norm. Another example from this same group is *the generalized cross-validation* [15].

However, when there is a knowledge of the approximate noise level available, it is convenient to use this estimate to find a fitting regularization parameter [5, Chapter 3.3.1]. One example of such method is the above-mentioned Discrepancy principle. The Discrepancy principle is based on the fact that, thanks to (1.2) and (1.3), for the exact solution of the noise free problem it holds that

$$\|A\tilde{x} - b\| = \|A\tilde{x} - \tilde{b} - e\| = \|e\|.$$

Therefore, if we have some estimate $\alpha \approx \|e\|$, and a *safety factor* $\xi > 1$, we can choose a regularization parameter so that the approximate solution $\hat{x}$ satisfies

$$\|A\hat{x} - b\| = \|\hat{r}\| \approx \xi\alpha.$$

The safety factor is introduced to balance the uncertainty in the approximation of the noise norm and it usually ranges between 1 and 1.5.

In the following chapters we will restrict ourselves to use only the Discrepancy principle as a stopping criterion for our computations.

2.2 How to solve Tichonov problem

In the previous section, we derived some theoretical properties of the Standard Tichonov regularization. In this section we will focus on the question that might arise, which is how to solve some real world problem by the Tichonov method numerically. Some notes on this topic can be found in [2, Chapter 5.1]. Despite the advantage that the singular value decomposition gives us when trying to understand the theoretical aspects, it is not that useful when it comes to real computations. Usually for large scale problems, the computation of SVD is expensive and inefficient, sometimes it is not available at all. We need to introduce some ways to implement the method without using the SVD.

2.2.1 Projecting on a Krylov subspace

There are two different approaches to this task. It is possible to treat the problem with methods for functional minimization, see [16], [17]. Second option is to apply some iterative method to the equation (2.5). We will limit ourselves to study only the second approach. In particular, we will consider *Krylov subspace methods* applied implicitly on the equation (2.5). The principle of any Krylov subspace method is to project the problem on a *Krylov subspace*, which makes it simpler, and then solve the projected system in order to obtain an approximation of the solution. We present the definition of a *Krylov subspace.*

Definition 3 (The Krylov subspace)**.** *Let us have a square matrix* $P \in \mathbb{R}^{n \times n}$*, a vector* $q \in \mathbb{R}^n$*, and* $k \leq n, k, n \in \mathbb{N}$*. We define the* k-th Krylov subspace generated by P and q *as*

$$\mathcal{K}_k(P, q) = span\{q, Pq, \ldots, P^{k-1}q\}.$$

It is obvious that $dim(\mathcal{K}_k(P, q)) \leq k$, for all further derivations we will assume that the dimension is exactly k.

Note that the matrix $A^T A$ is symmetric semidefinite and assume for a moment that it is also positive definite, i.e. $rank(A) = n$. We might then consider solving (2.2) using some Krylov subspace method suitable for symmetric matrices, e.g., *conjugate gradient method (CG)*, for details on CG see [14, Chapter 10.2-10.3], [2, Chapter 6.3-6.4], [8, Chapter 15], [18, Chapter 7.4.1] and in particular [19]. Let us assume a problem

$$P\hat{z} \approx q,$$

where P is symmetric positive definite, $q \in \mathbb{R}^n$ is an observation and $\hat{z} \in \mathbb{R}^n$ represents the true solution. In general in iterative methods, one might also consider an initial vector z_0, but in this thesis for simplicity of exposition, we assume that $z_0 = \vec{0}$. Then, for $k \in \mathbb{N}$, the k-th CG approximation of $\hat{z}$ is determined by

$$\|z_k - \hat{z}\|_P = \min_{z \in \mathcal{K}_k(P,q)} \|z - \hat{z}\|_P,$$

where $\| \cdot \|_P$ is the vector P-norm.

Since by our assumption $A^T A$ is symmetric and positive definite, the norm $\| \cdot \|_{A^T A}$ is well defined. CG method applied on the normal equations (2.2) is then formulated as

$$\|x_k - x_{naive}\|_{A^T A} = \min_{x \in \mathcal{K}_k(A^T A, A^T b)} \|x - x_{naive}\|_{A^T A}. \tag{2.6}$$

It holds that

$$\begin{aligned} \|x_k - x_{naive}\|^2_{A^T A} &= \langle (A^T A)(x_k - x_{naive}); x_k - x_{naive} \rangle \\ &= \langle Ax_k - Ax_{naive}; Ax_k - Ax_{naive} \rangle \\ &= \langle Ax_k - b; Ax_k - b \rangle \\ &= \|Ax_k - b\|^2, \end{aligned} \tag{2.7}$$

therefore the CG approximation actually minimizes the residual norm of the approximate solution of (1.1) over $\mathcal{K}_k(A^T A, A^T b)$, which is also the first term in (2.1). Different mathematically equivalent algorithms for computing the approximate x_k are based either on *the Symmetric Lanczos algorithm* (sometimes referred to as Lanczos tridiagonalization) applied on our problem, for general form and more details see [20], or the *Golub-Kahan iterative bidiagonalization*, see [5, Chapter 3.2.1], [21]. We will gradually introduce the basic ideas behind these two approaches.

The Symmetric Lanczos algorithm applied on a symmetric square matrix P and a vector q offers a three-term recurrence that iteratively constructs an orthonormal basis of the Krylov subspace $\mathcal{K}_k(P, q)$, usually called Arnoldi basis, that has interesting numerical properties. The process can be derived using the Gram-Schmidt orthogonalization process on the Krylov basis

$$q, Pq, \dots, P^{k-1}q$$

of the space $\mathcal{K}_k(P, q)$ in a specific way. For details see [22, Chapter 2.4]. We need to use the Symmetric Lanczos algorithm for the setting $P = A^TA, q = A^Tb$. In the k-th iteration, the process returns a matrix $V_k = (v_1, \dots, v_k) \in \mathbb{R}^{n\times k}$ with orthonormal basis of $\mathcal{K}_k(A^TA, A^Tb)$ in the columns, where $v_1 = A^Tb/\|A^Tb\|$, and a matrix

$$T_k = \begin{pmatrix} \alpha_1 & \beta_2 & & & \\ \beta_2 & \alpha_2 & \beta_3 & & \\ & \beta_3 & \ddots & \ddots & \\ & & \ddots & \ddots & \beta_k \\ & & & \beta_k & \alpha_k \end{pmatrix} \in \mathbb{R}^{k\times k}$$

satisfying

$$V_k^T(A^TA)V_k = T_k. \tag{2.8}$$

Denoting

$$T_{k+1,k} = \begin{pmatrix} T_k \\ 0\dots 0\,\beta_{k+1} \end{pmatrix} \in \mathbb{R}^{(k+1)\times k},$$

it holds

$$\begin{aligned}(A^TA)V_k &= V_{k+1}T_{k+1,k} \\ &= V_kT_k + \beta_{k+1}v_{k+1}e_k^T.\end{aligned}$$

We can use the relation (2.8) to derive the projected problem. Let's start with multiplying the normal equations (2.2) by V_k^T. Denoting $\tilde{x}_k \in \mathcal{K}_k(A^TA, A^Tb)$ the approximate we are searching for, we get

$$V_k^T(A^TA)\tilde{x}_k = V_k^TA^Tb.$$

Since $\tilde{x}_k \in \mathcal{K}_k(A^TA, A^Tb)$, we can substitute

$$\tilde{x}_k = V_k\tilde{y}_k, \tilde{y}_k \in \mathbb{R}^k,$$

therefore

$$V_k^T(A^TA)V_k\tilde{y}_k = V_k^TA^Tb.$$

Then, from (2.8), it follows that

$$\begin{aligned}T_k\tilde{y}_k &= \|A^Tb\|e_1, \\ \tilde{x}_k &= V_k\tilde{y}_k.\end{aligned} \tag{2.9}$$

We arrived at a form of the problem with tridiagonal symmetric matrix (so called Jacobi matrix), so the problem in this form is easily solvable by backward substitution (the matrix T_k is nonsingular, since we assumed A^TA to be positive definite). This method sustaining of Lanczos tridiagonalization and subsequent solution of the projected problem is referred to as *CGLS method* [1, Chapter 6.3.2] for the problem (1.1). But there is one issue. Since we assumed to deal with the original normal equations (2.2), this process will, with increasing dimension of the Krylov subspace, converge to the naive solution x_{naive}, which is not desirable.

We may apply similar process to the Tichonov normal equations (2.5) with positive definite matrix $A^TA + \lambda^2 I$, which leads to the minimization problem

$$\|x'_k - \hat{x}\|_{A^TA+\lambda^2 I} = \min_{x \in \mathcal{K}_k(A^TA+\lambda^2 I, A^Tb)} \|x - \hat{x}\|_{A^TA+\lambda^2 I}, \tag{2.10}$$

where $\hat{x}$ now represents the exact solution of the Tichonov normal equations (2.5). Since we applied the Symmetric Lanczos process to a different matrix-vector pair, we generally end up with different results. Denoting T'_k, V'_k the matrices from Lanczos tridiagonalization of $\mathcal{K}_k(A^TA + \lambda^2 I, A^Tb)$, equalities analogous to (2.8) and (2.9) hold, thus

$$\begin{aligned} T'_k y'_k &= \|A^Tb\| e_1, \\ x'_k &= V'_k y'_k, y'_k \in \mathbb{R}^k. \end{aligned} \tag{2.11}$$

Note that while the matrix T_k bears only the information about the matrix A^TA, the matrix T'_k incorporates the information about the regularization parameter λ as well. Also, for $\lambda > 0$, the matrix $A^TA + \lambda^2 I$ is symmetric and positive definite, therefore T'_k is a nonsingular matrix. This variant (also with more general regularization terms) is studied in [23]. For a fixed λ, the approximation x_k then converges to the Tichonov solution x_λ. Let's note, that it is not necessary to construct the matrix $A^TA+\lambda^2 I$. Since we only need the matrix-vector multiplication in every iteration for constructing the subsequent basis vector, it is possible to compute $A^TAv + \lambda^2 v$ gradually.

But still, the need to use λ in every iteration is a big disadvantage, since we may not be able to determine the regularization parameter a-priori and it would be very inefficient to repeat the whole computation of Krylov subspace basis for many variants of λ.

2.2.2 Shift invariance of Krylov subspaces

To our rescue comes the following lemma, that states the *shift-invariance* of Krylov spaces, formulated for a concrete Krylov space $\mathcal{K}_k(A^TA, A^Tb)$.

Lemma 2. *Taking the matrix $A \in \mathbb{R}^{m\times n}$, the vector $b \in \mathbb{R}^n$ as above, for any $\lambda > 0$ and $k \leq n$ it holds that the spaces $\mathcal{K}_k(A^TA + \lambda^2 I, A^Tb)$ and $\mathcal{K}_k(A^TA, A^Tb)$ are equal.*

Proof. Let us consider an arbitrary vector $x \in \mathcal{K}_k(A^TA + \lambda^2 I, A^Tb)$. Then there exist coefficients $\gamma_0, \ldots, \gamma_{k-1} \in \mathbb{R}$, so that, using the binomial theorem [24], we can derive

$$\begin{aligned} x &= \sum_{i=0}^{k-1} \gamma_i (A^TA + \lambda^2 I)^i (A^Tb) \\ &= \sum_{i=0}^{k-1} \gamma_i \sum_{j=0}^{i} \binom{i}{j} \lambda^{2(i-j)} (A^TA)^j (A^Tb) \\ &= \sum_{j=0}^{k-1} \left[\sum_{i=j}^{k-1} \gamma_i \binom{i}{j} \lambda^{2(i-j)} \right] (A^TA)^j (A^Tb), \end{aligned}$$

which implies that $x \in \mathcal{K}_k(A^TA, A^Tb)$ and therefore $\mathcal{K}_k(A^TA + \lambda^2 I, A^Tb) \subset \mathcal{K}_k(A^TA, A^Tb)$. We might use $-\lambda$ to prove the opposite statement, therefore the lemma holds. □

It follows from the structure of the Symmetric Lanczos algorithm, that, in precise arithmetic, the resulting basis $v_1, \ldots, v_k$ and $v'_1, \ldots, v'_k$ for the spaces $\mathcal{K}_k(A^TA, A^Tb)$ and $\mathcal{K}_k(A^TA+\lambda^2 I, A^Tb)$, respectively, are identical. Thus $V_k = V'_k$, while $T_k \neq T'_k$. However note that in finite precision, due to differences in the computations, the resulting basis vectors can differ in particular in later iterations.

Thanks to the Lemma 2 we can rewrite the minimization problem (2.10) equivalently as

$$\|x'_k - \hat{x}\|_{A^TA+\lambda^2 I} = \min_{x \in \mathcal{K}_k(A^TA, A^Tb)} \|x - \hat{x}\|_{A^TA+\lambda^2 I}.$$

It means that we can change the order of projection and Tichonov regularization when solving the inverse problem (1.1), which allows us to compute the Symmetric Lanczos algorithm only once for the Krylov subspace $\mathcal{K}_k(A^TA, A^Tb)$ and use the Tichonov regularization on the small projected problem (2.9). This results in a problem

$$\begin{aligned}(T_k + \lambda^2 I)y_k &= \|A^Tb\|e_1,\\ x_k &= V_k y_k,\end{aligned}$$

or alternatively

$$\begin{aligned}\min_{y_k \in \mathbb{R}^k} \{\|T_k y_k - \|A^Tb\|e_1\|^2 + \lambda^2 \|y_k\|^2\},\\ x_k = V_k y_k\end{aligned} \tag{2.12}$$

which is mathematically equivalent to (2.11), therefore $x_k = x'_k$. We no longer have the problem of using λ a-priori in every iteration of the Symmetric Lanczos process, and it is therefore possible to try different variants of λ once we have the projected problem computed. The possibility of changing the order of projection and Tichonov regularization in the basic form was introduced by Per Christian Hansen [1, Chapter 6.4].

2.2.3 Golub-Kahan iterative bidiagonalization

As we mentioned, there is an alternative way to accomplish the projection onto a Krylov subspace based on the Golub-Kahan iterative bidiagonalization. This process was introduced in [21] and, applied on the matrix A and the vector b, it offers a three-term recurrence generating two sets of basis vectors for the Krylov subspaces $\mathcal{K}_k(A^TA, A^Tb)$ and $\mathcal{K}_k(AA^T, b)$. The algorithm is introduced in the Algorithm 1, see also [5, Algorithm 3.3]. Note that the algorithm stops whenever one of the constants $\alpha_{i+1}, \beta_{i+1}$ is zero and it means that the new computed vector is almost linearly dependent on the previous vectors.

The bidiagonalization process returns the matrices $U_{k+1} = (u_1, \ldots, u_{k+1})$, $V_{k+1} = (v_1, \ldots, v_{k+1})$, which contain the basis of the spaces $\mathcal{K}_{k+1}(AA^T, b)$ and

Algorithm 1 Golub-Kahan iterative bidiagonalization

```
Require: $A \in \mathbb{R}^{m\times n}, b \in \mathbb{R}^m, k \in \mathbb{N}$
Ensure: $U_{k+1}, V_{k+1}, B_k$
  $\beta_1 = \|b\|$
  $u_1 = b/\beta_1$
  $\tilde{v} = A^T u_1$
  $\alpha_1 = \|\tilde{v}\|$
  $v_1 = \tilde{v}/\alpha_1$
  for $i = 1, \dots, k$ do
      $\tilde{u} = Av_i - \alpha_i u_i$
      $\beta_{i+1} = \|\tilde{u}\|$
      $u_{i+1} = \tilde{u}/\beta_{i+1}$
      $\tilde{v} = A^T u_{i+1} - \beta_{i+1} v_i$
      $\alpha_{i+1} = \|\tilde{v}\|$
      $v_{i+1} = \tilde{v}/\alpha_{i+1}$
  end for
```

$\mathcal{K}_{k+1}(A^TA, A^Tb)$, respectively, in the columns, and the matrix

$$B_k = \begin{pmatrix} \alpha_1 & & & \\ \beta_2 & \alpha_2 & & \\ & \beta_3 & \ddots & \\ & & \ddots & \alpha_k \\ & & & \beta_{k+1} \end{pmatrix} \in \mathbb{R}^{(k+1)\times k}.$$

Denoting $B_{k,k} \in \mathbb{R}^{k\times k}$ the matrix B_k without the last row, the following relations hold

$$\begin{aligned} AV_k &= U_{k+1}B_k \\ &= U_kB_{k,k} + \beta_{k+1}u_{k+1}e_k^T, \\ A^TU_{k+1} &= V_{k+1}B_{k+1,k+1}^T \\ &= V_kB_k^T + \alpha_{k+1}v_{k+1}e_{k+1}^T. \end{aligned} \tag{2.13}$$

We can now consider the CG applied to normal equations, which is equivalent to minimization of the residual norm as we have seen in (2.7). Then we can use the relations (2.13) to modify the problem. Substituting

$$\tilde{x}_k = V_k\bar{y}_k, \bar{y}_k \in \mathbb{R}^n$$

and using the fact that $b = U_{k+1}\beta_1e_1$ and that multiplying by U_k doesn't change the norm we can write

$$\begin{aligned} \|A\tilde{x}_k - b\|^2 &= \|AV_k\bar{y}_k - U_{k+1}\beta_1e_1\|^2 \\ &= \|U_{k+1}B_k\bar{y}_k - U_{k+1}\beta_1e_1\|^2 = \\ &= \|U_{k+1}(B_k\bar{y}_k - \beta_1e_1)\|^2 \\ &= \|B_k\bar{y}_k - \beta_1e_1\|^2. \end{aligned}$$

We arrived at the following least squares problem form of the minimization (2.6)

$$\begin{aligned} B_k\bar{y}_k &= \beta_1e_1, \\ \tilde{x}_k &= V_k\bar{y}_k, \end{aligned} \tag{2.14}$$

which is very similar to (2.9). This process of projecting the problem onto $\mathcal{K}_k(A^TA, A^Tb)$ via Golub-Kahan bidiagonalization process and solving the smaller problem is called the *LSQR method*, see [25], [2, Chapter 6.3.1]. Note that CGLS and LSQR methods are mathematically equivalent.

As in the case of CGLS method, we can apply the Tichonov regularizaion directly to the projected problem (2.14), resulting with

$$\begin{aligned}(B_k + \lambda^2 I)y_k &= \beta_1 e_1,\\ x_k &= V_k y_k.\end{aligned}$$

Note that using both the Conjugate gradient method for normal equations (either the CGLS or LSQR variant) and the Tichonov regularization brought us to a setting, where we have to deal with two parameters - the dimension of the Krylov subspace k and the strength of Tichonov regularization λ. The resulting method is, in fact, the so-called hybrid approach. We will get into details about hybrid methods in the next section.

2.3 Hybrid regularization

In the previous section, when describing the possibilities to solve the Standard Tichonov problem, we arrived at one example of a *hybrid method* [5], [1, Chapter 6.4]. In this section, we concentrate on the general setting. As was mentioned in Section 1.4, hybrid methods combine the variational and iterative methods to achieve even better results. We mentioned Lemma 2, that revealed the enjoyable fact that, for concrete combination of methods and fixed parameters, there is no difference in the two approaches *first project, then regularize* and *first regularize, then project.* In fact, this holds for many hybrid settings [5, Chapter 1.3], therefore it is not a disadvantage to restrict ourselves to the first project, then regularize setting. It is convenient for large-scale problems, since solving the projected problem of small dimension is cheaper, allowing us to try more variants of regularization parameter for one projection iteration. The general algorithmic approach of a hybrid method is introduced in the Algorithm 2, similar algorithm can be found in [5, Chapter 3].

Algorithm 2 Hybrid approach

```
Require: $A, b, k = 0$, projection method, variational method for solving the
  projected problem, parameter selection methods, stopping criterion
Ensure: approximate solution $x_k^\lambda$
  while stopping criterion not satisfied do
      Compute next basis vector to expand the projection space $\mathcal{V}_k$;
      Construct the problem projected to the space $\mathcal{V}_k$;
      Select the parameter $\lambda$ for inner regularization;
      Compute the solution $x_k^\lambda$ of the projected and regularized problem;
      Increase the dimension $k$
  end while
```

In this thesis, we concentrate on the sort of hybrid methods that combine Krylov subspace methods with General Tichonov regularization. Using the CGLS

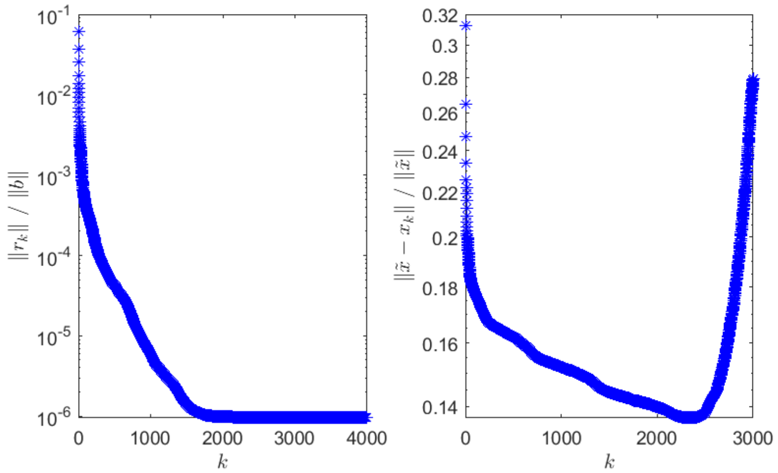

Figure 2.3: The comparison of the relative residual norms and the relative error norms for the CGLS method without Tichonov regularization for the problem from Example 2. On the left we can see that the relative residual norm is decreasing as we proceed with the computation. On the contrary, on the right we can see the relative error norm that decreases in the first part of the graph, but shortly after 2000th iteration it starts to increase again. This behaviour is called *semiconvergence* and it is typical for iterational methods as Krylov subspace methods. The semiconvergence causes the proces of choosing the suitable parameter k to be difficult.

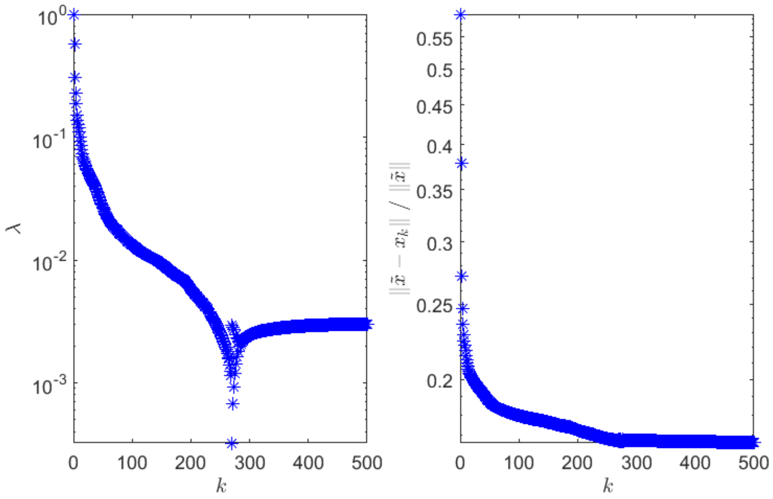

Figure 2.4: The values of the inner parameter λ computed during the hybrid LSQR method and the relative error norms during iterations for the problem from Example 2. The inner parameter (on the left) was chosen using the Discrepancy principle using the real noise level μ_e of the problem and a safety factor $\xi = 1.01$. We can see that the parameter λ settles after some iterations, which indicates, that the optimal value was found and the iterations can be stopped. The relative error (on the right) decreases during the whole process and we do not see any semiconvergence.

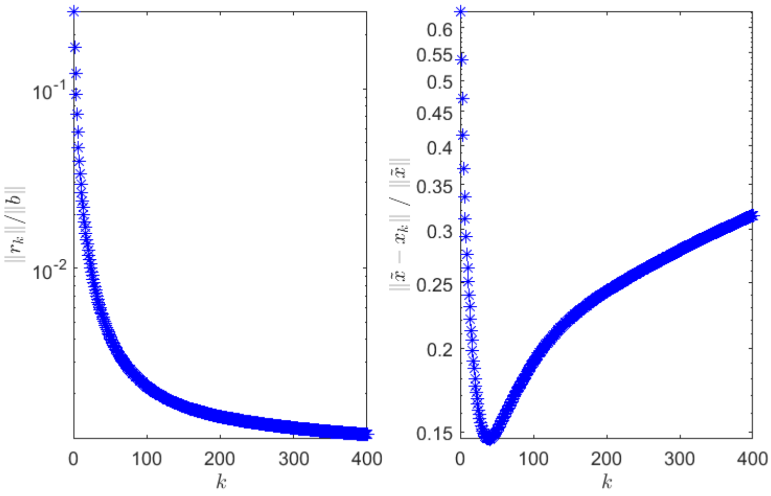

Figure 2.5: The comparison of the relative residual norms and the relative error norms for the LSQR method without Tichonov regularization for the problem from Example 1. On the left we can see that the relative residual norm is decreasing as we proceed with the computation. On the contrary, on the right we can see the semiconvergence of the relative error norm. When compared to the case of Example 2 we can see that the semiconvergence occured much earlier, due to the higher noise level.

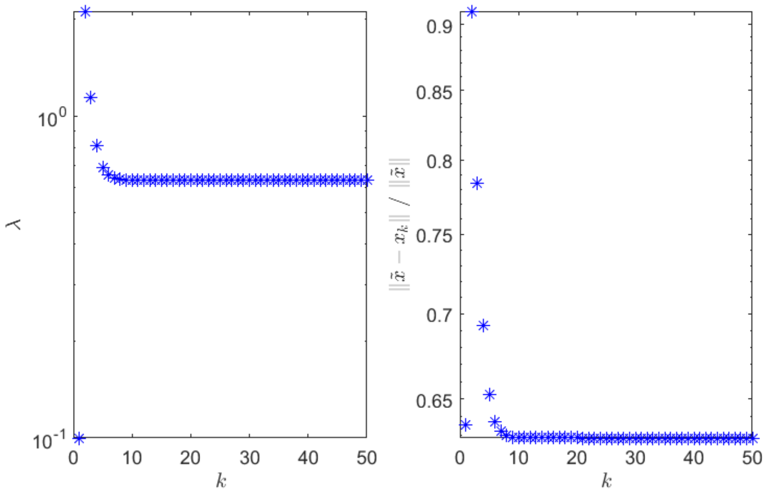

Figure 2.6: The values of the inner parameter λ and the relative error norms from the hybrid LSQR method applied on Example 1. We used the Discrepancy principle with the real noise level μ_e and a safety factor $\xi = 1.01$ as parameters. The regularization parameter λ settles after less than 10 iterations which is much earlier than in the case with Example 2 due to the higher noise level in b. The relative error also settles after some iterations and we do not experience the semiconvergence.

or LSQR, the problem (1.1) is projected on the space $\mathcal{K}_k(A^TA, A^Tb)$ of increasing dimension, and then the projected problem is regularized with Tichonov method and solved. The use of parameter choice methods based on the SVD is possible for the projected problem, since the computation of the SVD of the projected matrix is usually cheaper comparred to the computation of matrix-vector multiplication with A or A^T during the projection process [5, Chapter 3.3]. The Tichonov method with general regularization term is described in the next chapter.

We can see an illustration of a behaviour of a hybrid method compared to use of a solo Krylov subspace method in Figures 2.3 and 2.4. In Figure 2.3 we can see the relative residual norms and the relative error norms for the problem from Example 2, which are results of a CGLS method. The CGLS is an implementation of the approach, when the Conjugate gradient method is applied to the normal equations, introduced in Section 2.2. The relative error decreases for many iterations, as we get closer to the exact solution, but after some iterations, the error starts to increase very quickly, as the noisy components enter the approximate solution. This behaviour is called *semiconvergence* and it is typical for these types of methods. It is not easy to set the parameter k ideally, so that the error is minimal, since we do not have the knowledge of the real error in the real computations and the residual continues to decrease even when we actually get far from the exact solution.

The situation with hybrid methods is different. In Figure 2.4 we can see the development of the regularization parameter λ and the relative error norms for the problem from Example 2, which are results of a hybrid LSQR method. The problem is projected onto a Krylov subspace of a smaller dimension, which increases in every iteration and then the Tichonov regularization is applied to the projected problem, which is again mathematically the same process of CG applied to normal equations as we introduced in Section 2.2. We used the Discrepancy principle as a parameter choice method for the inner regularization, with the real noise level μ_e and a safety factor $\xi = 1.01$. The regularization parameter λ stabilized after some iterations and we do not see any semiconvergence in the process. The fact, that the inner regularization parameter stabilizes can be interpreted as that the value of the parameter is already sufficient even for the large-scale problem without projection, see [5, Chapter 3.3], [26]. Also the use of both projection and regularization has the effect that we get rid of the semiconvergence and the error decreases during the iterations, therefore we do not have to take such a good care about the outer parameter k and we can stop the iterations once the inner parameter λ stabilizes.

In Figures 2.3 and 2.4 we can see, that it took lots of iterations to encounter the semiconvergence in CGLS and the stabilization of the parameter λ in the hybrid LSQR. This is due to the fact that the noise level in Example 2 is really low ($\mu_e = 10^{-6}$). We can see similar illustrations for Example 1 that has much higher noise level in b ($\mu_e = 10^{-2}$) in Figures 2.5 and 2.6. In this case we used the LSQR and hybrid LSQR methods for computations. We can see that both the semiconvergence in LSQR and the stabilization of λ in hybrid LSQR appeared much quicker, in less then 100 iterations.

3. Tichonov regularization with general regularization term

Until now we have considered only the Standard Tichonov regularization in the form (2.1). It allowed us to control the norm of an approximate solution. However, according to some a-priori information, we might want to include some other information about the smoothness of the sought solution, therefore it is convenient to consider more general regularization terms. The Tichonov regularization in the general form can be formulated as

$$\min_{x\in\mathbb{R}^n}\{\|Ax-b\|^2+\lambda^2\mathcal{R}(x)\}. \tag{3.1}$$

There are several variants of regularization terms studied in literature. We described the variant with the 2-norm of an approximate solution in the previous chapter. We will further concentrate on studying methods that contain regularization term in the form $\mathcal{R}(x)=\|Lx\|^2$, where $L\in\mathbb{R}^{p\times n}, p\in\mathbb{N}$, is a regularization matrix. Such variant of a problem (3.1) is often called the *General Tichonov problem* [5, Chapter 4.1]. It takes the form

$$\min_{x\in\mathbb{R}^n}\{\|Ax-b\|^2+\lambda^2\|Lx\|^2\}. \tag{3.2}$$

3.1 Transformation to the Standard form

The formulation (3.2) can be transformed to Standard Tichonov form. As was demonstrated in [9], we can proceed as follows. We assume that the null spaces of A and L intersect trivially ($Ker(A)\cap Ker(L)=\{0\}$), therefore the resulting problem (3.2) has a unique solution.

We denote

$$r=\begin{pmatrix}A\\ \lambda L\end{pmatrix}x-\begin{pmatrix}b\\0\end{pmatrix}.$$

Then (3.2) is clearly equivalent to minimization of $\|r\|$. Now assume that L has full row rank and it is possible to compute the QR factorization [24, Chapter 8.4.3]

$$L^T=\begin{pmatrix}P_1 & P_2\end{pmatrix}\begin{pmatrix}R_1\\0\end{pmatrix},$$

where $P=\begin{pmatrix}P_1 & P_2\end{pmatrix}\in R^{n\times n}, R_1\in\mathbb{R}^{p\times p}$ are nonsingular, P is orthogonal and R_1 is upper triangular, $P_1\in\mathbb{R}^{n\times p}, P_2\in\mathbb{R}^{n\times\{n-p\}}$. Since in practice L is typically a band matrix, the computation of this QR factorization can be done efficiently.

First assume that $p=n=rank(L)$. Than the matrix PR_1^{-T} is nonsingular, so we can safely use a substitution $x=PR_1^{-T}\hat{x}, \hat{x}\in\mathbb{R}^n$ to get

$$\begin{aligned}r&=\begin{pmatrix}APR_1^{-T}\\ \lambda LPR_1^{-T}\end{pmatrix}\hat{x}-\begin{pmatrix}b\\0\end{pmatrix}\\&=\begin{pmatrix}APR_1^{-T}\\ \lambda(P^TL^T)^TR_1^{-T}\end{pmatrix}\hat{x}-\begin{pmatrix}b\\0\end{pmatrix}\\&=\begin{pmatrix}APR_1^{-T}\\ \lambda I\end{pmatrix}\hat{x}-\begin{pmatrix}b\\0\end{pmatrix},\end{aligned}$$

and we obtained a reformulation of (3.2) to the standard form

$$\min_{\hat{x}\in\mathbb{R}^n} \left\| \begin{pmatrix} APR_1^{-T} \\ \lambda I \end{pmatrix} \hat{x} - \begin{pmatrix} b \\ 0 \end{pmatrix} \right\|. \tag{3.3}$$

Now assume $p < n$, then we may use a substitution $x = Py = P_1y_1 + P_2y_2, y_1 \in R^p, y_2 \in \mathbb{R}^{n-p}$, and write

$$r = \begin{pmatrix} AP_1 & AP_2 \\ \lambda R_1^T & 0 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} - \begin{pmatrix} b \\ 0 \end{pmatrix}. \tag{3.4}$$

Since $Ker(P_1) = Ker(L)$, it follows that $rank(AP_2) = n - p$, therefore we can compute a QR factorization of AP_2:

$$AP_2 = \begin{pmatrix} Q_1 & Q_2 \end{pmatrix} \begin{pmatrix} R_2 \\ 0 \end{pmatrix},$$

where $Q = \begin{pmatrix} Q_1 & Q_2 \end{pmatrix} \in R^{m\times m}, R_2 \in \mathbb{R}^{\{n-p\}\times\{n-p\}}$, R_2 is nonsingular. We can multiply the first m components of (3.4) by Q^T and obtain a modified

$$\tilde{r} = \begin{pmatrix} \tilde{r}_1 \\ \tilde{r}_2 \\ \tilde{r}_3 \end{pmatrix} = \begin{pmatrix} Q_1^T AP_1 & R_2 \\ Q_2^T AP_1 & 0 \\ \lambda R_1^T & 0 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} - \begin{pmatrix} Q_1^T b \\ Q_2^T b \\ 0 \end{pmatrix}.$$

Note that both $\tilde{r}_1$ and $\tilde{r}_3$ are independent of y_2 and we can therefore compute y_2 so that $\tilde{r}_1 = 0$, and we can split the above expression into

$$\min_{y_1\in\mathbb{R}^p} \left\| \begin{pmatrix} Q_2^T AP_1 \\ \lambda R_1^T \end{pmatrix} y_1 - \begin{pmatrix} Q_2^T b \\ 0 \end{pmatrix} \right\|,$$
$$y_2 = R_2^{-1} Q_1^T (b - AP_1 y_1).$$

Finally, changing the variable $\bar{x} = R_1^T y_1$ we obtain a problem

$$\min_{y_1\in\mathbb{R}^p} \left\| \begin{pmatrix} Q_2^T AP_1 R_1^{-T} \\ \lambda I \end{pmatrix} \bar{x} - \begin{pmatrix} Q_2^T b \\ 0 \end{pmatrix} \right\| \tag{3.5}$$

in the standard form, from which, after obtaining the approximate solution $\bar{x}$, the solution of the original problem (3.2) can be computed as

$$x = P_1 R_1^{-T} \bar{x} + P_2 R_2^{-1} Q_1^T (b - AP_1 R_1^{-T} \bar{x}).$$

The question is how to solve the modified problem (3.3) or (3.5). We might simply apply some iterative solver as CGLS or LSQR to obtain an approximate solution. Of course the matrix of the problem is not necessary to be constructed explicitly, we can perform the matrix vector multiplication gradually. The formulation already incorporates the regularizing effect of λ and L. But again we arrived at a problem that this approach requires to know the regularization parameter λ a-priori, which is inconvenient. But we can notice that the transformed problem is in the same form as (2.4) and therefore we might repeat the derivation (2.3). E.g. for (3.5), denoting $\bar{A} = Q_2^T AP_1 R_1^{-T}$ and $\bar{b} = Q_2^T b$ we can arrive at

$$\min_{\bar{x}\in\mathbb{R}^p} \{ \|\bar{A}x - \bar{b}\|^2 + \lambda^2 \|\bar{x}\|^2 \}$$

and we can apply the same procedure as described in Sections 2.2 and 2.3, which allows us to first project, then regularize as is the common setting in hybrid methods.

Alternatively we might use a different transformation to the standard form than the one performed above. The transformation can be found in [1, Chapter 8.4] and it allows the user to transform the derivative matrix L to a preconditioner. It suggests to use the so-called *oblique pseudoinverse* of L that can be expressed as

$$L^{\#} = (I - W_L(AW_L)^{\dagger}A)L^{\dagger},$$

where W_L has a base of $Ker(L)$ in the collumns and the symbol † denotes the pseudoinverse of a matrix. In the case, where L is square and invertible, the oblique pseudoinverse is equal to the inverse of a matrix. Assuming that L has a full row rank and the exact solution $\tilde{x}$ does not have a part in $Ker(L)$, we can substitute

$$\overline{\overline{A}} = AL^{\#}, x = L^{\#}\overline{\overline{x}}$$

to (3.2). We end up with a modified problem

$$\min_{\overline{\overline{x}}}\{\|\overline{\overline{A}}\overline{\overline{x}} - b\| - \lambda^2\|\overline{\overline{x}}\|\},$$

on which we can apply the conjugate gradient method, searching for $\overline{\overline{x}}_k$ in

$$\begin{aligned}\mathcal{K}_k(\overline{\overline{A}}^T\overline{\overline{A}}, \overline{\overline{A}}^T b) &= \mathcal{K}_k((AL^{\#})^T(AL^{\#}), (AL^{\#})^T b)\\ &= \mathcal{K}_k((L^{\#})^T A^T AL^{\#}, (L^{\#})^T A^T b).\end{aligned}$$

We can write

$$\overline{\overline{x}}_k = \sum_{i-1}^{k} \left[(L^{\#})^T A^T AL^{\#}\right]^i (L^{\#})^T A^T b$$

and therefore

$$\begin{aligned}x_k &= L^{\#}\overline{\overline{x}}_k\\ &= \sum_{i-1}^{k} \left[L^{\#}(L^{\#})^T A^T A\right]^i L^{\#}(L^{\#})^T A^T b\\ &= \sum_{i-1}^{k} \left[MA^T A\right]^i MA^T b,\end{aligned}$$

where we denoted $M = L^{\#}(L^{\#})^T$. Then $x_k \in \mathcal{K}_k(MA^TA, MA^Tb)$ and M is in fact a preconditioner.

This second approach is implemented in the toolboxes [10] and [11].

3.2 Regularization by derivative penalization

As we have seen in the previous section, it is possible to transform the General Tichonov regularization into the standard form. In this section we offer some examples of the regularization matrices L for one dimensional and two dimensional problems. Detailed description of the problematics of choosing the right regularization matrix can be found in [27]. In order to get a smoother solution, it is often useful to take such L that would penalize non-smooth solutions. Therefore the frequent choices of L are discrete approximations of differential operators, which will be described in the following subsections. First we discuss approximations of function derivative in general.

3.2.1 Approximating derivatives by finite differences

The common method of approximating a derivative of a function is the finite-difference method, see [28]. Consider a real function $f : \mathbb{R} \to \mathbb{R}$. Recall that the derivative of f in a point $t \in \mathbb{R}$ is defined by the relation

$$f'(t) = \lim_{h \to 0} \frac{f(t+h) - f(t)}{h}. \tag{3.6}$$

Therefore, if we fix some $h > 0$, we can approximate the derivative $f'(t)$ by the expression

$$f'(t) \approx \frac{f(t+h) - f(t)}{h}, \tag{3.7}$$

which is called a *forward difference*. From (3.6) it is obvious that, if the function f has a derivative on some neighbourhood of t, the forward difference (3.7) will converge to $f'(t)$ when $h \to 0$. Taking $-h$ we get a *backward difference*

$$f'(t) \approx \frac{f(t) - f(t-h)}{h}, \tag{3.8}$$

which also approximates the derivative (3.6), since the limits when $h \to 0$ are equal for both the differences. By computing the average of (3.7) and (3.8) we arrive at a *centered difference*

$$f'(t) \approx \frac{f(t+h) - f(t-h)}{2h}. \tag{3.9}$$

We can observe the behaviour of the forward and centered difference for the first derivative on a simple example of the cosine function in Figure 3.1.

Let us have a look at the accuracy of the approximations (3.7), (3.8) and (3.9). Assuming $f \in \mathcal{C}^2$, we can use the Taylor expansion of f in the point $t + h$ with respect to the point t and write

$$f(t+h) = f(t) + hf'(t) + \frac{h^2}{2} f''(t) + \mathcal{O}(h^3), \tag{3.10}$$

therefore

$$f'(t) = \frac{f(t+h) - f(t)}{h} + \mathcal{O}(h)$$

and the forward difference is a first-order approximation of $f'(t)$. Here $\mathcal{O}(\cdot)$ denotes the big $\mathcal{O}$ of a function. The same result holds for the backward difference when taking the Taylor expansion of f in $t - h$ with respect to t, so

$$f(t-h) = f(t) - hf'(t) + \frac{h^2}{2} f''(t) + \mathcal{O}(h^3), \tag{3.11}$$

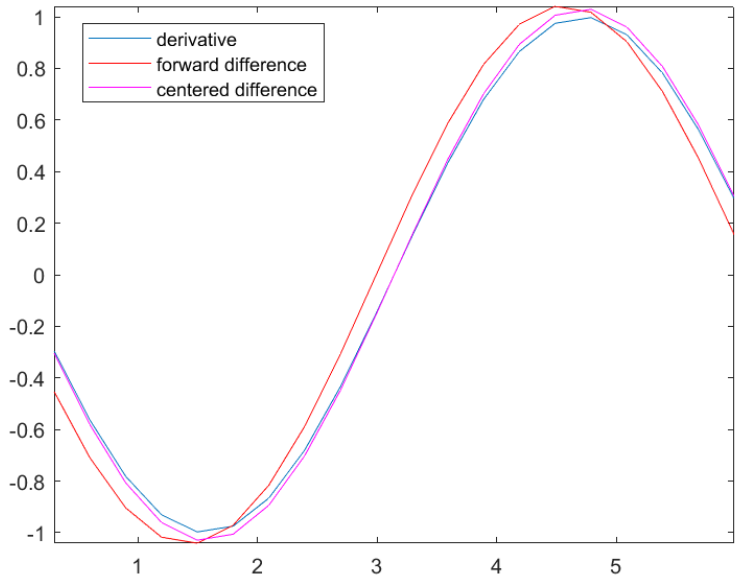

Figure 3.1: The comparison of the true derivative (blue line), the forward difference (red line) and the centered difference (violet line) of the function cosinus on the interval $(0, 2*\pi)$ divided into an equidistant grid of 22 points. Note that we only drew the values for 20 inner points of the grid.

and

$$f'(t) = \frac{f(t) - f(t-h)}{h} + \mathcal{O}(h)$$

By averaging the relations for $f'(t)$ from the Taylor expansions (3.10) and (3.11) we get

$$f'(t) = \frac{f'(t) + f'(t)}{2} = \frac{f(t+h) - f(t-h)}{2h} + \mathcal{O}(h^2),$$

so the centered difference is a second-order approximation of $f'(t)$.

For a second derivative, if $f \in \mathcal{C}^3$, let us sum the Taylor expansions

$$f(t+h) = f(t) + hf'(t) + \frac{h^2}{2}f''(t) + \frac{h^3}{3!}f'''(t) + \mathcal{O}(h^4),$$
$$f(t+h) = f(t) - hf'(t) + \frac{h^2}{2}f''(t) - \frac{h^3}{3!}f'''(t) + \mathcal{O}(h^4),$$

resulting with

$$f(t+h) + f(t-h) = 2f(t) + h^2 f''(t) + \mathcal{O}(h^4),$$

therefore

$$f''(t) = \frac{f(t+h) - 2f(t) + f(t-h)}{h^2} + \mathcal{O}(h^2)$$

and the centered difference defined as

$$f''(t) \approx \frac{f(t+h) - 2f(t) + f(t-h)}{h^2} \tag{3.12}$$

approximates the second derivative with second order of precision.

Taking more components in the Taylor expansions in the case $f \in C^k$, where $k \in \mathbb{N}$ is high enough, it is possible to arrive at a finite-difference approximations to higher derivatives. The corresponding formulas for higher derivatives require more grid points.

3.2.2 Discrete derivative matrices

In the previous section we summarized the most common finite-difference approximations of derivatives for continuous functions. Now we arrive at a question how to use these schemes in a discrete setting. For that purpose we consider vectors to be samples of continuous functions on some domain. Depending on the dimension we are in, there are different variants how to approach this, so we split into the two following sections.

Discrete derivatives in 1D

Let (1.1) be a model of some one dimensional inverse problem. Consider that a vector $x = (x_1, \dots, x_n)^T \in \mathbb{R}^n$ actually consists of samples of a continuous function $f(t)$ on an equidistant grid $0 = t_1 < \cdots < t_n = 1$ with the grid size $h = 1/(n-1)$, thus $f(t_i) = x_i$. It is possible to apply the finite-difference schemes to the function $f(t)$, that allows us to work on the grid only. We consider the finite-difference approximations of the first and second derivative of f, see [1, Chapter 8]. For the first derivative in a point $t_i, i \in \{2, \dots, n\}$, we get the approximation $(1/h) * [f(t_i) - f(t_{i-1})] = (1/h) * [x_i - x_{i-1}]$. If we define the matrix

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

then the vector $(1/h) * L'x$ contains the backward difference approximations of the first derivatives of f in the grid points. Note that, since we use Tichonov regularization for the solution of (1.1), we can include the constant $(1/h)$ to the parameter λ, therefore we can omit it. Thus the matrix L' represents a discrete approximation of the first derivative, its null space is $Ker(L') = span\{(1, \dots, 1)^T\}$. Similarly we can derive a common discrete approximation of the second derivative by the centered difference

$$L'' = \begin{pmatrix} 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \end{pmatrix} \in \mathbb{R}^{(n-2)\times n}.$$

The matrix L'' has a null space $Ker(L'') = span\{(1, \dots, 1)^T, (1, 2, \dots, n)^T\}$.

In some cases it might be reasonable to include some boundary conditions to the discrete derivatives. For a zero boundary condition, we end up with the modified matrices

$$L'_0 = \begin{pmatrix} 1 & & & \\ -1 & 1 & & \\ & \ddots & \ddots & \\ & & -1 & 1 \\ & & & -1 \end{pmatrix} \in \mathbb{R}^{(n+1)\times n}$$

and

$$L_0'' = \begin{pmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ & & & 1 & -2 \end{pmatrix} \in \mathbb{R}^{n\times n}.$$

One advantage of the matrices L_0' and L_0'' is that its null spaces are empty, therefore we do not have to care whether they intersect with the null space of A. In the case of a reflective boundary condition, the matrix of the discrete first derivative is $L_R' = L'$, see [1, Chapter 8.2], while the corresponding discrete second derivative matrix takes the form

$$L_R'' = \begin{pmatrix} -1 & 1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ & & & 1 & -1 \end{pmatrix} \in \mathbb{R}^{n\times n}$$

with $Ker(L_R'') = span\{(1,\dots,1)^T\}$.

In Figures 3.2 and 3.3 we can see the results of the use of different variants of regularization matrices on one-dimensional problems. Both problems were generated using the toolbox [11] by calling the functions *gravity(256,1)* and *gravity(256,2)*. These functions give us the matrix A, the exact solution $\tilde{x}$ and the exact observation $\tilde{b}$ as outputs. To construct (1.1), we added noise at the level $\mu_e = 5 * 10^{-3}$ using the function *PRnoise*. Then the noisy problem (1.1) was solved by the General Tichonov (3.2) computed by the hybrid LSQR method, see Section 2.3. In both cases it was useful to introduce a regularization matrix than enforced the zero boundary condition (see the exact solutions $\tilde{x}$ on the bottom right). For the first problem in Figure 3.2 we were able to achieve a nice approximation with L_0''. For the second example in Figure 3.3 we can see that all the solutions were too smooth and none of the tested matrices led to a perfect approximation. The use of more general regularization terms could help to find some better approximation to a non-smooth solution, see Section 3.3.

In Section 3.2.1 we discussed the orders of accuracy for finite difference schemes. We clarified that both the forward and backward differences for approximating the first derivative are of the first order. We introduce also the matrices

$$L_{cent}' = \begin{pmatrix} -1 & 0 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 0 & 1 \end{pmatrix} \in \mathbb{R}^{(n-2)\times n},$$

$$L_{0,cent}' = \begin{pmatrix} 0 & 1 & & & \\ -1 & 0 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 0 & 1 \\ & & & -1 & 0 \end{pmatrix} \in \mathbb{R}^{n\times n},$$

that represent the centered difference approximation of the first derivative without boundary conditions and with zero boundary conditions, respectively, in order to

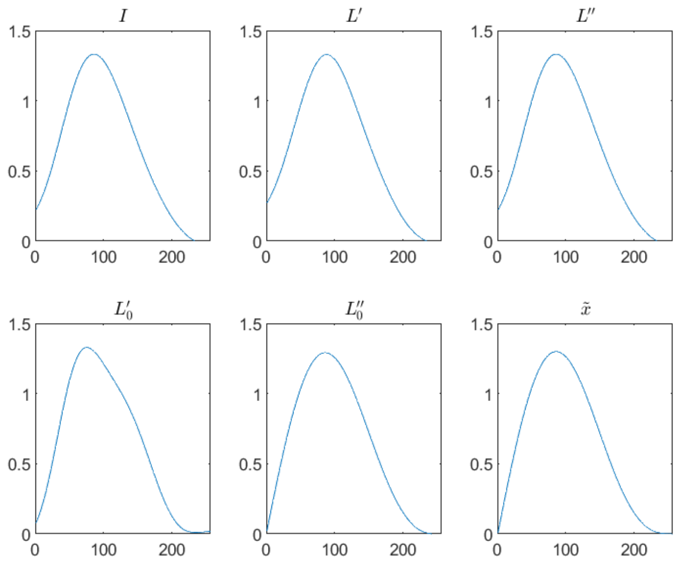

Figure 3.2: The hybrid LSQR solutions for different variants of L for a one dimensional problem from the toolbox [11]. The problem was constructed using the function call *gravity(256,1)* and noise was added at the level $\mu_e = 5{*}10^{-3}$. The hybrid LSQR was called with the estimate of the level of noise $\alpha = 1.1 * \mu_e$ and the safety factor $\xi = 1.2$. We can see that the approximations for $L \in \{I, L', L''\}$ could not capture the zero values on the boundary correctly and the left sides of the plots differ from the true solution $\tilde{x}$ (bottom right). The use of the zero boundary condition led to a more accurate approximation of the solution near the boundaries. The best variant of the regularization matrix was L''_0.

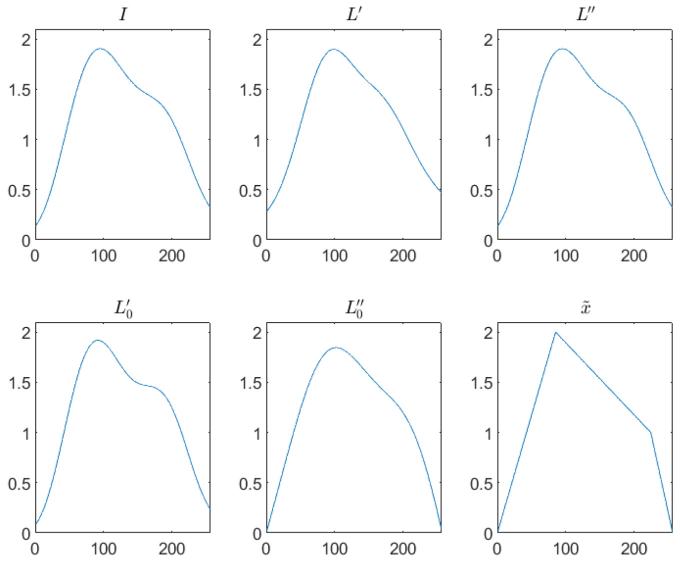

Figure 3.3: The same comparison as in Figure 3.2, for a problem generated by a call *gravity(256,2)* using the parameters $\alpha = 1.1{*}\mu_e, \xi = 1.5$. We can see similar results for the approximation near the boundary. These variants of L were not suitable to enforce a non-smooth solution. Better results might be achieved using more general regularization terms.

verify whether the higher order scheme has better regularization effect in the Tichonov setting. The null space of L'_{cent} is

$$Ker(L'_{cent}) = span\{(1,0,-1,0,\ldots)^T, (0,1,0,-1,\ldots)^T\}.$$

The null space of $L'_{0,cent}$ depends on n. If n is even, then the null space is trivial. For n odd the null space equals

$$Ker(L'_{0,cent}) = span\{(1,0,-1,0,\ldots)^T\}.$$

There are many other variants of discrete derivative matrices with various boundary conditions, since we may consider finite difference approximations of higher derivatives or schemes of higher order as we mentioned in the previous section. If there is a need to include more than one variant of a smoothing regularization matrix, one can also consider using a combination of different operators by simply putting the corresponding matrices below each other to form one matrix.

Discrete derivatives in 2D

In the previous section we introduced few examples of the regularization matrices based on the finite differences that served as the discrete approximations of derivatives in one dimension. However the situation in two dimensions is a bit more complicated, see [12, Chapter 7.3]. Consider (1.1), where $b = vec(B)$ and B is an image, see Section 1.2. Now we need to work with partial derivatives. Let us denote $X \in \mathbb{R}^{p\times q}$ the matrix version of an image stored in the vector x (therefore $n = p*q$). We can consider X to be a matrix of samples of a continuous function $f(s,t)$, where the variables s, t are taken on equidistant grids with the step size h, so that $s_i = i*h, i \in \{1,\ldots,p\}$ and $t_j = j*h, j \in \{1,\ldots,q\}$. We can write $X_{i,j} = f(s_i, t_j)$.

If we want to ensure the smoothness of the solution in both space directions, we need to consider partial derivatives with respect to both the variables s and t (vertical and horizontal direction, respectively). To that aim we will derive the approximations of $^{\partial f}/_{\partial s}$ and $^{\partial f}/_{\partial t}$ on the grid points. Consider a fixed grid point $[s_i, t_j]$. Then the forward difference approximation of the derivative with respect to s is

$$(1/h)\left[f(s_{i+1},t_j) - f(s_i,t_j)\right] = (1/h)\left[X_{i+1,j} - X_{i,j}\right].$$

Note that, as in the one-dimensional case, we can get rid of the constant $(1/h)$ by including it in λ. Now, when we want, for example, zero boundary condition, we can apply the matrix L'_0 to X. Clearly the matrix L'_0X contains the finite differences with respect to the variable s and zero boundary conditions on the grid. Similarly we can obtain the finite differences in the direction t when we apply L'_0 to the rows of X by computing $X(L'_0)^T$. To achieve the regularization in both directions, we can combine these two results in one regularization term

$$\begin{aligned}\mathcal{R}(x) &= \|L'_0X\|_F^2 + \|X(L'_0)^T\|_F^2 \\ &= \|(I_q\otimes L'_0)x\|_2^2 + \|(L'_0\otimes I_p)x\|_2^2 \\ &= \left\|\begin{pmatrix} I_n\otimes L'_0 \\ L'_0\otimes I_m\end{pmatrix} x\right\|_2^2,\end{aligned} \tag{3.13}$$

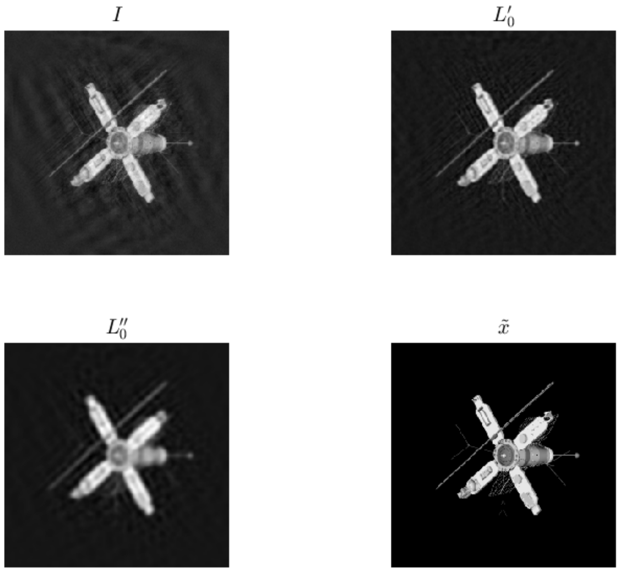

Figure 3.4: The hybrid LSQR solutions for different variants of L for the problem from Example 1. The method was called with the estimate of the level of noise $\alpha = 1.1 * \mu_e$ and the safety factor $\xi = 1.5$ and maximum number of iterations 2000. The variant with L_0'' did not satisfy the stopping criterion and stopped after 2000 iterations. The variant with L_0' offers a better approximate solution then the identity I, since L_0' enforces a smoother solution.

where $\| \cdot \|_F$ is a Frobenius norm of a matrix and the symbol $\otimes$ denotes the Kronecker product of two matrices. This term is in fact a discrete variant of a weighted Sobolev norm. The null space of the regularization matrix in (3.13) is empty, because of how the Kronecker product works and the fact that the null space of the matrix L_0' is empty.

The variants of this term for other combinations of finite-difference matrices that were described in Section 3.2.2 can be obtained in a similar way, but we need to be careful about the corresponding null spaces. When working with the second derivative one can also consider using a discrete variant of the Laplacian operator

$$\mathcal{R}(x) = \|(I_q \otimes L_0'' + L_0'' \otimes I_p)x\|, \tag{3.14}$$

that also has an empty null space. This variant of a regularization matrix is available in the toolbox [10].

We can see the example of using the 2D variants of regularization matrices in Figure 3.4 on the problem from Example 1. The Tichonov problem was again solved by the hybrid LSQR method. We can see that the approximate solution computed using the Standard Tichonov regulariation ($L = I$) is not smooth enough. Better approximation was computed with regularization matrix L_0', the computation with matrix L_0'' terminated after it reached the maximum number of iterations (2000). Note that we only tried the variants of regularization matrices with zero boundary conditions, since the picture has a black background and so the zero boundary condition is a clear choice. The reflective boundary conditions would give the same results for a picture with a black background. The 2D variant of L_0' was computed as the matrix in (3.13) and the 2D variant of L_0'' was implemented as in (3.14).

3.3 The use of different norms

As we already mentioned, we restricted ourselves to study the Tichonov method in the form (3.2). In this section we want to give a brief summary of other variants of regularization terms for Tichonov regularization, since they sometimes offer a more suitable regularization for certain problems [5, Chapter 4.3]. In the end, the choice of a suitable regularization depends on the properties of the particular problem.

In the case where one expects a sparse solution, it can be useful to introduce a regularization term in the form

$$\mathcal{R}(x) = \|x\|_p^p,$$

where $p \in (0, 1)$, see [29], [30].

Other variant is the use of *the total variation*, which takes in the discrete setting the form

$$\mathcal{R}(x) = \|L'x\|_1,$$

as a regularization term. This variant of the regularization term does not penalize steep gradients in the solution as strongly as the 2-norm, therefore it is more suitable when reconstructing vectors that are piecewise smooth, such as bar codes, see [1, Chapter 8.6], [31].

However, note that regularization terms defined by different norms than the 2-norm make the problem more complicated and it is usually necessary to employ different approaches when solving the corresponding General Tichonov problem than the ones introduced in this thesis.

4. Numerical experiments

In this chapter we focus on some numerical experiments. All the computations were performed on a computer Lenovo ThinkPad E14 (Gen 2, Type 20T6). The machine is equipped with AMD Ryzen 7 4700U processor with Radeon Graphics (2.00 GHz) and 8 cores. The size of RAM memory is 16 GB. Matlab R2020B was used with the toolboxes IR Tools [10] and Regularization Tools [11]. The scripts to create the experiments were newly defined as well as the functions for creating most of the regularization matrices, with the exception of discrete representations of Laplacian in 1D and 2D.

Let us denote

$$err_{rel} = \frac{\|\tilde{x} - x_k\|}{\|\tilde{x}\|}$$

the relative true error norm of an approximate x_k for simplicity. Note that $\tilde{x}$ is the exact solution of the problem without noise (1.3).

In the following experiments we often use the function *hybrid_LSQR* from the toolbox [10] with the predefined parameter $option.RegParam =' discrep'$ which specifies that the inner parameter λ should be computed via the Discrepancy principle. Nevertheless the function actually uses a sophisticated way of approximating the discrepancy principle in the hybrid setting, that is based on the secant method, computing the current parameter from the previous one. Therefore one needs to pass on an initial parameter λ, that we hereafter denote λ_0. The details on how the function computes the inner parameter can be found in [10] and [32].

4.1 The dependence of the optimal regularization parameter on the level of noise

In this section we illustrate how the optimal regularization parameter depends on the level of noise μ_e and how well does the Discrepancy principle approximate the optimal value λ. We took the model problem gravity(256,1) from the toolbox [11] and added different levels of noise μ_e. Then we computed the results of the hybrid LSQR method on these variants of the problem for various regularization parameters λ in order to find the optimal one and compared with the results of the hybrid LSQR algorithm with the Discrepancy principle as the parameter choice method in the inner iteration. In the second case for the Discrepancy principle the approximate noise level $\alpha = 1.1 * \mu_e$ was used in order to simulate a 10% error in the estimate of the noise level μ_e, the safety factor $\xi = 1.5$ and $\lambda_0 = 1$ were set.

In Figure 4.1 we can see the results. On the left column there are the approximate solutions for optimal parameter λ. On the right column we plot the results when calling the method with the integrated Discrepancy principle. We can observe that the approximate solutions are better for lower levels of noise μ_e, which illustrates the fact that with higher level of noise the problem becomes harder to solve. It is also visible when we take a look on Table 4.1, where we give relative errors.

The optimal λ for high μ_e is higher, which corresponds to the fact that there

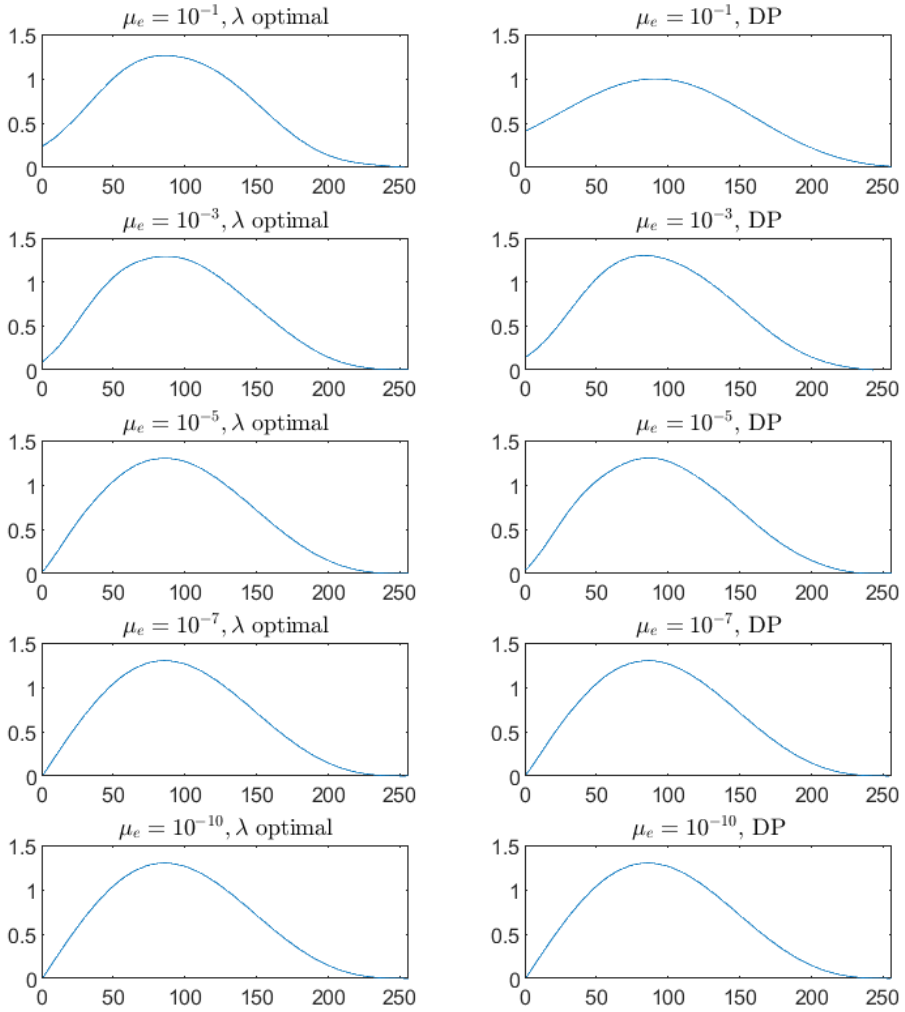

Figure 4.1: The comparison of the approximate solutions computed by the hybrid LSQR algorithm for the test problem gravity(256,1) with different levels of noise μ_e ranging from 10^{-1} to 10^{-10}. The function was called with parameters $\alpha = 1.1 * \mu_e, \xi = 1.5, \lambda_0 = 1$. The results for optimal λ used as a constant are on the left. The results from the adaptive Discrepancy principle in the inner iteration are on the right. We can see that for higher levels of noise it is more difficult to compute a suitable reconstruction of the exact solution. The corresponding relative error norms are stated in Table 4.1.

Table 4.1: The comparison of the quality of solution depending on the level of noise μ_e for the test problem gravity(256,1). Two variants of the hybrid LSQR method were used, first for the optimal λ set as a constant parameter, second for the Discrepancy principle (DP).

μ_e	opt. λ	err_{rel} for opt. λ	λ from the DP	err_{rel} for DP
10^{-1}	0.7	0.0537	2.2555	0.2190
10^{-3}	0.05	0.0154	0.1521	0.0269
10^{-5}	0.003	0.0025	0.0011	0.0076
10^{-7}	0.0001	$4.88 * 10^{-4}$	$4.74 * 10^{-4}$	0.0014
10^{-10}	$5 * 10^{-6}$	$1.01 * 10^{-4}$	$4.90 * 10^{-6}$	$1.19 * 10^{-4}$

is more uncertainty in the problem, therefore there is a greater need to regularize. On the contrary, for very low level of noise μ_e the need of regularization is not so strong and therefore the values of optimal λ are much smaller. Also the relative error err_{rel} is lower, which means that a problem with lower noise level allows us to compute a better approximation of the solution.

Similar trends also apply for λ and corresponding err_{rel} computed using the Discrepancy principle, although the results are generally worse, because the criterion only works with limited information about the noise level μ_e. In the last column of Table 4.1 we can observe that for lower noise levels the criterion is able to approximate the optimal λ very well, but for high levels of noise the quality of the approximations is declining, resulting with higher err_{rel} than in the case with optimal λ. However in Figure 4.1 we can see that the reconstructions are acceptable. Notice that for $\mu_e = 10^{-1}$ it is not possible to capture the zero on the boundary, not even when optimal λ is chosen. It shows that the problem is too spoiled.

4.2 Choosing suitable boundary conditions

The aim of this experiment is to illustrate the possible impacts of using boundary conditions that are not suitable for the specific problem we are solving. We took the test problem PRblur(256,options) from the toolbox [10], where we chose a particular image by setting *options.TrueImage='ppower'*. Then we added noise on the level $\mu_e = 10^{-2}$. Further we computed the hybrid LSQR solutions for different variants of the regularization matrix L and compared the results. In the inner iteration the Discrepancy principle was used with an approximate level of noise $\alpha = 1.1 * \mu_e$ that simulates the error in the noise estimate, the safety factor $\xi = 1.5$ and initial $\lambda_0 = 1$.

Figure 4.2 shows the computed approximate solutions. In the cases where the zero boundary condition was chosen we can observe that the light parts of the image are disturbed by some black shadows. We do not see this problem in the restorations where the reflective boundary condition was used. Also when we look at the relative error norms err_{rel} in Table 4.2, we can see that the reflective boundary condition allowed us to compute a solution that was closer to the exact solution $\tilde{x}$ of the problem (1.3) without noise.

It follows from the observations that it is very important to choose the boun-

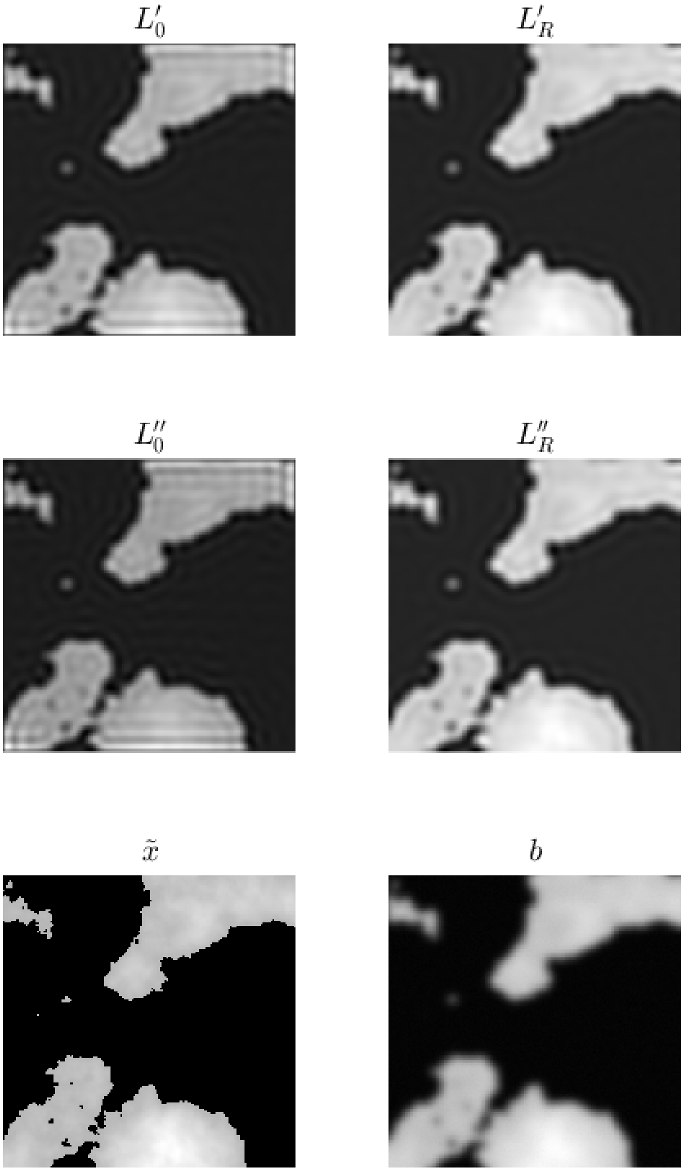

Figure 4.2: The comparison of the exact solution $\tilde{x}$ (bottom left), the blurred and noisy b (bottom right) and the solutions of hybrid LSQR for different variants of the regularization matrix L. We can see that the approximate solutions on the left, computed using regularization matrix with zero boundary conditions, offer worse approximations than the ones on the right, where L with reflective boundary conditions was used. The light spots in the image are disturbed by the black color that corresponds to the zero value. It is obvious that the reflective boundary condition is more suitable to restore the light parts of the tested image.

Table 4.2: The comparison of the relative error norms err_{rel} and the computed values of the regularization parameter λ of the approximate solutions from the hybrid LSQR method with the for zero and reflective boundary conditions.

	zero BC		reflective BC	
	λ from the DP	err_{rel}	λ from the DP	err_{rel}
L'	0.1053	0.1972	0.2724	0.1768
L''	0.1356	0.2047	0.6545	0.1766

dary conditions wisely, considering the information we have about the solution that is to be reconstructed. Notice that for the correct boundary condition the relative error norm of the reconstruction with L''_R was lower than the relative error norm for L'_R. Also the parameter λ is larger for L''_R than for L'_R, which is not surprising, since we absorbed the constants depending on h from the discrete derivatives into the parameter λ, see Section 3.2.2.

4.3 Centered and forward difference for the first derivative in 1D

When we introduced the concept of finite differences in Section 3.2.1, we also derived that the centered difference has higher order of precision than the forward difference for the first derivative. One might ask a question whether the schemes with higher orders of precision will offer better approximate solutions in the hybrid LSQR setting. Therefore we compute and compare the results of hybrid LSQR when using the regularization matrices L'_0 and $L'_{0,cent}$ to verify this statement in the case of the first derivative. Note that from the comparison of finite difference schemes (3.7) and (3.9) it follows that to perform a fair comparison, one should set λ for use with $L'_{0,cent}$ to be half of the λ for use with L'_0. For this reason we introduce an auxiliary coefficient τ and then compare the results computed for $\lambda = \tau$ and $\lambda = \tau/2$, respectively.

As a test problem we chose gravity(256,1) from the toolbox [11], with noise added on the level $\mu_e = 10^{-3}$. We computed the results of hybrid LSQR for three values of the parameter τ.

The results of the computations for $\tau = 0.1$, $\tau = 0.05$ and $\tau = 0.01$ can be seen in Figures 4.3 and 4.4. In all of the cases the centered difference $L'_{0,cent}$ brings worse approximation than the forward difference L'_0. We can see that all the solutions for $L'_{0,cent}$ suffer from oscillations. The conclusions from the Figures are also supported by the comparison of relative error norms in Table 4.3. The table shows that for all three variants of τ the error for centered difference $L'_{0,cent}$ is higher than the error for L'_0.

The experiment did not fulfill our expectations on the effect of the order of precision of the finite difference on the quality of the solution. The effect of the centered difference matrix $L'_{0,cent}$ to bring oscillations to the solution might be hidden in the shape of the centered difference. When we look at the shape of $L'_{0,cent}$ (see Section 3.2.2) we may realize that multiplication by this matrix compares the even and the odd components of the solution separately. When we take a closer look at the approximate solutions, they oscillate between each neighbor-

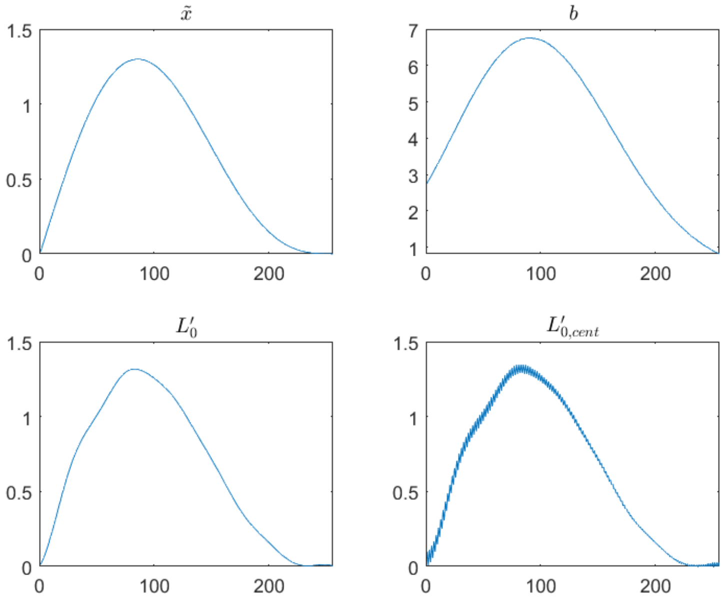

Figure 4.3: The comparison of the results of hybrid LSQR when using forward and backward difference schemes to approximate the first derivative. The solutions were computed with constant λ computed from the constant $\tau = 0.1$ according to Table 4.3. We can see that the solution for $L'_{0,cent}$ is oscillating. On the top we can see the precise $\tilde{x}$ and the right hand side b.

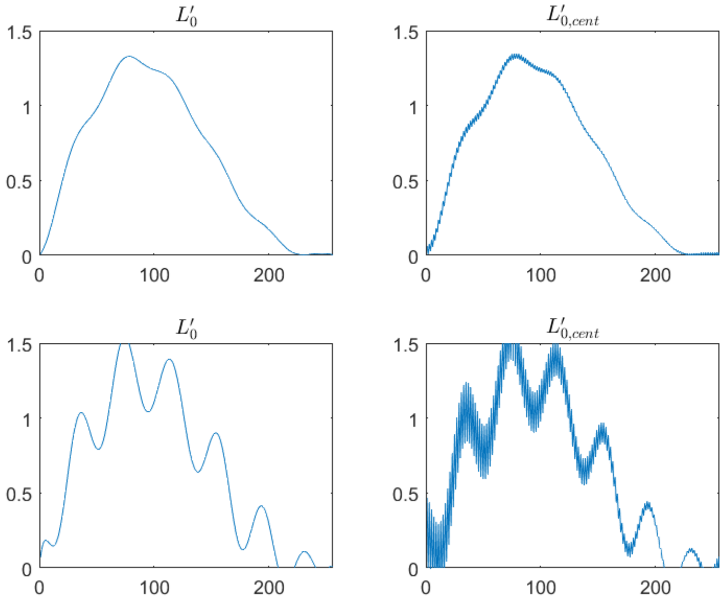

Figure 4.4: The same comparison as in Figure 4.3, but for the values $\tau = 0.05$ (in the first row) and $\tau = 0.01$ (in the second row). We observe similar results.

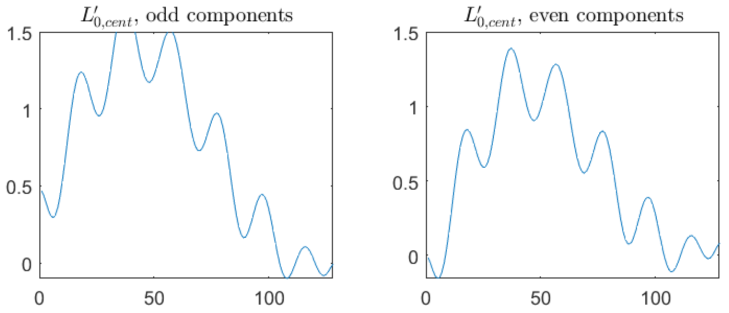

Figure 4.5: The odd and even components of the solution for $\tau = 0.01$ with $L'_{0,cent}$ from the figure 4.7. We can see that both the odd and even components form a vector that does not suffer from oscillations.

Table 4.3: The comparison of the relative error norms err_{rel} for more variants of the parameter τ and two regularization matrices L'_0 and $L'_{0,cent}$.

		$\tau = 0.1$	$\tau = 0.05$	$\tau = 0.01$
		err_{rel}	err_{rel}	err_{rel}
L'_0	$\lambda = \tau$	0.0196	0.0338	0.1998
$L'_{0,cent}$	$\lambda = \tau/2$	0.0380	0.0404	0.2638

hood component, as we can see in Figure 4.5, which supports our presumption. In the next section we offer a second experiment with similar goal that studies this question on a 2D example.

4.4 Centered and forward difference for the first derivative in 2D

In this experiment we want to study, whether the central difference provides better approximate solutions in 2D than we observed in the previous section in 1D. Since in 2D the regularization term requires comparing the components of the approximation in two directions, we expect that the oscillations will not appear here. Again we used different parameters λ to ensure a fair comparison.

We use the problem PRblur(256, options) with options.TrueImage='hst' and we added noise at the level $\mu_e = 5 * 10^{-3}$. For this problem we computed a hybrid LSQR solution for both 2D versions of L'_0 and $L'_{0,cent}$ for three variants of the auxiliary coefficient τ.

Table 4.4: The comparison of the relative error norms err_{rel} for more variants of the parameter τ and two regularization matrices L'_0 and $L'_{0,cent}$.

		$\tau = 1.5$	$\tau = 1.0$	$\tau = 0.1$	$\tau = 0.01$
		err_{rel}	err_{rel}	err_{rel}	err_{rel}
L'_0	$\lambda = \tau$	0.2701	0.2588	0.2182	0.2020
$L'_{0,cent}$	$\lambda = \tau/2$	0.2699	0.2586	0.2179	0.2052

The approximate solutions can be seen in Figures 4.6 and 4.7. We do not see

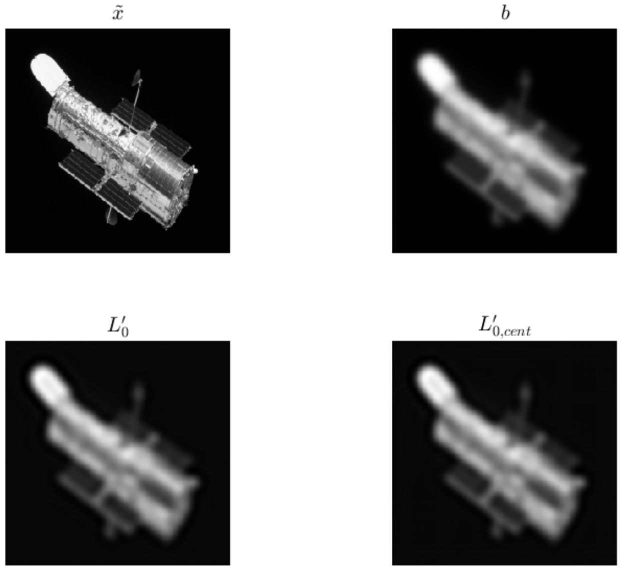

Figure 4.6: The comparison of the results of hybrid LSQR when using forward and backward difference schemes for approximating the first derivative. The solutions were computed with the constant λ set according to Table 4.4 with $\tau = 1$. The reconstructions for both matrices L_0' and $L_{0,cent}'$ are similar, the solution for the central derivative has slightly lower relative error norm as we can see in Table 4.4. In the top row we can see the precise image $\tilde{x}$ and the right hand side b.

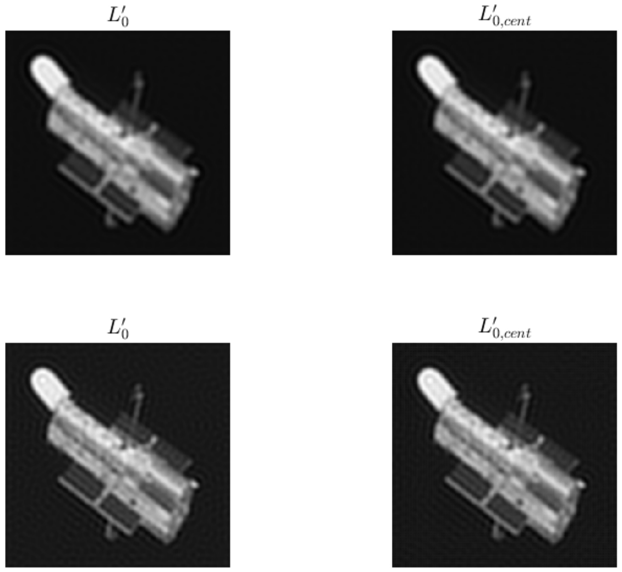

Figure 4.7: The same comparison as in Figure 4.6, but for the values $\tau = 0.1$ (in the first row) and $\tau = 0.01$ (in the second row). Again, the reconstructions are very similar. The relative error norm for $\tau = 0.1$ is slightly lower for $L_{0,cent}'$, the result for $\tau = 0.01$ is opposite, as we can see in Table 4.4.

any oscillations as in the 1D case. In the cases $\tau = 1.5$, $\tau = 1$ and $\tau = 0.1$ the relative error norm was lower when we used $L'_{0,cent}$, in the last case the relative error norm was lower for L'_0, see Table 4.4. Note that $\tau = 1.0$ is too large and thus the solutions in Figure 4.6 are overregularized. The reconstruction for $\tau = 0.1$ is visually favourable.

The results suggest that in some cases the central difference approximation of the first derivative in 2D can lead to better results than the forward difference approximation. In particular, in cases where more regularization is needed, i.e., λ is larger.

4.5 Higher order scheme for the second derivative

Based on the observations in the previous experiment we decided to take a look at a similar comparison for the second derivative. We tested the schemes of the second and the fourth order of precision. The first was introduced in Section 3.2.1, the second takes the form

$$f''(t) \approx \frac{-f(t+2h) + 16f(t+h) - 30f(t) + 16f(t-h) - f(t-2h)}{12h^2}. \tag{4.1}$$

The corresponding matrix L''_0 (4th order) can be derived similarly as in Section 3.2.2.

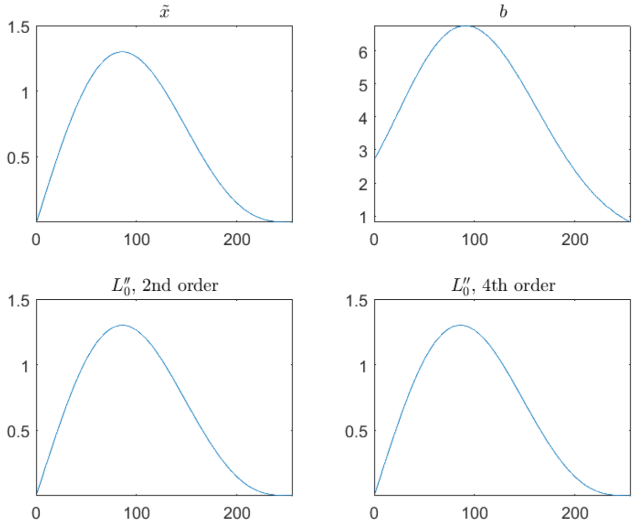

Figure 4.8: The comparison of the results of hybrid LSQR for the problem gravity(256,1) with noise added at the level $\mu_e = 10^{-3}$ when using the finite difference schemes for second derivative of the second and the fourth order of precision. The optimal parameter λ was chosen in each case separately, see the values in Table 4.5. The reconstructions for both schemes look similar, the solution for L''_0 (4th order) has slightly lower relative error norm. In the top row we can see the precise vector $\tilde{x}$ and the right hand side b.

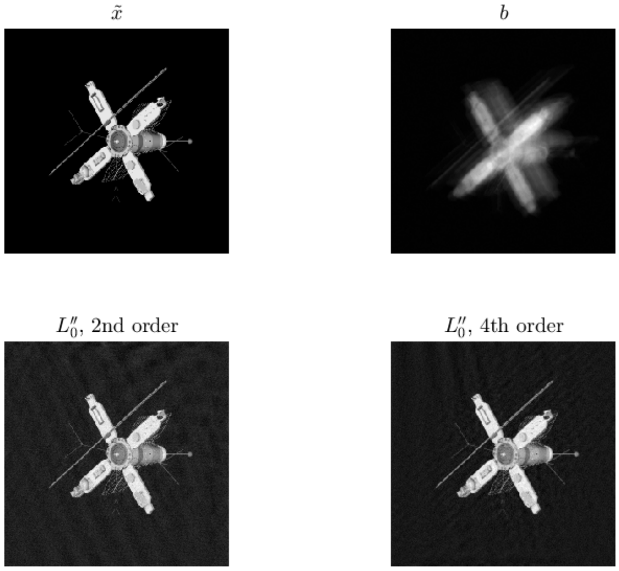

Figure 4.9: The same comparison as in Figure 4.8, but for the test problem from Example 1. Again, the values of λ can be seen in Table 4.5. The reconstruction computed with L_0'' (4th order) is visually better than the one computed with L_0'' (2nd order). This observation corresponds to the comparison of the relative error norms in Table 4.5.

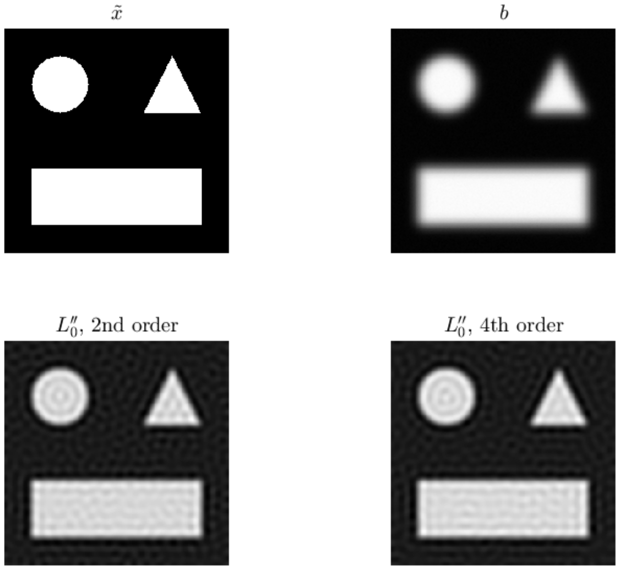

Figure 4.10: The same comparison as in Figure 4.8, but for the test problem from Example 2 with the noise level changed to $\mu_e = 10^{-2}$. The values of λ can be seen in Table 4.5. The reconstructions for both schemes look similar. The solution for L_0'' (2nd order) has slightly lower relative error norm, see Table 4.5.

We use three test problems. One is gravity(256,1) from the toolbox [11] with noise added at the level $\mu_e = 10^{-3}$, second is the problem from Example 1, third is based on Example 2, but with the noise level changed to $\mu_e = 10^{-2}$. The solutions were computed by the hybrid LSQR with constant λ. The parameter was chosen manually for the particular cases by computing the solutions for many variants of λ and choosing the optimal one. This way we could perform a fair comparison.

Table 4.5: The comparison of the relative error norms err_{rel} for two regularization schemes L_0'' of 2nd and 4th order for three test problems. The optimal parameter λ was chosen in the particular cases.

	gravity		PR1		PR2, $\mu_e = 10^{-2}$	
	opt. λ	err_{rel}	opt. λ	err_{rel}	opt. opt. λ	err_{rel}
L_0'', 2nd order	21	0.0026	$5 * 10^{-3}$	0.1538	$4 * 10^{-2}$	0.1812
L_0'', 4th order	2	0.0023	$6 * 10^{-4}$	0.1262	$4 * 10^{-3}$	0.1814

The results are visualized in Figures 4.8, 4.9 and 4.10. For the first two examples the reconstructions computed with the scheme with the fourth order of precision have lower relative error norms, in the third case it is the opposite, see Table 4.5. In Figure 4.9 we can see that the variant with L_0'' (4th order) led to visibly better restoration. Notice that the proportion of the optimal parameters λ for the two finite difference schemes is of the same order as the proportion of the constants in (3.12) and (4.1) that were absorbed to λ.

As in the previous section, the results suggest that the higher order schemes for the second derivative can lead to better approximations in some cases. A possible problem is the choice of a suitable regularization parameter, because the interval for the optimal λ differs from scheme to scheme.

Conclusion

The aim of this thesis was to study the behaviour of hybrid regularization methods combining Krylov subspace iterations and the Tichonov regularization with general regularization terms. We described inverse problems and illustrated the need to regularize. We proved the equivalence of the two approaches first regularize, then project and first project, then regularize in the hybrid setting. We focused on the Tichonov regularization with general regularization terms, summarized their well known variants based on finite differences and introduced variants with higher orders of precision for various boundary conditions.

Further on, we performed experiments using the hybrid method combining LSQR and Tichonov minimization. We illustrated the well known dependence of the regularization parameter on the level of noise and the importance of appropriate choice of the boundary conditions. We observe that an optimal parameter λ for the forward scheme for the second derivative is higher than for the first derivative, which corresponds to the constants from finite difference schemes that we absorbed to λ.

Then we studied, whether the centered difference for approximating the first derivative with a higher order of precision can lead to better results in the hybrid LSQR than the widespread forward difference. The results in 1D show that the approximate solutions computed with the centered difference slightly oscillate, resulting in larger relative true error. This can be caused by the shape of the centered difference matrix $L'_{0,cent}$. It can be seen that the regularization term with the centered difference measures the relations between the odd and the even components separately and does not relate these two groups of components at all.

For 2D problems oscillations do not appear. We suppose that this is caused by the fact that each component of the approximate solution is compared with other components in two space directions, which puts more requirements on the particular component compared to the 1D case. Here the centered difference leads to better approximations in some cases. Note that the use of the centered difference does not increase the computational costs of the regularization process, since the corresponding finite difference matrix is sparse as is the matrix corresponding to the forward difference.

Experiments with the finite difference scheme of the fourth order of precision for the second derivative in 1D and 2D were also performed. In some cases, it resulted in better approximations of the solution compared to the finite difference scheme of the second order of precision. Nevertheless, we observed that the choice of an optimal regularization parameter is more difficult with the higher order schemes.

In summary, particular choice of appropriate regularization term is strongly problem dependent.

Bibliography

[1] P. Ch. Hansen. *Discrete Inverse Problems: Insight and Algorithms*, volume 7 of *Fundamentals of Algorithms*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2010.

[2] P. Ch. Hansen. *Rank-Deficient and Discrete Ill-Posed Problems*. SIAM Monographs on Mathematical Modeling and Computation. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1998. Numerical aspects of linear inversion.

[3] P. Ch. Hansen and D. P. O'Leary. The Use of the L-Curve in the Regularization of Discrete Ill-Posed Problems. *SIAM Journal on Scientific Computing*, 14(6):1487–1503, 1993.

[4] C. R. Vogel. *Computational Methods for Inverse Problems*, volume 23 of *Frontiers in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2002.

[5] J. Chung and S. Gazzola. Computational Methods for Large-Scale Inverse Problems: A Survey on Hybrid Projection Methods. Version: August, 2021, forthcoming.

[6] A. N. Tikhonov. Solution of Incorrectly Formulated Problems and the Regularization Method. *Soviet Mathematics Doklady*, 4:1035–1038, 1963.

[7] A. N. Tikhonov and V. Y. Arsenin. *Solutions of Ill-Posed Problems*. V. H. Winston & Sons, Washington, D.C.: John Wiley & Sons, New York, 1977. Translated from the Russian, Preface by translation editor Fritz John, Scripta Series in Mathematics.

[8] M. Hanke. *A Taste of Inverse Problems - Basic Theory and Examples*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2017.

[9] L. Eldén. Algorithms for the Regularization of Ill-Conditioned Least Squares Problems. *BIT Numerical Mathematics*, 17(2):134–145, 1977.

[10] S. Gazzola, P. Ch. Hansen, and J. G. Nagy. IR Tools: A MATLAB Package of Iterative Regularization Methods and Large-Scale Test Problems. *Numerical Algorithms*, 81(3):773–811, jul 2019.

[11] P. Ch. Hansen. Regularization Tools: A MATLAB Package for Analysis and Solution of Discrete Ill-Posed Problems. *Numerical Algorithms*, 1994.

[12] P. Ch. Hansen. *Deblurring Images: Matrices, Spectra, and Filtering*. Fundamentals of algorithms. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2006.

[13] J. D. Tebbens, I. Hnětynková, M. Plešinger, Z. Strakoš, and P. Tichý. *Analýza Metod pro Maticové Výpočty: Základní Metody*. Matfyzpress, Praha, 2012.

[14] G. H. Golub and Ch. F. Van Loan. *Matrix Computations*. Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press, 1996.

[15] G. H. Golub, M. Heath, and G. Wahba. Generalized Cross-Validation as a Method for Choosing a Good Ridge Parameter. *Technometrics*, 21(2):215–223, 1979.

[16] R. Ramlau. A Steepest Descent Algorithm for the Global Minimization of the Tikhonov Functional. *Inverse Problems*, 18(2):381–403, 2002.

[17] R Griesse and D A Lorenz. A Semismooth Newton Method for Tikhonov Functionals with Sparsity Constraints. *Inverse Problems*, 24(3):035007, 2008.

[18] Å. Björck. *Numerical Methods for Least Squares Problems*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1996.

[19] M. R. Hestenes and E. Stiefel. Methods of Conjugate Gradients for Solving Linear Systems. *J Res NIST*, 49(6):409–436, 1952.

[20] 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(4):255–282, 1950.

[21] G. Golub and W. Kahan. Calculating the Singular Values and Pseudo-Inverse of a Matrix. *Journal of the Society for Industrial and Applied Mathematics: Series B, Numerical Analysis*, 2(2):205–224, 1965.

[22] J. Liesen and Z. Strakoš. *Krylov Subspace Methods: Principles and Analysis*. Numerical Mathematics and Scientific Computation. Oxford University Press, 2012.

[23] L. S. Borges, F. S.V Bazán, and L. Bedin. A Projection-Based Algorithm for $\ell 2 - \ell p$ Tikhonov Regularization. *Mathematical Methods in the Applied Sciences*, 41(15):5919–5938, 2018.

[24] L. Barto and J. Tůma. Lineární Algebra. `https://kam.mff.cuni.cz/~mikina/materialy/skripta_la5.pdf`. Accessed: May, 2022.

[25] Ch. Paige and M. Saunders. LSQR: An Algorithm for Sparse Linear Equations and Sparse Least Squares. *ACM Transactions on Mathematical Software*, 8(1):43–71, 1982.

[26] R. A Renaut, M. Horst, Y. Wang, D. Cochran, and J. Hansen. Efficient Estimation of Regularization Parameters via Downsampling and the Singular Value Expansion: Downsampling Regularization Parameter Estimation. *BIT Numerical Mathematics*, 57(2):499–529, 2016.

[27] J. Cullum. The Effective Choice of the Smoothing Norm in Regularization. *Mathematics of Computation*, 33(145):149–170, 1979.

[28] Ch. Jordan. *Calculus of Finite Differences*. Chelsea Publishing Company, New York, 3rd edition, 1965.

[29] G. Huang, A. Lanza, S. Morigi, L. Reichel, and F. Sgallari. Majorization-Minimization Generalized Krylov Subspace Methods for ℓ_p–ℓ_q Optimization Applied to Image Restoration. *BIT Numerical Mathematics*, 57(2):351–378, 2017.

[30] A. Lanza, S. Morigi, L. Reichel, and F. Sgallari. A Generalized Krylov Subspace Method for ℓ_p-ℓ_q Minimization. *SIAM Journal on Scientific Computing*, 37(5):S30–S50, 2015.

[31] S. Gazzola and M. Sabaté Landman. Flexible GMRES for Total Variation Regularization. *BIT Numerical Mathematics*, 59(3):721–746, 2019.

[32] S. Gazzola and P. Novati. Automatic Parameter Setting for Arnoldi-Tikhonov Methods. *Journal of Computational and Applied Mathematics*, 256:180–195, 2014.