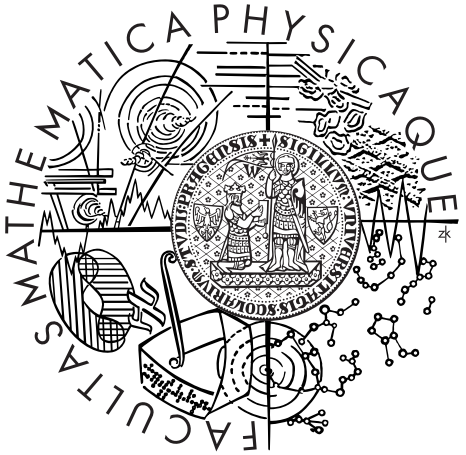
Univerzita Karlova v Praze
Matematicko-fyzikální fakulta

DIPLOMOVÁ PRÁCE



Marie Michenková

Regularizační metody založené na metodách nejmenších čtverců

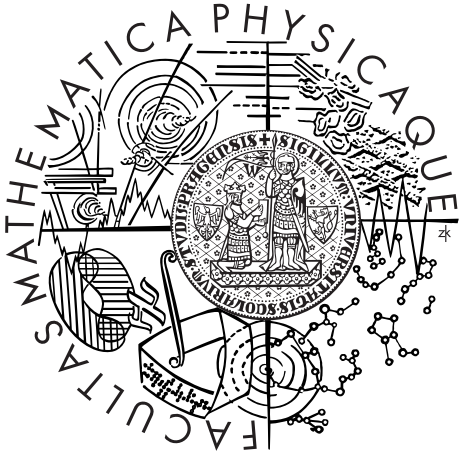
Katedra numerické matematiky

Vedoucí diplomové práce:	RNDr. Iveta Hnětynková, Ph.D.
Studijní program:	Matematika
Studijní obor:	Numerická a výpočtová matematika

Praha 2013

Charles University in Prague
Faculty of Mathematics and Physics

MASTER THESIS



Marie Michenková

Regularization Techniques Based on the Least Squares Method

Department of Numerical Mathematics

Supervisor of the master thesis: RNDr. Iveta Hnětynková, Ph.D.
Study programme: Mathematics
Specialization: Numerical and Computational Mathematics

Prague 2013

I would like to express my deepest gratitude to Dr. Iveta Hnětynková. Without her guidance and persistent help, this thesis would not have been possible.

I thank my family, in particular my parents, for their support. Special thanks go to my fiancé Petr.

I declare that I carried out this master thesis independently, and only with the cited sources, literature and other professional sources.

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

In Prague, April 11, 2013

Marie Michenková

Název práce: Regularizační metody založené na metodách nejmenších čtverců

Autor: Marie Michenková

Katedra: Katedra numerické matematiky

Vedoucí diplomové práce: RNDr. Iveta Hnětynková, Ph.D.

Abstrakt: V této práci se zabýváme lineárními inverzními problémy $Ax \approx b$, kde A je zhlazující lineární opearátor a b reprezentuje vektor pozorování zatížený neznámým šumem. V práci [Hnětynková, Plešinger, Strakoš, 2009] bylo ukázáno, že vysokofrekvenční šum se během Golubovy-Kahanovy iterační bidiagonalizace vyjevuje v levých bidiagonalizačních vektorech. V práci navrhujeme metodu, která identifikuje iteraci s maximálním vyjevením šumu a redukuje vysokofrekvenční šum odečtením příslušného (škálovaného) bidiagonalizačního vektoru od vektoru b. Tato metoda je následně testována pro různé typy šumu. Dále Hnětynková, Plešinger a Strakoš odvodili metodu k odhadování hladiny šumu v datech. V práci navrhujeme modifikaci této metody založenou na znalosti bodu maximalního vyjevení šumu.

Klíčová slova: ill-posed problémy, regularizace, Golubova-Kahanova iterační bidiagonalizace, vyjevení šumu, odhad šumu, odšumování

Title: Regularization Techniques Based on the Least Squares Method

Author: Marie Michenková

Department: Department of Numerical Mathematics

Supervisor: RNDr. Iveta Hnětynková, Ph.D.

Abstract: In this thesis we consider a linear inverse problem $Ax \approx b$, where A is a linear operator with smoothing property and b represents an observation vector polluted by unknown noise. It was shown in [Hnětynková, Plešinger, Strakoš, 2009] that high-frequency noise reveals during the Golub-Kahan iterative bidiagonalization in the left bidiagonalization vectors. We propose a method that identifies the iteration with maximal noise revealing and reduces a portion of high-frequency noise in the data by subtracting the corresponding (properly scaled) left bidiagonalization vector from b. This method is tested for different types of noise. Further, Hnětynková, Plešinger, and Strakoš provided an estimator of the noise level in the data. We propose a modification of this estimator based on the knowledge of the point of noise revealing.

Keywords: ill-posed problems, regularization, Golub-Kahan iterative bidiagonalization, noise revealing, noise estimate, denoising

Contents

List of Symbols

$\mathbb{R}$	set of real numbers
$\mathbb{R}^{m\times n}$	set of real matrices of size $m \times n$
$(\cdot,\cdot)$	inner product
$\|\cdot\|_2$, $\|\cdot\|$	Euclidian norm
$\|\cdot\|_F$	Frobenius norm
$\mathrm{span}(\cdots)$	subspace spanned by vectors
$\|\cdot\|$	absolute value of a
$\mathrm{diag}(b)$, $\mathrm{diag}(b_i)$	square matrix with entries b_i on its diagonal
A	coefficient matrix
A^T	transpose of A
$A^\dagger$	Moore-Penrose pseudoinverse of A
A^{-1}	inverse of A
$A(i,:)$	i-th row of A
b	right-hand side
b^{noise}	noise component in the right-hand side
b^{exact}	exact component in the right-hand side
e_i	i-th column of identity matrix
f	solution function in integral equations
g	data function in integral equations
I	identity matrix
K	kernel in integral equations
$k_{noise}+1$	noise revealing iteration
L_k	lower-bidiagonal matrix of size $k \times k$
L_{k+}	lower-bidiagonal matrix of size $(k+1) \times k$
m, n	matrix dimensions, $m \geq n$
$p_1^{(k)}$	left singular vector of L_k corresponding to the smallest singular value
s, t	independent variables in integral equations
s_i	i-th left bidiagonalization vector
S	matrix of right bidiagonalization vectors
u_i	i-th left singular vector/function
U	matrix of left singular vectors
v_i	i-th right singular vector/function
V	matrix of right singular vectors
w_i	i-th basis vector of projection method i-th right bidiagonalization vector

W_i	matrix of basis vectors
	matrix of right bidiagonalization vectors
x^{naive}	naive solution
x^{T-SVD}, x^k	T-SVD solution
x^{Tikh}, x^λ	Tikhonov solution
y_l	solution to projected problem
α_i	i-th diagonal entry of L_k
β_i	i-th subdiagonal entry of L_k
μ_i	singular value of kernel
Σ	diagonal matrix with singular values
σ_i	singular value of matrix
ω_i	quadrature weights
$\mathcal{N}(0, \sigma^2 I)$	normal distribution with zero mean and variance σ^2
$\mathcal{U}(a, b)$	uniform distribution in interval (a, b)
$\mathrm{Cov}(\cdot)$	covariance matrix

Preface

Consider a linear inverse problem

$$Ax \approx b\,, \qquad A \in \mathbb{R}^{n\times n}\,, \qquad b = b^{exact} + b^{noise} \in \mathbb{R}^n\,, \tag{1}$$

where A represents a discrete model (smoothing operator), b represents an observation vector, and b^{noise} represents unknown perturbations in the data, usually denoted as *noise*. Given A and b, the aim is to compute a numerical approximation of the exact solution x^{exact},

$$Ax^{exact} = b^{exact}\,.$$

Inverse problems of the form (1) arise in many fields of application, e.g. signal and image processing, geophysics, seismology, etc. [Han98, Vog02, Han10].

Typically, these problems are *ill-posed*, meaning that a small perturbation in the data may cause significant errors in computed approximations of x^{exact}. The ill-posed nature of the problem is revealed by the singular values of A, which decay gradually without a noticeable gap. Thus A is severely ill-conditioned, and a form of *regularization* is necessary to compute a stable approximation of x^{exact} [Eld77, Han98, EHN00, Vog02]. Regularization can take many forms, but target of all of them is to preserve sufficient information about the exact solution, while suppressing the influence of noise. The most well known regularization methods are the Tikhonov's regularization [Tik63, Gro90] and the truncated SVD [Han71, Var73, Han87] belonging to spectral filtering methods. However, they are usually confined to smaller problems, because they involve computing the (partial) SVD of A. An alternative is iterative regularization, also called projection regularization. Iterative regularization is often based on Krylov subspaces, where regularization is achieved via projection onto a Krylov subspace of smaller dimension. Such methods are for example LSQR [PS82b, Bjö88], CGLS [BES98] or CGNE [Cra55, FF63]. Hybrid methods combine both types of regularization. First, the original problem is projected onto a Krylov subspace, and subsequently, the projected problem is further regularized using spectral filtering [Bjö88].

Amount of regularization in every method is controlled by a regularization parameter and no regularization method is effective without an appropriate choice of this parameter. Methods for choosing regularization parameters can be divided into two groups: methods based on a priori knowledge of the noise level in the data, e.g. the discrepancy principle [Mor66, Mor84], and methods that work without this a priori information, e.g. the L-curve [HO93] or the generalized cross validation [GHW79].

In this thesis, we focus on the Golub-Kahan iterative bidiagonalization, which is a core of the LSQR method. In [HPS09], it was shown, how white noise

propagates through the iterative bidiagonalization. At some stage, a so-called noise revealing appears. At this point the corresponding left bidiagonalization vector becomes dominated by white noise. Identifying the point of noise revealing allows determining the noise level in the data.

The goal of this thesis is to derive a method to eliminate high-frequency noise in the data using the results of [HPS09]. The idea is to determine the iteration in which white noise reveals in the left bidiagonalization vector most significantly. Subsequently, we subtract this vector (properly scaled) from the noisy data b. The method is computationally verified on problems from Regularization Tools [Han07]. In the remainder of the thesis, we investigate how a change in spectral properties of noise influences the revealing of noise to verify the robustness of the proposed method.

The thesis is organized as follows. Chapter 1 brings an introduction into discrete inverse problems, and an overview of standard regularization methods. In chapter 2, the Golub-Kahan iterative bidiagonalization is recalled and a summary of the results of [HPS09] is given. In chapter 3, a method eliminating the troublesome high-frequency noise in the data is proposed and tested on problems with white noise. Chapter 4 studies, whether and how our method is affected when passing from white to a different type of noise. In chapter 5, a method possibly reducing the computational cost of the noise level estimate proposed in [HPS09] is derived. Conclusion summarizes the main ideas of the thesis and formulates open questions.

Chapter 1

Theoretical Background

This whole thesis deals with inverse problems, therefore it would be inappropriate to skip the definition of this class of problems. In mathematics, we are usually concerned with two general types of problems - *forward problems* and *inverse problems*. The forward problem is to compute the output, given a system and the input to this system, see the scheme 1.1. The inverse problem is to compute either the input or the system, given the other two quantities, see the scheme 1.2. Inverse problems generally arise when we wish to reconstruct the hidden data from some accessible outer measurements.

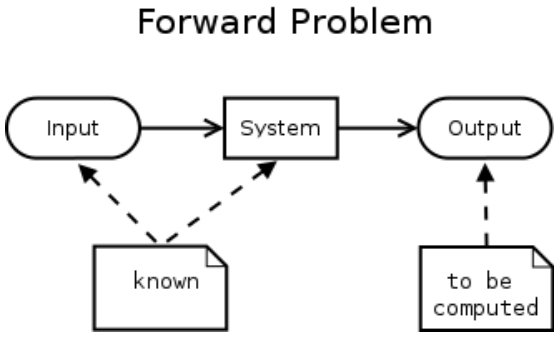


Figure 1.1: Froward problem.

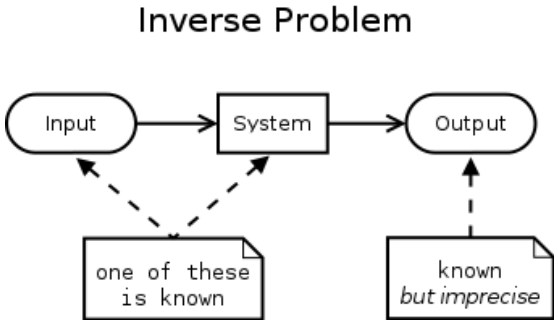


Figure 1.2: Inverse problem.

1.1 Ill-posed problems

In connection with inverse problems, also the definition of *well-posedness* of the problem should be mentioned. In 1902, Jacques Hadamard in [Had02] introduced the following definition.

Definition 1.1 (Hadamard[1])**.** *Mathematical models of physical phenomena are* well-posed *if the following three requirements are satisfied:*

- *A solution exists.*
- *The solution is unique.*
- *The solution's behaviour hardly changes, when there is a slight change in the initial condition.*

Mathematical problems that are not well-posed are called *ill-posed*. Alternatively, ill-posed problems are problems that violate one or more requirements in definition 1.1. Inverse problems are often ill-posed. A typical example of ill-posed problem is the inverse heat equation, deducing a previous distribution of temperature from final data, of which the solution is highly sensitive to changes in the final data.

If the problem is well-posed, then it stands a good chance of solution on a computer using a stable algorithm. If it is not well-posed, it needs to be reformulated for numerical treatment. For illustrative purposes, we will mention the most common ways of re-formulating, see [Han10] for more detailed discussion. The reformulations will be illustrated on systems of linear equations, for simplicity.

The *existence* of the solution can be enforced by weakening the requirements on the solution. Consider the following overdetermined system of linear equations

$$\begin{pmatrix} 1 \\ 2 \end{pmatrix} x = \begin{pmatrix} 1 \\ 3 \end{pmatrix}.$$

Obviously, there is no x, such that $1x = 1$ and $2x = 3$. However, we can reformulate the problem into the following associated least squares problem [Leg05].

$$\min_x \left\| \begin{pmatrix} 1 \\ 2 \end{pmatrix} x - \begin{pmatrix} 1 \\ 3 \end{pmatrix} \right\|_2.$$

This problem already has the solution $x = 1.4$.

The *uniqueness* can be most in cases fixed by adding additional requirements on the solution. Consider the following under-determined system of linear equations

$$(1 \;\; 1)\, x = 1.$$

Obviously, there are infinitely many xs, such that $x_1 + x_2 = 1$. However, requiring that the norm $\|x\|_2$ is minimal, there is a unique solution $x = (0.5,\ 0.5)^T$.

The *stability* condition is more tricky and can be usually achieved by some type of regularization.

[1] adopted from Wikipedia, the free encyclopedia, `http://en.wikipedia.org/wiki/Well-posed`

First, we will consider an inverse problem where a slight perturbation of the system causes a significant perturbation of the solution. Consider the following system of linear equations

$$\begin{pmatrix} 1 & 1 \\ 0 & \mu \end{pmatrix} x = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \text{ where } \mu > 0.$$

It is easy to show that the system has a unique solution $x = (1 - \mu^{-1}, \mu^{-1})^T$. Although the solution is unique, a small perturbation of μ may lead to a huge perturbation in the solution if μ is small. This is caused by the fact that for small μ, the system is close to the overdetermined system

$$\begin{pmatrix} 1 \\ 0 \end{pmatrix} x = \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

Substituting the original problem by its associated least squares problem and adding an extra condition bounding the norm of the solution re-formulates the original problem into a more stable one

$$\min_x \left\| \begin{pmatrix} 1 & 1 \\ 0 & \mu \end{pmatrix} x - \begin{pmatrix} 1 \\ 1 \end{pmatrix} \right\|_2, \text{ such that } \|x\| \leq c.$$

Now, we will consider an inverse problem where a slight perturbation of the right-hand side of the equation causes a significant perturbation of the solution. Consider the following system of linear equations

$$\begin{pmatrix} 1 & 2 \\ 2 & 4.1 \end{pmatrix} x = \begin{pmatrix} 3 \\ 6.1 \end{pmatrix}.$$

It is easy to show that the system has a unique solution $x = (1,\ 1)^T$. Changing the right-hand side slightly by adding the vector $(0,\ 0.1)^T$, we get the system

$$\begin{pmatrix} 1 & 2 \\ 2 & 4.1 \end{pmatrix} x = \begin{pmatrix} 3 \\ 6.2 \end{pmatrix}.$$

This system has a unique solution equal to $x = (-1,\ 2)^T$. Despite the small perturbation in the data, we obtained a solution far from the original one. This type of instability is caused by the properties of the system matrix, which is close to singular. It is not difficult to show that adding the vector $(-2,\ 1)^T$ to x changes the right-hand side only by $(0,\ 0.1)^T$, i.e. $(-2,\ 1)^T$ is 'almost' a null vector of the given matrix. There are more ways, how to introduce more stability into this type of problems. The most important ones will be described later in this chapter.

The following sections will cover fundamentals of solving discrete ill-posed problems resulting from discretization of Fredholm integral equations. This part will be mainly based on books [Han10, Han98, HNO06]. Although the author considers only problems based on Fredholm integral equations, the regularization techniques can be (after some adjustments) used for a broad range of discrete ill-posed inverse problems. We consider all functions and quantities to be real to keep the notation simple, but the theory is mostly extendible to the complex domain. Unless stated otherwise, we assume exact arithmetic.

1.2 Fredholm integral equations

In mathematics, the Fredholm integral equation is an integral equation of which solution gives rise to Fredholm theory [Fre03], the study of Fredholm kernels and Fredholm operators. Fredholm integral equation of the first kind is of the following form

$$g(s) = \int_{I_t} K(s,t) f(t) \mathrm{d}t, \tag{1.1}$$

and the aim is, given the *continuous kernel* $K(s,t)$ and the *data* - function $g(s)$, to find the source - function $f(t)$.

There also exists Fredholm integral equation of the second kind, which has the form of

$$\phi(s) = f(s) + \lambda \int_{I_t} K(s,t)\phi(t)\mathrm{d}t,$$

but these equations are not considered further in this thesis.

Fredholm equations arise naturally in the theory of signal processing - statistical signal processing, image processing and image deblurring (including medical imaging), seismic signal processing, data mining, etc. For illustrative purposes, we will show two basic problems resulting into Fredholm integral equation. The first of them will be an image deblurring the second one a gravity survey problem, both adopted from [Han10].

Image deblurring problem

Consider a 1D image (e.g. a barcode) scanned by some optical device (e.g. a barcode scanner). Although the original image is sharp, the record of the image is, due to imperfections in the scanner, usually blurred. Blurring process in an optical device is in most cases modeled by convolution with a Gauss kernel

$$k(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{x^2}{2\sigma^2}\right),$$

where the parameter σ controls the amount of blurring and its square corresponds to variance in statistics. The constant in front of the exponential, which normalize the Gaussian function, is often omitted. In image processing field, usually the term *point-spread function* instead of 'kernel' is used. Roughly speaking, it describes the response of an imaging system to a point source. Convolution kernels are special type of Fredholm kernels defined as $K(s,t) \equiv k(s-t)$. The idea of blurring process in the scanner is shown by scheme 1.3.

Gravity survey problem

Compared to the previous example, which deals with some process taking place in a technical device, this one describes a purely physical phenomenon. If we have measurements of the Earth's gravity field available, then we might ask the question: 'Given the vertical component of the gravity field $g(s)$, what can we deduce about the mass distribution with density $f(t)$ located at the depth d below the surface if we assume there is no mass outside this source?' The solution to this problem (i.e. the density distribution that best matches the data) is useful

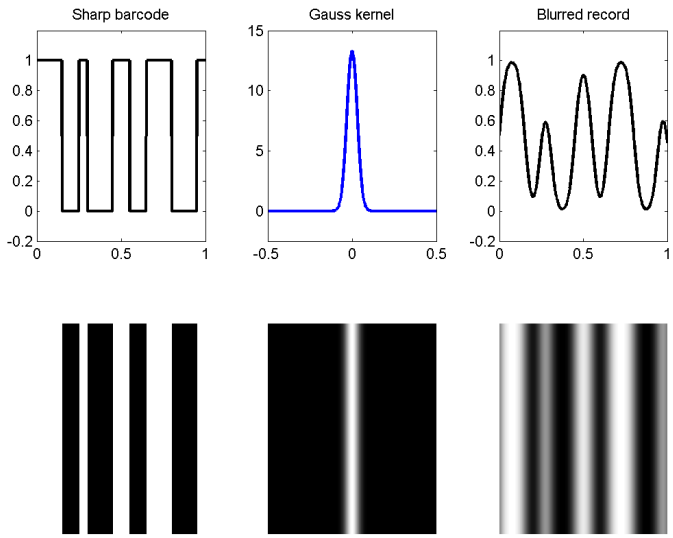


Figure 1.3: An illustration of blurring process in a bar-code scanner. Parameter σ is chosen as 0.03. See `example_barcode.m`.

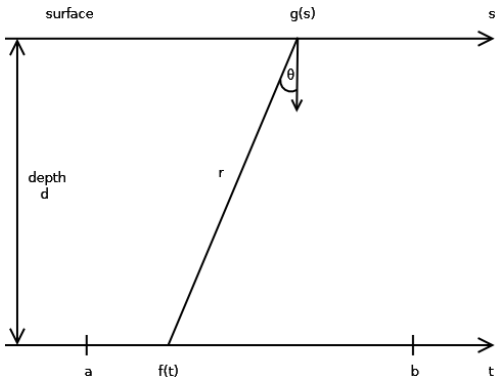


Figure 1.4: A scheme of the gravity surveying model with mass density $f(t)$ in the depth d under the surface. Here, $g(s)$ stands for the observed quantity – vertical component of the gravity field.

because it generally tells us something about a physical parameter that we cannot directly observe. We expect that the bigger distance d, the more difficult task it is.

According to the Newton's law of universal gravitation, the gravity field is inversely proportional to the squared distance between the objects (denoted by r here) and proportional to the mass, i.e.

$$\mathrm{d}g = G\,\frac{\cos\theta}{r^2}\,f(t)\mathrm{d}t,$$

where G is the universal gravitational constant. We multiplied the right-hand side by $\cos\theta$ as we only measure the vertical component of the gravitational field, see scheme 1.4. The total value of $g(s)$ for any s consists of contributions from

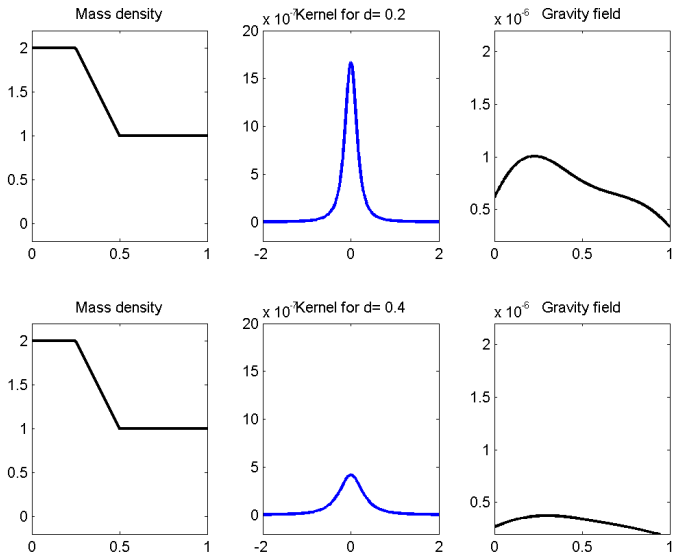


Figure 1.5: An illustration gravity survey problem for different depths d. Gravitation constant was chosen $6.674 \cdot 10^{-8}$, which corresponds to centimeter-gram-second system of units. See `example_gravity.m`.

all the mass along the t axis and therefore

$$g(s) = G \int_a^b \frac{\cos\theta}{r^2} f(t)\mathrm{d}t = G \int_a^b \frac{d}{(d^2 + (s-t)^2)^{3/2}} f(t)\mathrm{d}t\,, \tag{1.2}$$

where in the last expression we substituted for $\cos\theta$ and r. See [Han10] for more detailed derivation. We see that (1.2) has a form of Fredholm integral equation with the convolution kernel

$$K(s,t) \equiv G \frac{d}{(d^2 + (s-t)^2)^{3/2}}\,.$$

An example of gravity survey model is shown in figure 1.5. As we supposed, for smaller depth, we are at least able to guess that there was a steep decay in the density around the middle, but for depth $d = 0.4$, the jump is almost smoothed out. Intuitively, the second problem ($d = 0.4$) is 'more ill-posed', because even very different mass densities will lead to similar gravity fields.

In the two examples from different research fields we saw that a smoothing occurs while going from the source f to the observed data g. This is a universal phenomenon of integral equations. In mapping from f to g, *higher frequency components in f are dampened* compared to the lower ones. Mathematical formulation of this phenomenon is called Riemann-Lebesgue lemma and is included in the appendix of the thesis. Analogously, if we solve the inverse problem, i.e. compute the source f from the observed data g, high frequencies are amplified and the higher the frequency, the greater the amplification. This leads to the fact that even a small random perturbation of g can lead to a very large perturbation of f if it has a high frequency component. This illustrates the fundamental difficulty of solving inverse problems.

1.3 Singular value expansion

The *singular value expansion* (SVE) [Sch07] is a very important theoretical tool which helps us to understand the smoothing process in integral equations. It also provides a necessary and sufficient condition of the existence and uniqueness of the solution.

Consider a square integrable[2] kernel $K(s,t)$, then the kernel has an SVE, i.e. a convergent expansion of the following form

$$K(s,t) = \sum_{i=1}^{\infty} u_i(s)\, \mu_i\, v_i(t), \tag{1.3}$$

in which $\{\mu_i\}$ are singular values, and $\{u_i\}$ and $\{v_i\}$ are left and right singular functions respectively. Moreover, the following holds

- $\mu_1 \geq \mu_2 \geq \ldots, \mu_i \geq 0$,
- $\int_{I_s} u_i(s)u_j(s)\mathrm{d}s = \delta_{ij}$ and $\int_{I_t} v_i(t)v_j(t)\mathrm{d}t = \delta_{ij}$,
- $\int_{I_s} K(s,t)u_i(s)\mathrm{d}s = \mu_i v_i(t)$ and $\int_{I_t} K(s,t)v_i(t)\mathrm{d}t = \mu_i u_i(s)$.

It can be shown that the smoother the kernel, the faster the singular values decay to zero and that the singular functions resemble spectral bases. This is in agreement with Riemann-Lebesgue lemma. Without loss of generality the $\{u_i\}$ and $\{v_i\}$ form bases of $L^2(I_s)$ and $L^2(I_t)$ respectively. If we now expand both f and g in a standard manner (assuming that both of them lie in L^2), we get

$$f = \sum_{i=1}^{\infty} (v_i, f)_{L^2} v_i \quad \text{and} \quad g = \sum_{i=1}^{\infty} (u_i, g)_{L^2} u_i\,.$$

Substituting for g and f in (1.1) we get, due to orthonormality of $\{v_i\}$,

$$g = \sum_{i=1}^{\infty} \mu_i (v_i, f)_{L^2} u_i\,.$$

This leads formally to

$$\mu_i (v_i, f)_{L^2} = (u_i, g)_{L^2} \quad \text{and} \quad f = \sum_i \frac{(u_i, g)_{L^2}}{\mu_i}\, v_i\,. \tag{1.4}$$

The interpretation of the equation (1.4) is following: 'the solution exists if and only if all the projections of the right-hand side onto singular functions corresponding to zero singular values are zero'. Nevertheless, for every degenerate kernel (i.e. kernel with zero singular values) the solution is not unique, because right singular functions corresponding to the zero singular values form the null-space of the integral operator,

$$\int_{I_t} K(s,t)\, v_i(t)\, \mathrm{d}t = 0 \quad \text{if } \mu_i = 0.$$

[2] $\int_{I_s}\int_{I_t} K^2(s,t)\,\mathrm{d}t\,\mathrm{d}s < \infty$

Although the analysis in terms of SVE provides conditions of the existence and uniqueness of the solution, it is of purely theoretical concern because the SVE can be rarely determined analytically. However, it can always be approximated numerically as described in the next section.

Before finishing this part, we will discuss the *regularity* of the solution. To derive (1.4), we a priori assumed that the solution is square integrable, which guarantees the convergence of the infinite sum in (1.4). Conversely, for a square integrable solution f to exist, the sum of coefficients must be bounded, i.e.

$$\sum_i \frac{(u_i, g)_{L^2}}{\mu_i} < \infty. \tag{1.5}$$

Condition (1.5) has important consequences for numerical computations and is called *the Picard condition* [Pic10]. More on SVE can be found in [Han10].

1.4 Discretization of integral equations

The aim of discretizing integral equations lies in turning them into systems of linear equations

$$Ax \approx b, \quad A \in \mathbb{R}^{m \times n}, \quad x \in \mathbb{R}^n, \quad \text{and} \quad b \in \mathbb{R}^m, \tag{1.6}$$

which can be numerically solved to get approximate solutions[3]. This is often necessary, because we usually cannot solve (1.1) analytically and also because sometimes the right-hand side is evaluated in discrete points only. There is no exclusive way of discretizing, a review of the methods can be found in appendix of [VL05]. In the thesis, we will describe only the most common ones.

Quadrature rule

This is the most basic and straightforward approach. It is based on applying a numerical integration method to the integral equation, i.e.

$$\int_{I_t} K(s,t) f(t)\,\mathrm{d}t \approx \sum_{i=1}^{n} \omega_i K(s, t_j) f(t_j),$$

where t_j are the nodes of the quadrature and ω_j are the corresponding weights. We assume that we are able to observe the data g in some discrete points $s_1, \ldots, s_m$, hence

$$\sum_{j=1}^{n} \omega_j K(s_i, t_j) f(t_j) \approx g(s_i), \quad i = 1, \ldots, m.$$

Setting $a_{ij} \equiv \omega_j K(s_i, t_j)$, $x_j \equiv f(t_j)$, and $b_i \equiv g(s_i)$, we get (1.6).

It may be tricky to use quadrature rule to solve the integral equations, because there is not a clear link between the SVE analysis from the previous section and properties of A. One also has to pay attention to the fact that norm $\|A\|_F$ grows as $\mathcal{O}(\sqrt{m})$.

[3] We changed the notation to distinguish between functions K, f, and g and vector/matrix objects A, x, and b.

Expansion methods

The concept of expanding a function into a series is a pretty common method in mathematics to express a particular element as being part of a space. The ability of solving (1.1) consists in determining a good set of basis functions where to expand the data and the solution functions. Otherwise, it will be impossible to find even an approximation of this functions. The minimum requirement is that the elements of the basis are L^2 functions.

Orthonormal basis functions

The first approach is to approximate the data g and the unknown function f as a linear combination of some orthonormal [4] functions $\{\phi_j\}$ and $\{\psi_i\}$

$$f(t) \approx f_n(t) \equiv \sum_{j=1}^{n} x_j \phi_j(t),\ g(s) \approx g_m(s) \equiv \sum_{i=1}^{m} b_i\, \psi_i(s).$$

We assume that the residuals $f - f_n$ and $g - g_m$ are orthogonal to $\{\phi_j\}$ and $\{\psi_i\}$ respectively. Substituting for f and g in (1.1), we get

$$\sum_{j=1}^{n} \int_{I_t} K(s,t) x_j \phi_j(t)\, \mathrm{d}t \approx \sum_{i=1}^{m} b_i\, \psi_i(s).$$

Taking inner product with ψ_i, we get, from the orthonormality of the basis functions,

$$\sum_{j=1}^{n} \int_{I_s} \int_{I_t} \psi_i(s) K(s,t) x_j \phi_j(t)\, \mathrm{d}t\, \mathrm{d}s \approx b_i, \quad i = 1, \ldots, m.$$

Setting

$$a_{ij} \equiv \int_{I_s} \int_{I_t} \psi_i(s) K(s,t) \phi_j(t)\, \mathrm{d}t\, \mathrm{d}s, \tag{1.7}$$

we get (1.6).

The advantage of this approach consists in the link between SVE and the singular value decomposition (SVD) of A. It was proved in [Han88] that for any two sets of orthonormal functions, the SVD of A converges to the SVE of K as shown later in this section.

Galerkin method

This method is based on the previous one but the orthogonality condition has been relaxed and it is only necessary that the basis functions used in the series expansions are linearly independent and the residual is orthogonal to the bases, allowing the employment of such functions as B-splines or Chebyshev polynomials.

It is worth mentioning that not all the methods can always be used. If the data is given (or observed) only in a finite number of points, we cannot use the expansion method in a straightforward way, but some kind of discrete expansion must be used.

[4] orthonormal in L^2, that is $\int \phi_i \phi_j = \delta_{ij}$ and $\int \psi_i \psi_j = \delta_{ij}$

It may also be impossible to evaluate the a_{ij} explicitly in expansion methods and a quadrature rule must be employed to create the matrix A.

There exists a special case of basis functions which links the expansion methods and the quadrature rule. It is a set of so-called 'top hat' functions [Han87] – scaled indicator functions

$$\chi_i(x) = \begin{cases} \sqrt{x_i - x_{i-1}}, & x \in (x_{i-1}, x_i) \\ 0 & \text{elsewhere} \end{cases} .$$

The following two approaches coincide (up to scaling):

- expanding the functions f and g using 'top hat' functions and then using a quadrature rule to evaluate $a_{ij} \equiv \int_{I_s}\int_{I_t} \psi_i(s)K(s,t)\phi_j(t)\mathrm{d}t\mathrm{d}s$,
- using a composed quadrature rule to solve the whole integral equation (for a particular choice of t_j and s_i).

The relationship between SVD and SVE

As we mentioned, there exists a link between the SVD of the matrix A and the SVE of the kernel K for a particular type of discretization. This relation was thoroughly described in [Han88], we will repeat only the most important results. For simplicity, we will assume $m \geq n$, i.e. the matrix A is either square or has more rows than columns. Let A have the following (economic) SVD

$$A = U\Sigma V^T = \sum_{i=1}^{n} u_i \, \sigma_i \, v_i^T . \tag{1.8}$$

Here, $\Sigma \in \mathbb{R}^{n\times n}$ is a diagonal matrix with the singular values on the diagonal

$$\Sigma = \mathrm{diag}(\sigma_1, \ldots, \sigma_n), \quad \sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \geq 0,$$

and $U \in \mathbb{R}^{m\times n}$ and $V \in \mathbb{R}^{n\times n}$ are matrices with orthonormal columns constituting left and right singular vectors respectively. Since the SVD is generally not unique[5] [Bjö96], in (1.8) we consider an arbitrary but fixed SVD of the matrix A.

Formally, the solution of (1.6) can be written in the form of

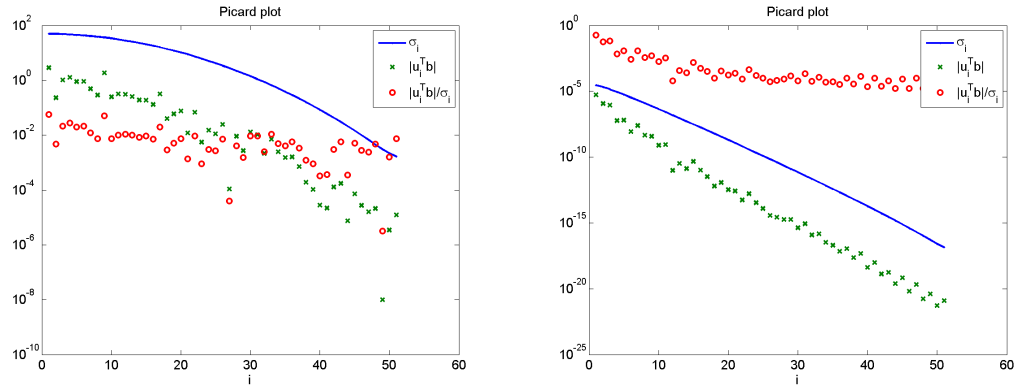
$$x = V\Sigma^{-1}U^T b = \sum_{i=1}^{n} \frac{u_i^T b}{\sigma_i} \, v_i \equiv A^\dagger b \, . \tag{1.9}$$

Analogously to the continuous model: 'the solution exists if and only if all the projections of the right-hand side onto singular vectors corresponding to zero singular values are zero', see (1.4). To ensure that the solution (1.9) approximates the solution (1.4) we expect a 'correspondence' between the SVE and SVD.

If $m = n$ and we compute the matrix A according to (1.7), then the singular values of A converge to those of K as $n \to \infty$. Left and right singular functions of K are approximated by $u_j^{(n)}$ and $v_j^{(n)}$, defined as follows

$$u_j^{(n)} \equiv \sum_{i=1}^{n} u_{ij}\psi_i \quad \text{and} \quad v_j^{(n)} \equiv \sum_{i=1}^{n} v_{ij}\phi_i \quad \text{for } j = 1 \ldots n.$$

[5] similarly to SVE



(a) Barcode reading ($\mu = 0.3$) (b) Gravity survey ($d = 0.2$)

Figure 1.6: The Picard plot for discretized problems from pages 10 and 11 – barcode reading and gravity survey. See `example_picard.m`.

The similar can be done for $m \neq n$[6]. This convergence, together with equations (1.4) and (1.9), has a fundamental consequence – the *discrete Picard condition*. The discrete Picard condition is a discrete version of the Picard condition (1.5) and can be in outline formulated as follows: 'on average, the sizes of projections $|u_i^T b|$ decay faster than the singular values of A'. The condition was originally proposed in [Han90]. See an illustration in figure 1.6.

Due to the fact that in most cases there is only a finitely dimensional data available (discrete measurements of some quantity, pixels in an image, already discretized sample problems etc.), we will work directly with the resulting system of linear equations (1.6). We will assume that the given system is good enough to approximate well the underlying integral equation. Roughly speaking, we will neglect the discretization error. Note that this is might be a bit tricky, because adding more points/functions to the discretizations decreases the discretization error but introduces more ill-posedness to the system (1.6) because the matrix A becomes more ill-conditioned and this has to be taken into account.

1.5 Noise

As mentioned in the end of sections 1.1 and 1.2, the fundamental difficulty of solving inverse problems lies in the fact that even a small random perturbation of the observed data g can lead to a very large perturbation of the solution f if it has high frequency components. The system (1.6) obviously inherits the ill-posedness of the original integral equation (1.1). The perturbation of the system is usually called *noise*. For the purpose of this thesis, we assume that noise is confined to the right-hand side b. This is a clear oversimplification. However, the perturbation of the system A is of different character than the perturbation in the data b. Matrix A is mostly influenced by model errors (the model is simplified relative to the real world) and discretization errors. Compared to that, the perturbation of b is usually dominated by random variations caused

[6] note that the number of left and right singular functions is always equal to $\min(m, n)$

by measurement errors (physical measurements). For solving systems, where the matrix A is contaminated by some errors, we refer the reader e.g. to the paper on scaled total least squares [PS02].

From this point on, we consider the following problem. Assume that there exists an exact solution x^{exact} and the corresponding 'exact' right-hand side $b^{exact} \equiv Ax^{exact}$. The data thus have the form

$$b = b^{exact} + b^{noise}, \quad \text{where} \quad b^{exact} = Ax^{exact}. \tag{1.10}$$

Moreover, we assume that the amount of noise is negligible compared to the right-hand side, i.e.

$$\left\|b^{noise}\right\| \ll \left\|b^{exact}\right\|, \tag{1.11}$$

which is a natural assumption and a prerequisite for any numerical method. The interpretation of x^{exact} is the following - x^{exact} is a solution to problem (1.6) with the right-hand side free from random perturbations. Note that, this is not completely true when a quadrature rule is used (either to discretize the problem or to evaluate (1.7)) but x^{exact} can still stand for a good approximation.

Analysis of the stochastic properties of the random noise b^{noise} represents the essential tool for solving discrete inverse problems. In the rest of the section, we cover the most important types of noise.

In this section, we use the same notation for both random and algebraical vectors.

White noise

White noise is the most commonly considered kind of noise. This is a type of perturbations in which all the elements of b^{noise} come from the same distribution and are uncorrelated.

Due to the fact that the 'observation error' is usually modeled by Gaussian distribution, in the rest of the thesis, white noise will stand for a noise from normal distribution with zero mean and variance σ^2, and will be denoted by

$$b^{noise} \sim \mathcal{N}(0, \sigma^2 I).$$

When the measurements are represented by a finite number of digits (linear analogue-to-digital conversion), then the errors are uniformly distributed in the interval $\left[-\frac{a}{2}, \frac{a}{2}\right]$, where a denotes the smallest unit of measurement[7]. This will be denoted by

$$b^{noise} \sim \mathcal{U}\left(-\frac{a}{2}, \frac{a}{2}\right).$$

Coloured noise

Coloured noise is noise in which some of the frequencies are dominant, i.e. the spectral density is not flat. The *power spectral density* (PSD), describes how the power of a signal or time series is distributed with frequency. Here, power is defined as the squared value of the signal. The formal definition of PSD can be found in [GD04]. Many of these definitions assume a signal with components at

[7] when the error is due to rounding; for truncation error, it is $[0, a]$

Table 1.1: Different colors of noise. The power spectral density is proportional to $\frac{1}{f^\beta}$.

colour	β
Brown(ian)/red	2
pink	1
white	0
blue	-1
violet	-2

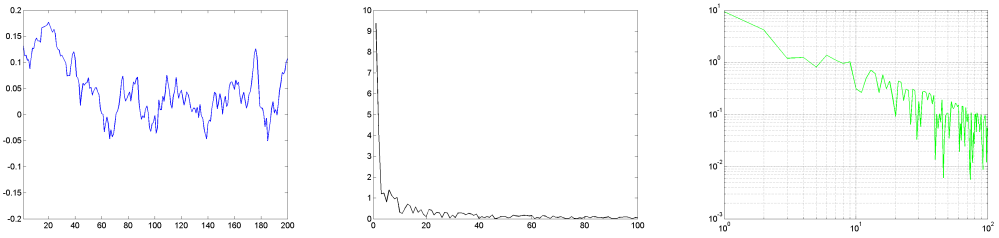
all frequencies[8], with a power spectral density per unit of bandwidth inversely proportional to the power of density

$$PSD \propto \frac{1}{f^\beta}\,.$$

Noise containing all frequencies is commonly referred to as broad-band.

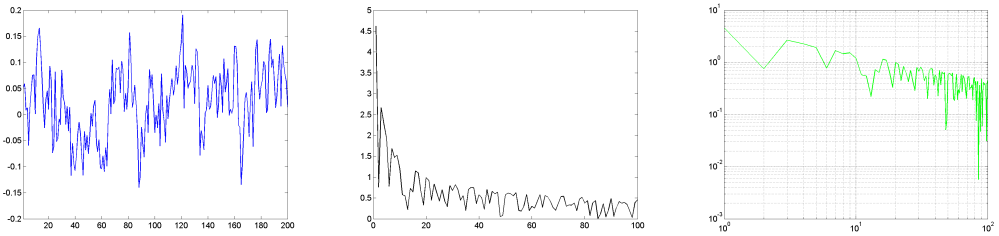
To generate a colored noise, the routine `powernoise.m` [LMR+07] is used in all experiment contained in the thesis. The most important noise colors are listed in table 1.1. Any noise with $\beta < 0$ will be referred to as *high-frequency noise (HF)*. *Low-frequency noise (LF)* will stand for noise with $\beta > 0$.

Following five figures illustrate the properties of different noise colors. The noise vectors b^{noise} are scaled to 1 in Euclidian norm.
See `example_coloured_noise.m`.



(a) noise vector b^{noise} (b) $|\text{fft}(b^{noise})|$ (c) PSD of b^{noise}

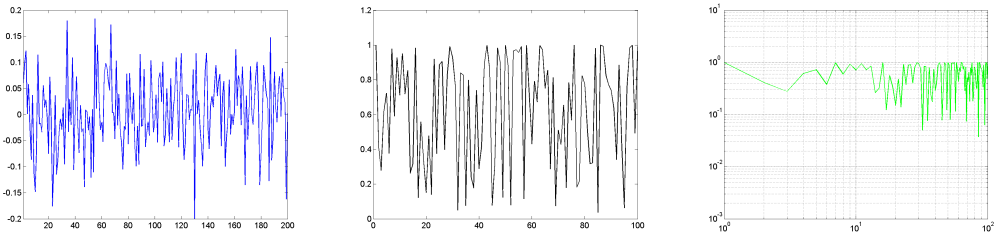
Figure 1.7: Brownian noise.



(a) noise vector b^{noise} (b) $|\text{fft}(b^{noise})|$ (c) PSD of b^{noise}

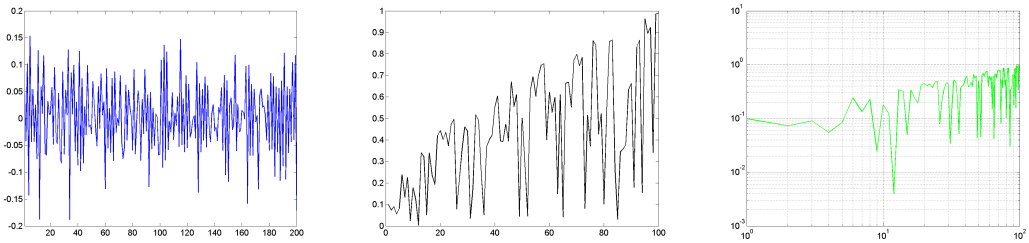
Figure 1.8: Pink noise.

[8] In the continuous case we consider all frequencies from some range (audible, visible etc.), in the discrete case, the frequency is bounded by the number of elements.



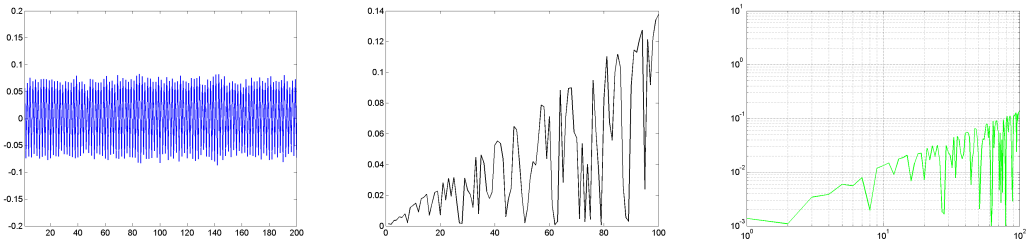
(a) noise vector b^{noise} (b) $|\text{fft}(b^{noise})|$ (c) PSD of b^{noise}

Figure 1.9: White noise.



(a) noise vector b^{noise} (b) $|\text{fft}(b^{noise})|$ (c) PSD of b^{noise}

Figure 1.10: Blue noise.



(a) noise vector b^{noise} (b) $|\text{fft}(b^{noise})|$ (c) PSD of b^{noise}

Figure 1.11: Violet noise.

Data correlated noise

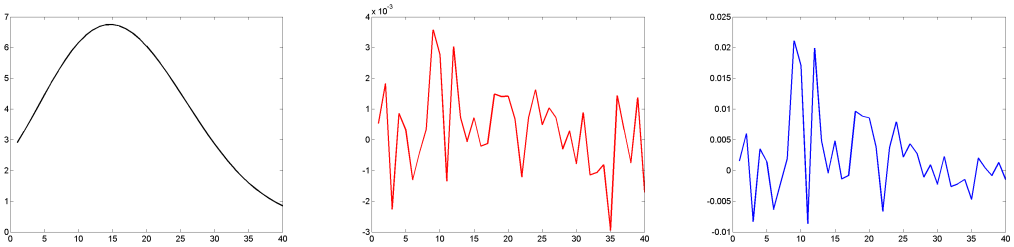
In the previous part of this section, we considered noise that is either uniformly or non-uniformly distributed but always independent of the 'true' data. Sometimes, however, it is reasonable to assume that noise is proportional to the data in some sense. For simplicity, we restrict ourselves to the case where noise is given by

$$b^{noise} \equiv \text{diag}(b^{exact})\ e, \quad \text{where} \quad e \sim \mathcal{N}(0, \sigma^2 I).$$

This kind of noise is obviously not white - the covariance matrix of b^{noise} is

$$\text{Cov}(b^{noise}) = \sigma^2 \text{diag}(b^{exact})^2.$$

Nevertheless, it has a property that is known as to be 'white-noise-like'. In short, white-noise-like means that the covariance matrix of the Fourier transform of b^{noise} is close to identity.



(a) Smooth exact data b^{exact} (b) White noise e (c) Noise b^{noise}

Figure 1.12: An example of a data correlated noise. The code is available under `example_data_correl_noise.m`.

Prewhitening

A number of methods is based on the assumption that noise in the right hand side is white. If noise is not white, then we are usually able to transform the problem into one with white noise. This technique is commonly called *prewhitening* [CS07]. The idea is to multiply the equation (1.10) by a non-singular matrix such that the noise becomes white, i.e. the covariance matrix is a (scaled) identity. Let now $\mathrm{Cov}(b^{noise}) = LL^T$ be the Cholesky decomposition of the covariance matrix. When we multiply the whole equation by the inverse Cholesky factor of the covariance matrix [9], the noise becomes white

$$\begin{aligned} L^{-1}Ax &\approx L^{-1}b \\ (L^{-1}A)x &\approx L^{-1}b^{exact} + L^{-1}b^{noise}. \end{aligned} \tag{1.12}$$

Using the property of covariance matrices

$$\mathrm{Cov}(Cx) = C\,\mathrm{Cov}(x)\,C^T,$$

it easily follows that the covariance matrix of $L^{-1}b^{noise}$ is identity. The transformation of the system also changes the spectral properties of the right-hand side. If the noise was originally high-frequency, then the inverse Cholesky factor dampens the high frequencies and $L^{-1}b^{exact}$ becomes even more low-frequency dominated and vice versa.

Prewhitening, however, does not have a regularizing effect – the transformed problem remains as ill-posed as it was before prewhitening.

1.6 Regularization methods – part 1: Spectral filtering methods

As explained in section 1.2, the main difficulty of solving discrete ill-posed problem consists in the fact that a small perturbation in the right-hand side b may lead to a very large perturbation of the solution x. This happens in the case when the perturbation has a high frequency component. Using the notation introduced in sections 1.4 and 1.5, we can describe the problem more formally. Let us recall that we search for x

$$Ax \approx b, \quad \text{where} \quad b = b^{exact} + b^{noise} \quad \text{and} \quad \left\|b^{noise}\right\| \ll \left\|b^{exact}\right\|. \tag{1.13}$$

[9] we assume that the covariance matrix is non-singular

According to (1.9), the solution to (1.13) has a form of

$$x = \sum_{i=1}^{n} \frac{u_i^T b}{\sigma_i} v_i = A^\dagger b. \tag{1.14}$$

Substituting for b and using the definition of x^{exact} (1.10), we get

$$\begin{aligned} x &= \sum_{i=1}^{n} \frac{u_i^T (b^{exact} + b^{noise})}{\sigma_i} v_i \\ &= \sum_{i=1}^{n} \frac{u_i^T b^{exact}}{\sigma_i} v_i + \sum_{i=1}^{n} \frac{u_i^T b^{noise}}{\sigma_i} v_i \\ &= x^{exact} + \sum_{i=1}^{n} \frac{u_i^T b^{noise}}{\sigma_i} v_i. \end{aligned}$$

We see that the desirable exact solution x^{exact} is perturbed by a linear combination of the right singular vectors. If we want x to approximate x^{exact}, we usually require

$$\left\| \sum_{i=1}^{n} \frac{u_i^T b^{noise}}{\sigma_i} v_i \right\| \ll \left\| \sum_{i=1}^{n} \frac{u_i^T b^{exact}}{\sigma_i} v_i \right\| \equiv \left\| x^{exact} \right\|,$$

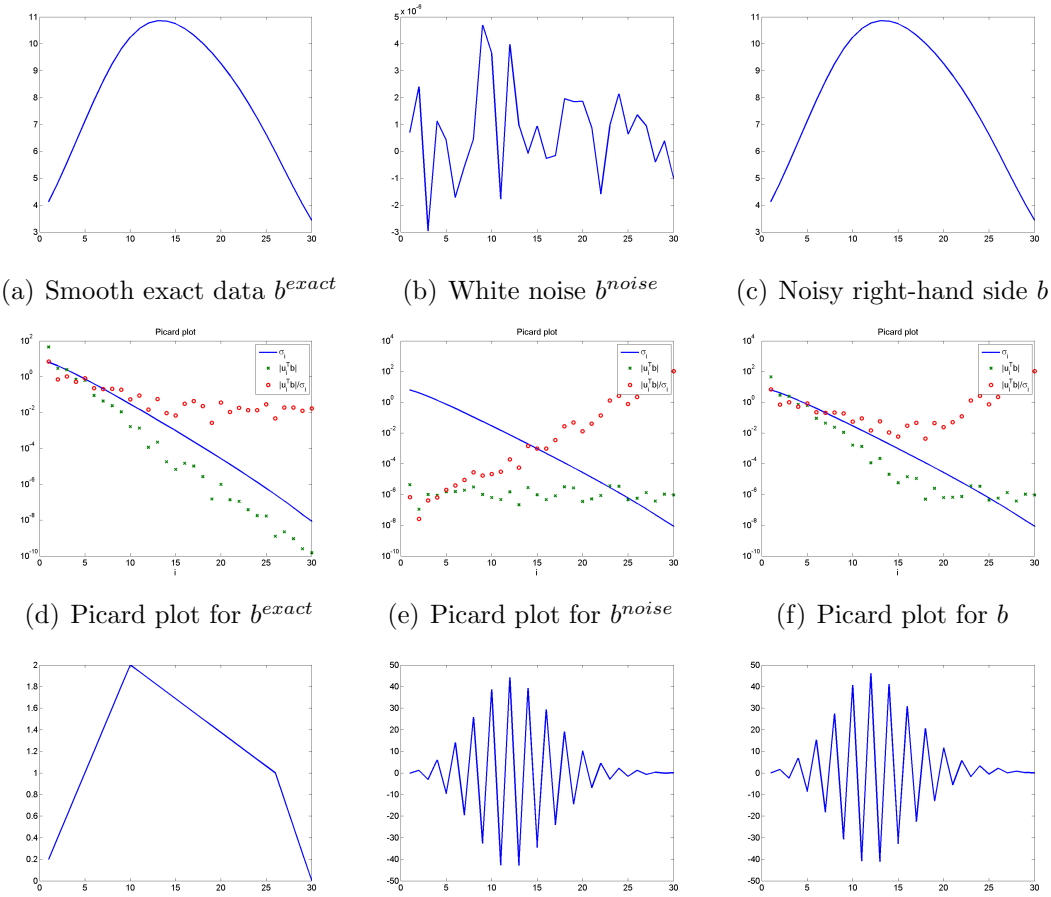
i.e. that the perturbation is small relative to the exact solution. This is, due to the orthonormality of the singular vectors, equivalent to

$$\sum_{i=1}^{n} \frac{|u_i^T b^{noise}|}{\sigma_i} \ll \sum_{i=1}^{n} \frac{|u_i^T b^{exact}|}{\sigma_i}.$$

Since the singular values σ_i decay (usually very fast) with i, we want $|u_i^T b^{noise}|$ to decay fast as well. In the view of the fact that the singular vectors resemble the spectral ones, this is equivalent to the requirement that the spectral components of b^{noise} are much smaller than the corresponding singular values. For the setting we consider, this is not the case. We assume that the kernel has a distinct smoothing effect which leads to a very fast decay of singular values, which causes the matrices to be extremely ill-conditioned. Compared to that, the noise is assumed to be broad-band, and even for the LF noise (e.g. Brownian), the high-frequency components are significant enough to make the last addends dominate the whole sum, unless the noise is truly minute. The true solution is then fully overlaid by the inverted noise. For this reason, solution (1.14) is commonly referred to as *naive* and will be further denoted by x^{naive}. The Picard plot proves as a useful tool to examine this behaviour.

In the figure 1.13 we see that even a white noise of order 10^{-6} can make the naive solution completely useless. This phenomenon is nicely depicted in the right Picard plot - the fractions $\frac{|u_i^T b|}{\sigma_i}$ increase for $i > 15$. These projections onto the left singular vectors are already dominated by the noise. This is the basis of spectral filtering methods – to preserve the projections that are solution dominated and dampen or remove the projections dominated by the noise. Generally, the regularized solution of this kind can be expressed in the form of filtered SVD

$$x^{reg} = \sum_{i=1}^{n} f_i \frac{u_i^T b}{\sigma_i} v_i. \tag{1.15}$$



(a) Smooth exact data b^{exact} (b) White noise b^{noise} (c) Noisy right-hand side b

(d) Picard plot for b^{exact} (e) Picard plot for b^{noise} (f) Picard plot for b

(g) x^{exact} (h) $A^{\dagger}b^{noise}$ (i) $x^{naive} = A^{\dagger}b$

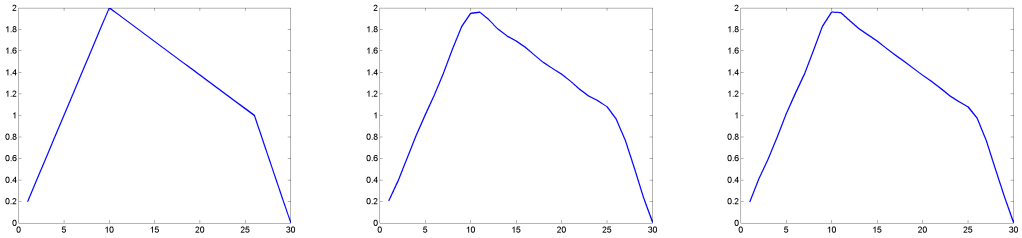
Figure 1.13: An example showing how the noise in the right-hand side is amplified when solving the inverse problem. The gravity matrix (see (1.2)) and a piece-wise linear source function were used. See `example_noise_amplification.m`.

Scalars f_i vary from 0 to 1 and are called the *filter factors*. Two most important filtering methods – truncated SVD and Tikhonov's method - are briefly described in the following subsections. There are more methods of this type but these are not covered here since they are not used in this thesis.

Truncated SVD

Truncated SVD [Han87, Han71, Var73] is the most straightforward method to regularize the solution. The idea is basically to preserve only a few first addends that correspond to the projections dominated by the exact solution and chop off the rest of the components, i.e. the components dominated by noise. More precisely, the truncated SVD method has one single parameter $k \leq n$ called the *truncation level* and the corresponding filter factors are

$$f_i = \begin{cases} 1 & \text{for } i \leq k \\ 0 & \text{for } i > k. \end{cases}$$



(a) Exact solution (b) x^{T-SVD} for $k = 15$ (c) x^{Tikh} for $\lambda = 10^{-4}$

Figure 1.14: Regularized solutions of gravity problem from figure 1.13. See `example_regul.m`.

The solution obtained by truncated SVD has the form

$$x^{T-SVD} = \sum_{i=1}^{k} \frac{u_i^T b}{\sigma_i} v_i. \tag{1.16}$$

The truncated SVD problem is less ill-posed than the original one, compare $\frac{\sigma_1}{\sigma_k}$ to $\frac{\sigma_1}{\sigma_n}$, but is biased, see [Han10] for more detailed discussion. The choice of regularization parameter(s) will be briefly mentioned in section 1.7. If we want to express the importance of the truncation level, we denote the solution simply x^k.

Tikhonov's method

Tikhonov's method [Tik63] explicitly incorporates the regularity requirement in the formulation of the problem. It balances the norm of residual and the norm of the corresponding solution, i.e.

$$\min_x \left\{ \|Ax - b\|^2 + \lambda^2 \|x\|^2 \right\}. \tag{1.17}$$

Using the formula (1.17), we define the Tikhonov solution in the form of spectral filtering (1.15) as follows

$$x^{Tikh} = \sum_{i=1}^{n} \frac{\sigma_i^2}{\sigma_i^2 + \lambda^2} \frac{u_i^T b}{\sigma_i} v_i. \tag{1.18}$$

For singular values σ_i larger than the parameter λ, the filter factors are close to 1. For singular values much smaller than λ, the filter factors are small and corresponding spectral components are dampened. Tikhonov's regularization introduces a bias into the problem similarly to truncated SVD. We will also use alternative notation x^λ if the need arises.

To demonstrate the usefulness, we solve the gravity problem from figure 1.13 using both truncated SVD and the Tikhonov's method. We used a black-box parameter estimator in this case.

Although the solution is not the best we could possibly get, there is a significant improvement compared to the naive solution. Unless stated otherwise, in the following experiments the methods `tsvd` and `tikhonov` implemented in Regularization tools [Han94] will be used.

1.7 Choice of regularization parameters

Finding the optimal regularization parameter k or λ is a difficult task and is not of particular interest of the thesis. Throughout the experiments, the parameter estimators will be used as black-box. Nevertheless, we decided to include a brief description of the methods, purely for the completeness of the work.

The aim of all parameter choosing methods is to minimize the errors in the regularized solution. For the spectral filtering methods, the error is of the following form

$$\begin{aligned} x^{exact} - x^{reg} &= x^{exact} + \sum_{i=1}^{n} f_i \frac{u_i^T b}{\sigma_i} v_i \\ &= x^{exact} + \sum_{i=1}^{n} f_i \frac{u_i^T A x^{exact}}{\sigma_i} v_i + \sum_{i=1}^{n} f_i \frac{u_i^T b^{noise}}{\sigma_i} v_i \\ &= \sum_{i=1}^{n} (1 - f_i)\, v_i^T x^{exact}\, v_i + \sum_{i=1}^{n} f_i \frac{u_i^T b^{noise}}{\sigma_i} v_i. \end{aligned}$$

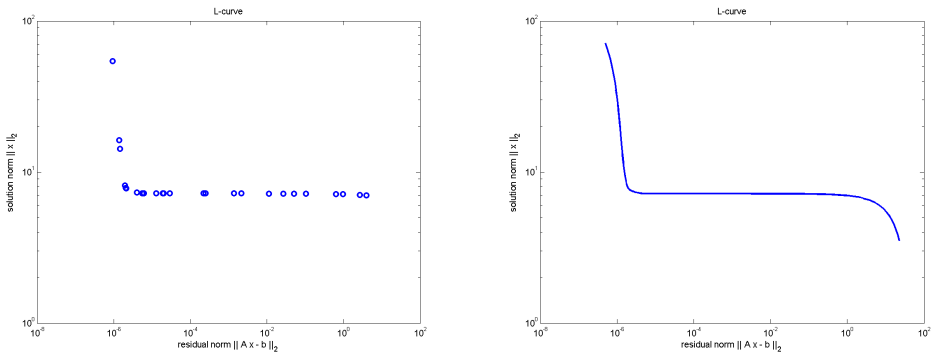
The first part of the error comes from the introduction of the filtering and introduces a bias into the problem. The error is generally called the *regularization error*. The second part of the error is the *perturbation error* resulting from inverting and filtering the noise component in the data.

Both perturbation and regularization error are always present in the regularized solution and their size depends on the regularization parameter. From the nature of the spectral filtering methods it follows that if one increases the other decreases and vice versa. Therefore we can say that the goal of choosing the regularization parameters k or λ is to balance the size of these two error terms.

At the moment, there is no parameter-choice method that is sufficiently robust. We will use three methods that are generally considered as methods of first choice. The disadvantage of all these methods is that they rely on the assumption that the noise in the right-hand side is white.

L-curve

L-curve is a method that simply balances the residual $\|Ax^{reg} - b\|$ against the norm of the regularized solution $\|x^{reg}\|$. Its application to discrete inverse problems was suggested by Hansen and O'Leary in [HO93]. For both regularization methods (1.16) and (1.18), the residual varies monotonically with the regularization parameters. So does the norm of the solution, which varies inversely. For truncated SVD, the residual decreases with k while the norm of the solution increases. For an ill-posed inverse problem, this usually works in a special way. The norm $\left\|x^k\right\|$ increases slowly with k until it reaches the level, where the components are dominated by the noise. Then it starts growing rapidly, while the residual does not change much any more. It works similarly for the Tikhonov method. The aim of L-curve is to find the corner, from which the norm of the solution grows dramatically compared to the decrease of the residual. Usually a log-log scale is used to emphasize the corner of the L-curve. The corner is defined as the point of maximal curvature. A discrete definition of curvature is used for the



(a) L-curve for T-SVD

(b) L-curve for Tikhonov

Figure 1.15: L-curves for gravity problem from figure 1.13. The code is available under `example_l_curve.m`.

truncated SVD. An example of such an L-curve is shown in figure 1.15. The 'L' in the name refers to the shape of the curve.

Discrepancy principle

The discrepancy principle (DP) is a method introduced by Morozov in [Mor66] and [Mor84], and is based on an a priori knowledge of the noise's variance σ^2. We choose the regularization parameter such that the residual norm equals the discrepancy in the data. The discrepancy is approximated by $c \cdot \sqrt{n}\sigma$, where $c \geq 1$ is a 'safety factor'. The method can be defined as follows

$$\text{choose the regularization parameter s.t. } \|Ax^{reg} - b\| = c \cdot \sqrt{n}\sigma.$$

For the truncated SVD method, this is hard to achieve, therefore we choose the largest k such that $\left\|Ax^k - b\right\| \geq c \cdot \sqrt{n}\sigma$. For simplicity, $c \equiv 2$ will be used in this thesis.

The obvious disadvantage of this method is that the variance of the noise or the noise level has to be known a priori. This is however very rarely the case. However, there are methods providing an estimate of the noise level. One of them will be described in section 2.3.

Generalized cross validation

Cross-validation is a statistical technique based on partitioning a sample of data into two complementary subsets, performing the analysis on one subset (called the training set), and validating the analysis on the other subset (called the validation set). To reduce variability, multiple rounds of cross-validation are performed using different partitioning, and the validation results are averaged over the rounds.

For simplicity, the i-th training set is set to be the system (1.13) with i-th row left out. And the results are averaged over all m rows. The goal is to find the regularization parameter which minimizes the prediction errors for all the validation sets, i.e. data elements. For Tikhonov we have

$$\min_{\lambda} \left[\frac{1}{m} \sum_{i=1}^{m} \left(A(i,:)x_{(i)}^{\lambda} - b_i \right)^2 \right], \tag{1.19}$$

where $x_{(i)}^{\lambda}$ is the regularized solution of the i-th training set. The same can be done for the truncated SVD. The generalized cross-validation (GCV) replaces the optimization (1.19) by the minimization

$$\min_{\lambda} \frac{\left\|Ax^{\lambda} - b\right\|^2}{\left(m - \sum_{i=1}^{m} f_i\right)^2}.$$

See [GHW79] for more details regarding GCV.

All three parameter-choice methods are implemented in Regularization Tools [Han94, Han07] as `l_curve`, `discrep`, and `gcv` and will be used in the following experiments.

1.8 Regularization methods – part 2: Projection and hybrid methods

The general idea of projection methods is to compute an approximation of x^{exact} that effectively lies in a (low-dimensional) subspace of $\mathbb{R}^n$. In some sense, this holds also for truncated SVD, where the subspace is spanned by the first k right singular vectors. For large-scale problems it is impossible to compute the SVD[10] of A. But from the SVD analysis we know that the singular vectors resemble spectral bases and that the solution is dominated by the low-frequency components, therefore we can define a priori a set $\mathcal{W}_l \equiv \{w_1, \ldots, w_l\}$ that has the same overall features as the first l singular vectors. These bases may be independent of the matrix A (basis vectors of discrete cosine transform, wavelets etc.), or adapted to a particular problem.

The construction of such a projection space leads to a *constrained least squares problem*

$$\min_{x} \|Ax - b\| \quad \text{subject to} \quad x \in \operatorname{span}(w_1, \ldots, w_l). \tag{1.20}$$

Defining $W_l \equiv (w_1, \ldots, w_l) \in \mathbb{R}^{n \times l}$, problem (1.20) can be reformulated as follows

$$x_l = W_l y_l, \text{ where } y_l = \operatorname*{argmin}_{y} \|(AW_l)y - b\|. \tag{1.21}$$

If $l \ll n$, which is usually the case, then we can compute the matrix AW_l explicitly and then solve the least squares problem with $m \times l$ matrix. For reasons of numerical stability, it is recommended that the matrix W_l has orthonormal columns.

In some cases, the restricted least squares problem (1.21) may still inherit some of the ill-posedness of the original problem, therefore it may be advantageous to use some of the regularization techniques mentioned in the section 1.6 to solve the projected problem $(AW_l)y \approx b$. The combination of a projection method and a spectral filtering method is generally referred to as *hybrid method* [Bjö88]. The hybrid methods have then two regularization parameters - one corresponding to the size of the projection matrix (denoted as l) and one corresponding to

[10] In case of the T-SVD, not the entire SVD is needed, but only k first components. However, we may need a few more components to find the regularization parameter, e.g. by L-curve.

the spectral filtering (denoted as k or λ). For an orthonormal matrix W_l, the Tikhonov regularization of the projected problem provides the identical solution as the same projection method applied on Tikhonov problem (1.17). The latter one is however not suited for large-scale problems, see [Han10].

As mentioned above, the basis can be a fixed set (usually some set of functions associated with fast transforms) or adapted to the problem. We will restrict ourselves to the bases that are problem-dependent, particularly to the *Krylov subspaces*.

The l-th Krylov subspace associated with a square non-zero matrix $C \in \mathbb{R}^{n \times n}$ and a non-zero vector $d \in \mathbb{R}^n$ is defined in the following way

$$\mathcal{K}_l(C, d) \equiv \text{span}\left(d, Cd, C^2d, \ldots, C^{l-1}d\right). \tag{1.22}$$

For the setting we have, i.e. the matrix A is generally rectangular, we cannot set C directly as A. Since $m \geq n$, we set $C \equiv A^T A$ and $d \equiv A^T b$.

The vectors that span the Krylov subspace converge to the principal eigenvector of the matrix C and therefore become close to linearly dependent. For that reason, they are not ideal for practical implementation and should be always orthonormalized. In real implementation, the Krylov subspace is not constructed in the way described in (1.22) but the solution is computed iteratively for $l = 1, \ldots, l_{max}$, where the l_{max} is determined by some stopping criterion.

Next chapter will focus on Krylov subspace method with particularly good computational properties - *LSQR* [PS82b, Bjö88]. It offers fast computation with short recurrences, low storage requirements and easy memory allocation. Moreover, this method will allow us to estimate the optimal l_{max} using several stopping criteria.

Chapter 2

LSQR and Golub-Kahan Iterative Bidiagonalization

As suggested in the previous chapter, we (formally) apply a Krylov subspace method to the *normal equations*

$$A^T A x = A^T b. \tag{2.1}$$

There are three main ways how to treat the problem (2.1), i.e. how to generate an orthonormal basis of $\mathcal{K}_k(A^T A, A^T b)$. We can apply the *conjugate gradient* (CG) method [HS52]. The resulting method is then referred to as *CGLS* or *CGNR*. However, for matrices that do not have full column rank a breakdown may occur. The *Lanczos method* [Lan50, MS06] is mathematically equivalent to CGLS but numerically more favourable. The most elegant and most stable method for solving (2.1) is *LSQR* [PS82b] based on the *Golub-Kahan iterative bidiagonalization* [GK65]. See [DTHP$^+$12, sec. 9.5], [Han10, sec. 6.4] or [PS82a] for more details and for the comparison of the methods.

2.1 Golub-Kahan iterative bidiagonalization

The aim of this section is to bring a brief overview of the Golub-Kahan bidiagonalization algorithm and the LSQR method. We adopted the notation of [HPS09]. Given the initial vectors $w_0 \equiv 0$, $s_1 \equiv b/\beta_1$, where $\beta_1 \equiv \|b\|$, the Golub-Kahan iterative bidiagonalization computes for

$$\begin{aligned} \alpha_j w_j &= A^T s_j - \beta_j w_{j-1}\,, \qquad \|w_j\| = 1\,, \\ \beta_{j+1} s_{j+1} &= A w_j - \alpha_j s_j\,, \qquad \|s_{j+1}\| = 1\,, \end{aligned} \tag{2.2}$$

until $\alpha_j = 0$ or $\beta_{j+1} = 0$, or until $j = n$. After k iterations, this algorithm has produced matrices $S_k \equiv [s_1, \ldots, s_k] \in \mathbb{R}^{m\times k}$ and $W_k \equiv [w_1, \ldots, w_k] \in \mathbb{R}^{n\times k}$ with orthonormal columns, further referred to as matrices of left and right bidiagonalization vectors, and a lower bidiagonal matrix

$$L_k \equiv \begin{bmatrix} \alpha_1 & & & \\ \beta_2 & \alpha_2 & & \\ & \ddots & \ddots & \\ & & \beta_k & \alpha_k \end{bmatrix} \in \mathbb{R}^{k\times k}. \tag{2.3}$$

Using this notation, the matrix version of the bidiagonalization (2.2) has the following form

$$A^T S_k = W_k L_k^T, \qquad AW_k = S_k L_k + \beta_{k+1} s_{k+1}. \tag{2.4}$$

To simplify the second part of (2.4), we introduce the matrix L_{k+},

$$L_{k+} \equiv \begin{bmatrix} \alpha_1 & & & \\ \beta_2 & \alpha_2 & & \\ & \ddots & \ddots & \\ & & \beta_k & \alpha_k \\ & & & \beta_{k+1} \end{bmatrix} \in \mathbb{R}^{k+1 \times k}. \tag{2.5}$$

Using this matrix, (2.4) can be further rewritten as

$$A^T S_k = W_k L_k^T, \qquad AW_k = S_{k+1} L_{k+}. \tag{2.6}$$

Moreover, columns of matrices S_k and W_k form the bases of the Krylov subspaces $\mathcal{K}_k(AA^T, b)$ and $\mathcal{K}_k(A^TA, A^Tb)$ respectively. Using the notation of section 1.8, we can rewrite our problem as

$$x_k = W_k y_k, \quad \text{where} \quad y_k = \underset{y}{\operatorname{argmin}} \left\| (AW_k)y - b \right\|,$$

which is, due to (2.6) and the orthonormality of the columns of S_{k+1}, equivalent to

$$x_k = W_k y_k, \quad \text{where} \quad y_k = \underset{y}{\operatorname{argmin}} \left\| L_{k+} y - \beta_1 e_1 \right\|. \tag{2.7}$$

In other words, x_k is the solution to the least squares problem restricted to $\mathcal{K}_k(A^TA, A^Tb)$. Equation (2.7) is solved via QR factorization[1]. The whole algorithm is referred to as LSQR and is thoroughly described in [PS82a]. This method is mathematically equivalent to CGLS and Lanczos tridiagonalization, but superior regarding stability, see Appendix.

2.2 Noise propagation in Golub-Kahan bidiagonalization

In this section, we describe how the noise in the data propagates through the bidiagonalization process. We will follow part 3 of [HPS09], where this phenomenon is thoroughly studied for white noise and is subsequently demonstrated on the problem `shaw` from Regularization Tools [Han07]. The idea is based on the smoothing property of the matrix A or more precisely AA^T (see section 1.2 of this thesis).

Consider the vectors s_k, w_k (left and right bidiagonalization vectors generated by the bidiagonalization algorithm). The starting vector $s_1 = b/\left\|b\right\|$ is the normalized noisy data. According to (2.2), the vector s_2 is obtained from s_1 as follows

$$\alpha_1 \beta_2 s_2 = AA^T s_1 - \alpha_1^2 s_1. \tag{2.8}$$

[1] In hybrid methods, a spectral filtering method is used instead.

This relation has an interesting consequence. The smoothing operator AA^T smooths out the high frequency components of s_1, i.e. $AA^T s_1$ is smooth. Subsequently, $AA^T s_1$ is orthogonalized against s_1, which is the normalized noisy data containing high-frequency components. This causes the high-frequency components to be transferred to s_2, while a portion of smooth part of s_1 is subtracted and therefore the *relative level of the high frequency part of the noise* can be expected to be higher in s_2 than in s_1. This occurs for any k with the vector s_{k+1} obtained from $AA^T s_k$ via the orthogonalization against the vectors s_{k-1} and s_k.

The high-frequency noise propagation can be described more precisely, when we treat the white noise separately in the bidiagonalization process. Therefore we decompose s_1 into the exact component $s_1^{exact} \equiv b^{exact}/\|b\|$ and the noise component $s_1^{noise} \equiv b^{noise}/\|b\|$, $s_1 = s_1^{exact} + s_1^{noise}$. Substituting into (2.2), we get

$$\beta_2 s_2 = Aw_1 - \alpha_1(s_1^{exact} + s_1^{noise}) = Aw_1 - \alpha_1 s_1^{exact} - \alpha_1 s_1^{noise}. \tag{2.9}$$

Due to the distinct smoothing property of A, the term Aw_1 is considered to be almost free of high-frequency noise. Furthermore, the low-frequency noise components of Aw_1 are negligible relatively to the low frequency components of the exact data. Therefore, the authors of [HPS09] suggest the following definition s_{k+1}^{exact} and s_{k+1}^{noise} for $k = 1, 2, \ldots$

$$\begin{aligned} \beta_{k+1} s_{k+1}^{exact} &\equiv \quad Aw_k - \alpha_k s_k^{exact}\,, \\ \beta_{k+1} s_{k+1}^{noise} &\equiv \qquad -\alpha_k s_k^{noise}\,. \end{aligned} \tag{2.10}$$

Obviously, $s_{k+1} = s_{k+1}^{exact} + s_{k+1}^{noise}$ and

$$\beta_{k+1} s_{k+1} = Aw_k - \alpha_k s_k,$$

which makes the definition (2.10) consistent with the algorithm (2.2).

Note that s_k^{exact} and s_k^{noise} *do not represent* the exact and noise components of s_k. The idea of this white-noise-propagation analysis is to 'neglect' the part of the noise vector that is multiplied by A (and therefore smoothed) within the bidiagonalization process and focus only on the part that is multiplied by a scalar. In other words, the smoothed noise components are included in the vectors s_k^{exact}. There is no analogy for right bidiagonalization vectors, because all vectors w_k are smoothed and do not contain significant information about the noise.

From (2.10) it immediately follows that

$$s_{k+1}^{noise} = -\frac{\alpha_k}{\beta_{k+1}}\, s_k^{noise} = (-1)^k \prod_{j=1}^{k} \frac{\alpha_j}{\beta_{j+1}}\, s_1^{noise}. \tag{2.11}$$

In order to estimate the behaviour of the *cumulative amplification ratio*

$$\rho_k^{-1} \equiv \prod_{j=1}^{k} \frac{\alpha_j}{\beta_{j+1}}\,, \tag{2.12}$$

we need some additional information about α_j and β_{j+1}. This can be obtained from the analysis of spectral components of the bidiagonalization vectors with

respect to the left and right singular vectors of A. Using the SVD (1.8), we rewrite the first step of the Golub-Kahan bidiagonalization (2.2) as

$$\alpha_1(V^T w_1) = \Sigma(U^T s_1)\,, \tag{2.13}$$
$$\beta_2(U^T s_2) = \Sigma(V^T w_1) - \alpha_1(U^T s_1)\,. \tag{2.14}$$

From (2.13) we see that $V^T w_1$ is dominated by the same components as $U^T s_1$, with the dominance amplified by the scaling by Σ. The subsequent orthogonalization (2.14) of $\Sigma(V^T w_1)$ against $U^T s_1$ requires that the dominance in $\Sigma(V^T w_1)$ and $U^T s_1$ is cancelled out, otherwise the orthogonality between $U^T s_2$ and $U^T s_1$ can not hold. Therefore we expect $\beta_2 \ll \alpha_1$.

Analogously for $k = 2, 3, \ldots$

$$\alpha_k(V^T w_k) = \Sigma(U^T s_k) - \beta_k(V^T w_{k-1})\,, \tag{2.15}$$
$$\beta_{k+1}(U^T s_{k+1}) = \Sigma(V^T w_k) - \alpha_k(U^T s_k)\,. \tag{2.16}$$

In (2.15) the dominance in $\Sigma(U^T s_k)$ and $(V^T w_{k-1})$ is shifted by one component and one can not expect a significant cancelation. By contrast, in (2.16), $\Sigma(V^T w_k)$ and $U^T s_k$ are dominated in the same components and the orthogonality between s_{k+1} and s_k can not be achieved without a significant cancelation. Summarizing,

$$\alpha_k \approx \beta_k \text{ and } \beta_{k+1} \ll \alpha_k\,.$$

This leads us to the conclusion that the cumulative amplification ratio (2.12) grows with increasing k, i.e. that the Golub-Kahan bidiagonalization for a smoothing operator A amplifies the relative level of noise in s_{k+1} as k increases. This, however, holds only until a certain stage. At some point, the discrete Picard condition is violated, see e.q. figure 1.13, and s_{k+1} starts to be dominated by the white noise component (s_{k+1}^{noise}). The phenomenon is demonstrated on problem `shaw` from [Han07] in figures 2.1 and 2.2. Problem `shaw` is used to stay consistent with the original paper [HPS09]. The point, where the dominating low-frequency components are projected out and the left bidiagonalization vectors become dominated by the high-frequency noise is referred to as the *noise revealing iteration*, denoted by k_{noise}, and will be further studied in the next section.

2.3 Noise revealing iteration and determination of the noise level

In the previous section, it was shown, that the bidiagonalization process (2.2), roughly speaking, in each step subtracts the low-frequency spectral components from s_k to obtain the successive left bidiagonalization vector s_{k+1}. Due to this elimination, the relative level of the high-frequency part of the noise is expected to be higher in s_{k+1}. This happens until the bidiagonalization process reaches the noise revealing iteration. At that iteration, the low-frequency component are projected out and vector s_k becomes dominated by the high-frequency noise.

The cancelation of low-frequency components can be nicely illustrated in the basis of left singular vectors of A. Such basis is, however, computationally expensive and for larger matrices almost infeasible. Therefore the authors

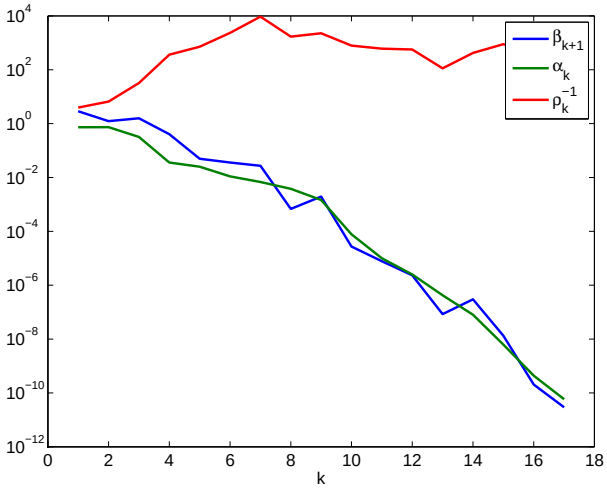


Figure 2.1: The amplification ratio for problem `shaw(400)` with the relative noise level 10^{-4}. See `example_s_k.m`.

of [HPS09] suggest using the standard Fourier trigonometric basis, which allows the coefficients being computed efficiently using *fast Fourier transform* via the MATLAB function `fft`. Sometimes it might be also useful and potentially more straightforward to compute the components in the basis of *discrete cosine transform* [ANR74], i.e. to use MATLAB function `dct`. First ten vectors of both bases and the first ten left singular vectors of matrix `shaw` are shown in figure 2.3 for comparison.

The projections of s_k onto the left singular vectors allowed us to determine visually when the low frequencies are projected out. The similar holds for Fourier and discrete cosine transform, although the transition is less pronounced, see figure 2.4.

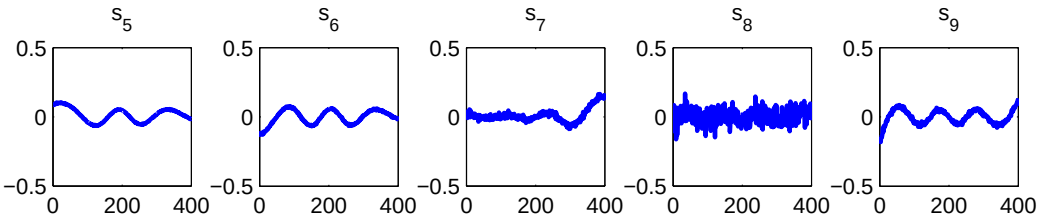
The aim of [HPS09] was to determine the noise revealing iteration $k_{noise}+1$.[2] This can be done manually, using a figure similar to 2.4 and visually determining the iteration, where the large frequencies are projected out. But such user's interference should be of course avoided.

Therefore, Hnětynková, Plešinger, and Strakoš derived an automated criterion based on relation between the Lanczos tridiagonalization and Riemann-Stieltjes integral. To understand the relation between the Riemann-Stieltjes integral and the Lanczos tridiagonalization, we refer the reader to [MS06]. Results of [MS06] related to our topic are also included in [HPS09] and [Vas11].

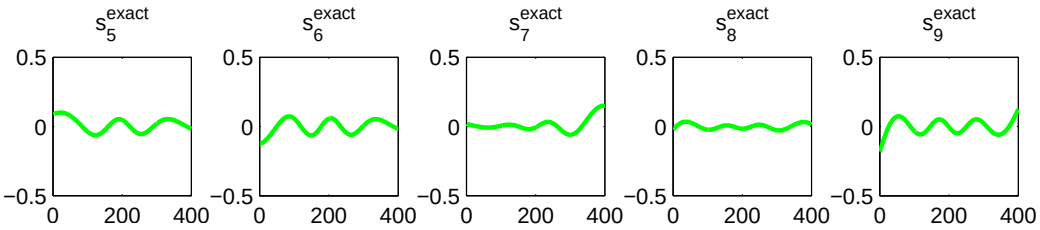
Without going here further into details, they used a criterion searching for a stagnation of particular entities - $|(p_1^{(k)}, e_1)|$, where $p_1^{(k)}$ denotes the left singular vector corresponding to the smallest singular value of the bidiagonal matrix L_k. The iteration k_{noise} was then determined as the first iteration step k for which

$$\frac{|(p_1^{(k+1)}, e_1)|}{|(p_1^{(k+1+step)}, e_1)|} < \left(\frac{|(p_1^{(k)}, e_1)|}{|(p_1^{(k+1)}, e_1)|} \right)^{\zeta}, \tag{2.17}$$

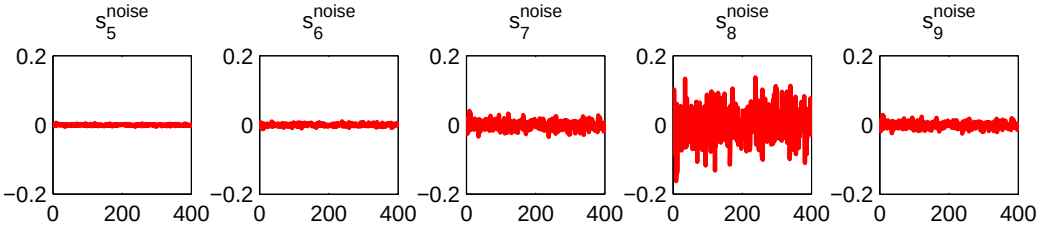
[2] the $+1$ is chosen to make the notation more convenient in the remainder of the thesis



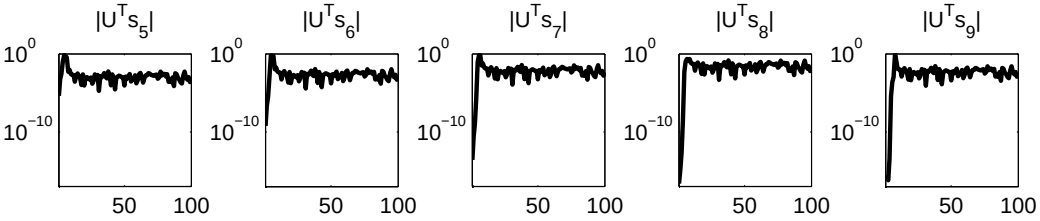
(a) s_k



(b) s_k^{exact}



(c) s_k^{noise}



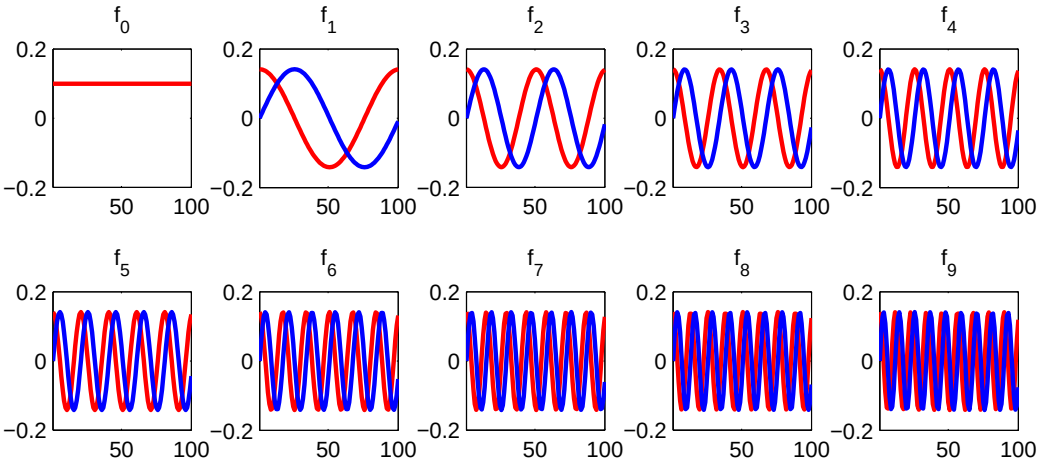
(d) $U^T s_k$

Figure 2.2: Left bidiagonalization vectors for the problem `shaw(400)` identical with the problem from figure 2.1, their low-frequency 'exact' parts s_k^{exact}, the white noise parts s_k^{noise}, and the first 100 spectral components. The noise revealing iteration is the iteration 8 - the large spectral components are projected out and the left bidiagonalization vector is dominated by white noise. See `example_s_k.m`.

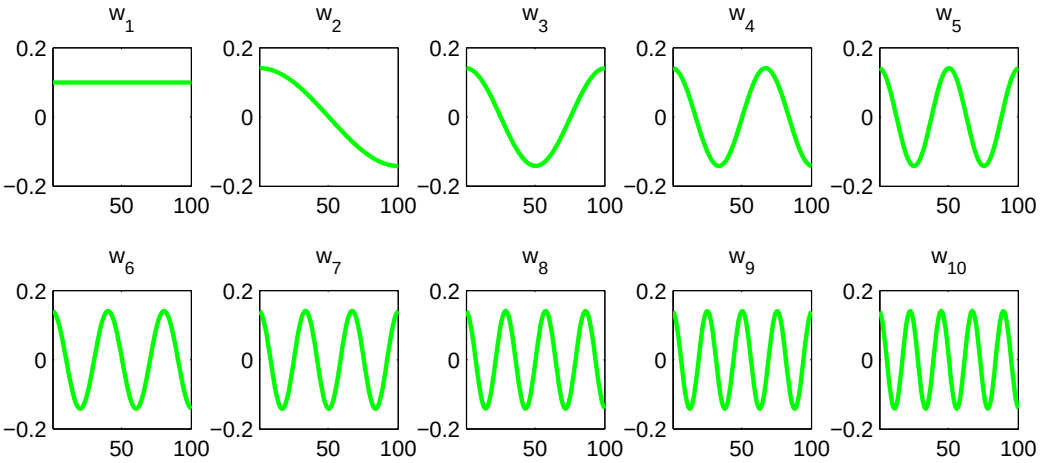
tested for $\zeta = 0.5$ and $step = 3$. The noise level estimate is then given by

$$\delta_{noise} \equiv \frac{\|b^{noise}\|}{\|b^{exact}\|} \approx |(p_1^{k_{noise}+1}, e_1)| \,. \tag{2.18}$$

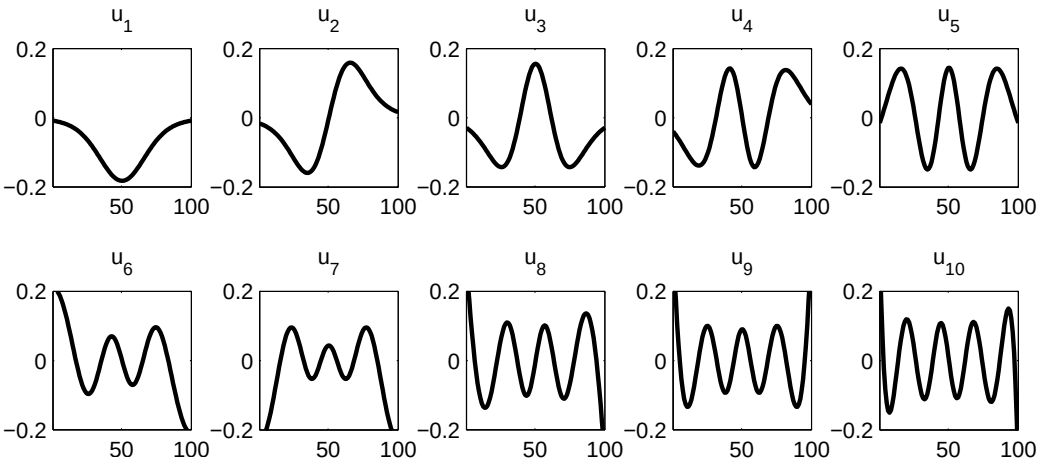
Subsequently, Vasilík derived several other methods for finding the noise revealing iteration $k_{noise} + 1$ that are based on the behaviour of the entities $|(p_1^k, e_1)|$. These are included in chapter 2 of [Vas11]. Note that Hnětynková, Plešinger, and Strakoš also derived a noise level estimate based on the amplification ratio (2.12). This estimate was however less reliable than (2.18).



(a) Fourier basis



(b) basis of discrete cosine transform



(c) left singular vectors

Figure 2.3: A comparison of Fourier basis, discrete cosine basis and left singular vectors of the matrix `shaw(100)`. See `example_FFT_basis.m`, `example_DCT_basis.m`, and `example_U_basis.m`.

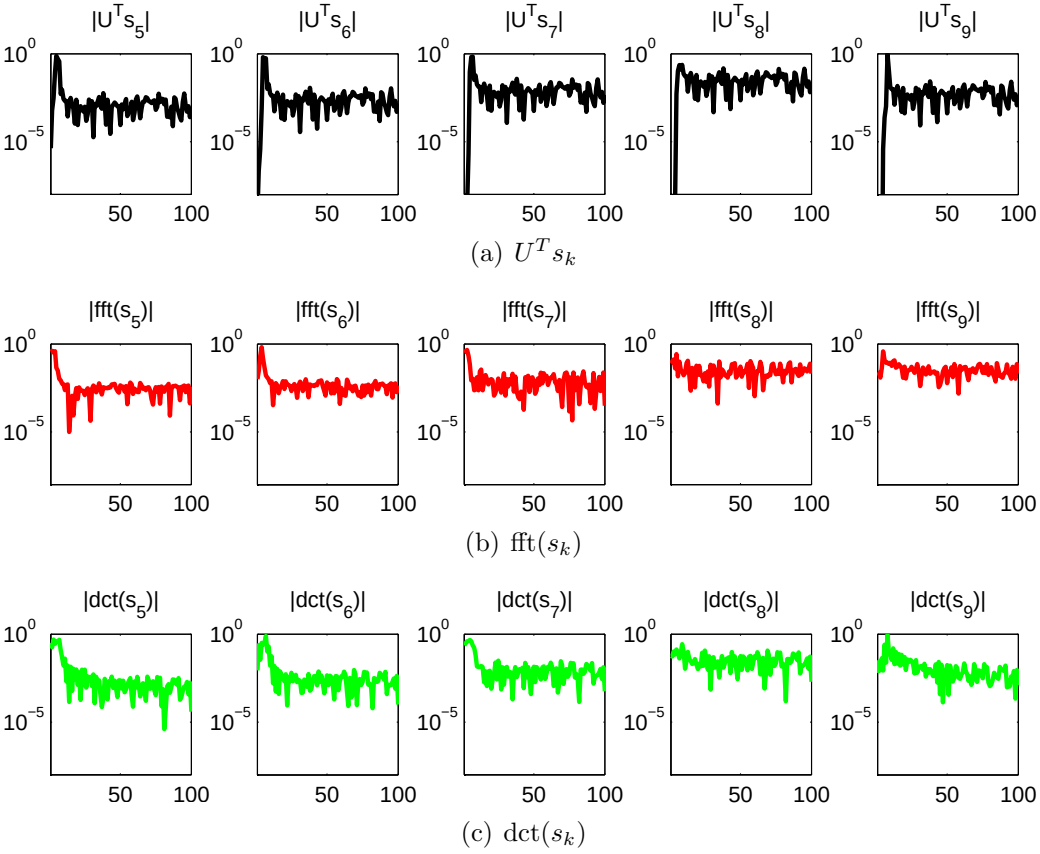


Figure 2.4: A comparison of the components of left bidiagonalization vectors in different bases for the problem `shaw(400)` identical with the problem from figure 2.2. See `example_s_k.m`.

Chapter 3

Denoising for Problems with White Noise

In this chapter, we return back to our original problem (1.13)

$$Ax \approx b, \quad b = b^{exact} + b^{noise} \quad \text{and} \quad \left\|b^{noise}\right\| \ll \left\|b^{exact}\right\|,$$

where multiplication by the matrix A has a distinct smoothing effect and b^{noise} is white noise from the normal distribution. Our aim is now to apply the results of [HPS09] covered in chapter 2 of this thesis to estimate the vector b^{noise}. Once we have the noise estimate $\tilde{b}^{noise}$, we can subtract it from the right-hand side b and compute the solution of the transformed problem

$$Ax \approx b^{denoised}, \quad \text{where} \quad b^{denoised} = b - \tilde{b}^{noise}. \tag{3.1}$$

We will refer to this process as *denoising* or *noise reduction*[1].

Obviously, we want the transformed problem (3.1) to have better overall properties than the original problem (1.13). In practice, the aim is either to decrease the relative noise level in the right-hand side

$$\frac{\left\|b^{denoised} - b^{exact}\right\|}{\|b^{exact}\|} < \frac{\|b - b^{exact}\|}{\|b^{exact}\|} = \frac{\|b^{noise}\|}{\|b^{exact}\|} \equiv \delta_{noise},$$

or to make the spectral properties of noise

$$b^{noise}_{trans} \equiv b^{denoised} - b^{exact}$$

more favorable, which usually means that the high frequencies in noise are dampened.

Implementation

All experiments in the remainder of the thesis are conducted on the same workstation under Windows XP with Matlab 7.10.0 (R2010a). The linear systems were adopted from the Regularization Tools [Han07, Han94] as well as the functions `tsvd`, `tikhonov`, and `l_curve` for spectral filtering methods. The Golub-Kahan bidiagonalization was computed using the code `bidiag_gk` available at

[1] Please note that in literature, denoising/noise reduction usually relates to reducing white noise in images. We hope that the reader will not find this little inconsistency with the remaining literature misleading.

`http://www.cs.cas.cz/krylov/` with default setting, i.e. with full double reorthogonalization. The noise level is measured relatively against the exact data, i.e. $\delta_{noise} = \frac{\|b-b^{exact}\|}{\|b^{exact}\|}$. White noise is generated by the built-in function `randn`, for coloured noise we adopted the function `spatialPattern.m` from MATLAB Central - File Exchange.

3.1 Noise reduction via spectral filtering

In some sense, spectral filtering method (1.15)

$$x^{reg} = \sum_{i=1}^{n} f_i \frac{u_i^T b}{\sigma_i} v_i$$

can be also viewed as a solution to a denoised problem. Modifying the equation (1.15), we get

$$\begin{aligned} x^{reg} &= \sum_{i=1}^{n} f_i \frac{u_i^T b}{\sigma_i} v_i \\ &= \sum_{i=1}^{n} \frac{u_i^T b}{\sigma_i} v_i - \sum_{i=1}^{n} (1-f_i) \frac{u_i^T b}{\sigma_i} v_i \\ &= \sum_{i=1}^{n} \frac{u_i^T b}{\sigma_i} v_i - \sum_{i=1}^{n} \frac{u_i^T (1-f_i)(u_i^T b) u_i}{\sigma_i} v_i \\ &= \sum_{i=1}^{n} \frac{u_i^T [b - (1-f_i)(u_i^T b) u_i]}{\sigma_i} v_i \\ &= A^{\dagger} \left[b - \sum_{i=1}^{n} (1-f_i)(u_i^T b) u_i \right] . \end{aligned} \tag{3.2}$$

The interpretation of (3.2) is following. Spectral filtering methods are mathematically equivalent to the Moore-Penrose pseudoinverse applied to the right hand side from which a part of spectral components was subtracted. Figure 3.2 illustrates the denoising process for the problem `shaw` using the truncated SVD and the Tikhonov's method. Figure 3.3 shows, how the Picard plot changes, when the new right-hand side $b - \sum_{i=1}^{n}(1-f_i)(u_i^T b)u_i$ is used (compare to figure 3.1). See `example_spectral_denoising.m`.

The spectral filtering methods, in the prospective of denoising, dampen the high-frequency part in the right-hand side, which allows $A^{\dagger}$ to be applied directly. We are not able to distinguish which part of the high frequencies in b belongs to b^{exact} and which belongs to noise. Thus, dampening all high-frequency components of b introduces a bias to the problem as described at the beginning of section 1.7. This means that the solution of spectral filtering method applied to the exact data b^{exact} does not match x^{exact}. Changing the regularization parameter we trade off the size of the bias and the dampening of large frequencies in b^{noise}.

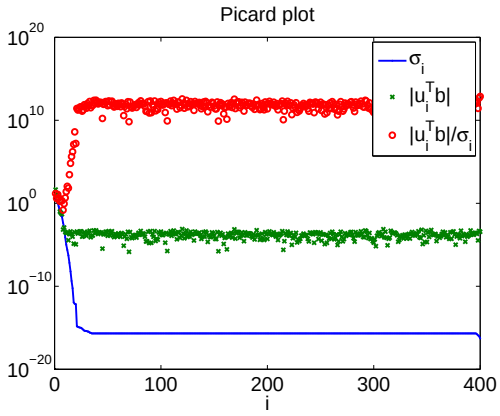
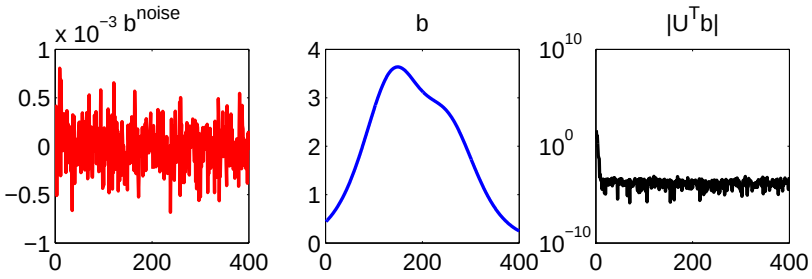
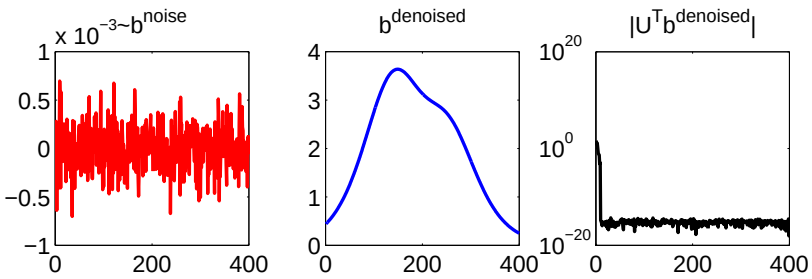


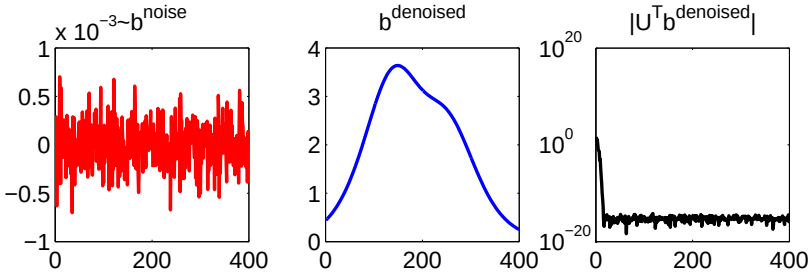
Figure 3.1: Picard plot for the problem `shaw(400)` with the relative noise level 10^{-4}.



(a) original problem

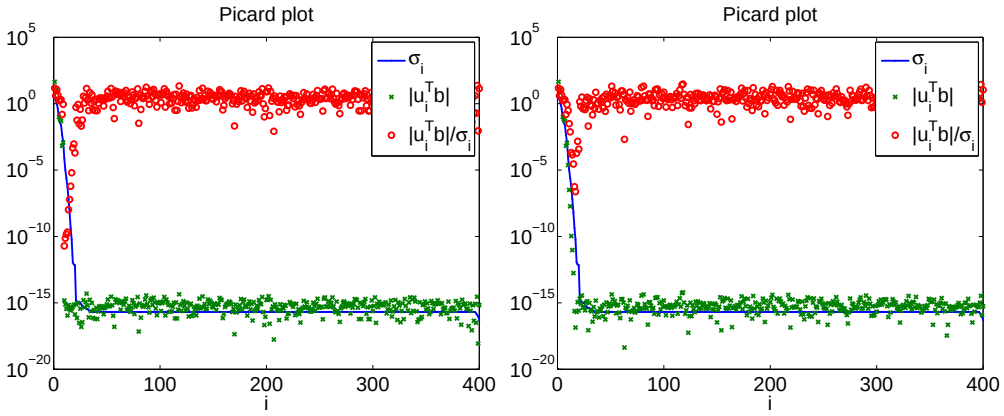


(b) problem denoised via T-SVD with $k = 9$



(c) problem denoised via Tikhonov regularization with $\lambda = 2.96 \cdot 10^{-4}$

Figure 3.2: Denoising via spectral filtering for the problem `shaw(400)` identical with the problem from figure 3.1. The left column shows the real noise (the first row) and the vectors that are (formally) subtracted from the right hand-side (the last two rows). The middle column contains plots of the right-hand side (in the last two rows after denoising). In the right column, we see the spectral components of the right-hand sides from the middle column.



(a) Picard plot for T-SVD (b) Picard plot for Tikhonov

Figure 3.3: Picard plot after denoising for `shaw(400)`

The spectral components are dampened depending only on their order in the singular value decomposition. This may lead to an elimination of a significant frequency in the data when some of the lower frequencies[2] are already dominated by noise. In other words, we can assume that these methods will work considerably worse for problems, where the projections $u_i^T b^{exact}$ oscillate and the discrete Picard condition (1.5) holds only 'on average', than for problems where $u_i^T b^{exact}$ decrease monotonically.

3.2 Noise reduction via noise revealing: Analysis

Let us now recall some of the results from chapter 2. We defined

$$\begin{aligned} s_1^{exact} &= b^{exact} / \|b\| \ , \\ s_1^{noise} &= b^{noise} / \|b\| \ , \\ \beta_{k+1} s_{k+1}^{exact} &\equiv A w_k - \alpha_k s_k^{exact} \ , \\ \beta_{k+1} s_{k+1}^{noise} &\equiv -\alpha_k s_k^{noise} \ , \end{aligned}$$

which yields

$$s_{k+1}^{noise} = -\frac{\alpha_k}{\beta_{k+1}} s_k^{noise} = (-1)^k \prod_{j=1}^{k} \frac{\alpha_j}{\beta_{j+1}} s_1^{noise} \equiv (-1)^k \rho_k^{-1} s_1^{noise} \ .$$

We emphasize that s_k^{noise} does not stand for the whole noise component of s_k but only for the part that stays white during the computation, i.e. has been multiplied only by a scalar. This partitioning of s_k plays the crucial role in the estimation of the noise vector.

In (3.1), we estimated the noise vector b^{noise}, and the estimate was then subtracted from the right-hand side. This is however equivalent to approximating

[2] frequency corresponding to the singular vector associated with a larger singular value

the exact data b^{exact} directly. In our case, using the equations above, we can rewrite the (scaled) exact data as follows

$$\begin{aligned}\frac{b^{exact}}{||b||} = s_1^{exact} &= s_1 - s_1^{noise} \\ &= s_1 - (-1)^k \rho_k s_{k+1}^{noise} \\ &= s_1 - (-1)^k \rho_k \left(s_{k+1}^{noise} + s_{k+1}^{exact} \right) + (-1)^k \rho_k s_{k+1}^{exact} \\ &= s_1 - (-1)^k \rho_k s_{k+1} + (-1)^k \rho_k s_{k+1}^{exact} ,\end{aligned}$$

yielding

$$b^{exact} = \|b\| \left[s_1 - (-1)^k \rho_k s_{k+1} + (-1)^k \rho_k s_{k+1}^{exact} \right] . \tag{3.3}$$

Although it may seem that we just substituted unknown entity b^{exact} with some other unknown entity s_{k+1}^{exact}, the properties of s_{k+1}^{exact} will allow us to transform the problem (1.13) in to a more convenient one.

Let us define a new right-hand side of the system as the known part of the equation (3.3), i.e.

$$b^{denoised} \equiv \|b\| \left[s_1 - (-1)^k \rho_k s_{k+1} \right] = b - (-1)^k \|b\| \rho_k s_{k+1} , \tag{3.4}$$

which means that we took

$$\tilde{b}^{noise} = \|b\| (-1)^k \rho_k s_{k+1} \tag{3.5}$$

as our noise estimate and

$$b^{denoised} - b^{exact} = - \|b\| (-1)^k \rho_k s_{k+1}^{exact} \tag{3.6}$$

represents noise in our new system $Ax \approx b^{denoised}$. In other words, we subtract the $(k+1)$-th left bidiagonalization vector multiplied by scalar $\|b\| (-1)^k \rho_k$ from the right-hand side of the system, which transforms the white broad-band white noise b^{noise} into $(-1)^{k+1} \|b\| \rho_k s_{k+1}^{exact}$. At this point it is not clear whether doing such a transformation is of any use.

As mentioned at the beginning, we want the noise $b^{denoised} - b^{exact}$ either to be of smaller size or to have better spectral properties than the original one.

Spectral properties

Our aim is now to find some 'convenient subspace' in which s_{k+1}^{exact} is contained. The phrase 'convenient subspace' should be understood as a subspace from which the spectral properties of s_{k+1}^{exact} can be deduced. Such a subspace can be found iteratively:

- $s_1^{exact} \in \text{span}(b^{exact})$ from its definition;
- $\beta_2 s_2^{exact} = A w_1 - \alpha_1 s_1^{exact}$, where $w_1 \in \text{span}(A^T b)$, which gives $s_2^{exact} \in \text{span}(AA^T b, b^{exact})$;
- $\beta_3 s_3^{exact} = A w_2 - \alpha_2 s_2^{exact}$, where $w_2 \in \text{span}((A^T A)A^T b, A^T b)$ and $s_2^{exact} \in \text{span}(AA^T b, b^{exact})$, which gives $s_3^{exact} \in \text{span}((AA^T)^2 b, AA^T b, b^{exact})$; etc.

Summarizing,

$$s_{k+1}^{exact} \in \operatorname{span}((AA^T)^k b, AA^T b, b^{exact}) \,. \tag{3.7}$$

Thus s_{k+1}^{exact} lies in a space spanned solely by smooth vectors (due to the smoothing properties of A). This means, in the perspective of denoising, that the white noise b^{noise} is replaced by noise that is more or less confined to low frequencies, when $k+1 \ll n$.

Size

Even when we already know that the spectral properties of noise are improved by replacing b by $b^{denoised}$, we do not want the relative size of $(-1)^{k+1} \|b\| \rho_k s_{k+1}^{exact}$ to be considerably larger than the original noise level δ_{noise}. We see that

$$\begin{aligned} \|b\| \left\|\rho_k s_{k+1}^{exact}\right\| &= \rho_k \|b\| \left\|s_{k+1} - s_{k+1}^{noise}\right\| \\ &\leq \rho_k \|b\| \|s_{k+1}\| + \rho_k \|b\| \left\|s_{k+1}^{noise}\right\| \\ &= \rho_k \|b\| + \rho_k \|b\| \left\|\rho_k^{-1} s_1^{noise}\right\| \\ &= \rho_k \|b\| + \|b\| \left\|s_1^{noise}\right\| \\ &= \rho_k \|b\| + \left\|b^{noise}\right\| \,. \end{aligned} \tag{3.8}$$

Thus in the worst case, we increase the error in the data by $\rho_k \|b\|$. We want, the $\rho_k \|b\|$ to be small, preferably the same order as the original noise $\|b^{noise}\|$. This is the point, where the noise revealing iteration $k_{noise}+1$ comes into play.

From section 2.3, we know that at the noise revealing iteration, s_{k+1} is significantly corrupted by the white noise , i.e.

$$1 = \|s_{k_{noise}+1}\| \leq c \left\|s_{k_{noise}+1}^{noise}\right\| = c \left\|\rho_{k_{noise}}^{-1} s_1^{noise}\right\| = c\rho_{k_{noise}}^{-1} \left\|b^{noise}\right\| / \|b\| \,, \tag{3.9}$$

where c is a moderate number. If $s_{k_{noise}+1}$ is dominated by the $s_{k_{noise}+1}^{noise}$, we can take $c = 2$. Equation (3.9) yields immediately

$$\rho_{k_{noise}} \|b\| \leq c \left\|b^{noise}\right\| \,. \tag{3.10}$$

Combining (3.8) with (3.10), we get

$$\|b\| \left\|\rho_{k_{noise}} s_{k_{noise}+1}^{exact}\right\| \leq (1+c) \left\|b^{noise}\right\| \,, \tag{3.11}$$

which means that the size of our new noise $(-1)^{k+1} \|b\| \rho_k s_{k+1}^{exact}$ is (at the noise revealing iteration) at most a small factor of the original $\|b^{noise}\|$. The size of the factor depends on how much the noise reveals at the noise revealing iteration.

Note that in (3.8), the worst scenario was assumed. Usually for small k_{noise}, we do not expect much cancellation between $s_{k_{noise}+1}^{exact}$ and $s_{k_{noise}+1}^{noise}$, therefore $\left\|s_{k_{noise}+1}^{exact}\right\|^2 + \left\|s_{k_{noise}+1}^{noise}\right\|^2 \approx 1$. Substituting into the first line of (3.8), we obtain

$$\|b\| \left\|\rho_k s_{k_{noise}+1}^{exact}\right\| \approx \rho_{k_{noise}+1} \|b\| - \left\|b^{noise}\right\| \leq (c-1) \left\|b^{noise}\right\| \,, \tag{3.12}$$

which means that we can expect the noise level in the right-hand side to be even reduced when $s_{k_{noise}+1}$ is dominated by the white noise.

Let us now summarize the denoising method proposed above. To eliminate white noise in the data, we suggest that $\|b\| (-1)^k \rho_k s_{k+1}$ is subtracted from the

right-hand side b at the noise revealing iteration. This transforms noise into low-frequency one with size of the same order as $\|b^{noise}\|$. In many cases, we can even expect that the relative noise level will become smaller after the denoising.

Till now, we supposed that the noise revealing iteration is found by some black-box function. In the following section, we develop a method that will determine the noise level iteration at a really negligible cost.

3.3 Determining the point of noise revealing

Let us recall that the noise revealing iteration is the iteration where the white noise s_{k+1}^{noise} significantly corrupts the left bidiagonalization vector s_{k+1}. From the equation (2.11) we know that the size of s_{k+1}^{noise} depends linearly on the amplification ratio ρ_k^{-1}, i.e.

$$\left\|s_{k+1}^{noise}\right\| = \rho_k^{-1}\left\|s_{k+1}^{noise}\right\| = \rho_k^{-1}\frac{\|b^{noise}\|}{\|b\|}. \tag{3.13}$$

This leads us to conclusion that the most significant noise revealing occurs, when the amplification ratio $\rho_k^{-1} = \prod_{j=1}^{k}\frac{\alpha_j}{\beta_{j+1}}$ is maximal. Therefore we decided to redefine the term 'noise revealing iteration'. From this point on, $k_{noise}+1$ denotes the iteration, where ρ_k^{-1} reaches its maximum, i.e.

$$k_{noise}+1 \equiv \operatorname*{argmax}_k \rho_k^{-1} + 1. \tag{3.14}$$

Computing ρ_k^{-1}s and finding their maximum requires $\mathcal{O}(n)$ operations, which represents a negligible part of the total computational cost.

This definition of noise revealing iteration is *different* from the definition of [HPS09] in the sense that (3.14) is the *most* noise revealing iteration, while Hnětynková, Plešinger, and Strakoš defined the noise revealing iteration as the *first* iteration, where the left bidiagonalization vector becomes dominated by s_{k+1}^{noise}. In many cases, these two definition coincide. However, sometimes ρ_k^{-1} oscillates and they differ. An example, when ρ_k^{-1} oscillates, is shown in figure 3.4.

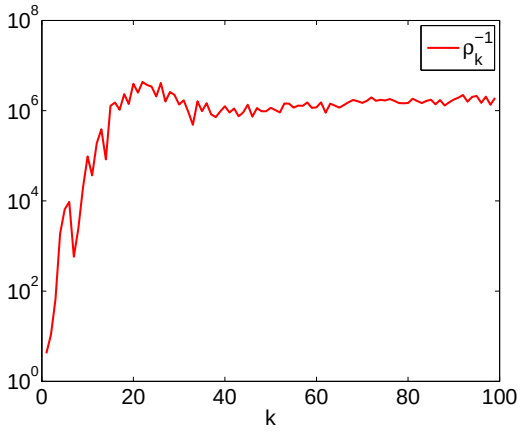
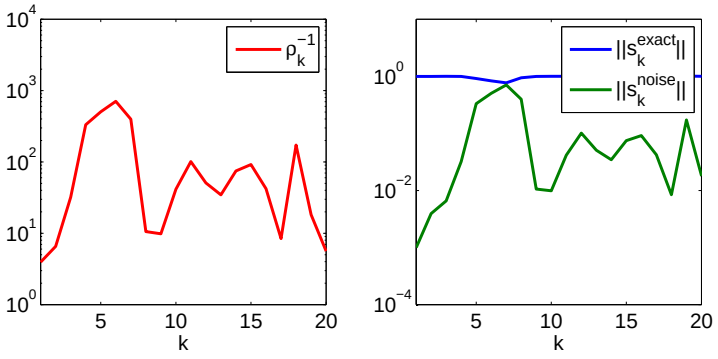


Figure 3.4: An example of oscillating amplification ratio – problem `phillips(400)` with the relative noise level 10^{-7}. See `example_ampl_ratio2.m`.

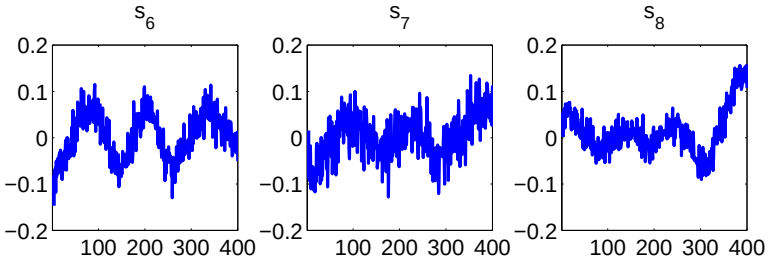
3.4 Noise reduction via noise revealing: Experiments

The aim of this section is to computationally verify the theoretical results of section 3.2. We have chosen three problems from Regularization Tools: `shaw(400)`, `phillips(400)`, and `foxgood(100)`, and carried out 5 various experiments.

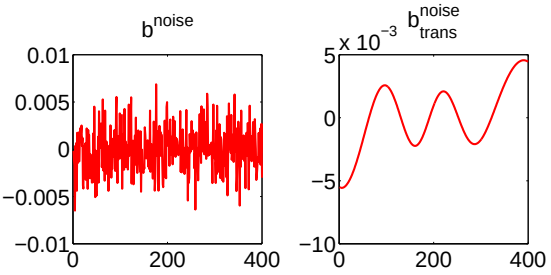
Experiment 1. The first experiment demonstrates the qualitative properties of the denoising. Noise of the relative size 10^{-3} was added to the right-hand



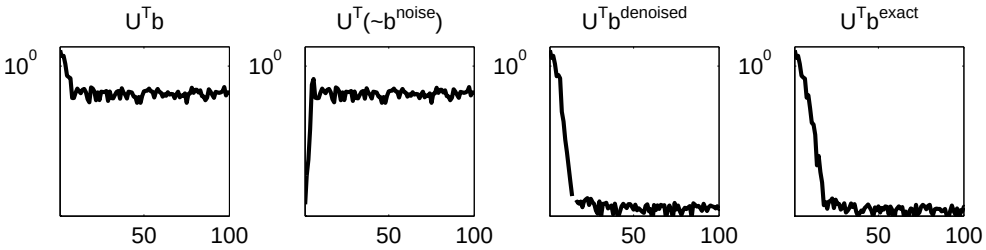
(a) amplification ratio and the size of s_{k+1}^{noise}



(b) the noise revealing vector (middle)



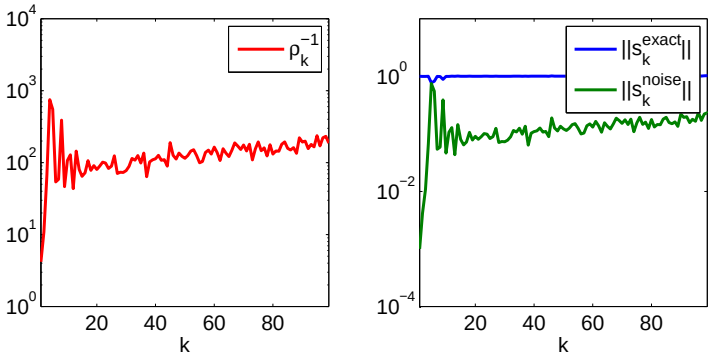
(c) noise transformation



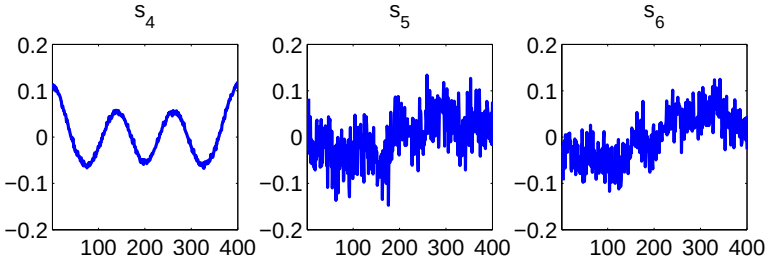
(d) spectral analysis (noisy data, noise estimate, denoised data, exact data)

Figure 3.5: Denoising for the problem `shaw(400)` with the noise level 10^{-3}.

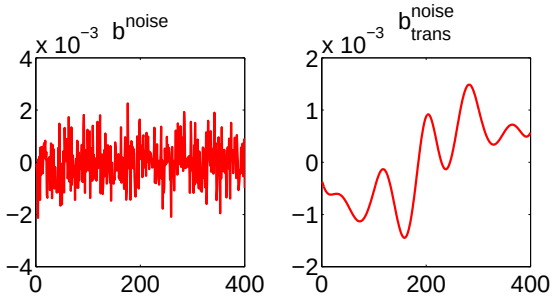
side. Results are shown in figures 3.5, 3.6, and 3.7. Each plot (a) depicts, how the behavior of the relative size of white noise resembles the behavior of the amplification ratio ρ_k^{-1}. Plot (b) shows the left bidiagonalization vector at the noise revealing iteration together with its preceding and succeeding vector to emphasize the revealing of noise. In plot (c), we see how white noise is transformed to a low-frequency one. And finally, in the last plot (d), the change of spectral properties is demonstrated (only the first quarter is plotted). Code is available under `test1.m`.



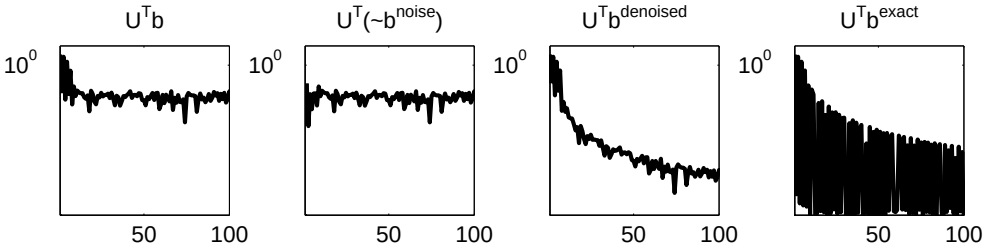
(a) amplification ratio and the size of s_{k+1}^{noise}



(b) the noise revealing vector (middle)



(c) noise transformation

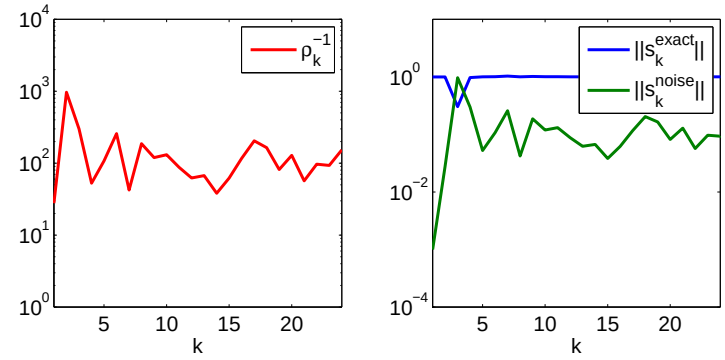


(d) spectral analysis (noisy data, noise estimate, denoised data, exact data)

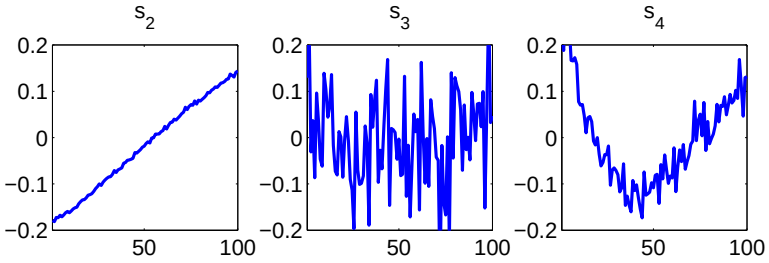
Figure 3.6: Denoising for the problem `phillips(400)` with the noise level 10^{-3}.

For `shaw`, the peak of the amplification curve is blunt, i.e. $\rho_{k_{noise}-1}$, $\rho_{k_{noise}}$, and $\rho_{k_{noise}+1}$ are of comparable size. This relates to a less significant noise revealing at $k_{noise} + 1 = 7$. For the other two matrices, the peak is sharp, which means that the noise revealing vector $s_{k_{noise}+1}$ is significantly more oscillating than the two surrounding vectors.

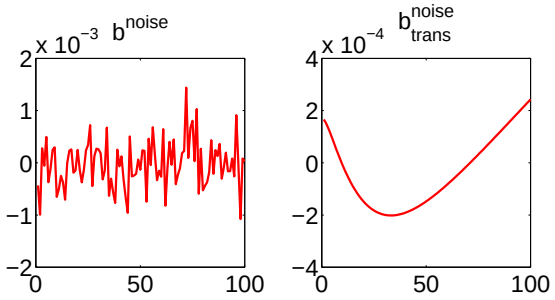
The right plots in part (a) demonstrate, how much white noise actually dominates the noise revealing vector. For the first two matrices at the noise revealing iteration, the vectors $s^{exact}_{k_{noise}+1}$ and $s^{noise}_{k_{noise}+1}$ are of about the same size, so we cannot



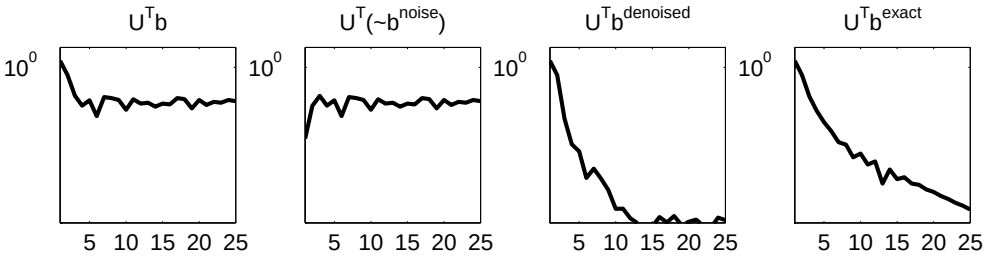
(a) amplification ratio and the size of s^{noise}_{k+1}



(b) the noise revealing vector (middle)



(c) noise transformation



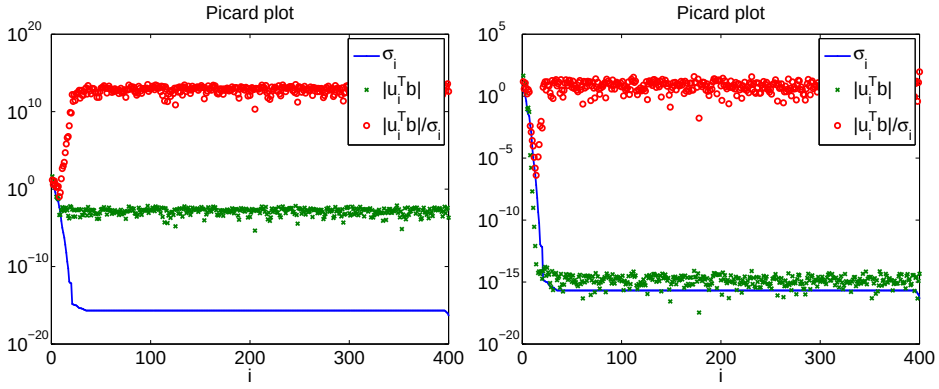
(d) spectral analysis (noisy data, noise estimate, denoised data, exact data)

Figure 3.7: Denoising for the problem `foxgood(100)` with the noise level 10^{-3}.

expect a reduction of noise concerning the norm. For `foxgood`, the noise revealing vector is strongly dominated by $s^{noise}_{k_{noise}+1}$, which means that $s_{k_{noise}+1}$ resembles the original b^{noise} and we can expect a significant decrease of the noise level after denoising. Moreover, the noise reveals very quickly - in the third iteration. This results in a very smooth noise vector b^{noise}_{trans}.

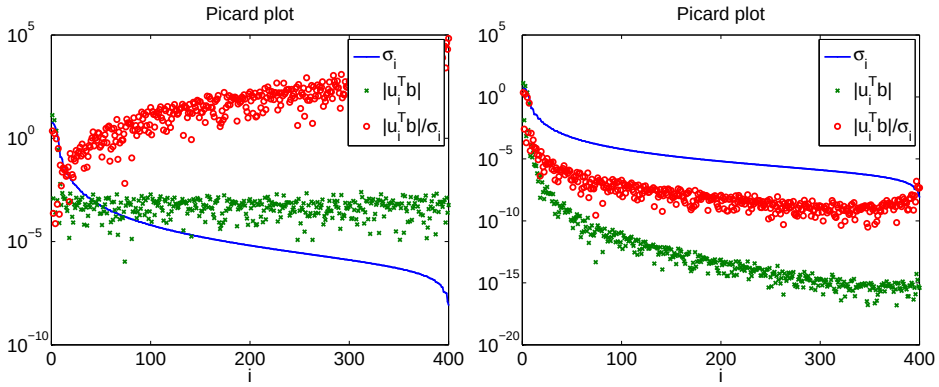
Experiment 2. In the second experiment, we focus on spectral properties of the right-hand side $b^{denoised}$. As mentioned in section 1.4, the discrete Picard condition plays a crucial role in solving discrete ill-posed problems. How the Picard plot changes after denoising is shown in figures 3.8, 3.9, and 3.10. Code is a part of `test1.m`.

We see that in all cases, the Picard plot improved in the sense that the projections of $u_i^T b^{denoised}$ decay faster than the singular values σ_i. Note that for problems `shaw` and `foxgood`, the discrete Picard condition becomes violated when the computation reaches machine precision $\sim 10^{-16}$. This is purely a problem of finite precision arithmetic, which is not a concern of this work.



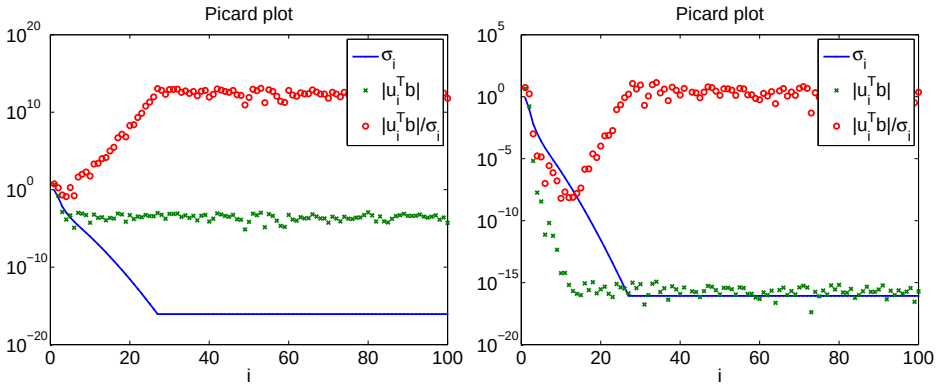
(a) before denoising (b) after denoising

Figure 3.8: Picard plots for `shaw(400)` identical with the problem from the figure 3.5.



(a) before denoising (b) after denoising

Figure 3.9: Picard plots for `phillips(400)` identical with the problem from the figure 3.6.



(a) before denoising (b) after denoising

Figure 3.10: Picard plots for `foxgood(100)` identical with the problem from the figure 3.7.

Experiment 3. In this experiment, we study quantitative properties of the denoising based on noise revealing. To our three testing problems, we added two

`shaw(400)`				
δ_{noise}	1e-002	1e-004	1e-006	1e-008
$k_{noise}+1$	5	8	10	13
$\rho_{k_{noise}}$	1.09e-002	1.05e-004	1.32e-006	1.15e-008
$\left\|\left\|b_{trans}^{noise}\right\|\right\| / \left\|\left\|b^{exact}\right\|\right\|$	4.57e-003	3.67e-005	8.73e-007	6.30e-009
`phillips(400)`				
δ_{noise}	1e-002	1e-004	1e-006	1e-008
$k_{noise}+1$	5	9	16	32
$\rho_{k_{noise}}$	1.43e-002	1.44e-004	1.40e-006	2.28e-008
$\left\|\left\|b_{trans}^{noise}\right\|\right\| / \left\|\left\|b^{exact}\right\|\right\|$	1.02e-002	1.08e-004	1.07e-006	2.08e-008
`foxgood(100)`				
δ_{noise}	1e-002	1e-004	1e-006	1e-008
$k_{noise}+1$	3	4	5	7
$\rho_{k_{noise}}$	1.03e-002	1.07e-004	1.18e-006	1.18e-008
$\left\|\left\|b_{trans}^{noise}\right\|\right\| / \left\|\left\|b^{exact}\right\|\right\|$	3.26e-003	4.31e-005	6.90e-007	7.54e-009
`i_laplace(100,1)`				
δ_{noise}	1e-002	1e-004	1e-006	1e-008
$k_{noise}+1$	6	10	14	17
$\rho_{k_{noise}}$	1.70e-002	1.42e-004	1.30e-006	1.22e-008
$\left\|\left\|b_{trans}^{noise}\right\|\right\| / \left\|\left\|b^{exact}\right\|\right\|$	1.44e-002	1.15e-004	9.51e-007	8.71e-009
`baart(400)`				
δ_{noise}	1e-002	1e-004	1e-006	1e-008
$k_{noise}+1$	3	5	6	7
$\rho_{k_{noise}}$	1.36e-002	1.08e-004	1.00e-006	1.12e-008
$\left\|\left\|b_{trans}^{noise}\right\|\right\| / \left\|\left\|b^{exact}\right\|\right\|$	9.51e-003	4.19e-005	1.65e-007	5.46e-009

Table 3.1: Noise level in the data (first row), nose revealing iteration (second row), amplification ratio at the noise revealing iteration (third row) and the relative size of the transformed noise (last row). The results were averaged over 10 randomly chosen vectors b^{noise}.

more problems from Regularization tools - `i_laplace(100,1)` and `baart(400)`. The results can be found in table 3.1. Code is available under `test2.m`.

The aim is to investigate, for different noise levels, when the amplification ratio ρ_1^{-1} reaches its maximum and how the relative size of the noise changes after the denoising. The results of this experiment support our hypothesis that $\rho_{k_{noise}}$ is a small factor of the original noise level δ_{noise} and that the size of noise can be significantly reduced via denoising, when noise reveals in the first few steps[3].

Experiment 4. Let us recall that the aim of denoising is to dampen the high frequencies of noise in the right-hand side, which would be otherwise amplified when solving the system of linear equations (1.13). Although it is not in the main focus of the thesis, we will investigate now what happens when we compute the naive solution of the transformed problem, i.e.

$$x^{denoised} \equiv A^{\dagger} b^{denoised}. \tag{3.15}$$

This solution will be subsequently compared to the naive solution $A^{\dagger}b$. We would like to emphasize that applying the pseudoinverse of A to the denoised right-hand side is not the way how the solution should be computed in practice and here we did it only for *illustrative purposes*. The results in figures 3.11, 3.12, and 3.13 confirm a distinct regularizing effect of the proposed denoising. Code is available under `test3.m`.

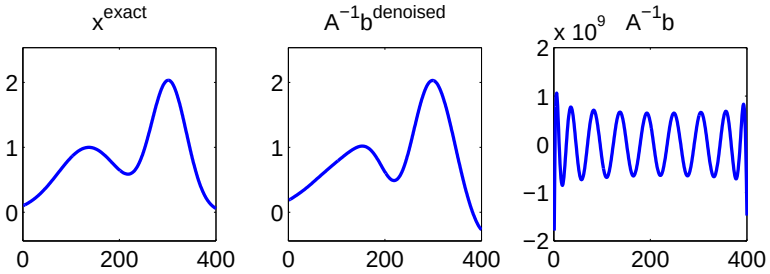


Figure 3.11: Regularizing effect of denoising based on noise revealing for the problem `shaw(400)` with the noise level 10^{-3}.

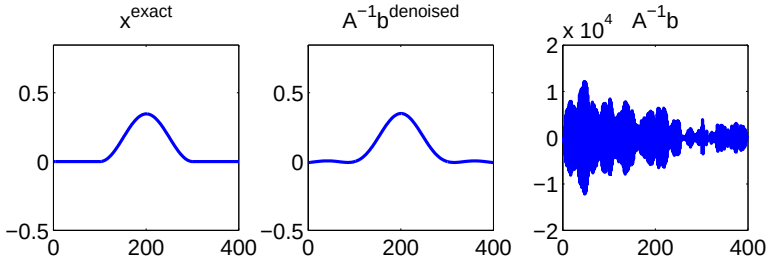


Figure 3.12: Regularizing effect of denoising based on noise revealing for the problem `phillips(400)` with the noise level 10^{-3}.

[3]For some matrices or for small noise levels, the noise revealing does not have to appear, or it appears in the last few iterations. Then $\rho_{k_{noise}}$ can be even smaller than δ_k, because there is a large cancellation between $s^{exact}_{k_{noise}+1}$ and $s^{noise}_{k_{noise}+1}$, and $\left\|s^{exact}_{k_{noise}+1}\right\|^2 + \left\|s^{noise}_{k_{noise}+1}\right\|^2$ is significantly larger than $\left\|s_{k_{noise}+1}\right\|^2 = 1$.

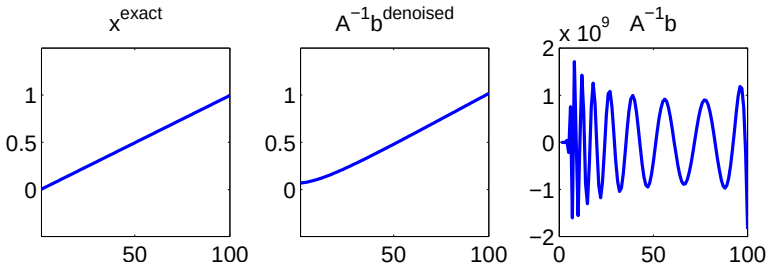


Figure 3.13: Regularizing effect of denoising based on noise revealing for the problem `foxgood(100)` with the noise level 10^{-3}.

Experiment 5. Our aim will be now, in connection with experiment 4, to compare the regularizing effect of the denoising procedure and the spectral filtering methods - truncated SVD and Tikhonov. Here, $x^{denoised}$ is computed as described above, regularization parameters for the T-SVD and Tikhonov method are determined via the L-curve. We do realize that standard spectral filtering methods

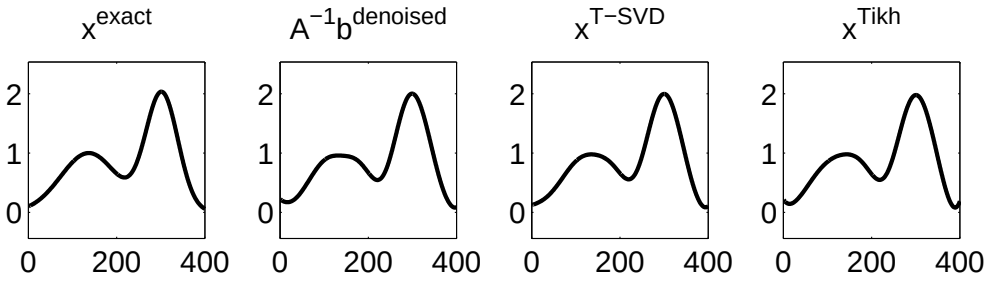


Figure 3.14: A comparison of regularization methods for the problem `shaw(400)` with the noise level 10^{-5}.

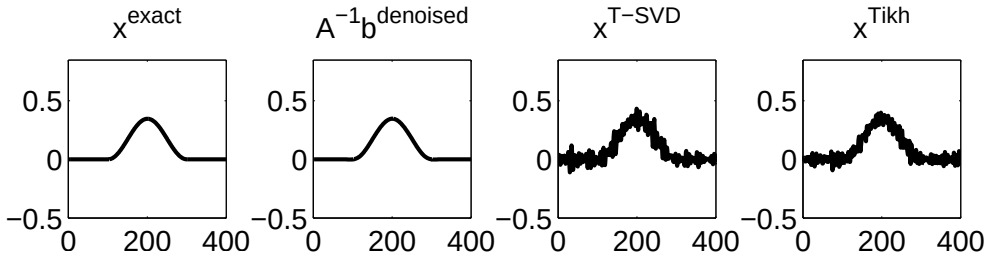


Figure 3.15: A comparison of regularization methods for the problem `phillips(400)` with the noise level 10^{-5}.

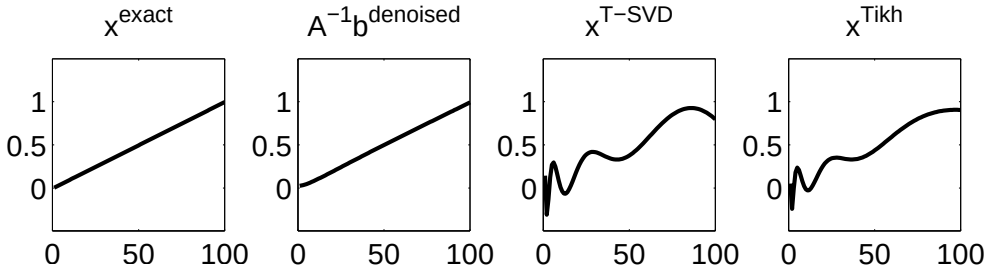


Figure 3.16: A comparison of regularization methods for the problem `foxgood(100)` with the noise level 10^{-5}.

combined with the L-curve are not the state-of-the-art methods. However, they are often the methods of the first choice, therefore we find this comparison relevant. The noise level was decreased to 10^{-5} to amplify the differences between the methods. The results can be found in figures 3.14, 3.15, and 3.16. Code is available under `test4a.m`. We can see that for the last two problems, denoising provides much better results.

A more comprehensive study, measuring the relative error of the solution, forms the second part of this experiment. The results are included in table 3.2. Code is available under `test4b.m`.

`shaw(400)`				
δ_{noise}	1e-002	1e-004	1e-006	1e-008
denoising	1.69e-001	4.75e-002	3.20e-002	9.09e-003
T-SVD	7.26e-002	3.30e-002	4.26e-002	1.50e-002
Tikhonov	7.42e-002	3.49e-002	2.86e-002	1.43e-002

`phillips(400)`				
δ_{noise}	1e-002	1e-004	1e-006	1e-008
denoising	4.68e-002	8.50e-003	1.03e-003	1.41e-004
T-SVD	3.57e-002	1.95e-001	1.76e+000	1.88e-001
Tikhonov	5.43e-002	1.17e-001	3.01e-001	9.93e-002

`foxgood(100)`				
δ_{noise}	1e-002	1e-004	1e-006	1e-008
denoising	4.01e-002	8.41e-003	2.20e-003	7.30e-004
T-SVD	3.24e-002	2.79e-002	9.29e-002	8.81e-002
Tikhonov	4.19e-002	4.95e-002	6.47e-002	5.90e-002

Table 3.2: Noise level in the data (first row), relative errors of solutions (second to fourth row). The results were averaged over 10 randomly chosen vectors b^{noise}.

As we can see, for `shaw`, denoising is comparable to the other two methods. For `phillips` and `foxgood`, denoising seems to be superior to spectral filtering. Solving problem `phillips` is generally considered a difficult task, because the projections $u_i^T b^{exact}$ oscillate extensively. The L-curve does not have then the shape of the letter L (particularly for small noise levels) and it is difficult to determine the optimal regularization parameter.

Let us now summarize the results of performed experiments. We observed a distinct regularization effect of denoising without a substantial increase of the noise level in the data. Moreover, for the presented problems, the proposed denoising is at least comparable to spectral filtering. In addition, finding the optimal parameter $k_{noise}+1$ (the point of noise revealing) can be done at a negligible cost.

We recall that all the experiments were performed via the Golub-Kahan bidiagonalization with full double reorthogonalization. However, similar results can be obtained for bidiagonalization without reorthogonalization. In this case we expect the noise revealing to be delayed as studied in [HPS09]. We would like

to emphasize that to investigate the robustness of this kind of denoising, a more comprehensive testing would be necessary. At this point, we find the results very promising.

Chapter 4

Propagation of Other Types of Noise

So far, we assumed that in (1.13) the data is polluted by Gaussian white noise. This is however not always the case. In this chapter, we illustrate how noise propagation and noise revealing described in chapter 2 transforms when noise comes from a different distribution. The aim is to present some ideas about what we can expect for different types of noise, not to provide a thorough analysis. Therefore, most of the content will be comprised of experiments.

To stay consistent with the work of Hnětynková, Plešinger, and Strakoš [HPS09], and Vasilík [Vas11], we restrict ourselves to the problem `shaw`. The following types of noise will be considered: white noise from uniform distribution, data correlated noise, broad-band high-frequency and low-frequency noise. Their definition and properties can be found in section 1.5 of this thesis. The size of noise will be measured in the Euclidian norm, because measurements in the norm corresponding to particular distribution (which is in practice often unknown) would make the results of experiments hard to compare.

4.1 Data correlated and uniform white noise

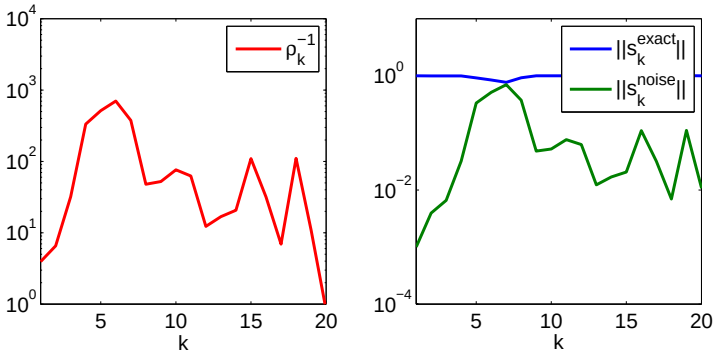
In this part, we investigate noise propagation of two types of noise:

1. uniform white noise,
2. data correlated Gaussian noise[1].

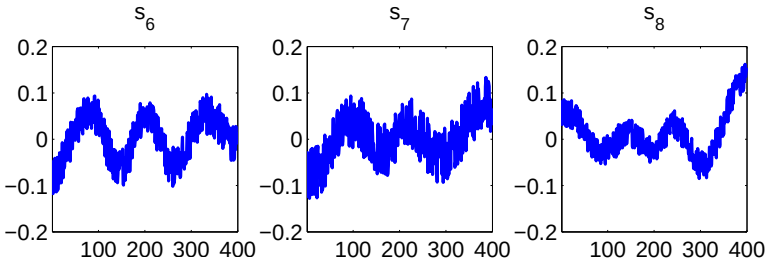
We decided to include both types of noise in one section because they have similar spectral properties - both are white-noise-like[2], i.e. the covariance matrix of their Fourier transform is close to identity. Due to this property, we expect their propagation in the Golub-Kahan bidiagonalization to be similar to the propagation of Gaussian white noise. Therefore, we repeated the experiments of section 3.4. Their results are shown in figures 4.1 and 4.2.

[1] The right-hand side of the problem `shaw` can be found in figure 3.2. From its shape, we expect the data correlated noise to have the middle components amplified and the border components dampened.

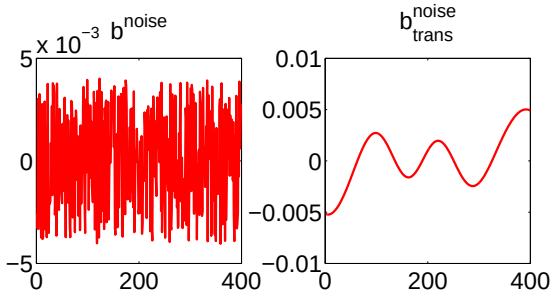
[2] White noise is white-noise-like from its definition.



(a) amplification ratio and the size of s_{k+1}^{noise}



(b) the noise revealing vector (middle)

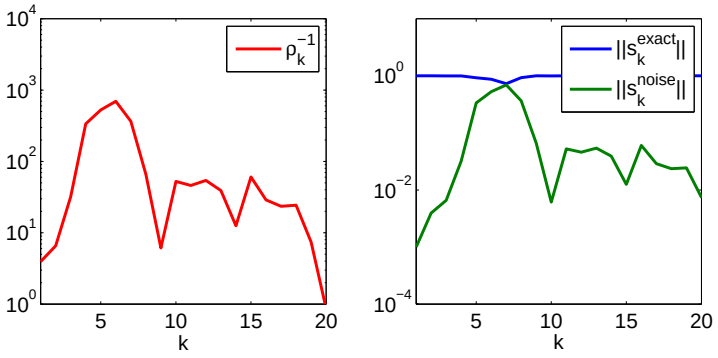


(c) noise transformation

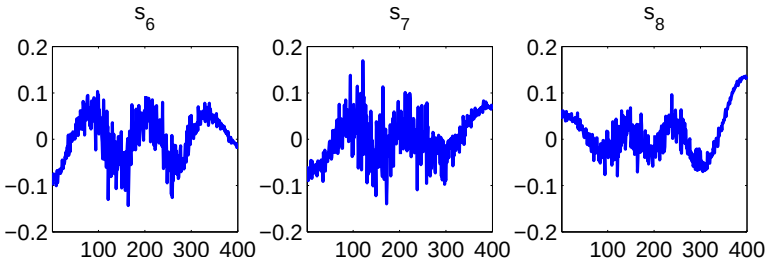
Figure 4.1: Denoising for the problem `shaw(400)` with uniform white noise of relative size $\delta_{noise} = 10^{-3}$. See `test5.m`.

As we see, the behaviour of ρ_k^{-1} for uniform and data correlated noise resembles the behaviour of the amplification ratio for white noise from normal distribution, see figure 3.5. For this particular setting, noise reveals at the same iteration for all considered types of noise. Although the individual left bidiagonalization vectors look different, their low-frequency part s_k^{exact} remains almost unchanged. Therefore, the transformed noise b_{trans}^{noise} more or less coincides with the transformed Gaussian noise.

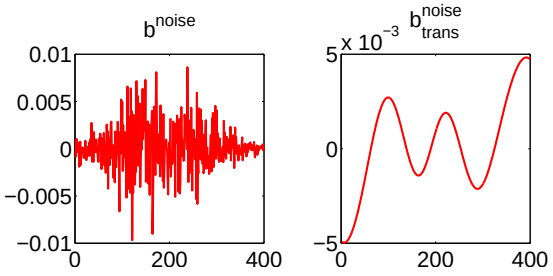
These preliminary results let us believe that passing from Gaussian white noise to another white-noise-like noise will not change the propagation of noise through the bidiagonalization process significantly and the results of chapter 3 could be possibly generalized in this sense. This however requires a thorough analysis and more comprehensive testing, which we hope will be done in near future.



(a) amplification ratio and the size of s_{k+1}^{noise}



(b) the noise revealing vector (middle)



(c) noise transformation

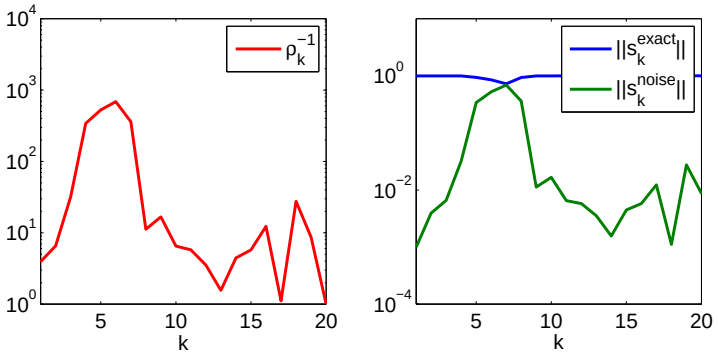
Figure 4.2: Denoising for the problem `shaw(400)` with data correlated noise of relative size $\delta_{noise} = 10^{-3}$. See `test5.m`.

4.2 Coloured noise

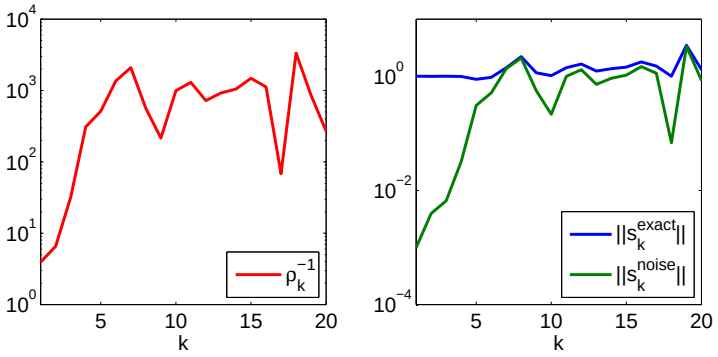
Coloured noise, compared to uniform or data correlated noise, amplifies/dampens some of the frequencies. Generally speaking, high-frequency noise is considered the more convenient alternative when solving a discrete inverse problem because it is usually easier to determine when the projections $u_i^T b$ become dominated by the noise component and the discrete Picard condition becomes violated.

Some results regarding coloured noise and noise revealing are already included in [Vas11]. However, they are confined only to the noise level estimate (2.18). We would like to focus more on the noise propagation itself.

For our experiments, the high-frequency noise will be represented by violet noise and the low-frequency noise by Brownian noise, see table 1.1 for their definitions. We expect the intermediate colours of noise, i.e. blue and pink, to inherit the tendencies of violet and Brownian noise, but these tendencies will be less pronounced.



(a) violet noise



(b) Brownian noise

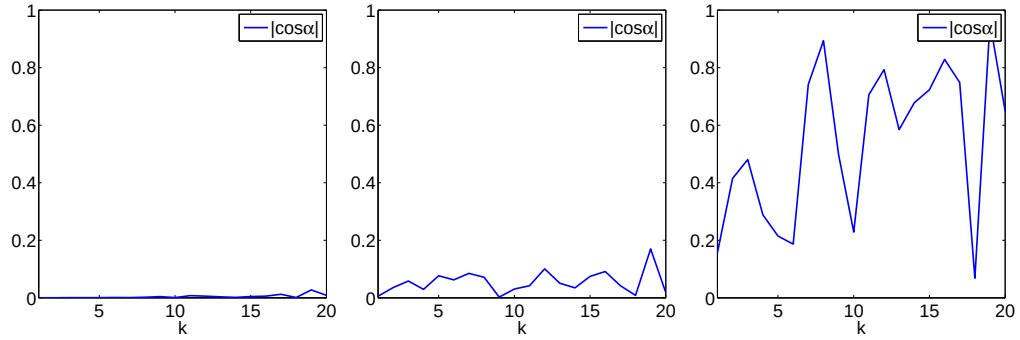
Figure 4.3: Noise propagation for problem `shaw(400)` with high-frequency and low-frequency noise of relative size $\delta_{noise} = 10^{-3}$. See `test5.m`.

First of all, we investigate, how the colour of noise influences the behaviour of the amplification ratio ρ_k^{-1}. Their plots are shown in figure 4.3.

For violet noise the amplification ratio looks similar to the amplification ratio for white noise in figure 3.5. However, for violet noise ρ_k^{-1} after reaching its maximum oscillates back around its starting value ρ_1^{-1}. For white noise it stays above this value. This is caused by different size of cancelation between s_k^{exact} and s_k^{noise}. Roughly speaking, for white noise at the point of noise revealing, the noise is approximated by s_k up to the low-frequency part s_k^{exact}. For violet noise, the low-frequencies are dampened and noise is therefore approximated more accurately. Subsequent orthogonalization of s_{k+1} against s_k causes noise to be almost fully projected out.[3] Therefore, we expect denoising to work for high-frequency noise comparably to white noise or even better.

In contrast, for Brownian low-frequency noise the cancellation between s_k^{exact} and s_k^{noise} is noticeable, see figure 4.4. Having revealed for the first time, noise stays present in the s_ks without being projected out. In this case, the relation (3.12) does not hold and we cannot expect a significant reduction of the noise level after denoising. Moreover, the amplification ratio reaches its maximum much later than for the white-noise case, i.e. the regularization effect is limited. In other words, for low-frequency noise the proposed denoising procedure subtracts s_k too late and we eliminate only the highest frequencies. This delay

[3]Note that for the problem `shaw` this is done in three subsequent iterations because the noise reveals only partially, i.e. s_k is not fully dominated by s_k^{noise}.



(a) violet noise (b) white noise (c) Brownian noise

Figure 4.4: Relative projection of s_k^{noise} onto s_k^{exact} for problem `shaw(400)` with noise of relative size $\delta_{noise} = 10^{-3}$; $\cos\alpha \equiv \frac{(s_k^{noise}, s_k^{exact})}{\|s_k^{noise}\| \|s_k^{exact}\|}$. See `test6.m`.

causes the problem (3.1) to stay still severely ill-posed. Insufficient regularization is a shortcoming of many other methods for solving discrete inverse problems with low-frequency noise including the methods covered in this thesis. Therefore, if one expect the noise in the right-hand side to be low-frequency, we recommend using a method that is adapted for such type of noise.

Chapter 5

Noise Level Estimate

One of the results of Hnětynková, Plešinger, and Strakoš [HPS09] was a method to estimate the noise level in the data. As mentioned in section 2.3, they studied the behaviour of $|(p_1^{(k)}, e_1)|$ – the first component of the left singular vector corresponding to the smallest singular value of the bidiagonal matrix L_k. At some point, this value starts to stagnate. Let us denote this point as k_{stag}, see

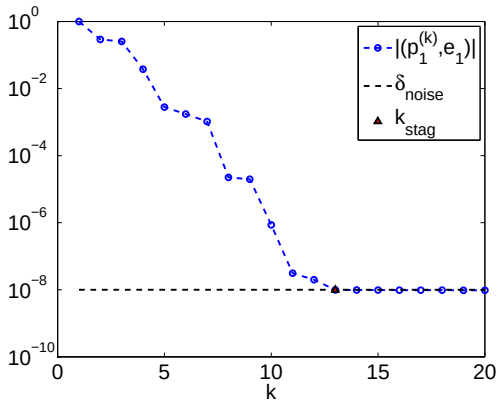


Figure 5.1: Iteration k_{stag} for the problem `shaw(400)` with the noise level 10^{-8}. See `example_stagnation.m`.

figure 5.1. The noise level is then estimated as

$$\delta_{noise} \approx |(p_1^{(k_{stag})}, e_1)|. \tag{5.1}$$

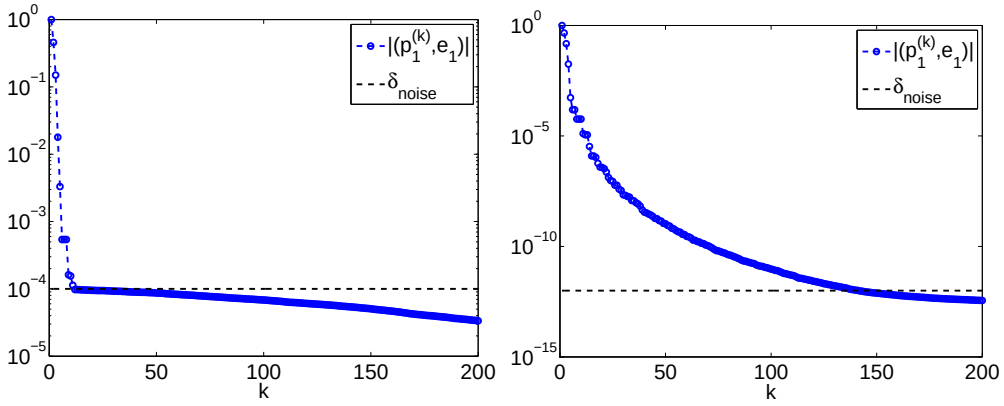
In [HPS09], an automated method for finding k_{stag} was derived, see (2.17). Subsequently, Vasilík in [Vas11] proposed some other ways of determining k_{stag}. However, they were all based on the behaviour of $|(p_1^{(k)}, e_1)|$. Therefore, this entity has to be computed in each step. This represents additional computational cost, even though this cost is low.

In section 3.3, we introduced a new definition of the point of noise revealing,

$$k_{noise} + 1 \equiv \operatorname*{argmax}_k \rho_k^{-1} + 1.$$

Due to the fact that, according to [HPS09], the iteration k_{stag} corresponds to the noise revealing iteration, we propose to estimate the noise level as

$$\delta_{noise} \approx |(p_1^{(k_{noise}+1)}, e_1)|, \quad \text{where} \quad k_{noise} + 1 = \operatorname*{argmax}_k \rho_k^{-1}. \tag{5.2}$$



(a) $\delta_{noise} = 10^{-4}$ (b) $\delta_{noise} = 10^{-12}$

Figure 5.2: A disappearance of stagnation for the problem `phillips(400)` for a low noise level (right), compare to the plot on the left. See `example_stagnation_phillips.m`.

The subject of this chapter will be to verify reliability and robustness of the estimate (5.2). Note that in some cases, the stagnation may not occur, especially when the noise level is low[1]. In this case, we are not able to determine the noise level using neither (2.17) nor (5.2). An example is shown in figure 5.2.

To compare the proposed method with the stagnation criterion (2.17), we repeated the experiment from section 4 of [HPS09]. To the problems `shaw(400)` and `i_laplace(100,1)` we added the problem `gravity(400)` and `baart(400)` that are used in [Vas11]. Results are shown in table 5.1, the code is available under `test7.m`.

We see that the estimates usually overestimate the noise level in the data. When the noise level is underestimated, the difference is only a few percent. In these cases, the point where the stagnation starts is determined correctly. Whether the stagnation starts above or bellow the actual noise level depends mostly on the problem and the noise level. The influence of the waveform of the vector b^{noise} is minor, provided noise is white. We believe that the over-/underestimation property depends somehow on the distribution of the singular/Ritz values around the noise level, but at the moment we are not able to provide a trustworthy explanation.

From the results of the experiments we conclude that for the tested problems, the criterion based on the noise revealing is comparably reliable as the stagnation criterion (2.17). Both techniques allow to estimate the noise level very accurately.

Let us now look at some of the cases for which the two estimates differ – `gravity(400)` with the noise level 10^{-8} and `baart(400)` with the noise level 10^{-4}.

As already observed in [Vas11], for the problem `gravity` the corner we are looking for is rounded, i.e. it is difficult to determine even visually where the stagnation starts. This relates to the fact that at two successive iterations, the noise revealing is of comparable rate, see figure 5.3. This situation will be further discussed later in this section.

For `baart`, situation is different. The entity $|(p_1^{(k)}, e_1)|$ decays very quickly

[1]We observed that in cases when there is no visually detectable stagnation and the noise level is small, the Picard condition is usually not violated and regularization is not necessary.

shaw(400)					
δ_{noise}	1e-002	1e-004	1e-006	1e-010	1e-014
k_{stag}	5	8	10	14	17
$\lvert(p_1^{k_{stag}}, e_1)\rvert$	1.03e-002	1.02e-004	1.31e-006	9.88e-011	1.80e-014
rel. error	3 %	2 %	31 %	1 %	80 %
$k_{noise}+1$	5	8	10	14	18
$\lvert(p_1^{k_{noise}+1}, e_1)\rvert$	1.03e-002	1.02e-004	1.30e-006	9.88e-011	9.83e-015
rel. error	3 %	2 %	30 %	1 %	2 %

i_laplace(100,1)					
δ_{noise}	1e-001	1e-002	1e-007	1e-010	1e-013
k_{stag}	3	6	15	20	23
$\lvert(p_1^{k_{stag}}, e_1)\rvert$	1.12e-001	1.01e-002	1.34e-007	1.26e-010	9.17e-014
rel. error	12 %	3 %	37 %	34 %	8 %
$k_{noise}+1$	3	5	16	20	23
$\lvert(p_1^{k_{noise}+1}, e_1)\rvert$	1.12e-001	1.18e-002	9.92e-008	9.45e-011	9.17e-014
rel. error	12 %	19 %	6 %	6 %	8 %

gravity(400)					
δ_{noise}	1e-001	1e-002	1e-004	1e-008	1e-012
k_{stag}	3	5	10	19	28
$\lvert(p_1^{k_{stag}}, e_1)\rvert$	1.22e-001	1.25e-002	1.10e-004	1.12e-008	1.06e-012
rel. error	22 %	25 %	10 %	12 %	6 %
$k_{noise}+1$	3	5	10	18	27
$\lvert(p_1^{k_{noise}+1}, e_1)\rvert$	1.22e-001	1.25e-002	1.32e-004	1.40e-008	1.42e-012
rel. error	22 %	25 %	32 %	40 %	42 %

baart(400)					
δ_{noise}	1e-001	1e-002	1e-004	1e-008	1e-012
k_{stag}	2	3	4	6	9
$\lvert(p_1^{k_{stag}}, e_1)\rvert$	1.96e-001	1.37e-002	2.68e-004	8.37e-008	2.15e-012
rel. error	96 %	37 %	168 %	737 %	115 %
$k_{noise}+1$	3	3	5	7	10
$\lvert(p_1^{k_{noise}+1}, e_1)\rvert$	9.96e-002	1.33e-002	1.00e-004	1.11e-008	9.89e-013
rel. error	1 %	33 %	0 %	11 %	1 %

Table 5.1: Noise level in the data (first row), the point of stagnation determined by (2.17) (second row), the point of noise revealing (3.14) (fifth row), the corresponding noise level estimates (third and sixth row), and the relative errors of the estimates (fourth and seventh row). Results were averaged over 100 randomly chosen vectors b^{noise}.

to the noise level, but before it reaches the level, 'quasi-stagnation' occurs – difference between two successive entities $|(p_1^{(k)}, e_1)|$ is much smaller than it was for smaller ks. See figure 5.4. The stagnation criterion evaluates this situation as 'real' stagnation and the noise estimate is therefore inaccurate. In the view of amplification ratio ρ_k^{-1}, a distinct noise revealing occurs before the noise revealing iteration $k_{noise}+1$. In this case, the criterion based on noise revealing outperforms the criterion based on stagnation. The performance of stagnation criterion could

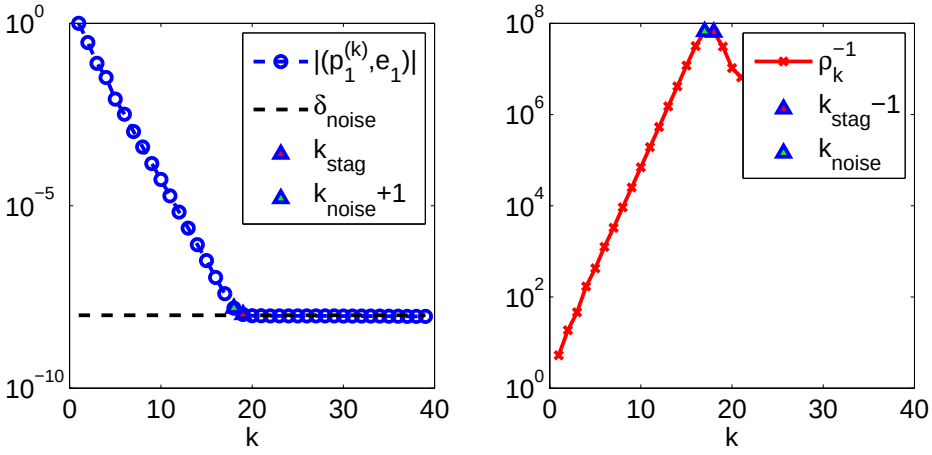


Figure 5.3: A comparison of the two noise level estimators for the problem `gravity(400)` with the noise level 10^{-8}. See `test8.m`.

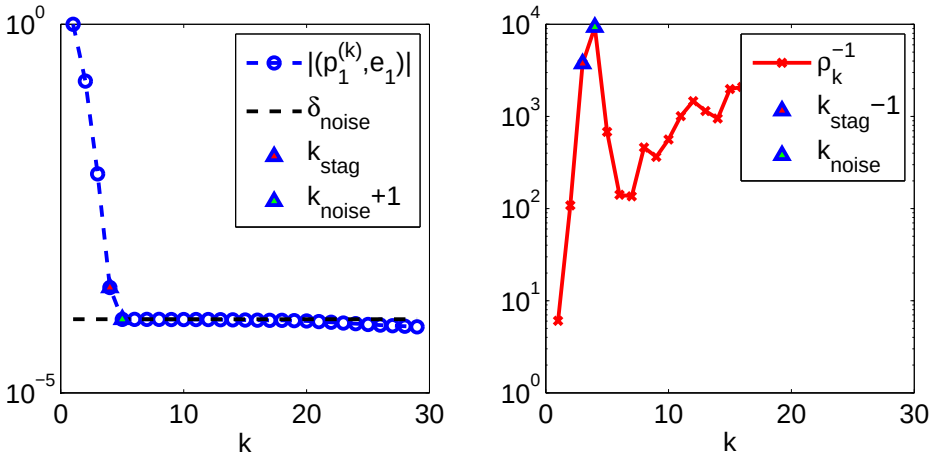


Figure 5.4: A comparison of the two noise level estimators for the problem `baart(400)` with the noise level 10^{-4}. See `test8.m`.

be possibly improved by a change of the parameter ζ.

There are inverse problems for which the noise estimator (5.2) fails (even when a visually detectable stagnation is present). This, as observed, happens in cases when noise reveals in the iteration $k_{noise}+1$, but after a few iterations it is revealed again. In other words, the amplification ratio after reaching its maximum stays or oscillates close to its maximal value. To some extent it also holds for the problem `gravity(400)`, see figure 5.3. Another example is shown in figure 5.5. Here, the point of maximal noise revealing is the iteration 5. But the actual stagnation starts at iteration 9, i.e. at the point where the amplification ratio ρ_k^{-1} is back close to its maximum and noise is revealed again. In such cases, the noise level would be estimated more accurately if we take the last k for which noise reveals significantly as our $k_{noise}+1$ in (5.2). This would however require a human interaction we want to avoid.

Roughly speaking, we can expect the criterion (5.2) to provide a very reliable estimate of the noise level in cases when after the noise revealing itera-

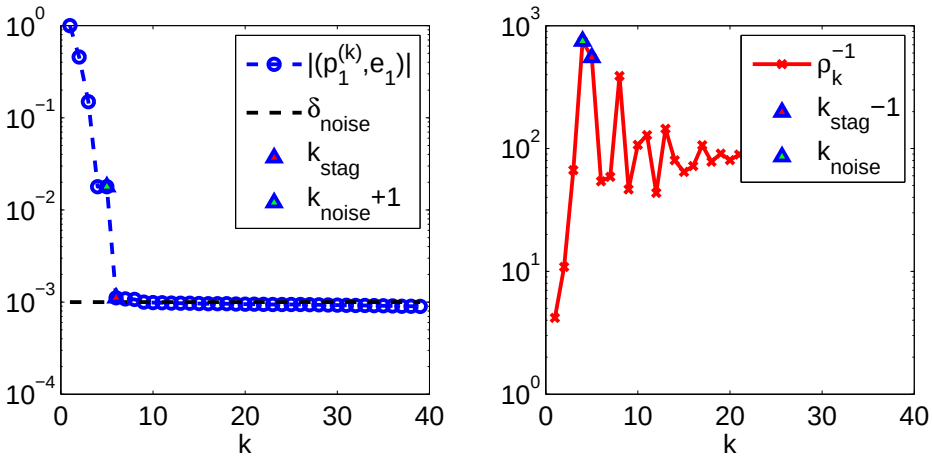


Figure 5.5: An example of a problem for which the noise level estimate based on maximal noise revealing fails - the problem `phillips(400)` with the noise level 10^{-3}. See `test8.m`.

tion $k_{noise}+1$, noise is almost projected out and the subsequent left bidiagonalization vectors are much smoother than $s_{k_{noise}+1}$.

It is worth noting that there is a slight difference between the computational aspects of stagnation and the noise revealing criterion. To determine the iteration k_{stag} by the criterion (2.17), always only few additional iterations (their number corresponds to the parameter *step*) have to be computed. By contrast, finding the maximum of ρ_k^{-1} might be a tricky task, as there is theoretically no upper bound for the number of iterations to be computed. In practice, we assume that the maximum is reached within a few iterations and when ρ_k^{-1} starts to decrease[2], we terminate the bidiagonalization process.

We emphasize that all results in this chapter are confined to white noise and to the bidiagonalization with full double reorthogonalization. As observed in [Vas11], in the bidiagonalization without reorthogonalization, a series of short stagnations may occur and it may be difficult to determine the final stagnation even visually.

[2]on average as the values ρ_k^{-1} may oscillate

Conclusion

In this thesis, we focused on linear inverse problems, where the coefficient matrix A has a distinct smoothing effect and the right-hand side b is contaminated by unknown noise.

Following [HPS09], we studied noise propagation through the Golub-Kahan iterative bidiagonalization. We identified the iteration in which white noise reveals in the left bidiagonalization vector. This knowledge was then used to denoise the right-hand side, i.e. to subtract high-frequency noise from b. This method is considered optimal in the sense that it regularizes the problem while introducing (locally) minimal perturbation into b^{exact}.

Several numerical experiments were performed in order to show sufficient regularization effect of the proposed method combined with reasonable perturbation of the data. The method was also compared to spectral filtering, showing its superiority for some testing ill-posed problems. The aim of denoising is to eliminate high-frequency noise in the data. Solving the resulting denoised problem remains an open problem, because in some cases further regularization by spectral filtering might by advantageous.

We also investigated propagation of noise of other types. We conclude that the method could be possibly used for noise that is white-noise-like or for high-frequency noise. For low-frequency noise, the proposed method provides insufficient regularization and should not be used in the current form.

Knowing the point of noise revealing, we modified the noise level estimator of [HPS09]. We proposed a method that is fully automated, and at a negligible cost provides estimate that is comparable to the original one.

Throughout the thesis we considered the Golub-Kahan iterative bidiagonalization with full double reorthogonalization. We believe that similar results could be obtained using bidiagonalization without reorthogonalization. This will however require further investigation.

Bibliography

[ANR74] N. Ahmed, T. Natarajan, and K. R. Rao. Discrete cosine transform. *Computers, IEEE Transactions on*, C-23(1):90–93, 1974.

[BES98] Å. Björck, T. Elfving, and Z. Strakoš. Stability of Conjugate Gradient and Lanczos methods for linear least squares problems. *SIAM J. Matrix Anal. Appl.*, 19(3):720–736, 1998.

[Bjö88] Å. Björck. A bidiagonalization algorithm for solving large and sparse ill-posed systems of linear equations. *BIT Numerical Mathematics*, 28(3):659–670, 1988.

[Bjö96] Å. Björck. *Numerical Methods for Least Squares Problems*. SIAM, 1996.

[Cra55] E. J. Craig. The N-step iteration procedures. *J. Math. and Physics*, 1955.

[CS07] D. Calvetti and E. Somersalo. *An Introduction to Bayesian Scientific Computing: Ten Lectures on Subjective Computing*. Springer-Verlag, 2007.

[DTHP+12] J. Duintjer Tebbens, I. Hnětynková, M. Plešinger, Z. Strakoš, and P. Tichý. *Analýza metod pro maticové výpočty: Základní metody*. Matfyzpress, 2012.

[EHN00] H. W. Engl, M. Hanke, and A. Neubauer. *Regularization of Inverse Problems*. Kluwer Academic Publishers, 2000.

[Eld77] L. Eldén. Algorithms for the regularization of ill-conditioned least squares problems. *BIT Numerical Mathematics*, 17(2):134–145, 1977.

[FF63] D. K. Fadeev and V. N. Fadeeva. *Computational Methods of Linear Algebra*. W. H. Freeman and Company, 1963.

[Fre03] I. Fredholm. Sur une classe d'équations fonctionnelles. *Acta Mathematica*, 27(1):365–390, 1903.

[GD04] B. M. Gray and L. D. Davisson. *An Introduction to Statistical Signal Processing*. Cambridge University Press, 2004.

[GHW79] 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.

[GK65] G. Golub and W. Kahan. Calculating the singular values and pseudo-inverse of a matrix. *SIAM Journal on Numerical Analysis*, 2(2):205–224, 1965.

[Gro90] C. W. Groetsch. *The Theory of Tikhonov Regularization for Fredholm Integral Equations of the First Kind*. Pitman Advanced Publishing Program, 1990.

[Had02] J. Hadamard. Sur les problèmes aux dérivés partielles et leur signification physique. *Princeton University Bulletin*, 13:49–52, 1902.

[Han71] R. J. Hanson. A numerical method for solving Fredholm integral equations of the first kind using singular values. *SIAM Journal on Numerical Analysis*, 8(3):616–622, 1971.

[Han87] P. C. Hansen. The truncated SVD as a method for regularization. *BIT Numerical Mathematics*, 27(4):534–553, 1987.

[Han88] P. C. Hansen. Computation of the singular value expansion. *Computing*, 1988.

[Han90] P. C. Hansen. The discrete Picard condition for discrete ill-posed problems. *BIT Numerical Mathematics*, 30(4):658–672, 1990.

[Han94] P. C. Hansen. A MATLAB package for analysis and solution of discrete ill-posed problems. *Numerical Algorithms*, 6(1):1–35, 1994.

[Han98] P. C. Hansen. *Rank-Deficient and Discrete Ill-Posed Problems: Numerical Aspects of Linear Inversion*. SIAM, 1998.

[Han07] P. C. Hansen. Regularization tools version 4.0 for Matlab 7.3. *Numerical Algorithms*, 46(2):189–194, 2007.

[Han10] P. C. Hansen. *Discrete Inverse Problems: Insight and Algorithms*. SIAM, 2010.

[HNO06] P. C. Hansen, J. G. Nagy, and D. P. O'Leary. *Deblurring Images: Matrices, Spectra, and Filtering*. SIAM, 2006.

[HO93] P. C. 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.

[HPS06] I. Hnětynková, M. Plešinger, and Z. Strakoš. Lanczos tridiagonalization, Golub-Kahan bidiagonalization and core problem. *PAMM*, 6:717–718, 2006.

[HPS07] I. Hnětynková, M. Plešinger, and Z. Strakoš. Lanczos tridiagonalization and core problems. *Linear Algebra and Its Applications*, 421:243–251, 2007.

[HPS09] I. Hnětynková, M. Plešinger, and Z. Strakoš. The regularizing effect of the Golub-Kahan iterative bidiagonalization and revealing the noise level in the data. *BIT Numerical Mathematics*, 49(4):669–696, 2009.

[HS52] M. R. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, 49(6):409–436, 1952.

[Lan50] 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.

[Leg05] A. M. Legendre. Sur la méthode des moindres quarrés. In *Nouvelles méthodes pour la détermination des orbites des comètes*. 1805.

[LMR+07] M. A. Little, P. E. McSharry, S. J. Roberts, D. A. E. Costello, and I. M. Moroz. Exploiting nonlinear recurrence and fractal scaling properties for voice disorder detection. *BioMedical Engineering OnLine*, 6(23), 2007.

[Mor66] V. A. Morozov. On the solution of functional equations by the method of regularization. *Soviet Math. Dokl.*, 7:414–417, 1966.

[Mor84] V. A. Morozov. *Methods for Solving Incorrectly Posed Problems*. Springer Verlag, 1984.

[MS06] G. Meurant and Z. Strakoš. The Lanczos and conjugate gradient algorithms in finite precision arithmetic. *Acta Numerica*, 15:471–542, 2006.

[Pic10] É. Picard. Sur un théorème général relatif aux équations intégrales de première espèce et sur quelques problèmes de physique mathématique. *Rendiconti del Circolo Matematico di Palermo*, 29:79–97, 1910.

[PS82a] C. C. Paige and M. A. Saunders. Algorithm 583: Sparse linear equations and least square problems. *ACM Trans. Math. Soft.*, 8:195–209, 1982.

[PS82b] C. C. Paige and M. A. Saunders. LSQR: An algorithm for sparse linear equations and sparse least squares. *ACM Trans. Math. Soft.*, 8:43–71, 1982.

[PS02] C. C. Paige and Z. Strakoš. Scaled total least squares fundamentals. *Numerische Mathematik*, 91(1):117–146, 2002.

[Sch07] E. Schmidt. Zur theorie der linearen und nichtlinearen integralgleichungen. *Mathematische Annalen*, 63(4):433–476, 1907.

[Tik63] A. N. Tikhonov. Solution of incorrectly formulated problems and the regularization method. *Soviet Math. Dokl.*, 4:1035–1038, 1963.

[Var73] J. M. Varah. On the numerical solution of ill-conditioned linear systems with applications to ill-posed problem. *SIAM Journal on Numerical Analysis*, 10(2):257–267, 1973.

[Vas11] K. Vasilík. Linear error-in-variable modeling. Master's thesis, Charles University in Prague, 2011.

[VL05] J. Vergés Llahí. *Color Constancy and Image Segmentation Techniques for Applications to Mobile Robotics*. PhD thesis, Universitat Politècnica de Catalunya, 2005.

[Vog02] C. R. Vogel. *Computational Methods for Inverse Problems*. SIAM, 2002.

Appendix

Riemann-Lebesgue lemma

Theorem (Riemann-Lebesgue lemma)**.** *Let $f \in L^1(a,b)$. Then*

$$\lim_{n\to\infty} \int_a^b f(x) e^{-inx}\, dx = 0.$$

That is, the Fourier transform of an L^1 function vanishes at infinity.

Lanczos tridiagonalization and Golub-Kahan bidiagonalization

An interesting relation between Lanczos and bidiagonalization process is briefly explained here. A detailed discussion can be found in [HPS06, HPS07].

The Lanczos tridiagonalization of the matrix AA^T with the starting vector $s_1 = b/\beta_1$, $\beta_1 = \|b\|$, yields in k steps the symmetric tridiagonal matrix T_k such that (using the notation of section 2.1)

$$AA^T S_k = S_k T_k + \alpha_k \beta_{k+1} s_{k+1} e_k^T,$$

and

$$T_k \;=\; L_k L_k^T \;=\; \begin{bmatrix} \alpha_1^2 & \alpha_1\beta_2 & & \\ \alpha_1\beta_2 & \alpha_2^2+\beta_2^2 & \ddots & \\ & \ddots & \ddots & \alpha_{k-1}\beta_k \\ & & \alpha_{k-1}\beta_k & \alpha_k^2+\beta_k^2 \end{bmatrix},$$

i.e. the bidiagonal matrix L_k from the Golub-Kahan bidiagonalization can be considered the Cholesky factor of the tridiagonal matrix T_k from the Lanczos tridiagonalization.