Parameter Estimation for Large-Scale Reconstruction

Steffen Brun Kjøller

Kongens Lyngby 2007
IMM-M.Sc.-2007-75

Technical University of Denmark
Informatics and Mathematical Modelling
Building 321, DK-2800 Kongens Lyngby, Denmark
Phone +45 45253351, Fax +45 45882673
reception@imm.dtu.dk
www.imm.dtu.dk

IMM-M.Sc: ISSN 0909-3192

Summary

The focus of this thesis is on solving linear inverse problems, using Tikhonov Regularization. For the Tikhonov Regularization, a regularization parameter is needed, describing the balance between being true to the data and getting a smooth solution. When solving small problems, it is possible to compute a curve, called an L-Curve outlining this balance. A good regularization parameter can be determined as the corner point on such a curve.

We will come up with an estimated L-Curve from just a few points. This will make it possible to locate a corner point and thereby a good regularization parameter, for large-scale problems where it is not possible to compute the L-Curve.

It is important for this method to work, that the estimated L-Curve has the same properties as the exact L-Curve. We start by showing that the basic idea is useful by solving small inverse problems. Since the problems are small, we can compute the L-Curve, and thereby we compare the computed L-Curve and corner point, with the estimated L-Curve and the corner point from this.

Hereafter we will try some attempts to improve the method. Not all these attempts improve the method of estimating an L-Curve. Instead some of the attempts will be used for showing the limitations of the method. Among these are the choice of data points, which is important to get reasonable results. They have to represent the whole curve, so there is enough information to estimate the L-Curve.

We will describe the requirements of a good choice of data points, as well

as, how to ensure that even for a badly chosen set of data points a reasonable solution can be obtained. That is, to add points, making sure that the data points are placed reasonably.

Finally two large-scale problems will be solved to demonstrate the method in use, before a conclusion on the results can be made. Moreover some thoughts on areas for future improvements are presented.

Resumé

Denne rapport fokuserer på løsning af linære inverse problemer ved hjælp af Tikhonov regularisering. Ved Tikhonov regularisering skal man bruge en regulariseringsparameter til at beskrive balancen mellem at være troværdig overfor data og for at finde en glat løsning. Ved små problemer kan man beregne en kurve, kaldet en *L-Kurve*, som beskriver denne balance. Et udtryk for en god regulariseringsparameter kan findes ved hjælp af et hjørnepunkt på denne kurve.

Vi vil finde en tilnærmet L-Kurve ud fra få punkter. Dette gør det muligt at finde et hjørnepunkt og dermed en god regulariseringsparameter for storskala problemer, hvor det ikke er muligt at beregne en L-Kurve.

For at denne fremgangsmåde virker, er det vigtigt, at den tilnærmede L-Kurve har samme egenskaber som den eksakte L-Kurve. Først viser vi, at den grundlæggende idé er brugbar ved at løse nogle små inverse problemer. Da problemerne er små, kan L-Kurven beregnes, og dermed kan denne samt dens hjørnepunkt sammenlignes med denne tilnærmede L-Kurve og dennes hjørnepunkt.

Herefter vil vi følge nogle forsøg på forbedringer af fremgangsmåden. Ikke alle disse forsøg vil kunne forbedre metoden med at lave en tilnærmet L-Kurve. I stedet vil nogle af disse kunne bruges til at belyse nogle af de begrænsninger, der findes ved metoden. Dette inkluderer, blandt andet, valget af datapunkter, som er vigtigt for at få et fornuftigt resultat. Disse skal være placeret så hele kurven er repræsenteret, så der findes information nok til at kunne estimere L-Kurven.

Vi vil vise kravene, der skal være til et godt valg af punkter samt nogle ideer til, hvordan man kan sikre, selv med et dårligt valg, at en fornuftig løsning

opnås. Dette kan gøres ved at tilføje punkter løbende og dermed sikre, at data punkternes placeringer vil være fornuftige.

Afslutningsvis i rapporten vil der blive løst to storskala problemer for at demonstrere metoden i praksis, inden der kan konkluderes på de opnåede resultater. Ydermere er der gjort nogle tanker om fremtidige områder til forbedringer.

Preface

This thesis documents the work done as a final step in becoming a civil engineer in applied mathematics from the Technical University of Denmark. The project was started the 2^{nd} of January and handed in the 23^{23} of August and is a collaboration between the two departments IMM and MAT. It represents a workload of 30 European Credit Transfer System for one person.

During the project I have received much help for numerous people. Some as techical discussions, others as nutritious treats when the sugar level was low. I want to thank these people but especially my two supervisors, Jens Gravesen for inspirational discussion, and Per Christian Hansen for being a constant source of motivation and his amount of insight in the subject
bigskip

Kgs. Lyngby, October 9, 2007

Steffen Brun Kjøller
Student no. s011304

List of Symbols

Throughout this thesis, different symbols and operators will be used. Here follows a list of the symbols. In general non-capital letters have been used for vectors and scalars, whereas capital letters are used for matrices, except in Section 6, where they are used for parameters. We use superscript as part of the symbol name in order to indicate a specific situation for the symbols. Subscripts are used for indices.

Symbol	Description
A	Transformation matrix
a_{ij}	Element in A
x	Sought model
x^{exact}	Exact model
b	Transformed data
b^{exact}	Noise free data
λ	Regularization parameter
k	SVDtruncation parameter
L	Additional information for the Tikhonov Regularization
κ	Curvature
f_i	i^{th} filter factor
ξ	Error level
ε_i	Error element
U	Matrix holding the left singular vectors, u_i
u_i	i^{th} left singular vector
V	Matrix holding the right singular vectors, v_i

Continued on the next page

– continued from previous page

Symbol	Description
v_i	i^{th} right singular vector
Σ	Matrix holding the singular values, σ_i
σ_i	i^{th} singular value
σ_l	Singular value, corresponding to the noise level
$\tilde{u_i}$	Discretized u_i
$\tilde{v_i}$	Discretized v_i
μ_i	Singular values in the continuous case
s	Independent variable
t	Independent variable
$k(s,t)$	Kernel function
$f(t)$	Continuous model
$g(s)$	Continuous data
K	Operator
$k(s,t)$	Kernel function
$\langle\ .\ ,\ .\ \rangle$	Inner product
arg min	Argument for the minimum
trace	Sum of the elements on the main diagonal in the matrix
$\|\ .\ \|$	Two-Norm
$\|\ .\ \|_2$	Two-Norm
$\|\ .\ \|_{\mathrm{F}}$	Frobenius-Norm
I	Identity matrix
$^{-1}$	Matrix inverse operator
$^{\mathtt{T}}$	Transpose operator
$m \times n$	Size of A
L_2	Hilbert space
$\mathbb{R}$	Real space
`t`	x-coordinate of a data fitting point
`Y`	y-coordinate of a data fitting point
γ	Function, describing the data points, `Y`
p	Set of parameters, $p_1, p_2, p_3, \ldots$ for the curve used in the data fitting
M_p	Curve used in the data fitting
R	Residual, the difference between the M_p and `Y`
α	Measurement error when computing data points for the data fitting
β	Approximation error for the data fitting
N	Number of data points
g^{res}	Parametric curve used for data fitting on the Residual Norm Curve
g^{sol}	Parametric curve used for data fitting on the Solution Norm Curve
p^{res}	Parameters for g^{res}

Continued on the next page

– continued from previous page

Symbol	Description
p^{sol}	Parameters for g^{sol}
λ^{F}	Regularization parameter for the corner point in the fitted curve
λ^{err}	Regularization parameter for the smallest solution error
λ^{res}	Regularization parameter for the smallest residual error
λ^{L}	Regularization parameter for the corner point in the L-Curve
x^{λ_i}	Model found using regularization parameter λ_i
x^{λ}	Model found using Tikhonov Regularization
x^{k}	Model found using TSVD
x^{F}	Model found using Tikhonov Regularization with λ^{F}
x^{err}	Model found using Tikhonov Regularization with λ^{err}
x^{res}	Model found using Tikhonov Regularization with λ^{res}
x^{L}	Model found using Tikhonov Regularization with λ^{L}
x^{naive}	Model found naively
ρ	Residual Norm
η	Solution Norm
ϵ	Small number used for stop criteria
h	Vector used for isolating linear parameters
G	Matrix used for isolating linear parameters
c	Fudge parameter
A	Parameter, used for derivation of the parametric curve
B	Parameter, used for derivation of the parametric curve
C	Parameter, used for derivation of the parametric curve
D	Parameter, used for derivation of the parametric curve
E	Parameter, used for derivation of the parametric curve
F	Parameter, used for derivation of the parametric curve
x	Variable, used for derivation of the parametric curve
y	Variable, used for derivation of the parametric curve

Contents

Chapter 1

Introduction

To provide a gentle start, the layout of this thesis will be briefly outlined. We start in Chapter 1 by giving a simple introduction to the problem and the motivation behind this. Then the problem will be more formally introduced. Here the theory of the problem will be shown and in Chapter 2 an approach for solving the problem is presented before describing the idea behind the work done. One of method are briefly described in 3. The result of this thesis is shown in Chapter 4, and the considerations which lead to this result are presented in Chapter 5 and in Chapter 6. Finally, we show the obtained results in Chapter 7 and present a discussion on future work in Chapter 8.

1.1 A Gentle Start

The problem this thesis deals with is to find a model in the real world based on some data. Here data refer to measured data after an transformation has been applied. The model is the original data before the transformation. Finding such a model is done by modelling the problem using mathematics. The mathematical problem is a transformation of some model into some measureable data. Figure 1.1 shows a situation where the data are what can be seen with the naked eye and the model is hidden.

Figure 1.1: *Illustration of the problem. Normally the original model is hidden, and only the transformed model, the data, can be seen.*

A very simplified example of such a situation is to look at an object in sunlight. The images one can observe are the object itself, corresponding to the model, and a shadow on the ground corresponding to a transformation of the object. When looking at the object it is straightforward to interpretate what is perceived. However, the general problem is to reconstruct the original object from its shadow[1]. This thesis describes how to obtain an observation which cannot be seen directly.

The process of reconstruction is called an inverse problem, since the calculations are the inverse of perceiving the object directly. This type of problem can occur in different situations. It can be physically impossible to acquire the observations of the model directly, or it might be due to the fact that the model is contaminated with noise. Consequently it is necessary to do the estimations based on the second hand observations, the data. A great deal of problems can be considered inverse problems, for instance image reconstruction, geophysics and more. Moreover, the inverse problems share the fact that a naively found solution will be meaningless. This is due to the limitations of computers and calculations on these as well as the unavoidable errors in measurements. In order to be able to obtain a useful solution, the problem has to be changed. The new problem is formulated such that the solution will not only be true to the original problem, but also be a meaningful solution.

This is called regularization and is a trade off between being loyal to the original model and acquiring a useful solution. To get a good solution, one needs to find the right balance between these two characteristics. The focus in this thesis is to determine a parameter for this balance. Determining this parameter can, however, be a very difficult computational problem in the large-scale case. In the following chapters the problem will be discussed further and an approach to solve the underlying regularization problem will be presented.

[1] Actually, there is not enough information to do a real reconstruction besides the shape of the object.

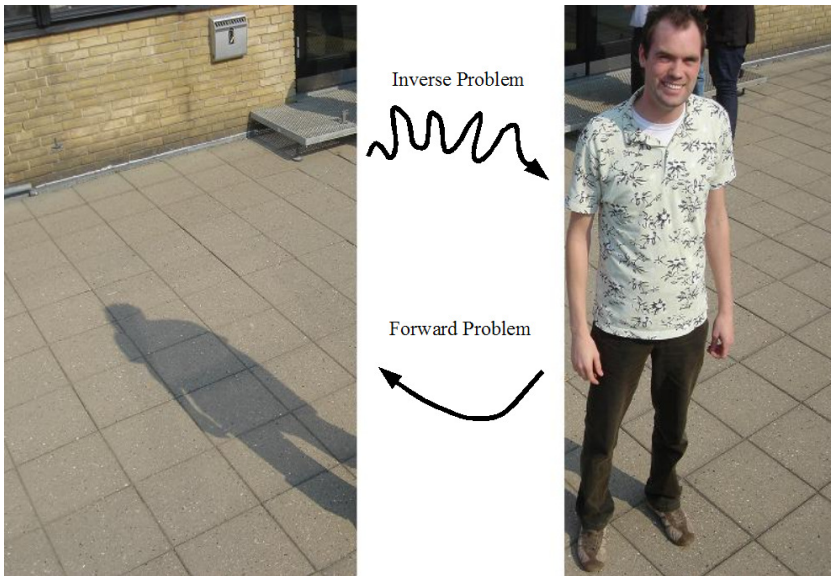

Figure 1.2: *Illustration of the problem. The forward problem is to look at the person and derive the shadow. The inverse problem is to look at the shadow and estimate the person.*

1.2 What is the Problem?

In the following pages, we will describe the underlying problem considered in this thesis. First, we present a general type of problem, followed by a more specific, which we will use later. We will use figures to illustrate the described situations created, using different problems. The problem chosen is the one, best illustrating the situation. Most of the derivation in the following pages is taken from [4].

The general type of problem is an ill-posed problem. According to Hadamard [4] a problem is called well-posed if all of the following properties are fulfilled:

1. A solution exists.
2. The solution is unique.
3. The solution depends continuously on the data.

A problem, which is not well-posed is said to be ill-posed.

We will solve ill-posed problems by getting around those properties that are not fulfilled. The first two requirements can fairly easily be fulfilled by reformulating the problem, for instance adding additional requirements. This could be done by adding a constraint to the problem or formulating it as a minimization problem to a Least Squares problem. The last requirement, on the other hand, is a bit more tricky to come around, while the problem is ill-conditioned. A small error in the data can therefore result in much larger errors in the solution of the problem.

The specific ill-posed problem we are focusing on, is the linear inverse problem. This can occur when dealing with an integral equation, known from mathematical physics. If we define an interval $\mathbb{I} = [a, b]$ with $a, b \in \mathbb{R}$, and $\mathbb{J} = \mathbb{I} \times \mathbb{I}$ then the Fredholm integral equation of the first kind[9] can be written as:

$$\int_a^b k(s,t)f(t)dt = g(s), \tag{1.1}$$

where $f(t),\ g(s) \in L_2(\mathbb{I})$ and $k(s,t) \in L_2(\mathbb{J})$. We call this the underlying problem, since the problem to solve is derived from it.

We can also describe the under-lying problem as a linear operator K acting on $L_2(\mathbb{I})$ with kernel $k(s,t) \in L_2(\mathbb{J})$:

$$Kf = g, \qquad f,\ g \in L_2(\mathbb{I}).$$

Here an integral operator, K, describes the mapping $f \mapsto g$. When this operator, i.e., the kernel K is sufficiently nice, it has a smoothing effect on $f(t)$. An example of the smoothening can be seen in Figure 1.3. When calculating $g(s)$ from the kernel $k(s,t)$ and $f(t)$, it is called the forward problem. When $g(s)$ is known and $f(t)$ is sought, it is called the inverse problem.

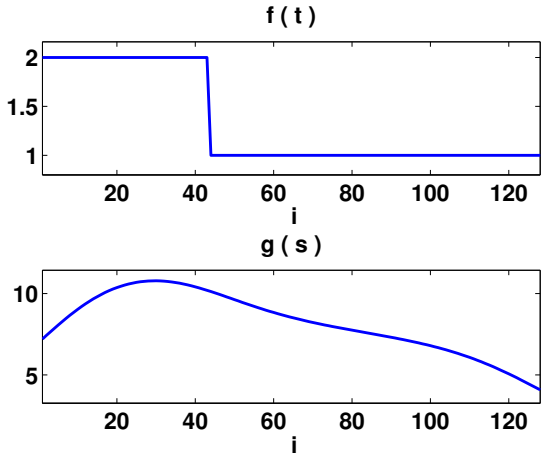

Figure 1.3: *The damping of $f(t)$. The problem shown is the* `1-D gravity surveying model problem` [5].

1.2.1 The Riemann-Lebesgue Lemma

In general, the underlying problem that we will be looking at, is the one where the mapping from $f \mapsto g$ is damping the higher frequencies more than the lower frequencies in $f(t)$. This is due to the *Riemann-Lebesgue lemma*:

If f is an integrable, measurable function, then [9]

$$\int_a^b f(x)e^{\mathbf{i}nx}\,dx \to 0 \text{ for } \quad n \to \pm\infty. \tag{1.2}$$

The *Riemann-Lebesgue lemma* implies, that an integral for an oscillating function will be small. This can be seen if we look at

$$f_p(t) = \sin(2p\pi t), \quad p = 1, 2, \ldots$$

and for an arbitrary kernel $k(s,t)$, we have:

$$g_p(s) = \int_0^1 k(s,t) f_p(t) dt \to 0 \quad \text{for} \quad p \to \infty. \tag{1.3}$$

In this case, when the frequency p increases, the amplitude of g_p decreases. In this way, there is a damping effect of f_p for the forward problem. An illustration of this can be seen in Figure 1.4, where f_p and g_p from equation (1.3) are shown for different values of p.

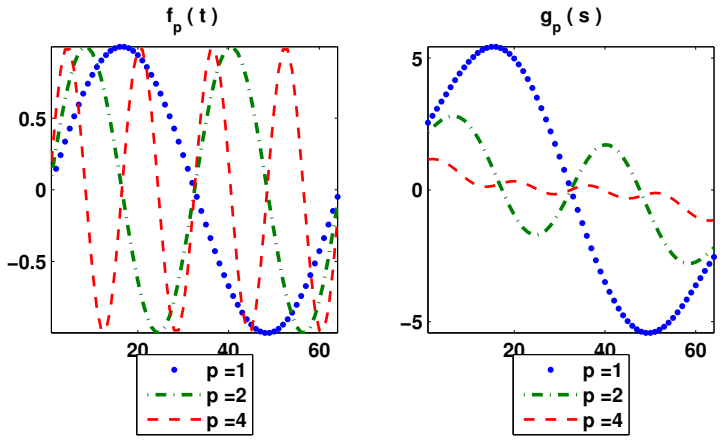

Figure 1.4: *Illustration of the Riemann-Lebesgue lemma. We see that when p increases the amplitude of g_p decreases. Notice that the scales are different.*

For the inverse problem, where we want to find f_p given g_p, there is somewhat the opposite effect. The amplitude of g_p will be amplified. In other words, higher frequencies are amplified more than lower frequencies. This is why it is an ill-posed problem we are dealing with. The solution does not depend continuously on the data, since a small change in the data can lead to a large change in the solution, if the amplitude of g_p is high enough.

1.3 The Singular Value Expansion

Now we will do a further examination of the calculations for equation (1.1). In order to do this, we look at the kernel $k(s,t) \in L_2(\mathbb{J})$ again and use the following definition of the inner product:

$$\langle \phi, \psi \rangle = \int_0^1 \overline{\phi(x)} \psi(x) dx,$$

and the norms

$$\|\phi\| = \langle \phi, \phi \rangle^{\frac{1}{2}}, \quad \|k\| = \left(\int_0^1 \int_0^1 |k(s,t)|^2 ds dt \right)^{\frac{1}{2}}.$$

Since $k(s,t)$ is square integrable, it is possible to write a singular value expansion (SVE):

$$k(s,t) \doteq \sum_{i=1}^{\infty} \mu_i u_i(s) \overline{v_i(t)}, \quad s,t \in \mathbb{I}. \tag{1.4}$$

Here $\doteq$ is used to show that the right hand side is converging in the mean to the left hand side. μ_i are the singular values of $k(s,t)$ and are sorted in a non-increasing order $\mu_1 \geq \mu_2 \geq \cdots \geq \mu_i \geq \mu_{i+1} \geq \cdots 0$. The u_i and v_i are called singular functions, and they are orthonormal with respect to the inner product,

$$\langle v_i, v_j \rangle = \langle u_i, u_j \rangle = \delta_{ij}, \quad i,j = 1,2,3,\ldots$$

By using equation (1.4) and (1.1), it is possible to formulate the fundamental relation

$$\int_0^1 k(s,t) v_i(t) dt = \mu_i u_i(s), \quad i = 1,2,3,\ldots$$

From [2] we know the singular values always decay to zero, and the speed of this decay is a matter of how smooth $k(s,t)$ is. An example of this decay can be seen in Figure 1.5.

If we write the two terms, f and g, in the orthonormal basis spanned by the left and right singular functions, $u_i(s)$ and $v_i(t)$, we get:

$$f(t) \doteq \sum_{i=1}^{\infty} \langle v_i, f \rangle v_i(t), \quad g(s) \doteq \sum_{i=1}^{\infty} \langle u_i, g \rangle u_i(s).$$

Inserting this into equation (1.1) gives:

$$g = Kf, \quad g(s) = \int_0^1 k(s,t) f(t) dt \quad \doteq \quad \sum_{i=1}^{\infty} \mu_i \langle v_i, f \rangle u_i(s). \tag{1.5}$$

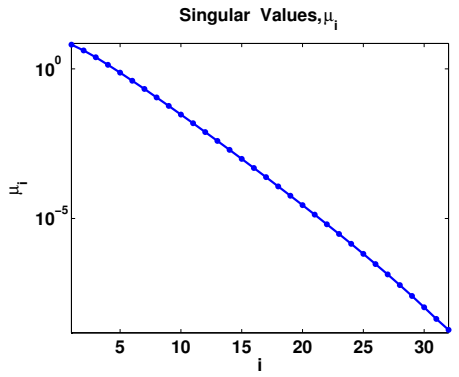

Figure 1.5: *Illustration of the decay in singular values, when i increases. The problem used is the* `1-D gravity surveying model problem` [5].

The final equation, (1.5), shows that higher oscillations are being damped more than lower oscillations and explains why K has a smoothing effect. Now by writing the expansion for both $f(t)$ and $g(s)$, we obtain:

$$\sum_{i=1}^{\infty} \mu_i \langle v_i, f \rangle u_i(s) \doteq \sum_{i=1}^{\infty} \langle u_i, g \rangle u_i(s).$$

Considering it as an equation gives:

$$\langle v_i, f \rangle = \frac{\langle u_i, g \rangle}{\mu_i}, \quad i = 1, 2, 3, \ldots.$$

Which leads to the following expression of the solution to the inverse problem:

$$f(t) \doteq \sum_{i=1}^{\infty} \frac{\langle u_i, g \rangle}{\mu_i} v_i(t). \tag{1.6}$$

In this way we see that in order for a solution to exist, the two-norm of f must be finite. This can be expressed as:

$$\|f\|_2^2 = \int_0^1 f(t)^2 dt \quad = \quad \sum_{i=1}^{\infty} \langle v_i, f \rangle^2 = \sum_{i=1}^{\infty} \left(\frac{\langle u_i, g \rangle}{\mu_i} \right)^2 < \infty. \tag{1.7}$$

This last condition is called the Picard Condition. Which states:

> In order for a solution to the Fredholm integral equation of the first kind to exist, $\langle u_i, g \rangle$ must decay faster than μ_i.

The Picard Condition is necessary to fulfill equation (1.7). Such a situation is illustrated in Figure 1.6.

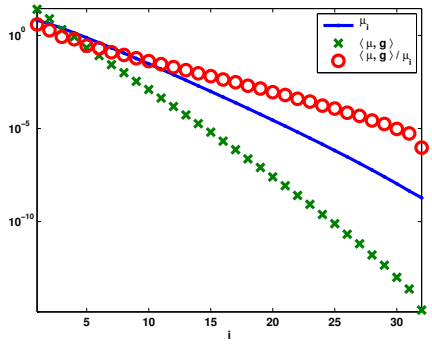

Figure 1.6: *Illustration of the Picard Condition. It can be seen for increasing i, μ_i are decaying and that $\langle u_i, g \rangle$ decays, in a proper way, faster than μ_i. Therefore the Picard Condition is satisfied. The problem shown is* `1-D gravity surveying model problem` [5].

This is, however, rarely the case for inverse problems. Either because of the nature of the problem being ill-posed, because the calculations done with finite precision or due to noise in the data. To use finite precision, we need a discretization and other computations, making the problem ill-posed.

1.4 Discretizing the Problem

Now we have seen the difficulties with continuous problems. In order to obtain a solution using computers, we need a discrete problem. When we discretize the problem in equation (1.1), there are different approaches. The quadrature method and the expansion method are described in [4]. Both methods are approximations of the underlying problem, the Fredholm integral equation of the first kind, and they will have an approximation error.

When focusing on the quadrature method, the integral is evaluated in different points, $t_1, \ldots, t_n$ as the collocation points, and $s_1, \ldots, s_m$ as the abscissas for the quadrature rule, and the corresponding weights are introduced, $\omega_1, \ldots, \omega_n$.

When doing the discretization, it turns out to be a linear system of equations:

$$\begin{pmatrix} \omega_1 K(s_1,t_1) & \omega_2 K(s_1,t_2) & \dots & \omega_m K(s_1,t_m) \\ \omega_1 K(s_2,t_1) & \omega_2 K(s_2,t_2) & \dots & \omega_m K(s_2,t_m) \\ \vdots & \vdots & & \vdots \\ \omega_1 K(s_n,t_1) & \omega_2 K(s_n,t_2) & \dots & \omega_m K(s_n,t_m) \end{pmatrix} \begin{pmatrix} \tilde{f}_1 \\ \tilde{f}_2 \\ \vdots \\ \tilde{f}_m \end{pmatrix} = \begin{pmatrix} \tilde{g}(s_1) \\ \tilde{g}(s_2) \\ \vdots \\ \tilde{g}(s_n) \end{pmatrix}.$$

Please note that $\tilde{f}$ and $\tilde{g}$ are used instead of f and g. This is due to the fact that the discretization is evaluated in sampled values and therefore resulting in a discretization error.

The discretization can be written in a more simple form

$$Ax = b. \tag{1.8}$$

It can be seen that $A \in \mathbb{R}^{m \times n}$, $x \in \mathbb{R}^m$ and $b \in \mathbb{R}^n$. For the sake of simplicity, we will only consider the situation with the matrix A having the same number of rows as columns, thus only consider $m = n$.

Matrix A corresponds to the transformation the kernel gives, vector x is the model discretized and b is the data. Here we notice why the problem of computing x is called the inverse problem. If we assume A is invertible the naive solution to the system is

$$x^{\text{naive}} = A^{-1}b. \tag{1.9}$$

1.5 The Singular Value Decomposition

Like with the SVE for the continuous case, it is possible to decompose matrix A using singular value decomposition (SVD):

$$A = U\Sigma V^{\mathsf{T}} = \sum_{i=1}^{n} u_i \sigma_i {v_i}^{\mathsf{T}} \tag{1.10}$$

with Σ being a matrix with the singular values σ_i in the diagonal and zeros at other places, while the columns of U and V consist of the left and right singular vectors respectively, as illustrated here:

$$\Sigma = \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \sigma_n \end{bmatrix}, \quad \mathbf{V} = \begin{bmatrix} v_1 & \dots & v_n \end{bmatrix} \quad \text{and } \mathbf{U} = \begin{bmatrix} u_1 & \dots & u_m \end{bmatrix}.$$

σ_i are sorted such that $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \geq 0$. The singular vectors, u_i and v_i are oscillating more and more as i increases. This is illustrated in Figure 1.7. It can be seen that, if the inverse exist,

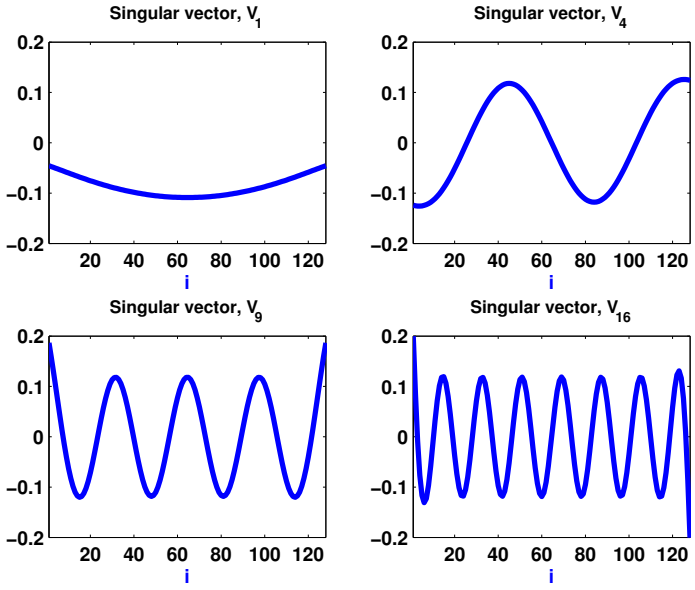

Figure 1.7: *Illustration of the increase in oscillations for the singular vectors when i increases. The problem shown is the* `1-D gravity surveying model problem` [5].

$$A^{-1} = \left(U\Sigma V^{\mathrm{T}}\right)^{-1} = \left(V^{\mathrm{T}}\right)^{-1}\Sigma^{-1}U^{-1} = V\Sigma^{-1}U^{\mathrm{T}}$$

with Σ^{-1} being a diagonal matrix with ${\sigma_i}^{-1}$ in the diagonal. This SVD can be used to write the solution to $Ax = b$ similar to (1.6), as:

$$x \quad = \quad A^{-1}b = V\Sigma^{-1}U^{\mathrm{T}}b = \sum_{i=1}^{n} \frac{{u_i}^{\mathrm{T}}b}{\sigma_i}v_i. \tag{1.11}$$

From [4] it is known that the SVD can be used as an approximation to the SVE with the help of the discretization. This gives specifically σ_i as approximation to μ_i and the singular vectors, $\tilde{u_i}$ and $\tilde{v_i}$, as approximations to the singular functions, $u_i(s)$ and $v_i(t)$. Again $\tilde{}$ is used to show that there are inaccuracies. We write it as:

$$\tilde{u_i}(s) = \sum_{i=1}^{n} u_{ij}\psi_i(s)$$
$$\tilde{v_i}(t) = \sum_{i=1}^{n} v_{ij}\phi_i(t)$$

where $\phi_i(t), \psi_i(s)$ are the chosen basis functions for the discretization.

In fact, the singular vectors converge in the mean to the singular functions when more sample values are used, that is for $n \to \infty$ and $m \to \infty$. Using this to compare the solution from the SVE and the SVD, we get:

$$\begin{aligned}
\langle u_i, g\rangle &= \left\langle \sum_{j=1}^{n} u_{ij}\psi_i(s), \sum_{k=1}^{n} b_k\psi_k(s) \right\rangle \\
&= \int_0^1 \sum_{j=1}^{n} u_{ij}\psi_i(s) \sum_{k=1}^{n} b_k\psi_k(s)ds = \sum_{j=1}^{n} u_{ij} \sum_{k=1}^{n} b_k \langle \psi_i, \psi_k\rangle \\
&= \sum_{j=1}^{n} u_{ij}b_i = {u_i}^{\mathrm{T}}b.
\end{aligned}$$

It is therefore possible to introduce a condition we call the Discrete Picard Condition. Similarly to the Picard Condition from the SVE, requires ${u_i}^{\mathrm{T}}b$ to decay faster than the decay of σ_i.

No matter which discretization method is used, the Discrete Picard Condition is not satisfied. This is due to the singular values which, like in the SVE, decay towards zero. However when the singular values approach the calculation precision, that is machine accuracy, the singular values level off. This is illustrated in Figure 1.8.

Therefore these singular values can not be trusted, and will violate the Discrete Picard Condition. The same thing happens for the $|{u_i}^{\mathrm{T}}b|$ when they reach the machine accuracy. Like in the SVE, we will now examine whether the Discrete Picard Condition is satisfied or not, by plotting the singular values together with $|{u_i}^{\mathrm{T}}b|$. This is shown in Figure 1.9. We call such a figure a Picard Plot. There we see, for this specific problem, for approximately $i < 55$ the Picard Condition is satisfied.

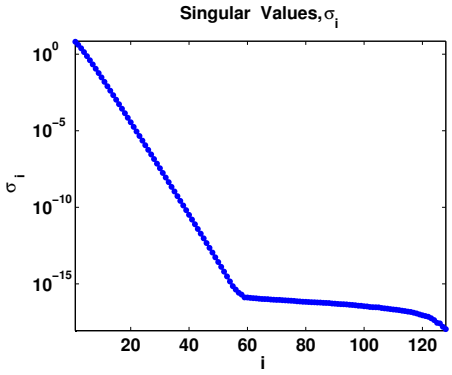

Figure 1.8: *Illustration of the singular values leveling off, when approaching the machine accuracy. The problem shown is the* `1-D gravity surveying model problem` [5].

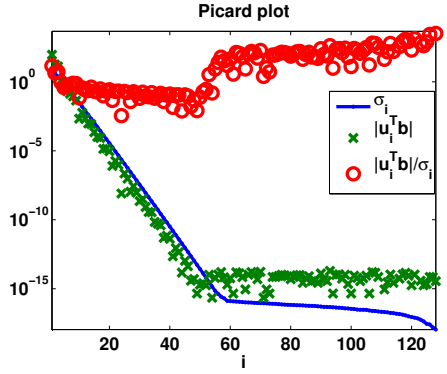

Figure 1.9: *Illustration of the Picard Plot. It can be seen that the singular values σ_i level off when approaching the machine accuracy, thus not satisfying the condition of $|u_i^T b|$ decaying faster that σ_i. The problem shown is the* `1-D gravity surveying model problem` [5].

Chapter 2

Finding a Solution

Knowing what causes problems to be ill-posed, we can now concentrate on the specific problem of the discretized Fredholm integral equation of the first kind. As already seen, this discretization can be written as the system of equations

$$Ax = b,$$

where b corresponds to the measurements of data, and A describes the transformation of the model x. These measurements will normally be contaminated by some random noise.

We will concentrate on the ill-posed problems, where the measurements of data, the right hand side, are contaminated. The system of equations will be ill-posed when discretizing the original problem into sampled values, such that the the measurements are contaminated. That is, the right hand side is $b = b^{\text{exact}} + \varepsilon$, with b^{exact} as the exact data, and ε is some random Gaussian noise with a varians of ξ. An example of the components of the right hand side can be seen in Figure 2.1.

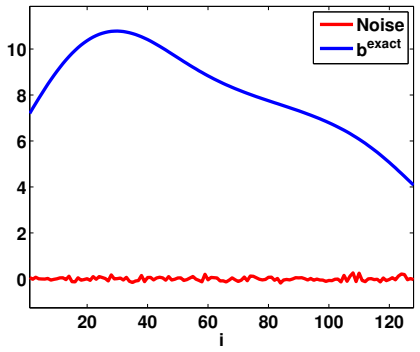

Figure 2.1: *Illustration of the components of the measurements, the exact data b and the noise ε_i. The noise shown has $\xi = 10^{-3}$. The problem shown is the* `1-D gravity surveying model problem` [5].

2.1 Introducing Regularization

One way to find a solution to the linear inverse problem from equation (1.9) is to solve the Least Squares minimization:

$$\min ||Ax - b||$$

with the solution

$$x^{\text{naive}} = (A^{\mathsf{T}}A)^{-1}A^{\mathsf{T}}b.$$

We will call this the naive solution, since it is done straight forward and can lead to a solution which is not useful. Such a situation is illustrated in Figure 2.2 where the naive solution has been shown together with the exact solution.

To find out whether the problem has a useless solution, using the naive approach we have to examine the Discrete Picard Condition. If this condition is not satisfied, the solution will be meaningless. If we split the solution into a noisy part and an exact part, we have

$$x^{\text{naive}} = (A^{\mathsf{T}}A)^{-1}A^{\mathsf{T}}b^{\text{exact}} + (A^{\mathsf{T}}A)^{-1}A^{\mathsf{T}}\varepsilon,$$

and writing it using the SVD gives

$$x^{\text{naive}} = \sum_{i=1}^{n} \frac{u_i{}^{\mathsf{T}}b^{\text{exact}}}{\sigma_i}v_i + \frac{u_i{}^{\mathsf{T}}\varepsilon}{\sigma_i}v_i.$$

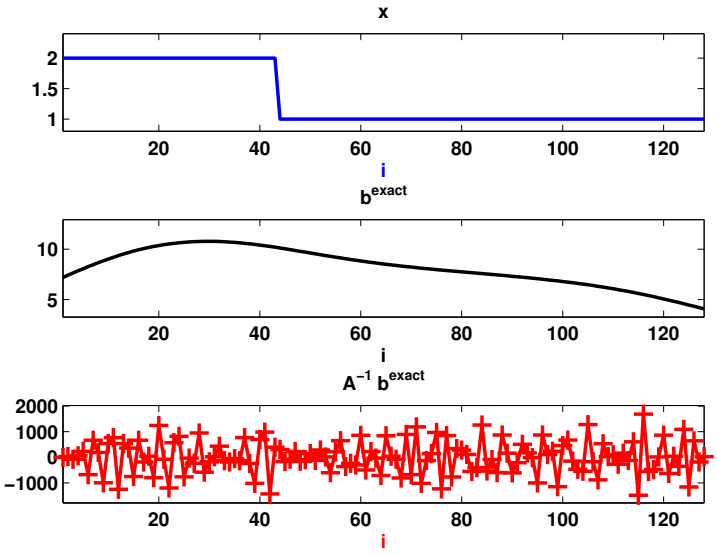

Figure 2.2: *The damping of x along with the naive solution. The figure on top is x, the middle one is b and the bottom figure is a naive solution $x^{\text{naive}} = A^{-1}b$. The problem shown is the* `1-D gravity surveying model problem` [5].

Here we see the problem. We know the first part is decreasing as i increases. The second term increases since ε is constant and σ_i decreases. Therefore the last term, the error, will dominate for large i, and the Discrete Picard Condition is not satisfied. Such a situation is shown in Figure 2.3, where a Picard Plot has been shown for a contaminated right hand side. It can easily be seen that $|u_i{}^{\mathrm{T}}b|$ levels off at the noise level.

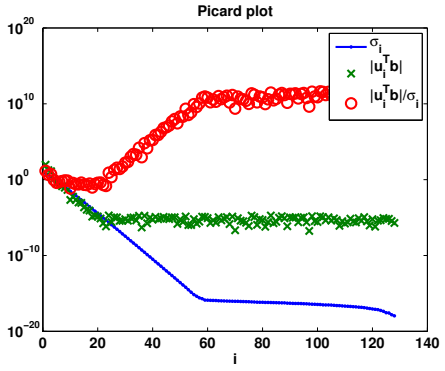

Figure 2.3: *The Picard Plot where the right hand side is contaminated with noise of* $\xi = 10^{-6}$. *We see* $|u_i{}^{\mathrm{T}}b^{\mathrm{exact}}|$ *, levels off at the noise level, and the singular values* σ_i *levels off at the machine accurracy. The problem shown is the* `Baart problem` [5].

When the Discrete Picard Condition is not satisfied some regularization is needed. The regularization will change the problem into a related problem. In our case it means changing the problem into one with better numerical characteristics, such we avoid having the solution dominated by the errors. The related problem should be formulated such that the inverted noise is suppressed, as well as the solution still being close to x^{exact}.

2.2 The Truncated Singular Value Decomposition

The first regularization we will look at is the Truncated Singular Value Decomposition (TSVD). In order to make the related problem, we will look at the SVD:

$$A = U\Sigma V^{\mathrm{T}}.$$

We write up the solution to the underlying problem:

$$x^{\text{naive}} = \sum_{i=1}^{n} \frac{{u_i}^{\mathsf{T}} b}{\sigma_i} v_i.$$

Keeping in mind that from a Picard Plot, such as in Figure 2.3, we can see if the Discrete Picard Condition is satisfied or not. For ill-posed problems the $|{u_i}^{\mathsf{T}} b|$ start by fulfilling the condition, until the value of these reaches the noise level, ξ, where it flattens out[1]. Due to this, it is reasonable to make a related problem, which skips the last factors in the summation, such that the parts which are dominated by noise, are neglected.

This is a simple regularization which is written as

$$x^k = \sum_{i=1}^{k} \frac{{u_i}^{\mathsf{T}} b}{\sigma_i} v_i, \quad 1 \leq k \leq n. \tag{2.1}$$

Here, the value k is called the regularization parameter or the truncation parameter and should be chosen so that it minimizes $\|x^{\text{exact}} - x^k\|_2$. This corresponds to be looking at the singular values, which mostly contribute to the solution but not to the noise.

2.2.1 Choosing a Parameter for TSVD

From the Picard Plot, Figure 2.3, it can be seen that k can be chosen from the abscissa. Here it is possible to see how many singular values are trustworthy, and where the Discrete Picard Condition is no longer satisfied.

The influence of k has been illustrated in Figure 2.4, showing $\|x^{\text{exact}} - x^k\|_2$. The solutions get better and better until the Discrete Picard Condition is violated. Hereafter, the error between the solutions increases. A good choice of k is therefore the number of singular values for which the Discrete Picard Condition is still satisfied. Since the Picard Plot is not always available for large-scale problems, some method for finding this optimal point is needed. Finding a good regularization parameter will not be part of this project. An example of different regularizations can be seen in Figure 2.5, where solutions to a problem has been shown together with the exact model. We see that both a too small and a too large truncation parameter can give a bad solution.

[1] In fact, if the noise level is less than the machine accuracy, it will flatten out when reaching this level.

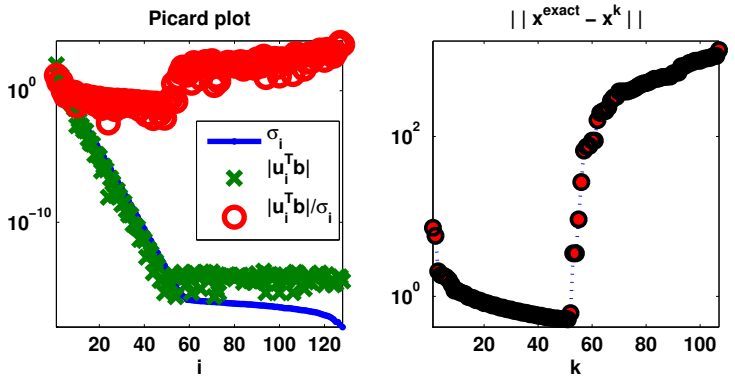

Figure 2.4: *Illustration of the Picard Plot, left, and the optimal value of x^k, right. It can be seen that the optimal truncation parameter corresponds to the point where the Discrete Picard Condition is no longer satisfied. The problem shown is the* `1-D gravity surveying model problem` [5]

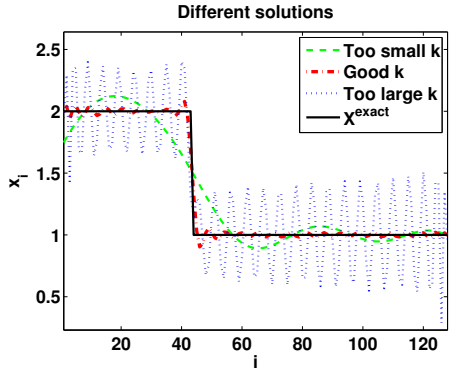

Figure 2.5: *Different solutions using TSVD as the regularization method. We see that both a too large and a too small k can give a bad solution compared with a good truncation parameter. The problem shown is the* `1-D gravity surveying model problem` [5].

2.3 Tikhonov Regularization

We see that TSVD filters out all components with $\sigma_i < \sigma_k$ and moreover the regularization parameter, k is discrete. Therefore the error in Figure 2.4 is discrete as well. If we want a more smooth filter, Tikhonov Regularization can be used. This regularization offers a continuous regularization parameter. It not only tries to fit the regularized solution to the exact solution, but also penalizes large norms. Thereby the regularized solution will be a trade-off between being loyal to the data, b, and acquiring a smooth solution. This is done by not finding the solution to the Least Squares, $\arg\min_x \|Ax - b\|_2^2$, but instead to the regularized problem:

$$x^\lambda = \arg\min_x \left(\|Ax - b\|_2^2 + \lambda^2 \|Lx\|_2^2 \right). \tag{2.2}$$

Here λ is the regularization parameter or the Tikhonov Regularization parameter and must be non-negative. The L can be chosen as a weighting matrix or as a discrete derivative operator, surplying additional information to the problem. In this thesis we will focus on L being the identity matrix, $L = I$. We see that λ is a weight between the two terms in equation (2.2). The two terms can be expressed as:

- $\|Ax - b\|_2^2$, which is a measure of how well x fits the measurement b
- $\|x\|_2^2$ which is a regularization term, preventing the solution from exploding.

We see that if $\lambda = 0$, we have the Least Squares problem. Moreover when $\lambda \to \infty$ the solution $x^\lambda \to 0$, and we will filter out all components.

Now writing the Tikhonov Regularization in normal equations and using the SVD to write the solution:

$$\begin{aligned} x^\lambda &= \left(A^\mathsf{T} A + \lambda^2 I\right)^{-1} A^\mathsf{T} b \\ &= \left(V \Sigma^\mathsf{T} U^\mathsf{T} U \Sigma V^\mathsf{T} + \lambda^2 I\right)^{-1} V \Sigma U^\mathsf{T} b \\ &= V \left(\Sigma^2 + \lambda^2 I\right)^{-1} \Sigma U^\mathsf{T} b. \end{aligned}$$

We write it out as:

$$x_\lambda = \sum_{i=1}^{n} f_i \frac{{u_i}^\mathsf{T} b}{\sigma_i} v_i \tag{2.3}$$

with $f_1, \ldots, f_n$ being Tikhonov Regularization filter factors, given by

$$f_i = \frac{{\sigma_i}^2}{{\sigma_i}^2 + \lambda^2} \simeq \begin{cases} 1, & \text{if } \sigma_i \gg \lambda \\ {\sigma_i}^2/\lambda^2, & \text{if } \sigma_i \ll \lambda. \end{cases} \tag{2.4}$$

These filter factors damp the contributions from most singular values when λ is large, whereas if λ is small, most of the SVD components are used. It is easy to see, that if f_i is set to

$$f_i = 1, \ \forall i$$

the solution to Tikhonov Regularization is in fact the Least Squares solution. If it is substituted with

$$f_i = \begin{cases} 1, & \text{if } \sigma_i \geq \sigma_k \\ 0, & \text{if } \sigma_i < \sigma_k \end{cases}$$

it is the TSVD. Figure 2.6 shows the different filter factors for Least Squares, TSVD and Tikhonov Regularization.

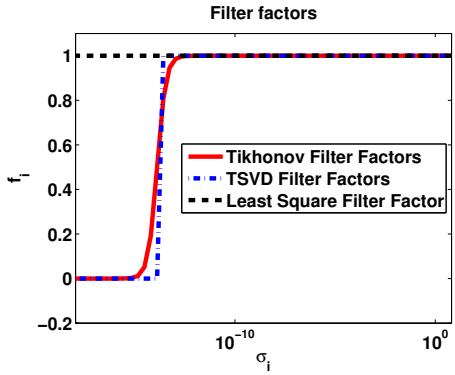

Figure 2.6: *Comparison of the filter factor for Least Squares, TSVD and Tikhonov Regularization.*

In general, we see that similar to the TSVD, the Tikhonov Regularization damps the smallest singular values, and therefore the contributions from the components are dominated by noise. However, the filter becomes more smooth.

The Tikhonov Regularization parameter determines if the regularized solution should fit the data or if it should remove most of the noise. When λ is too big, too much regularization is added (oversmoothing) and the term $||Ax_\lambda - b||_2^2$ will be large and the solution will not fit the data b. On the other hand, if λ is small and too little regularization is applied (undersmoothing), the solution will be dominated by the noise, just like in the naive solution. Therefore it is necessary to find a good way to determine a suitable value for λ. An example of different solutions found using Tikhonov Regularization can be seen in Figure 2.7.

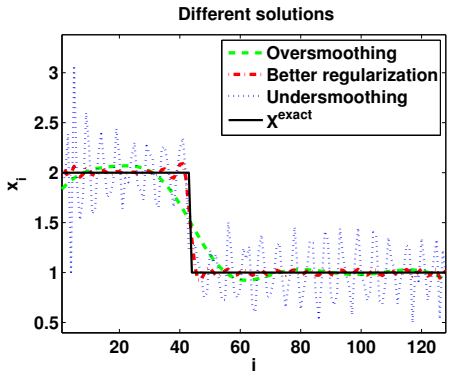

Figure 2.7: x^{exact} *as well as three different solutions using Tikhonov Regularization as regularization method has been shown . The problem shown is the* `1-D gravity surveying model problem` [5].

2.3.1 Choosing a Parameter for Tikhonov Regularization

In order to find the optimal λ value for the Tikhonov Regularization, we need to look at two factors. These are the solution norm, $\|x^\lambda\|_2$, and residual norm, $\|Ax^\lambda - b\|_2$. They satisfy:

$$\begin{aligned} x^\lambda &= \sum_{i=1}^{n} f_i \frac{u_i{}^{\mathrm{T}} b}{\sigma_i} v_i \\ \|x^\lambda\|_2^2 &= \sum_{i=1}^{n} \left(f_i \frac{u_i{}^{\mathrm{T}} b}{\sigma_i} \right)^2, \end{aligned} \tag{2.5}$$

and

$$\begin{aligned} b - Ax_\lambda &= \sum_{i=1}^{n} (1 - f_i)\, u_i{}^{\mathrm{T}} b u_i \\ \|Ax^\lambda - b\|_2^2 &= \sum_{i=1}^{n} \left((1 - f_i) u_i{}^{\mathrm{T}} b \right)^2 + \sum_{i=n+1}^{m} \left((u_i{}^{\mathrm{T}} b) u_i \right)^2. \end{aligned} \tag{2.6}$$

When minimizing each of the two norms, we see that

- $\|Ax^\lambda - b\|_2^2$ is small when f_i is chosen close to 1.
- $\|x^\lambda\|_2^2$ is small when f_i is chosen close to 0.

To achieve a small solution norm, λ should therefore be big, and to achieve a small residual norm, λ should be small.

In Figure 2.8, $\|x^{\text{exact}} - x^{\lambda}\|_2$ is shown. In comparison to the corresponding figure for the TSVD, Figure 2.4, we see that they both have a clear minimum, and that the TSVD is less smooth.

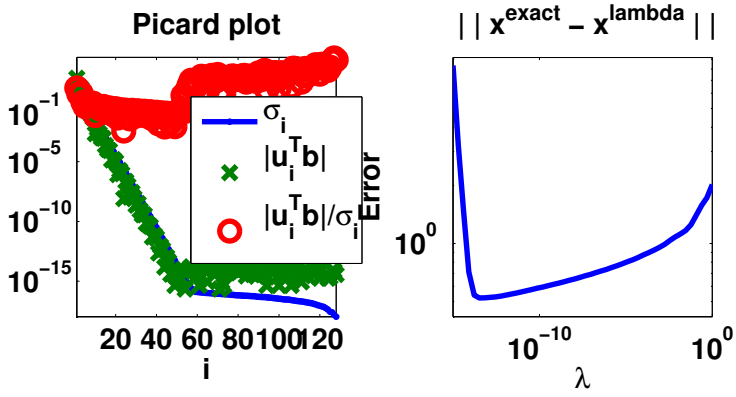

Figure 2.8: *Illustration of the Picard Plot and the behavior of* $\|x^{\text{exact}} - x^{\lambda}\|_2$. *We see that the behavior is more smooth than for the TSVD. The problem shown is the* `1-D gravity surveying model problem` [5].

A way to locate the optimal parameter is to look at the two terms in equation (2.2). This will be explained in the following section.

2.4 The L-Curve Criterion

In order to find a good compromise between norms and thereby the minimum of the Tikhonov Regularization, a plot can be used. Such a figure is a trade-off curve, illustrating the effect on both parameters when varying λ. Plotting the

solution norm together with the residual norm in logarithmic scale will show the effect on both norms and is useful for finding the best λ. To understand how such a curve behaves, we look at the behavior of the two norms. From equations (2.5) and (2.6) we see:

$$\lambda \to 0 \quad \begin{cases} \text{the solution norm } \|x^\lambda\| \to \infty \\ \text{the residual norm } \|Ax^\lambda - b\| \to 0 \end{cases}$$

$$\lambda \to \infty \begin{cases} \text{the solution norm } \|x^\lambda\| \to 0 \\ \text{the residual norm } \|Ax^\lambda - b\| \to \infty. \end{cases}$$

An illustration of this can be seen in Figure 2.9. Notice that the shape of the curve is similar to a capital L. From this shape it gets its name, The L-Curve. It describes the trade-off between the solution norm $\|x^\lambda\|_2$ and the residual norm $\|Ax^\lambda - b\|_2$.

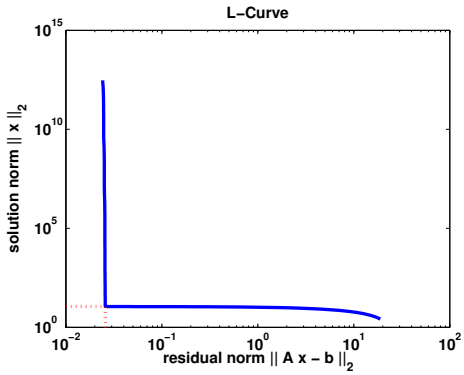

Figure 2.9: *Illustration of the parametric curve with the L shape, called the L-Curve. The corner point is a point in the lower left corner. The problem shown is* `Shaw problem` [5].

Since a good solution is a balance between these two terms, it is interesting to find a way of locating a good combination. To locate a reasonable point, we are interested in both terms to be small. The lower left corner of the curve is such a point.

There are different ways to define a corner point of a curve. We will use the extremum of the curvature to define a corner point. In the case of a curve with a shape like an L, the corner is convex. This means the curvature will be a curve with a peak at the corner.

When looking at the Tikhonov Regularization we can derive the curvature from the parametric curve description. To simplify the derivation, we define

$\eta = \log_{10} \|x^\lambda\|_2^2$ and $\rho = \log_{10} \|Ax^\lambda - b\|_2^2$ then the L-Curve is given by $(\frac{1}{2}\rho, \frac{1}{2}\eta)$. Letting $\eta' = \frac{d\eta}{d\lambda}$, $\rho' = \frac{d\rho}{d\lambda}$, $\eta'' = \frac{d^2\eta}{d\lambda^2}$ and $\rho'' = \frac{d^2\rho}{d\lambda^2}$, the curvature is:

$$\kappa = \frac{\rho'\eta'' - \rho''\eta'}{\left(\rho'^2 + \eta'^2\right)^{\frac{3}{2}}}. \tag{2.7}$$

From [4] we know inserting η and ρ gives

$$\kappa = 2\frac{\eta\rho}{\eta'} \; \frac{\lambda^2\eta'\rho + 2\lambda\eta\rho + \lambda^4\eta\eta'}{(\lambda^2\eta^2 + \rho^2)^{3/2}}. \tag{2.8}$$

κ can be calculated if the SVD is available. Accordingly, if the SVD is available, a λ for the corner of the L-Curve can be found analytically as the one giving the greatest κ. An example of the curvature for the L-Curve can be seen in Figure 2.10.

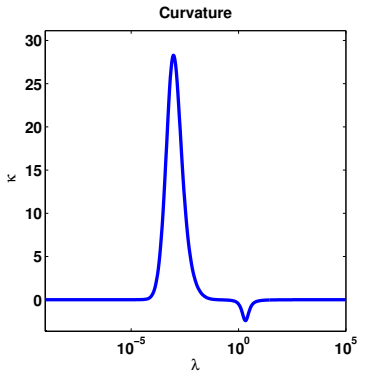

Figure 2.10: *Illustration of the curvature. The L-Curve has a convex corner, corresponding to the peak. Problem shown is the* `Shaw problem` [5].

In this way, the L-Curve Criterion is a good method of finding the regularization parameter. However the trade-off curve must have the characteristic L shape.

Chapter 3

Data Fitting

So far, we have seen how to solve ill-posed problems using different regularization methods. In order to find a good regularization parameter for the methods, we need to compute the SVD which is very demanding. However, the Tikhonov Regularization can be calculated without computing the SVD. Computing x^λ for many different values of λ makes it possible to create an L-Curve and thereby locate a reasonable λ parameter as the corner point.

Though, like with the computation of the SVD, computing many x^λ is demanding, we will therefore use a method which does not need many x^λ. When we are able to draw the L-Curve, we know a way to locate a good value for the Tikhonov Regularization parameter. For that reason, we want to estimate such a curve without having to calculate every point on the L-Curve.

A way to receive such a curve is to use data fitting. Data fitting is to estimate parameters to a parametric curve such that this curve is an optimal approximation to a given set of points. In the following, the data fitting will be described before applying it to some points on the L-Curve. These points will be referred to as data points.

In order to describe the data fitting, we start by introducing the points we want to fit. These will be referred to as the N data points and are located in $(\mathtt{t}_i, \mathtt{Y}_i)$, for $i = 1 \dots N$. If we assume that the data points can be approximated

as a function,

$$\mathtt{Y} = \gamma(\mathtt{t}),$$

and we call a parametric curve with parameters, p,

$$M_p(\mathtt{t}),$$

then we wish to find the curve with parameters, p such that $M_p(\mathtt{t}_i)$ is close to $\gamma(\mathtt{t}_i)$. This can be written using residuals as follows:

$$R(\mathtt{t}_i) = \gamma(\mathtt{t}_i) - M_p(\mathtt{t}_i).$$

$R(\mathtt{t}_i)$ consists of the two errors $\alpha_i + \beta_i$, where α_i is the error due to measurement errors in the data points and β_i is the approximation error. The optimal parameter is when $R(\mathtt{t}_i)$ only consist of α_i, such the only error origins from the measurement of the data points and not from the data fitting. In other words, $M_p(\mathtt{t})$ will follow the trends in the data points.

In order to find the best parameters, various ways can be chosen, depending on how well the data points are measured. That is, how big α_i can be considered. In general, it is done by minimizing the k-norm of the residuals:

$$\arg\min\left(\|R\|_k\right) = \arg\min_p \left(\sum_{i=1}^{N} \left(M_p(\mathtt{t}_i) - \mathtt{Y}_i\right)^k\right)^{\frac{1}{k}}, \quad k \geq 1. \tag{3.1}$$

The measurements of the data points are actually computations and therefore considered good, with respect to the L-Curve, and as such the two-norm is well-suited. This gives the following minimization problem:

$$\arg\min_p \|R\|_2 = \arg\min_p \left(\sum_{i=1}^{N} \left(M_p(\mathtt{t}_i) - \mathtt{Y}_i\right)^2\right)^{\frac{1}{2}}. \tag{3.2}$$

The parameter vector p that minimizes this, corresponds to the parameters for which $M_p(\mathtt{t}_i)$ is closest to $\mathtt{Y}(\mathtt{t}_i)$ for $i = 1 \ldots N$ in the least squares sense. This implies that $M_p(\mathtt{t})$ has the same shape as $\gamma(\mathtt{t})$.

Chapter 4

Using Data Fitting and the L-Curve Criterion

Now we have seen the basic theory, which this thesis is based on. In the following, we will show an algorithm for locating a regularization parameter. We call it *Finding Regularization Parameter Using Data Fitting*(FRe-PUF).

This is a general algorithm which combines the L-Curve Criterion with data fitting. We will start by showing the algorithm in this section. Then in Chapter 5 and 6 we focus on some limitations and examinations of the algorithm.

4.1 A Brief Walkthrough

In the following, an algorithm for finding a good Tikhonov Regularization parameter will be presented. To illustrate the method, we will show a flowchart of the algorithm. Such a graphical illustration of the algorithm can be seen in Figure 4.1. After a brief walkthrough of the flowchart, the algorithm will be shown in pseudo code.

The algorithm utilizes λ-values that are equidistantly distributed on a logarithmic scale to create an L-Curve using data fitting. On this L-Curve, it is

Input	Description
a, b	Initial interval
N	Number of data points
c	Fudge parameter
ϵ_i	Constant used for determining stop criteria
k^{max}	Handbrake for the loop
n^{max}	Handbrake for the loop

Table 4.1: *Table of input parameters for the algorithm.*

then possible to use the L-Curve Criterion to locate a regularization parameter, as described in Section 2.4. To get the best possible result, new data points are added one by one until a satisfactory solution to the fitting problem is found. In the following the method will be explained. The algorithm requires some user specified parameters given in Table 4.1.

These input are boundary points for the initial data points being inside, $[a, b]$, the number of initial data points, N, a constant used for adding new points, c, and some parameters, ϵ_i, n^{max} and k^{max} used for the stopping criteria.

Given these parameters, the algorithm finds N data points on two curves (the Solution Norm Curve and the Residual Norm Curve), based on the interval given. These two curves will be referred to as M^η and M^ρ. Each data point is a tuple of three elements corresponding to $\{\lambda, \eta(\lambda), \rho(\lambda)\}$ given by the point, $(\lambda, \eta(\lambda))$ on the Solution Norm Curve and $(\lambda, \rho(\lambda))$ on the Residual Norm Curve.

The number of initial data points, N, should be at least equal to the number of parameters in the parametric curve used for the data fitting. If we divide the two curves into three parts corresponding to the line segments seen in Figure 6.3 and 6.4, then the interval, $[a, b]$, should be chosen such that the data points are distributed across the three line segments each of the two curves consist of.

After having initialized the variables, the data fitting is done solving a problem in the Least Squares sense. Hereafter the loop starts by finding a new point according to the distribution of the λ-values. Different strategies can be used for finding such a point, and they will be discussed in Section 6.9.1. With the new point added, a new Least Squares problem is solved. Each new fit is compared with the previous one in order to determine the quality of the new point. If the new fit is a lot different from the previous, it is a symptom of the point being chosen badly and therefore does not contribute to a better fit. This shows that the parametric curve does not have a shape like the data points. Such situations will be discussed in Section 6.7.

Output	Description
found	Flag, telling whether the algorithm stopped due to convergence or not
p^{sol}	Parameters for the fitted Solution Norm Curve
p^{res}	Parameters for the fitted Residual Norm Curve
λ^{F}	Regularization parameter found

Table 4.2: *Table of output parameters for the algorithm.*

When this is the case the point is removed and next iteration starts. In order to prevent returning the same point, a fudge parameter, c, is introduced. This parameter is used when finding a new point, and only changed if the previous point was not satisfactory.

The loop continues until either the regularization parameter or the fit has become stable, i.e. it does not change between two consecutive iterations. For this stop criteria ϵ is used.

The Output parameters can be seen in Table 4.2.

4.2 The FRe-PUF Algorithm

Now the algorithm will be shown. Boldface font will be used when a variable is a vector and not a scalar value, ϵ is used as a stopping criteria, k^{max} and n^{max} are handbrakes for stopping, if the solution does not converge. The $\mathbf{p^{sol}}[\mathbf{k}]$ and $\mathbf{p^{res}}[\mathbf{k}]$ are vectors holding the parameters for the parametric curves used when fitting the Solution Norm Curve and the Residual Norm Curve, respectively, for the k^{th} iteration. κ is the curvature of the fitted L-Curve given by the curves with parameters p^{sol} and p^{res}.

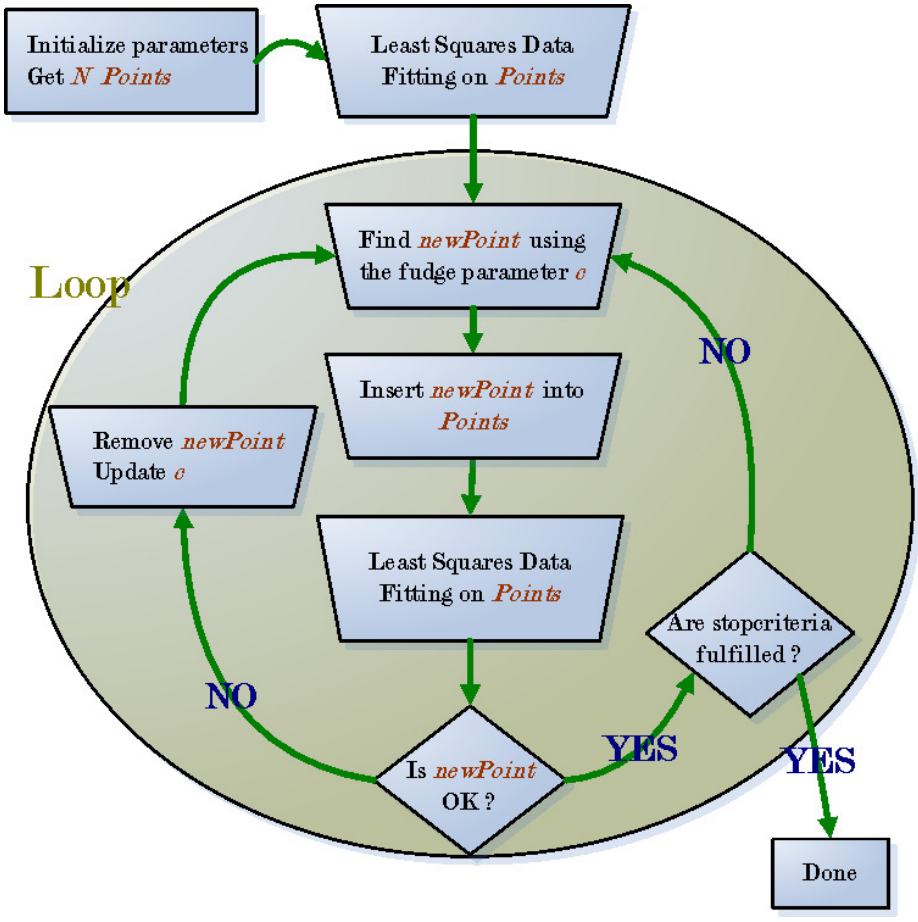

Figure 4.1: *Flow chart of the algorithm.*

Algorithm 1 Finding Regularization Parameter Using Fit

INPUT $[a, b]$, k^{max}, n^{max}, c, ϵ_i
begin
$found \leftarrow$ false
$k \leftarrow 0$
for $i = 0$ to $n - 1$ **do**
$points[i] \leftarrow \{\lambda[i], \eta(\lambda[i]), \rho(\lambda[i])\}$;
end for
set initial guess for $\mathbf{p}^{\mathbf{sol}}[k]$
set initial guess for $\mathbf{p}^{\mathbf{res}}[k]$
$\mathbf{p}^{\mathbf{sol}}[k] \leftarrow \arg\min_{\mathbf{p}^{\mathbf{sol}}} \|M^{\eta}(\lambda, \mathbf{p}^{\mathbf{sol}}[k]) - \mathbf{points}^{\eta}\|$
$\mathbf{p}^{\mathbf{res}}[k] \leftarrow \arg\min_{\mathbf{p}^{\mathbf{res}}} \|M^{\rho}(\lambda, \mathbf{p}^{\mathbf{res}}[k]) - \mathbf{points}^{\rho}\|$
while (**NOT** $found$) $\wedge$ ($k < k^{max}$) $\wedge$ (Size of **points** $< n^{max}$) **do**
$k \leftarrow k + 1$
$newPoint \leftarrow$ find a new point from $\lambda, p^{res}, p^{sol}$, and c
INSERT $newPoint$ into **points**
$\mathbf{p}^{\mathbf{sol}}[k] \leftarrow \arg\min_{\mathbf{p}^{\mathbf{sol}}} \|M^{\eta}(\lambda, \mathbf{p}^{\mathbf{sol}}[k]) - \mathbf{points}^{\eta}\|$
$\mathbf{p}^{\mathbf{res}}[k] \leftarrow \arg\min_{\mathbf{p}^{\mathbf{res}}} \|M^{\rho}(\lambda, \mathbf{p}^{\mathbf{res}}[k]) - \mathbf{points}^{\rho}\|$
$\lambda^{\mathrm{F}} \leftarrow \arg\min_{\lambda} \kappa$
if ($\mathbf{p}^{\mathbf{res}}[k-1] \approx \mathbf{p}^{\mathbf{res}}[k]$) $\wedge$ ($\mathbf{p}^{\mathbf{sol}}[k-1] \approx \mathbf{p}^{\mathbf{sol}}[k]$) **then**
Update $found$ according to stop criteria
else
REMOVE $newPoint$ from **points**
UPDATE c
end if
end while
end begin
OUTPUT $found$, p^{sol}, p^{res}, λ^{F}

4.2.1 Comments to the Pseudo Code

As described in the previous section, there are some requirements for the algorithm to work robustly. It is important that at *(Line number 5)*, the data points are chosen so that λ values are on both sides of the right bend in the Solution Norm Curve, preferably two points to the right of this bend.

λ-value corresponding to the right bend can be found as approximations as well as some of the parameters for the curves. When doing the data fitting, *(Line number 9-10)*, different methods can be used, normally an iterative minimizer on the two-norm of the residuals is used.

At *(line number 13)* a new point must be found. Different strategies will be discussed later, but in general the point can be found by looking at the fitted curve. Similar to finding the initial points, each line segment the two curves consist of, should be represented. c parameter controls the method of finding the new point. This parameter should be altered if there is an indication of a badly chosen point. An indicator of this is if the fit is changed drastically, suggesting that the point results in a very different curve. It is especially important to monitor those parameters, with a good guess. This is checked at *(line number 18)*.

At *(line number 18)* the stopping criteria is checked. To do this an estimate of a good λ is calculated as the regularization parameter of maximum curvature.

$$\lambda^{\mathrm{F}} = \underset{a<\lambda<b}{\operatorname{argmax}}\, \kappa$$

where κ is the curvature of the L-Curve.

Having this regularization parameter, we can get the stop criteria. The loop must stop when no further convergence can be obtained, that is when a stable solution has be found. An indication of this is when two consecutive solutions have not changed. Therefore we will look at the parameters p^{sol} and p^{res} and also at the regularization parameter, λ^{F}. When none of these have changed much, the algorithm has found a stable solution. If however no stable solution can be reached, the algorithm should still stop, but this time with a flag showing no solution is found.

The stop criteria depends on the number of data points, the number of iterations, the change in λ^{F} and the obtained change in the parameters for the L-Curve. The stopping criteria is written more formally as follows:

1. $k > k^{max}$

2. Size of $\mathbf{points} > n^{max}$
3. $\|\mathbf{p}^{\mathbf{res}}[k-1] - \mathbf{p}^{\mathbf{res}}[k]\|_2 < \epsilon_1$
4. $\|\mathbf{p}^{\mathbf{sol}}[k-1] - \mathbf{p}^{\mathbf{sol}}[k]\|_2 < \epsilon_2$
5. $\|\lambda^{\mathrm{F}}[k] - \lambda^{\mathrm{F}}[k-1]\|_2 < \epsilon_3$.

When any of these criteria are satisfied, the algorithm stops. If it is one of the first two criteria, the algorithm has not succeeded in converging, and has stopped due to the handbrake. On the other hand, if all of the last 3 criteria causes the algorithm to stop, it has converged successfully.

Chapter 5

Getting to a Curve

Now we have seen the FRe-PUF and a brief description of some details in it. In the following, we discuss, as well as show, some of the analysis which has lead to the algorithm.

Starting with the parametric curve, the implementation of the data fitting will be described and followed by a brief discussion on different optimal regularization parameters, λ and their corresponding sougth models, x^λ. Then a data fitting solution will be shown and through that, the basic idea behind the algorithm will be shown working. Different regularization parameters will be compared with the one from FRe-PUF.

Having seen data fitting as a useful tool for locating a good regularization parameter, we try different methods for fine tuning this idea. These optimizations include looking at the initial values, parameters and variables in the algorithm, as well as analyzing the method through testing on different problems.

All the optimization ideas have rised from the analysis of the results obtained when locating an optimal regularization parameter, using the FRe-PUF. They have been done on some or all of the problems from Regularization Tools [5].

5.1 Curve Description

To be able to do the data fitting, we need a proper description of the data points, $(\mathtt{t}, \mathtt{Y})$. That is, we want to describe the $\gamma(\mathtt{t})$ with a parametric curve, $M_p(\mathtt{t})$. Firstly, we examine the L-Curve, the Residual Norm Curve and the Solution Norm Curve, in the following.

To get the best fit, we need to know the shape of the curve to be fitted and we need some starting values for the parameters. These are found by analyzing the norms. To keep the analysis simple, we will look at the TSVD, instead of the Tikhonov Regularization. In order to show the close relation between these two, the analysis will be made for one of the norms for the Tikhonov Regularization as well.

To describe the shape of the Residual Norm Curve and the Solution Norm Curve, we begin by looking at the two norms for the TSVD. To start the analysis, the behavior can be divided into different situations. We get these situations from looking at a Picard Plot. Please recall that σ_i are non-increasing and non-negative and n corresponds to the number of singular values. We see from the Picard Plot in Figure 5.1, the singular values are decaying till they reach the noise level, ξ. At this point, the noise dominates over the σ_i. Introducing l as the number of singular values which is dominating over the noise contributions, we see that $|{u_i}^{\mathrm{T}} b^{\mathrm{exact}}|$ dominates for $i < l$, and $|\xi|$ will be the contribution for $i > l$. Having introduced these two situations, we analyze the two norms that the L-Curve consists of.

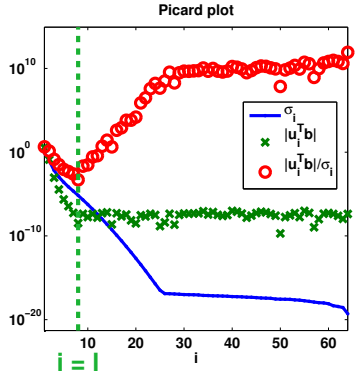

Figure 5.1: *The Picard Plot for the* `Foxgood problem` [5]. *The point where* $|{u_i}^{\mathrm{T}} b^{\mathrm{exact}} + \xi|$ *is dominated by* ξ*, is* l*, and has been shown.*

5.2 Residual Norm Curve

If we focus on the Residual Norm Curve for the TSVD, it can be written as follows:

$$\|Ax_k - b\|_2^2 = \sum_{i=k+1}^{n} (u_i{}^{\mathsf{T}} b)^2. \tag{5.1}$$

This shows, together with Figure 5.1, that, depending on how k is chosen, there are two situations: Situation 1 where k is chosen as $1 < k < l$ and situation 2 where k is chosen as $l < k < n$.

5.2.1 Situation 1

Taking the first situation, presented above, under consideration, we can divide the norm into a noisy part and an exact part:

$$\begin{aligned} \|Ax_k - b\|_2^2 &= \sum_{i=k+1}^{n} (u_i{}^{\mathsf{T}} b)^2 \\ &\simeq \sum_{i=k+1}^{l} (u_i{}^{\mathsf{T}} b^{\text{exact}})^2 + \sum_{i=l+1}^{n} (u_i{}^{\mathsf{T}} \xi)^2 \\ &\simeq \sum_{i=k+1}^{l} (u_i{}^{\mathsf{T}} b^{\text{exact}})^2 + (n-l+1)(u_i{}^{\mathsf{T}} \xi)^2. \end{aligned} \tag{5.2}$$

This latter equation shows that when k increases, the value of $\|Ax_k - b\|_2^2$ decreases. The residual norm for this interval will then be increasing when plotted versus σ_i .

5.2.2 Situation 2

Hereafter we look at $l < k < n$. This gives:

$$\begin{aligned} \|Ax_k - b\|_2^2 &= \sum_{i=k+1}^{n} (u_i{}^{\mathsf{T}} b)^2 \\ &\simeq \sum_{i=k+1}^{n} (u_i{}^{\mathsf{T}} \xi)^2 \\ &\simeq (n-k+1)(u_i{}^{\mathsf{T}} \xi)^2. \end{aligned}$$

Now taking the square root:

$$\begin{aligned} \|Ax_k - b\|_2 &\simeq \sqrt{(n-k+1)}({u_i}^T\xi) \simeq \sqrt{n-k}({u_i}^T\xi) \\ &\simeq \sqrt{n}\sqrt{1-\frac{k}{n}}({u_i}^T\xi). \end{aligned} \tag{5.3}$$

We use the first term of a Taylor expansion of $\sqrt{1-\frac{k}{n}}$ giving:

$$\|Ax_k - b\|_2 \simeq \sqrt{n}\left(1 - \frac{1}{2}\frac{k}{n}\right)({u_i}^T\xi). \tag{5.4}$$

This shows that for $k \ll n$, the Residual Norm Curve is a constant $\sqrt{n}\,({u_i}^T\xi)$, and for other values it is decreasing.

To illustrate the curve parts just derived they have been shown along with the data points and σ_i in figure 5.2.

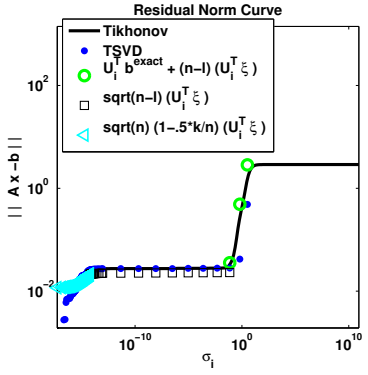

Figure 5.2: *Illustration of the derived curve segments for the Residual Norm Curve. The Residual Norm Curve for computed TSVD and Tikhonov Regularization for the* `Baart problem` [5] *has been shown as well.*

5.3 Solution Norm Curve

In the following part we will look at the Solution Norm Curve and derive it in a similar manner. Like with the Residual Norm Curve we introduce l and divide it into the same situations.

The solution norm can be written as:

$$\|x_k\|_2^2 = \sum_{i=1}^{k} \frac{{u_i}^{\mathrm{T}} b}{\sigma_i}. \tag{5.5}$$

5.3.1 Situation 1

At first, we look at $1 < \mathrm{k} < l$:

$$\|x_k\|_2^2 \;=\; \sum_{i=1}^{k} \left(\frac{{u_i}^{\mathrm{T}} b}{\sigma_i}\right)^2 \simeq \sum_{i=1}^{k} \left(\frac{{u_i}^{\mathrm{T}} b^{\mathrm{exact}}}{\sigma_i}\right)^2 = \|x^{\mathrm{exact}}\|_2^2. \tag{5.6}$$

At this point, because σ_i is decaying faster than ${u_i}^{\mathrm{T}} b^{\mathrm{exact}}$, the solution norm decreases, when k increases.

5.3.2 Situation 2

Next we look at $l < k < n$. Here the solution norm can e written as:

$$\begin{aligned} \|x_k\|_2^2 &= \sum_{i=1}^{k} \left(\frac{{u_i}^{\mathrm{T}} b}{\sigma_i}\right)^2 \\ &\simeq \sum_{i=1}^{l} \left(\frac{{u_i}^{\mathrm{T}} b^{\mathrm{exact}}}{\sigma_i}\right)^2 + \sum_{i=l+1}^{k} \left(\frac{{u_i}^{\mathrm{T}} \xi}{\sigma_i}\right)^2 \\ &\simeq \|x^{\mathrm{exact}}\|_2^2 + (k-l+1)\left(\frac{{u_i}^{\mathrm{T}} \xi}{\sigma_i}\right)^2. \end{aligned} \tag{5.7}$$

So we see when k increases, the Solution Norm Curve increases as well.

This two situations are illustrated in Figure 5.3.

5.4 Tikhonov Analysis

Now we have been able to derive the behavior of the Residual Norm Curve and the Solution Norm Curve for TSVD. As a comparison, the Solution Norm Curve is derived using Tikhonov Regularization in the following.

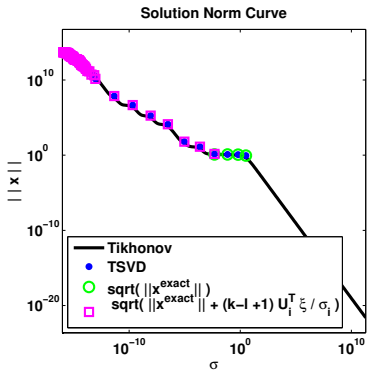

Figure 5.3: *Illustration of the derived curve segments for the Solution Norm Curve. The Solution Norm Curve for computed TSVD and Tikhonov Regularization for problem the* `Baart problem` [5] *have been shown as well.*

From the SVD we know that the solution norm is written as:

$$\|x\lambda\|_2^2 = \sum_{i=1}^{n} \left(f_i \frac{{u_i}^{\mathrm{T}} b}{\sigma_i} \right)^2, \tag{5.8}$$

with

$$f_i = \frac{{\sigma_i}^2}{{\sigma_i}^2 + \lambda^2}. \tag{5.9}$$

We divide the analysis into different situations, depending on how λ is chosen. The different possibilities of choosing λ derived from the filter factors in Equation (5.9) and the Picard Plot in Figure 5.1. Now we see that λ can be chosen such f_i is close to one or close to $\frac{{\sigma_i}^2}{\lambda^2}$. Moreover, it is possible to choose $\lambda \ll \sigma_n$ and $\lambda \gg \sigma_1$.

5.4.1 Situation 1

If $\lambda \gg \sigma_i$ and $1 \geq i \geq n$ then it implies that $\lambda \gg \sigma_1$. We write the solution norm as:

$$\begin{aligned} \|x^\lambda\|_2^2 &= \sum_{i=1}^{n} \left(f_i \frac{{u_i}^{\mathrm{T}} b}{\sigma_i} \right)^2 \\ &\simeq \sum_{i=1}^{n} \left(\frac{{\sigma_i}^2}{\lambda^2} \frac{{u_i}^{\mathrm{T}} b}{\sigma_i} \right)^2 = \sum_{i=1}^{n} \frac{{\sigma_i}^2}{\lambda^4} \left({u_i}^{\mathrm{T}} b \right)^2 \Rightarrow \\ \|x^\lambda\|_2 &\simeq \frac{1}{|\lambda|^2} \sum_{i=1}^{n} \sigma_i \left({u_i}^{\mathrm{T}} b \right) \qquad , \lambda > \sigma_1. \end{aligned} \tag{5.10}$$

From this equation, we observe that the Solution Norm Curve, $\|x^\lambda\|_2$, for $\lambda > \sigma_1$ will be a straight line with a negative slope, when plotted on a logarithmic scale.

5.4.2 Situation 2

Next we look at $\sigma_n < \lambda < \sigma_1$. If we introduce l, $n > l > 1$, such that l corresponds to the number of f_i close to 1 and $p:\ 1 < p < l$ such that $\lambda \simeq \sigma_p$. Then we make the following approximation of the contributions of ${u_i}^{\mathrm{T}} b$:

$$\begin{aligned} \|x^\lambda\|_2^2 &= \sum_{i=1}^{n} \left(f_i \frac{{u_i}^{\mathrm{T}} b}{\sigma_i} \right)^2 \\ &\simeq \sum_{i=1}^{l} \left(\frac{{\sigma_i}^2}{{\sigma_i}^2 + \lambda^2} \frac{{u_i}^{\mathrm{T}} b^{\mathrm{exact}}}{\sigma_i} \right)^2 + \sum_{i=l+1}^{n} \left(\frac{{\sigma_i}^2}{{\sigma_i}^2 + \lambda^2} \frac{{u_i}^{\mathrm{T}} \xi}{\sigma_i} \right)^2. \end{aligned}$$

Now looking at $f_i = \frac{{\sigma_i}^2}{{\sigma_i}^2 + \lambda^2}, \quad \lambda \simeq \sigma_p$ we see:

$$\begin{aligned} \|x^\lambda\|_2^2 &\simeq \\ &\simeq \sum_{i=1}^{p} \left(\frac{{\sigma_i}^2}{{\sigma_i}^2} \frac{{u_i}^{\mathrm{T}} b^{\mathrm{exact}}}{\sigma_i} \right)^2 + \sum_{i=p+1}^{l} \left(\frac{{\sigma_i}^2}{\lambda^2} \frac{{u_i}^{\mathrm{T}} b^{\mathrm{exact}}}{\sigma_i} \right)^2 + \sum_{i=l+1}^{n} \left(\frac{{\sigma_i}^2}{\lambda^2} \frac{{u_i}^{\mathrm{T}} \xi}{\sigma_i} \right)^2 \\ &\simeq \sum_{i=1}^{p} \left(\frac{{u_i}^{\mathrm{T}} b^{\mathrm{exact}}}{\sigma_i} \right)^2 + \frac{1}{\lambda^4} \left(\sum_{i=p+1}^{l} \left({\sigma_i}^2 \frac{{u_i}^{\mathrm{T}} b^{\mathrm{exact}}}{\sigma_i} \right)^2 + \sum_{i=l+1}^{n} \left({\sigma_i}^2 \frac{{u_i}^{\mathrm{T}} \xi}{\sigma_i} \right)^2 \right). \end{aligned}$$

At this point we see that the first factor $\sum_{i=1}^{p} \left(\frac{{u_i}^{\mathrm{T}} b^{\mathrm{exact}}}{\sigma_i} \right)^2$ actually is the Truncated SVD. Remembering σ_i is non-increasing, we see the second factor can be

approximated to the largest σ_i :

$$\begin{aligned} & \frac{1}{\lambda^4}\left(\sum_{i=p+1}^{l}\left({\sigma_i}^2\frac{{u_i}^{\mathsf{T}}b^{\text{exact}}}{\sigma_i}\right)^2+\sum_{i=l+1}^{n}\left({\sigma_i}^2\frac{{u_i}^{\mathsf{T}}\xi}{\sigma_i}\right)^2\right) \\ \simeq\ & \frac{1}{\lambda^4}\left(\left({\sigma_{p+1}}^2\frac{{u_{p+1}}^{\mathsf{T}}b^{\text{exact}}}{\sigma_{p+1}}\right)^2+\left({\sigma_{l+1}}^2\frac{{u_{l+1}}^{\mathsf{T}}\xi}{\sigma_{l+1}}\right)^2\right). \end{aligned}$$

We know that $\sigma_{p+1}\geq\sigma_{l+1}$ and ${u_{p+1}}^{\mathsf{T}}b^{\text{exact}}\geq\xi$, and using the definition of p, $\lambda\simeq\sigma_p$ and $\sigma_p\simeq\sigma_{p+1}$ gives:

$$\begin{aligned} \|x^{\lambda}\|_2^2 &\simeq \|{x^{\text{exact}}}_p\|_2^2+\frac{1}{\lambda^4}\left(\left({\sigma_{p+1}}^2\frac{{u_{p+1}}^{\mathsf{T}}b^{\text{exact}}}{\sigma_{p+1}}\right)^2+\left({\sigma_{l+1}}^2\frac{{u_{l+1}}^{\mathsf{T}}\xi}{\sigma_{l+1}}\right)^2\right) \\ \|x^{\lambda}\|_2^2 &\simeq \|{x^{\text{exact}}}_p\|_2^2+\frac{1}{\lambda^4}\left(\sigma_{p+1}{u_{p+1}}^{\mathsf{T}}b^{\text{exact}}\right)^2 \\ &\simeq \|{x^{\text{exact}}}_p\|_2^2+\frac{\left(\sigma_{p+1}{u_{p+1}}^{\mathsf{T}}b^{\text{exact}}\right)^2}{{\sigma_{p+1}}^4}\Rightarrow \\ \|x^{\lambda}\|_2 &\simeq \|{x^{\text{exact}}}_p\|_2+\frac{{u_{p+1}}^{\mathsf{T}}b^{\text{exact}}}{\sigma_{p+1}} \\ &\simeq \|{x^{\text{exact}}}_{p+1}\|_2, \qquad \sigma_l<\lambda<\sigma_1. \end{aligned} \tag{5.11}$$

$\|{x^{\text{exact}}}_{p+1}\|_2$ increases slowly as λ decreases. This can easily be seen as when λ decreases, p will increase, and more $\frac{{u_i}^{\mathsf{T}}b^{\text{exact}}}{\sigma_i}$ is included. However, as $\frac{{u_i}^{\mathsf{T}}b^{\text{exact}}}{\sigma_i}$ will decay for $\sigma_l<\lambda<\sigma_1$ then the increase is relative small.

5.4.3 Situation 3

When looking at $\lambda<\sigma_l$, and starting by introducing $\lambda\simeq\sigma_p$ with $l<p<n$. We make the following approximation:

$$\begin{aligned} \|x^{\lambda}\|_2^2 &= \sum_{i=1}^{l}\left(\frac{{\sigma_i}^2}{{\sigma_i}^2}\frac{{u_i}^{\mathsf{T}}b^{\text{exact}}}{\sigma_i}\right)^2+\sum_{i=l+1}^{p}\left(\frac{{\sigma_i}^2}{{\sigma_i}^2}\frac{{u_i}^{\mathsf{T}}\xi}{\sigma_i}\right)^2+\sum_{i=p+1}^{n}\left(\frac{{\sigma_i}^2}{\lambda^2}\frac{{u_i}^{\mathsf{T}}\xi}{\sigma_i}\right)^2 \\ &\simeq \sum_{i=1}^{l}\left(\frac{{u_i}^{\mathsf{T}}b^{\text{exact}}}{\sigma_i}\right)^2+\sum_{i=l+1}^{p}\left(\frac{{u_i}^{\mathsf{T}}\xi}{\sigma_i}\right)^2+\frac{1}{\lambda^4}\sum_{i=p+1}^{n}\left(\sigma_i{u_i}^{\mathsf{T}}\xi\right)^2 \\ &\simeq \|x^{\text{exact}}\|_2^2+\left(\frac{{u_i}^{\mathsf{T}}\xi}{\sigma_p}\right)^2+\frac{1}{\lambda^4}\left(\sigma_{p+1}{u_i}^{\mathsf{T}}\xi\right)^2. \end{aligned}$$

Since the last approximation is due to the fact that $\sum_{i=1}^{l}\left(\frac{{u_i}^{\mathsf{T}}b^{\text{exact}}}{\sigma_i}\right)^2\simeq\|x^{\text{exact}}\|_2^2$ and σ_i is non-increasing, the largest element will dominate in such a summation.

Since we know that $\sigma_p \simeq \lambda$, we can make the following approximation:

$$\begin{aligned}
\|x^{\lambda}\|_2^2 &\simeq \|x^{\text{exact}}\|_2^2 + \left(\frac{{u_i}^{\mathrm{T}}\xi}{\sigma_p}\right)^2 + \frac{1}{\lambda^4}\left(\sigma_{p+1}{u_i}^{\mathrm{T}}\xi\right)^2 \\
&\simeq \|x^{\text{exact}}\|_2^2 + \left(\frac{{u_i}^{\mathrm{T}}\xi}{\lambda}\right)^2 + \frac{1}{\lambda^4}\left(\lambda{u_i}^{\mathrm{T}}\xi\right)^2 \\
&\simeq \|x^{\text{exact}}\|_2^2 + \left(\frac{2({u_i}^{\mathrm{T}}\xi)^2}{\lambda^2}\right).
\end{aligned}$$

It can be observed that for $\sigma_n < \lambda < \sigma_l$ the Solution Norm Curve will be a straight line, with a slope of twice the slope from situation 1, a logarithmic plot.

5.4.4 Situation 4

Finally, we will look at $\lambda \ll \sigma_n$. Here

$$\|x^{\lambda}\|_2^2 \simeq \sum_{i=1}^{n}\left(\frac{{\sigma_i}^2}{\sigma_i}\frac{{u_i}^{\mathrm{T}}b}{\sigma_i}\right)^2 = \sum_{i=1}^{n}\left(\frac{{u_i}^{\mathrm{T}}b}{\sigma_i}\right)^2.$$

This is the Least Squares solution:

$$\begin{aligned}
\|x^{\lambda}\|_2^2 &\simeq \sum_{i=1}^{n}\left(\frac{{u_i}^{\mathrm{T}}b}{\sigma_i}\right)^2 = \\
\|x^{\lambda}\|_2 &\simeq \|x^{\text{naive}}\|_2. \qquad (5.12)
\end{aligned}$$

Now, as we know the shape of the of the Solution Norm Curve, it is possible to show each line segment. This can be seen in Figure 5.4, where the computed Solution Norm Curve has been shown as data points for the comparison with the line segments.

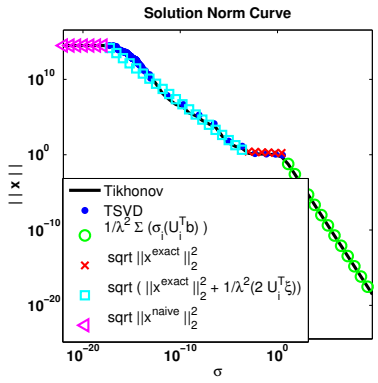

Figure 5.4: *Illustration of the derived curve segments for the Solution Norm Curve. The Solution Norm Curve for computed TSVD and Tikhonov Regularization for the* `Baart problem` [5] *have been shown as well.*

Chapter 6

Deriving a Proper Parametric Curve

We have now seen the shape of the two curves and want to derive a parametric curve which has the same shape. The data fitting will be made using some calculated points on the L-Curve. In order to make the fit as good as possible, the parametric curve has to be able to fit the data points. As for the L-Curve, there is two possible sets of data points which can be fitted. There are the L-Curve itself, which is one curve with λ being the parameter, for each of the two function, $\|Ax^\lambda - b\|_2$ and $\|x^\lambda\|_2$, and there is the two curves separately.

We will concentrate on fitting each of the two norms. When doing the fitting it is necessary to have as many data points as there are parameters in the parametric curve. For that reason the curve should have few parameters but still describe the overall behavior of the norm.

Now we will make a derivation of the parametric curve used for the data fitting of the Residual Norm Curve and Solution Norm Curve.

6.1 Locating Optimal Regularization Parameters

We are able to derive the shape of the two curves from the theory, see 5.1. Now we will locate the interesting parts of the two curves such that we can determine a parametric curve. In order to do this, we start by locating the optimal regularization parameter. This optimal regularization parameter should give the best solution to the inverse problem, when x^{exact} is known. Two values can be considered the optimal parameter and they will both be shown on the L-Curve and on the Solution Norm Curve and the Residual Norm Curve. Thus, we start by looking at the optimal parameters. For the inverse problem of $Ax = b$, we are interested in finding x^{λ} from the Tikhonov Regularization giving the best model. Two such regularization parameters giving good models can be considered:

- $\min \|x^{\text{exact}} - x^{\lambda}\|_2$, which is the point where the regularized solution is closest to the original model. That is, the point where the error of the solution is smallest.
- $\min \|Ax^{\lambda} - b^{\text{exact}}\|_2$, which gives the point of the smallest residual norm.

Both these quantities can only be calculated, when the exact solution is known, which is one of the reasons for testing the method on small problems.

In this thesis, the model obtained using the parameters giving the above two terms will be referred to as x^{err} and x^{res}, respectively, and the corresponding parameters will be λ^{err} and λ^{res}. Besides the two optimal parameters, another regularization parameter is desired. Since the data fitting is imitating the analytical L-Curve Criterion, the regularization parameter found using this method, is of interest. The model obtained using the parameter from the corner point of the analytical L-Curve, will be referred to as x^{L} and the regularization parameter will be λ^{L}. As a final regularization parameter, we will introduce the value found using the FRe-PUF. This will be denoted λ^{F} with the solution x^{F}.

Using the first three regularization parameters, λ^{L}, λ^{err} and λ^{res}, different L-Curves have been examined, in order to locate the interesting parts. An example of such an examination can be seen in Figures 6.1 and 6.2, where the three curves have been shown together with the three interesting parameters.

We see that the part of the L-Curve are without the two concave parts. As such the two end areas are of no interest. Looking at the Residual Norm Curve and Solution Norm Curve, we see the interesting parts of the two curves both

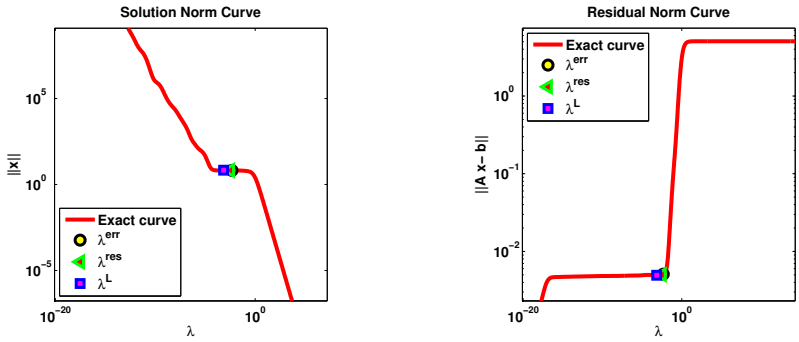

Figure 6.1: *The Solution Norm Curve(left) and Residual Norm Curve(right), together with the three regularization parameters. The problem shown is the* `Foxgood problem` [5].

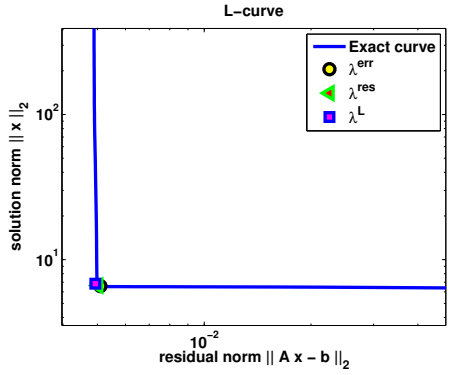

Figure 6.2: *The L-Curve, together with the 3 regularization parameters. The problem shown is the* `Foxgood problem` [5].

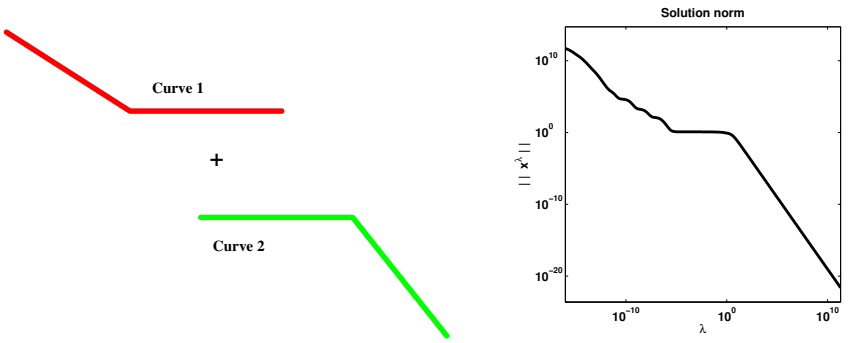

Figure 6.3: *Illustration of the two curves used for the data fitting(left). Combined they give the Solution Norm Curve(right).*

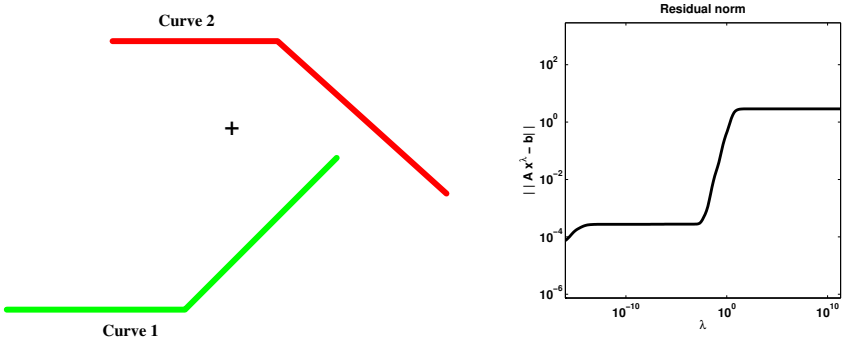

Figure 6.4: *Illustration of the two curves used for the data fitting(left), Combined they give the Residual Norm Curve(right).*

consist of three parts. The Residual Norm Curve has two almost horizontal parts with a vertical part connecting them. As the Solution Norm Curve there is two negative slope parts with a horizontal part in between. This corresponds to the derivation in Section 5.1

We see the interesting curves are made by two curves each with two line segments. We will find such a parametric curve to use for data fitting, corresponding to the shape of the two norms. An idea of such parametric curve is a combination of two curves, each with a flat line and a sloped line. An illustration of the situation can be seen in Figure 6.3, where the two curves, which in combination can give the Solution Norm Curve, are shown. A similar figure for the Residual Norm Curve has been shown in Figure 6.4.

Now we will derive a parametric description for such a curve.

6.2 Getting to a Curve

The combined curves from Figure 6.3 and 6.4 give a total of five for parameters for the Residual Norm Curve and six parameters for the Solution Norm Curve. The reason for fewer parameters for the Residual Norm Curve is the slopes for the curves are equal. Two combined conic sections can be used as parametric curves.

To derive a useful parametric curve we start by looking at a curve with a bend, as curve 2 in Figure 6.3.

The conic section is described as [1] :

$$Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0.$$

To be able to make a line with a bend, some of the parameters is set to zero. When looking at the wanted curve we see for $x \to \infty, \quad y \to 0$. Therefore the terms consisting only of x but not y:

$$Bxy + Cy^2 + Ey + F = 0.$$

Solving this equation with respect to y gives:

$$y = \frac{-(Bx + E) \pm \sqrt{(Bx + E)^2 - 4CF}}{2C}.$$

Since the parameter can be chosen freely, the equation is simplified. At first B can be isolated and some of the parameters substituted giving:

$$y = \pm\hat{B}\left(x - \hat{E} \pm \sqrt{\left(x - \hat{E}\right)^2 + \hat{F}^2}\right), \tag{6.1}$$

where the $\hat{}$ denotes change of the parameter.

[1] Please notice, that for the derivation we will use a notation with capital letter for the parameters and x and y for the variables. This will conflict with the notation used elsewhere in this thesis, but since it is only used for the derivation it should not cause any confusion.

An examination of equation (6.1) tells us:

$$\begin{aligned}
\hat{B}\left(x-\hat{E}\pm\sqrt{\left(x-\hat{E}\right)^2+\hat{F}^2}\right) &\cong \hat{B}\left(x-\hat{E}\pm|x-\hat{E}|\right),\ |x-\hat{E}|\gg\hat{F}\\
x>\hat{E}\ :\quad \hat{B}\left(x-\hat{E}\pm|x-\hat{E}|\right) &= \begin{cases} 2\hat{B}\left(x-\hat{E}\right)\ , & ''+''\\ 0\ , & ''-''\end{cases}\\
x<\hat{E}\ :\quad \hat{B}\left(x-\hat{E}\pm|x-\hat{E}|\right) &= \begin{cases} 0\ , & ''+''\\ 2\hat{B}\left(x-\hat{E}\right)\ , & ''-''\end{cases}
\end{aligned}$$

We see for this curve, $\hat{B}$ determines the slope of the steep part, $\hat{F}$ determines how sharp the bend is and $\hat{E}$ tells where the bend is located. This corresponds to the curve we are seeking.

6.2.1 The Parametric Curve

Equation (6.1) describes one line with a bend, for the data fitting, the parametric curve we are interested in, consists of a combination of two such lines. Therefore renaming and combining curves of the type from equation (6.1) and adding a translation parameter gives:

$$y=\pm(p_1x-p_2)\pm\sqrt{(p_1x-p_2)^2+p_3^2}\pm(p_4x-p_5)\pm\sqrt{(p_4x-p_5)^2+p_3^2}+p_6.$$

The $\pm$ should be chosen so it corresponds to the Solution Norm Curve and Residual Norm Curve, see Figure 6.3 and 6.4. We can immediately see in order to make the Residual Norm Curve, from Figure 6.1, we have

$$p_4=-p_1,$$

giving the flat parts of the curve.

The parametric curve for the Residual Norm Curve will be:

$$g^{res}(x)=p_1\left(-p_2+\sqrt{(x-p_2)^2+p_3^2}+p_5+\sqrt{(x-p_5)^2+p_3^2}\right)+p_6. \quad (6.2)$$

Whereas the Solution Norm Curve is:

$$\begin{aligned}
g^{sol}(x) &= -p_1\left(x-p_2+\sqrt{(x-p_2)^2+p_3^2}\right)-\\
&\quad p_4\left(x-p_5-\sqrt{(x-p_5)^2+p_3^2}\right)+p_6. \qquad (6.3)
\end{aligned}$$

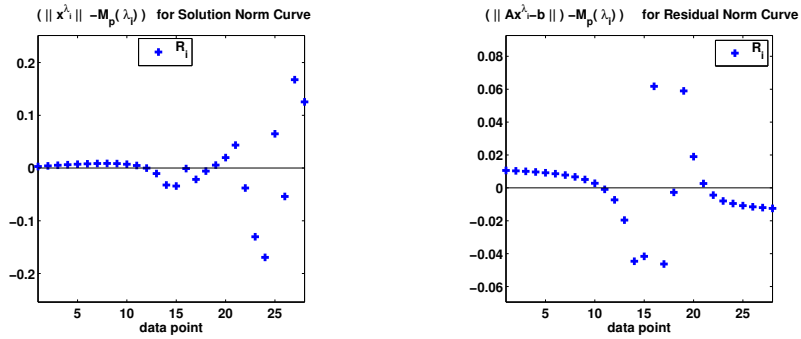

Figure 6.5: *The residuals for the fitted curves for the Residual Norm Curve and the Solution Norm Curve. The problem shown is the* `Shaw problem` [5] *with 28 data points.*

To see if the parametric curve is chosen poorly, we will examine the residuals R_i. If there are no trends in these, the parametric curve is suited for the data points. Here *trend* refers to R_i and R_j having same sign for $j = i + 1$. That is, two neighboring residuals should have different signs. A figure of the residuals for a fit made using the FRe-PUF with one loop can be seen in Figure 6.5.

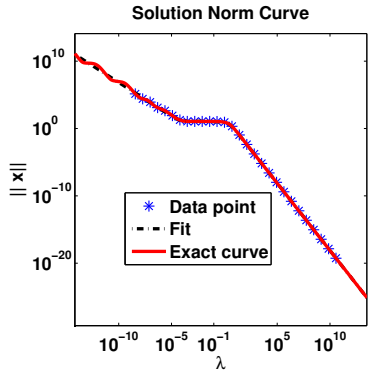

Figure 6.6: *A fit of Solution Norm Curve. The data points used for the data fitting as well as the real Solution Norm Curve have been shown. The problem shown is the* `Shaw problem` [5].

If we compare the residuals with the data fitting seen in Figure 6.6 and 6.7, we see the residuals have trends. Though the trends are mostly for the flat parts of the Residual Norm Curve and Solution Norm Curve. The errors here are relatively small compared to the residuals for the data points where curves

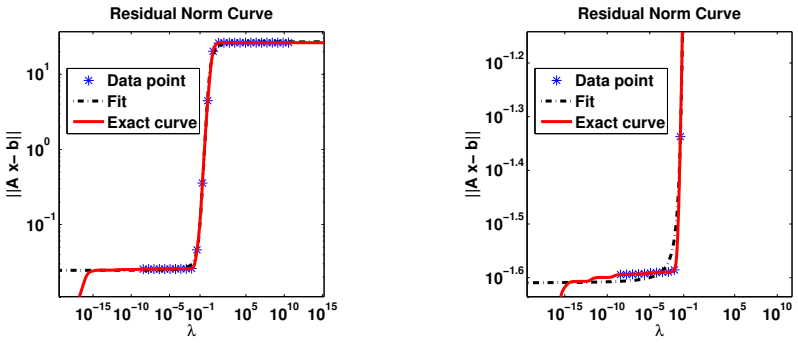

Figure 6.7: *A fit of Residual Norm Curve. The data points used for the data fitting as well as the real Residual Norm Curve have been shown. The figure to the right is the same as the left zoomed in on one bend. The problem shown is the* `Shaw problem` [5].

are most difficult to fit. For the Solution Norm Curve this is at one bend and at the left part, where the Solution Norm Curve has steps but the parametric curve is a straight line. The Residual Norm Curve also has the biggest residuals where the curve seems most difficult to fit. This is between the two bends.

6.3 Doing the First Fitting

As we have just observed that the data fitting is possible. Now we will try using the data fitting to make a fit of the L-Curve and locate a regularization parameter in the corner as the maximum curvature. At first, we do the data fitting corresponding to one sweep through the loop described in the Section 4.2. Here, ten data points are used, and the interval bounds approximately correspond to $a = 100$ and $b = 10^{-8}$. A fit of the Residual Norm Curve and Solution Norm Curve can be seen in Figure 6.8. We see the fit corresponds to the data points and their respective curves. However, the two curves are not as interesting as the L-Curve and especially the location of the regularization parameter, λ^{F}. In Figure 6.9 we see the curve almost fits the L-Curve for the whole area, and it does have somewhat the correct shape. In the figure the calculated λ^{F} and λ^{L} have been shown. Both are located in the corner, almost on top of each other, which can be seen in the zoomed figure. Here we see, though the curve itself does not visually look like the L-Curve, the curve behaves in the same way, with respect to λ. We can therefore conclude:

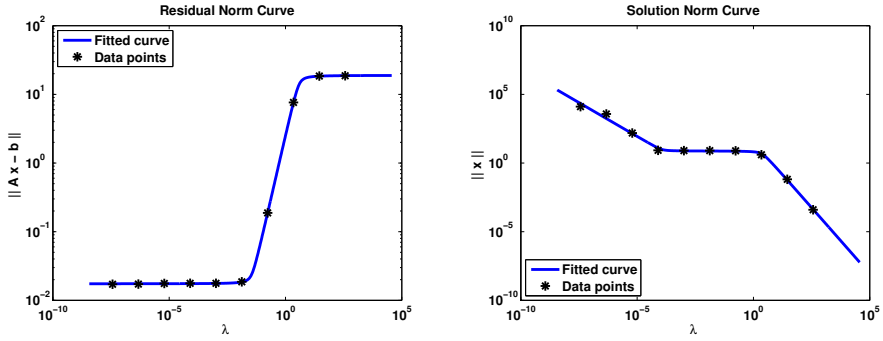

Figure 6.8: *One loop of FRe-PUF. We see that the fitted curve follows the data points. The problem shown is the* `Shaw problem` [5].

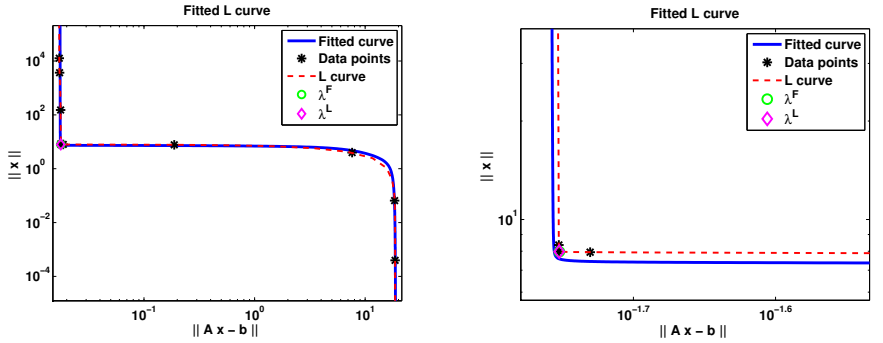

Figure 6.9: *The fitted curve shown together with the data points and the two corner points. The two corner points are close to each other. The problem shown is the* `Shaw problem` [5].

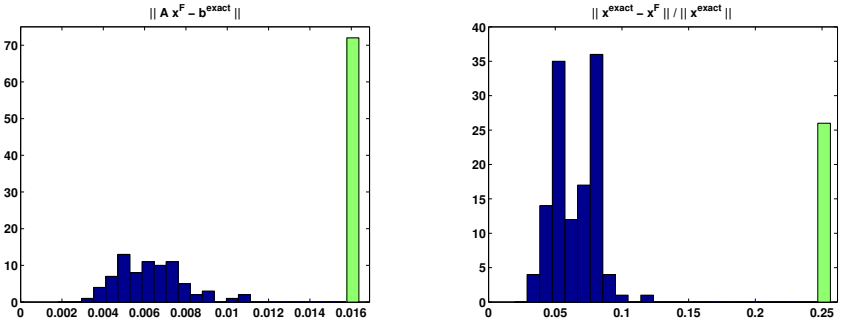

Figure 6.10: *A histogram of the residual error norm and the error in the solution for a problem with 150 different patterns of random noise. The right most bin is a summation of the tail. The problem shown is the* `Shaw problem` [5].

The basic idea of using data fitting on ten data points from the L-Curve for locating a regularization parameter is possible.

To make sure this statement holds, we will examine some more problems. Starting with the problem used for making the L-Curve in Figure 6.9. This problem was created by adding some noise, where ε was created using randomization, such $b = b^{\text{exact}} + \varepsilon$. Due to the randomization, the noise can be suited for the method, and some more testing needs to be done.

We make different generated noise distributions and calculate $\dfrac{\|x^{\text{F}} - x^{\text{exact}}\|_2}{\|x^{\text{exact}}\|_2}$ and $\|Ax^{\text{F}} - b^{\text{exact}}\|_2$. In Figure 6.10, the noise have been generated 150 times. The last bin in the histograms is a summation of when the method has given a huge error.

We see in the histogram that approximately 50 percent of the times, the found solution is good. But we also see that approximately 50 percent of the times there is a tail, which tells us the solution is extremely bad. In the following, different aspects will be discussed in order to eliminate this tail in the error, and in this way it is subject to the optimization of the method.

6.4 Choosing the Parameters

Having found a useful form of the parametric curve, we will look at some initial guesses for the parameters. These will be discussed in the following part.

Firstly, we notice, that the parameters describe the slope of the line segments, the location of the two bends and the sharpness of the bends. If the data points are distributed as suggested in Section 4.2.1, two data points are on the right sloped line segment in the Solution Norm Curve, the slope is found from the data points. That is using the general slope formula for a straight line. A good guess for the slope, and therefore the parameters in p_4^{sol} for the Solution Norm Curve, is:

$$p_4^{sol} = \frac{\log_{10}(\eta[n] - \eta[n-1])}{\log_{10}(\lambda[n] - \lambda[n-1])}$$

and

$$p_1^{sol} = 0.5 p_4^{sol},$$

in logarithmic scale.

From the derivation of the curve, in Section 6 we know that the location of the bend is at σ_1. From [7] we know that $\|A\|_2 = \sigma_1$. An approximation of the greatest singular value is found using the SVD:

$$\begin{aligned} A &= U\Sigma V^{\mathrm{T}} \\ \|A\|_{\mathrm{F}} &= \left(\sum_{i=1}^{m} \sum_{j=1}^{n} a_{i,j}^2 \right)^{\frac{1}{2}} \\ &= \left(\text{ trace } \left(V\Sigma^2 V^{\mathrm{T}} \right) \right)^{\frac{1}{2}} \\ &= \left({\sigma_1}^2 + {\sigma_2}^2 + \ldots + {\sigma_n}^2 \right)^{\frac{1}{2}} \\ &\simeq \|A\|_2 = \sigma_1. \end{aligned} \tag{6.4}$$

Here the last approximation is due to the fact that $\sigma_1 \geq \sigma_2$. Here we see the right bend point can be approximated as:

$${p^{sol}}_2 = {p^{res}}_4 = \|A\|_{\mathrm{F}}.$$

The other parameters cannot be found directly, but we can choose these properly. For instance, we know that the bend points are located relatively close to each other which can be used.

We have that seen the basic idea of using data fitting to make the L-Curve and locate a regularization parameter works. In the following, we will look a bit further at the specific areas of such the FRe-PUF.

First thing to do is to examine the cases where the method does not work. Such a case is the tail in the histogram in Figure 6.10. In Figure 6.11, the

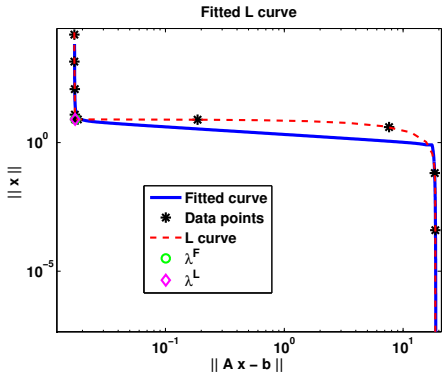

Figure 6.11: *Illustration of a case where the data fitting is not working. The problem shown is the* `Shaw problem` [5].

L-Curve has been shown. We see the problem is that the fit is not good enough. An examination of what is wrong will be done in the following.

6.5 Isolation of Linear Parameters

The first aspect to examine is the possibility of isolating of the linear parameters. By doing this, the minimization method works better and the data fitting is optimized. This is especially helpful, if the minimizer does not use the first order derivative. If this is the case, the minimizer is greatly helped by removing the linear parameters from the minimization.

We know that the data fitting is done by minimizing the distance between the data points and the curve, using the residuals from equation (3.2). Now we will focus on the Solution Norm Curve, since the Residual Norm Curve is done in the same manner. In the case of the Solution Norm Curve the $\mathtt{t}_i = \log_{10} \lambda_i$ and $\mathtt{Y}_i = \log_{10} \|x^{\lambda_i}\|$ for $i = 1, 2, \ldots, N$, with N being the number of data points used for the data fitting. The curve is called $g^{sol}(p^{sol}, \mathtt{t}_i)$, where for the data fitting, the $\mathtt{t}_i$ is constant.

By isolating the linear elements, it will be possible to reduce the number of unknowns and therefore making the minimization more efficient. The minimization for the data fitting of the Solution Norm Curve, is done by solving a

linear system of equations, where we will use G and h as matrices. The Solution Norm Curve is given by:

$$\begin{aligned}\arg\min_{p^{sol}} \sum_{i=1}^{N} \left(g^{sol}(p_1, p_4, p_6, \mathtt{t}_i) - \lambda_i\right)^2 \\ = \sum_{i=1}^{N} \Bigg(p_1 \left[\mathtt{t}_i - p_2 + \sqrt{(\mathtt{t}_i - p_2)^2 + p_3^2}\right] - \\ p_4 \left[\mathtt{t}_i - p_5 - \sqrt{(\mathtt{t}_i - p_5)^2 + p_3^2}\right] + p_6 - \lambda_i \Bigg)^2 . \qquad (6.5)\end{aligned}$$

Differentiating this with respect to the linear parameters p_1, p_4 and p_6 gives:

$$\begin{aligned}\frac{\delta g^{sol}}{\delta p_1} &= \sum_{i=1}^{N} 2 \frac{\delta g^{sol}(p^{sol}, \mathtt{t}_i)}{\delta p_1} \left(g^{sol}(p^{sol}, \mathtt{t}_i) - \mathtt{Y}_i\right) \\ &= \sum_{i=1}^{N} \Bigg(2 \left((\mathtt{t}_i - p_2) + \sqrt{(\mathtt{t}_i - p_2)^2 - p_3^2} \right) \cdot \\ &\quad \Bigg[p_1 \left(\mathtt{t}_i - p_2 + \sqrt{(\mathtt{t}_i - p_2)^2 + p_3^2} \right) - \\ &\quad p_4 \left(\mathtt{t}_i - p_5 - \sqrt{(\mathtt{t}_i - p_5)^2 + p_3^2} \right) + p_6 - \mathtt{Y}_i \Bigg] \Bigg) \\ \frac{\delta g^{sol}}{\delta p_4} &= \sum_{i=1}^{N} 2 \frac{\delta g^{sol}(p^{sol}, \mathtt{t}_i)}{\delta p_4} \left(g^{sol}(p^{sol}, \mathtt{t}_i) - \mathtt{Y}_i\right) \\ &= \sum_{i=1}^{N} \Bigg(2 \left((\mathtt{t}_i - p_5) - \sqrt{(\mathtt{t}_i - p_5)^2 - p_3^2} \right) \cdot \\ &\quad \Bigg[p_1 \left(\mathtt{t}_i - p_2 + \sqrt{(\mathtt{t}_i - p_2)^2 + p_3^2} \right) - \\ &\quad p_4 \left(\mathtt{t}_i - p_5 - \sqrt{(\mathtt{t}_i - p_5)^2 + p_3^2} \right) + p_6 - \mathtt{Y}_i \Bigg] \Bigg) \\ \frac{\delta g^{sol}}{\delta p_6} &= \sum_{i=1}^{N} 2 \frac{\delta g^{sol}(p^{sol}, \mathtt{t}_i)}{\delta p_6} \left(g^{sol}(p^{sol}, \mathtt{t}_i) - \mathtt{Y}_i\right) \\ &= \sum_{i=1}^{N} \Bigg(2 \cdot 1 \cdot \Bigg[p_1 \left(\mathtt{t}_i - p_2 + \sqrt{(\mathtt{t}_i - p_2)^2 + p_3^2} \right) - \\ &\quad p_4 \left(\mathtt{t}_i - p_5 - \sqrt{(\mathtt{t}_i - p_5)^2 + p_3^2} \right) + p_6 - \mathtt{Y}_i \Bigg] \Bigg) .\end{aligned}$$

These equations are linear in p_1, p_4 and p_6 and can be rewritten as a system of linear equations which are easy to solve.

The matrix G and the vector h has the following elements:

$$\begin{aligned}
G_{1,1} &= \sum_{i=1}^{N} \left(\mathtt{t}_i - p_2 + \sqrt{(\mathtt{t}_i - p_2)^2 + p_3^2} \right)^2 \\
G_{1,2} &= \sum_{i=1}^{N} \left(\mathtt{t}_i - p_5 + \sqrt{(\mathtt{t}_i - p_5)^2 + p_3^2} \right) \left(\mathtt{t}_i - p_2 + \sqrt{(\mathtt{t}_i - p_2)^2 + p_3^2} \right) \\
G_{1,3} &= \sum_{i=1}^{N} \left(\mathtt{t}_i - p_2 + \sqrt{(\mathtt{t}_i - p_2)^2 + p_3^2} \right) \\
G_{2,1} &= G_{1,2} \\
G_{2,2} &= \sum_{i=1}^{N} \left(\mathtt{t}_i - p_5 + \sqrt{(\mathtt{t}_i - p_5)^2 + p_3^2} \right)^2 \\
G_{2,3} &= \sum_{i=1}^{N} \left(\mathtt{t}_i - p_5 + \sqrt{(\mathtt{t}_i - p_5)^2 + p_3^2} \right) \\
G_{3,1} &= G_{1,3} \\
G_{3,2} &= G_{2,3} \\
G_{3,3} &= \sum_{i=1}^{N} 1^2
\end{aligned}$$

and

$$\begin{aligned}
h_1 &= \sum_{i=1}^{N} \mathtt{Y}_i \left(\mathtt{t}_i - p_2 + \sqrt{(\mathtt{t}_i - p_2)^2 + p_3^2} \right) \\
h_2 &= \sum_{i=1}^{N} \mathtt{Y}_i \left(\mathtt{t}_i - p_5 + \sqrt{(\mathtt{t}_i - p_5)^2 + p_3^2} \right) \\
h_3 &= \sum_{i=1}^{N} \mathtt{Y}_i,
\end{aligned}$$

and will give the parameters, p_1, p_4, p_6 by solving

$$G \begin{bmatrix} p_1 \\ p_4 \\ p_6 \end{bmatrix} = h.$$

Hereby the Solution Norm Curve is reduced to be only dependent on three parameters.

Now we will look at what changes this gives for finding solutions.

6.5.1 Results without Linear Parameters

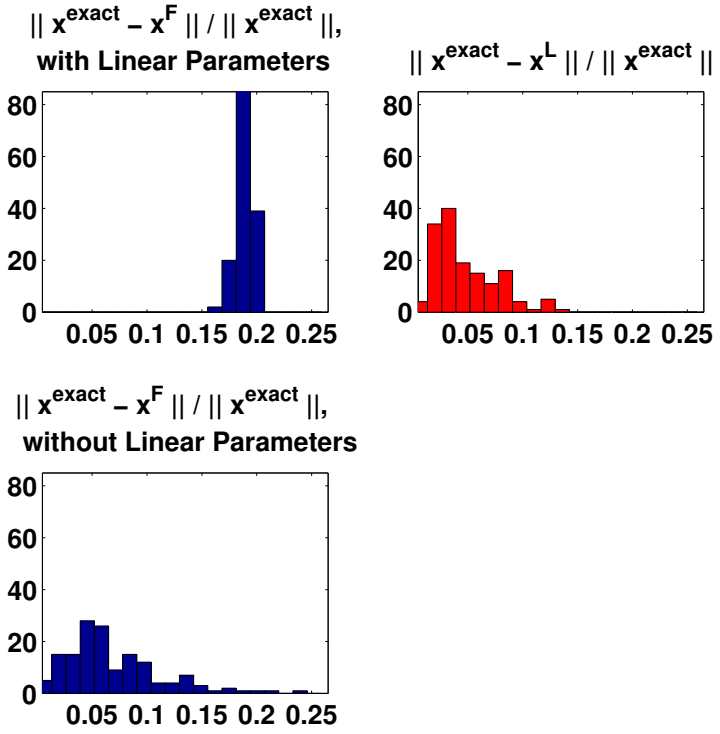

Figure 6.12: *Comparison of the data fitting with and without the linear parameters. A histogram of the error in the solution for a problem with 150 different patterns of random noise. The right most bin is a summation of all bin further to the right. The problem shown is the* `Foxgood problem` [5].

After having isolated the parameters, we examine the effects. Since the method is made to work for large scale problems, the time used for data fitting will be short compared to the time needed for computing a data point. It is therefore of no interest to look at the evaluation time of the data fitting, nor the number of iterations. The correctness of the new results is on the other hand of more interest. We make the same type of histograms as seen in Figure 6.10 to

compare the results. Again, the right most bin is a summation of all huge errors. The new histogram can be seen in Figure 6.12. These histograms show that we acquire a more robust method after the linear parameters are isolated. The histogram is greatly improved compared with the one from L-Curve Criterion and therefore the isolation improves the robustness of the method.

6.6 Making a Contour Plot

After having isolated the linear parameters, the curve for the data fitting $M_p(\mathtt{t})$ depends on only three parameters. This gives the idea of keeping one parameter constant, and thereby letting the minimization problem depend on only two parameters.

When this is the case, we are able to make a contour plot of the minimization problem, and thereby analyze the problem better. Moreover, we will be able to understand the effects on the fit, when changing some of the parameters.

Looking at the curve without the linear parameters, we see the parameters are, p_2, p_3 and p_5. Here p_3 determines the sharpness of the bends in the curve. This gives reason to believe holding p_3 constant reasonable value will not have a large influence on the solution to the general minimization problem. On the other hand, changing one of the other two parameters can change the result more, since these determine the slopes for the curve. In order to do the analysis for different problems, each with different patterns of noise, we find an appropriate value of p_3.

Keeping p_3 constant, we make a contour plot for each of the two curves, see Figure 6.13 and 6.14. We see both minimization problems having a long valley which makes the data fitting difficult. We know from the minimization theory[8] that such a valley gives problems. This problem will be slow convergence, and the number of iterations exploding, if the initial guess is bad and the minimizer relies only on the steepest descent method. Moreover, if the stopping criteria is too conservative, the found solution can be far from the sougth solution. This can easily happen when the valley is too flat.

To avoid the first problem, we use a minimization method, which is not completely dependent on the descent direction. The second problem can cause a bit more troubles, since there are many solutions, which will be approximately equally good. In order to find the minimum, two stopping criteria must be set properly. First, the number of iterations should be relatively large, such that this does not stop the minimization prematurely. Secondly, the step improve-

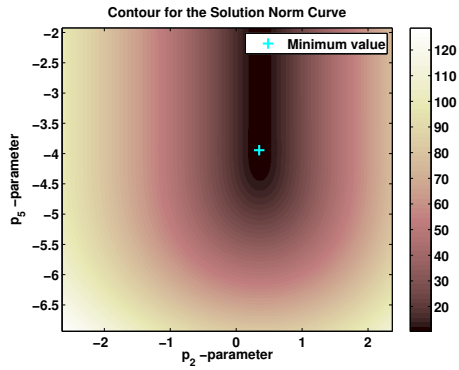

Figure 6.13: *The contour plot for the Solution Norm Curve as a function of p_2 and p_5 with a fixed p_3. The minimum of the function is shown as $+$. The problem shown is the* `Shaw problem` [5].

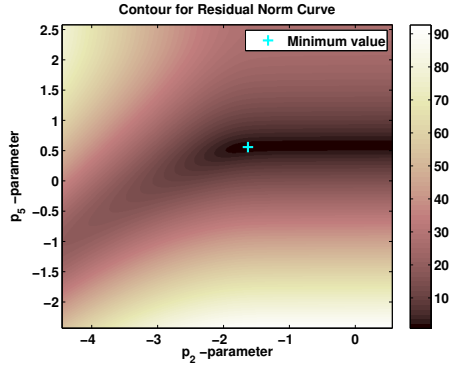

Figure 6.14: *The contour plot for the Residual Norm Curve as a function of p_2 and p_5 with a fixed p_3. The minimum of the function is shown as $+$. The problem shown is the* `Shaw problem` [5].

ment should be small, since a step inside the valley will only cause a small improvement in the function value.

6.7 Looking for Trouble

Now we will look at those situations, where the histograms have shown that the method is not as reliable as the L-Curve Criterion. A situation where the error is large can be seen in Figure 6.15. The initial interval is too large and the leftmost data points represent an area, which has not been considered when deriving the parametric curve used for the data fitting.

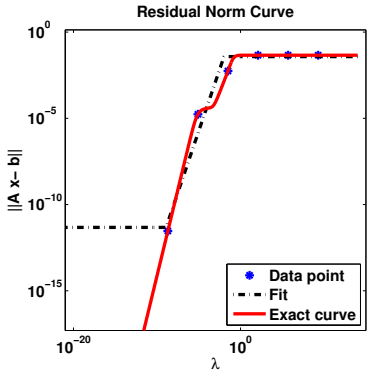

Figure 6.15: *An example of data points distributed poorly. The problem shown is the* `Blur problem` [5].

The biggest problem with the data fitting is that the parametric curve actually does not follow the data points. As we have seen in Figure 6.3 and 6.4, the two curves used for the data fitting are simple curves compared to the behavior of the data points. The simple parametric curve was chosen such the data fitting requires very few data points. This simplification helps the data fitting, but it also requires a more precise initial interval for the data points.

If the data points have been chosen poorly, the data fitting will not be correct and the corner point might be wrong. Such a situation can be seen in Figure 6.15. But it is not only a matter of keeping the interval tight around the two bends in the Residual Norm Curve. The Solution Norm Curve can cause trouble for the data fitting as well, since it has a stepped part on the left side, and not a straight line. This can be seen in Figure 6.16.

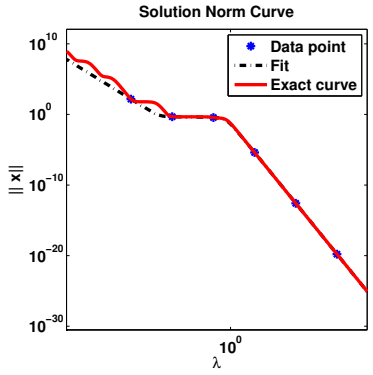

Figure 6.16: *An example of the stepped area on the Solution Norm Curve. The problem shown the* `Wing problem` [5].

	With badly distributed points	With good distributed points	For the x^{err}
$\frac{\|x^{\lambda} - x^{\mathrm{exact}}\|_2}{\|x^{\mathrm{exact}}\|_2}$	0.608	0.524	0.491

Table 6.1: *The calculated normed error for solutions with two different distributions of data points. The problem shown is the* `Wing problem` [5].

The steps causes troubles for the data fitting, since the curve to be fitted does not include these steps, and only has a straight line. Therefore the best result would be achieved with more than one, but preferably as many points as possible, in this area.

The data points should be placed such that at least two points are at the left part of the Solution Norm Curve, but they are still on the flat part of the Residual Norm Curve.

A good choice of data points for the data fitting can be seen in Figure 6.17, where the problem as before has been shown with a better distribution of data points. As a comparison $\frac{\|x^{\mathrm{F}} - x^{\mathrm{exact}}\|_2}{\|x^{\mathrm{exact}}\|_2}$ is calculated for the two distributions of data points and compared with $\frac{\|x^{\mathrm{err}} - x^{\mathrm{exact}}\|_2}{\|x^{\mathrm{exact}}\|_2}$. These results can be seen in Table 6.1.

It is worth mentioning that there is reason to believe, that a better solution

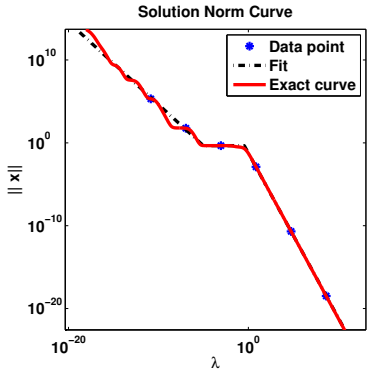

Figure 6.17: *An example of a better distribution of the data points for the stepped area on the Solution Norm Curve. The problem shown is the* `Wing problem` [5].

can be obtained if the parametric curve is more complex. If it takes the left sloped part of the the Residual Norm Curve into account, the data fitting will be more robust, but might be less precise.

6.8 Removing a Parameter

When looking at the fitted curves throughout the testing, we notice that the two sloped parts have a correlation for the Solution Norm Curve. As we saw in the derivation in Section 5.1 this correlation is:

$$p_1 \simeq 2p_4.$$

If we use this as a strict constraint for the data fitting, we can reduce the number of parameters. As with the isolation of the linear parameters, this makes the data fitting faster, which, however, is not the direct objective of the optimization. Instead we add this enhancement in order to make the data fitting more robust.

The reason for the solution being more robust is due to the fact that more residuals, R_i, depend on the new parameter. Therefore a change in the new parameter for both slopes after applying the enhancement will have greater impact on the minimization problem. The change will be better for the overall $\|R\|_2$.

A histogram of the Solution Norm Curve with the same interval as used in Figure 6.12 can be seen in Figure 6.18. Here it is visually easy to see that the

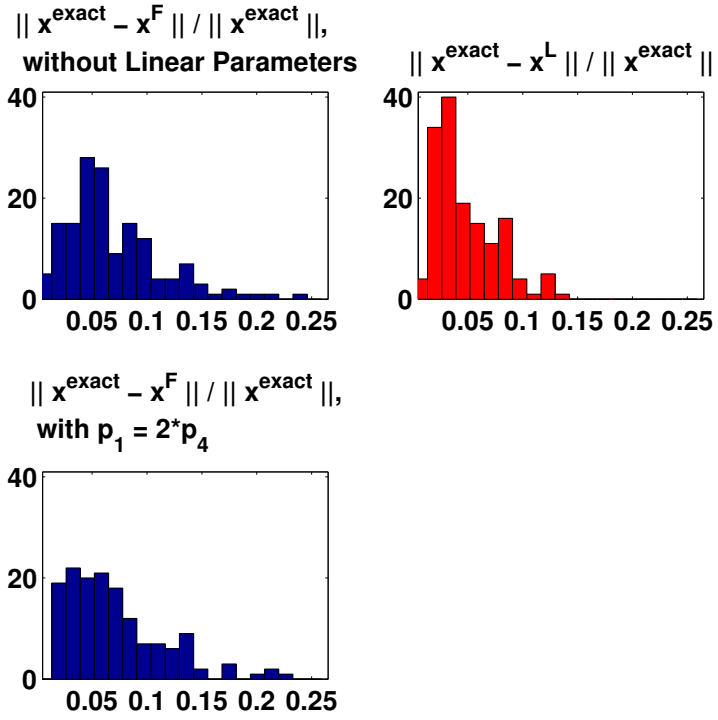

Figure 6.18: *Comparison of errors when applying the $p_1 \simeq 2p_4$ enhancement. Problem Shown is* `Foxgood problem` [5].

data fitting has improved for some of the noise realizations, whereas others have not.

However, although most problems show this improvement, they do not all show improvements. In Figure 6.19, one exception is shown. The reason for this might be that with fewer parameters the minimizer cannot go in as many directions, and in that way it can end up at a wrong minimum.

In Figure 6.20 a situation has been shown for a noise pattern where the error is big. Here we see that the two sloped parts are wrong, making the L-Curve bad and thereby the found regualarization parameter, λ, is also bad.

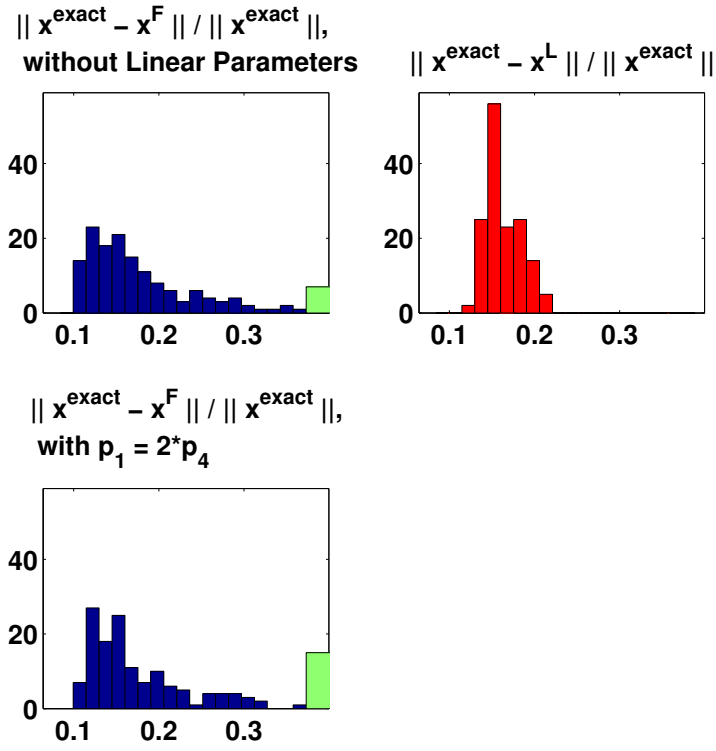

Figure 6.19: *Comparison of errors when applying the $p_1 \simeq 2p_4$ enhancement. A histogram of the error in the solution for a problem with 150 different patterns of random noise. The right most bin is a summation of the error further to the right. The problem shown is the* `Baart problem` [5].

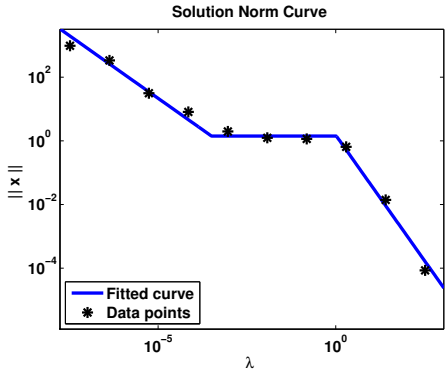

Figure 6.20: *Illustration of the problem when the enhancement of the* $p_1 \simeq 2p_4$ *does not improve the method. Problem shown is* `Baart problem` [5].

Therefore, this enhancement should be optional and depending on the initial knowledge of the problem whether it is used.

6.9 Finding a New Point

As for the algorithm, the way of adding a new point can be crucial for the convergence. Therefore different ways to place the next point will be examined. We start by noticing that both the Residual Norm Curve and the Solution Norm Curve can be divided into three line segments. For the Solution Norm Curve, a good estimate for the right bend point is known, but the left bend point is more difficult to get a value for. The next point should be chosen such that it improves the fitting, especially the parameter for the left bend.

We analyze different strategies of adding a new point.

6.9.1 Using 'Near a Point-of-Interest'-Criterion

The first strategy is to add the new point near the expected interesting part of the interval. In that case the new point is chosen to be near either of the bends or at the point corresponding to the corner of the L-Curve. The last two points will be interesting for making a better fit from badly distributed points. Whereas the last point will be interesting if the fit is reasonably good.

An illustration, in Figure 6.21, shows the data points(the blue points) and the points of interest are at the two bends (triangles). Therefore the new points should preferably be placed at one of these points.

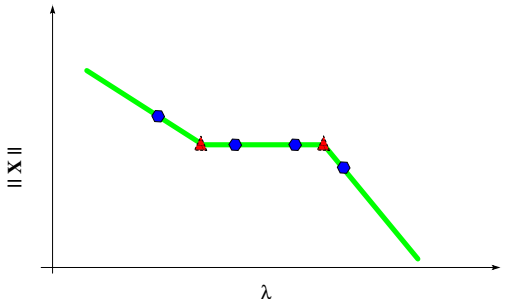

Figure 6.21: *Illustration of adding a new point where there are points of interest. The blue points are data points, and the red points are possible new points.*

This strategy is not always good, since the interval for the starting points does not necessarily cover the shape of the Solution Norm Curve and the Residual Norm Curve. If this is not the case, the information needed for a proper data fitting will not be available, and the fit will not be good enough. Another problem with a strategy like this one, arises when the fit is reasonably good, then the new point will be close to the previously found points and as such not give good convergence.

6.9.2 Using 'Equally Placed'-Criterion

A straight forward way to avoid the last problem is to put the new point in between two other points. That way two points do not end up being relative close to each other. However, doing this has little chance of returning a point which actually gives a better fit, since the point does not cover the interesting area of the curves. When placing the new point it should be done at a place where the new point adds some more information about the curve. That is a place, which either has not many points yet or near an interesting part in the curve.

Such a strategy could be to divide the curve into three parts for each line segment, and ensure that there are data points in each of the three line segments. If a line segment is not represented by data points, the next point should be

placed there. But if all three line segments contain data points, the new point should be near a point of interest. An illustration, see Figure 6.22, shows the data points(the blue points), and the points of interest are the two bends. But this time, there are no data points at the left part, and therefore the new point should be placed either here or at one of the bends (the red triangle).

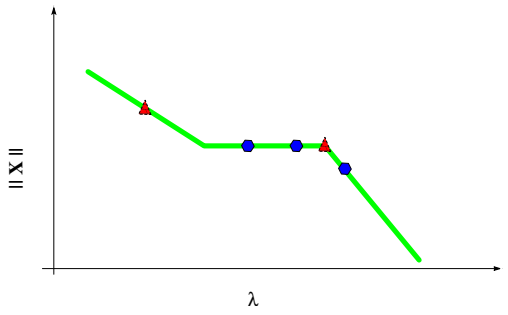

Figure 6.22: *Illustration of adding a new point where there are no other points. The blue points are data points, and the red points are possible new points.*

Problems arise for this strategy. Firstly, the three parts have to be located. This can be done fairly easy, since the first fit gives a rough estimate of the two bends, and therefore the borders between the three line segments.

Secondly we need to consider, where to put the point, after having found the line segment in the curve. As mentioned it should not be too close to an existing point. To avoid this an outer borders must be introduced, such that all three parts become closed intervals.

An immediate idea for finding such an interval is to make the parts with same width. Since the middle line segment is known, from the two bend points, this can easily be computed. If the data points are not spread across the three parts properly, the fit will not be good compared to the original shape. However, when following this strategy the next point will make the data points spread more. Therefore the fudge parameter c, mentioned in FRe-PUF is utilized here. It should make sure that the width is corrected if the new point does not give convergence.

Chapter 7

Large-Scale Problems

We have seen some of the examinations which have lead to the FRe-PUF algorithm and also seen it work. So far only small problems have been used, but the method is developed for large-scale problems. Therefore the FRe-PUF will be applied on large-scale problems in this chapter.

7.1 The Simple Blurring Problem

At first we test the method on the `Blur problem` [5]. It can be seen in Figure 7.1. The problem is made by reshaping the image into a column stacked vector, x, and adding a Gaussian point spread function giving the data b. It is made with the knowledge of the x^{exact} and the SVD.

We use FRe-PUF with the next point chosen by the strategy from Section 6.9.2. The fudge parameter removes new points, if they do not contribute to a better solution. This give the result seen in Figure 7.2, which can be compared with Figure 7.3. Here we see that the x^{F} is closer than x^{L} to x^{err}. This is due to the fitted L-Curve having a particularly nice shape as seen in Figure 7.4, where we see that the corner is very flat. This leads, by coincidence, to good performance by the FRe-PUF- algorithm.

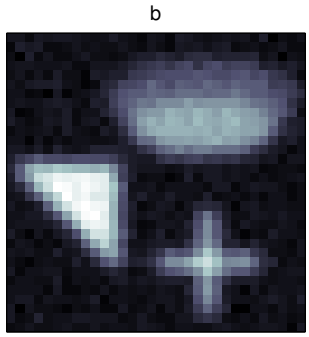

Figure 7.1: *The data b, from the* `Blur problem` [5].

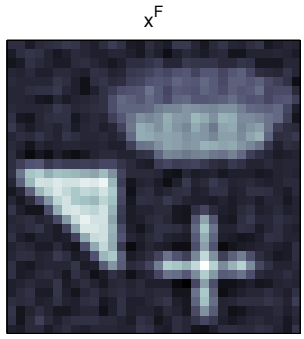

Figure 7.2: *The solution found by FRe-PUF to the* `Blur problem` [5]. x^{F} *is found after four iterations.*

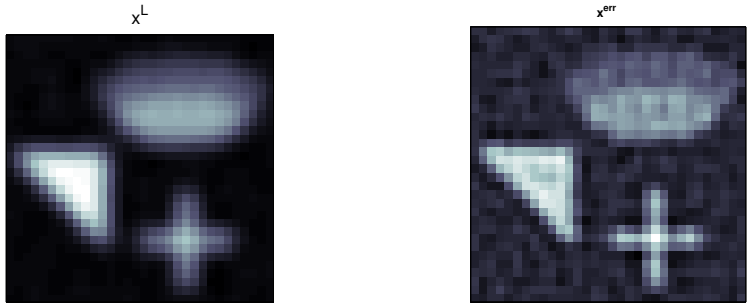

Figure 7.3: *Solution to the* `Blur problem` [5]. *The solution from the L-Curve Criterion to the left and to the right, the solution corresponding to using* λ^{err}.

For this solution no stopping criterion have been used. Instead the FRe-PUF added data points until manually stopped when the solution was visually good. The figure of the data fitting for the L-Curve for each iteration can be seen in Figure 7.4. We see that the curve is improved for each new data point added. The λ^{F} is also getting closer to λ^{L} for each iteration.

To see the improvements given for each newly added data point the $\frac{\|x^{\mathrm{F}} - x^{\mathrm{exact}}\|_2}{\|x^{\mathrm{exact}}\|_2}$ has been shown in Figure 7.5. For comparison reasons the $\frac{\|x^{\mathrm{L}} - x^{\mathrm{exact}}\|_2}{\|x^{\mathrm{exact}}\|_2}$ and $\frac{\|x^{\mathrm{err}} - x^{\mathrm{exact}}\|_2}{\|x^{\mathrm{exact}}\|_2}$ has been shown as well.

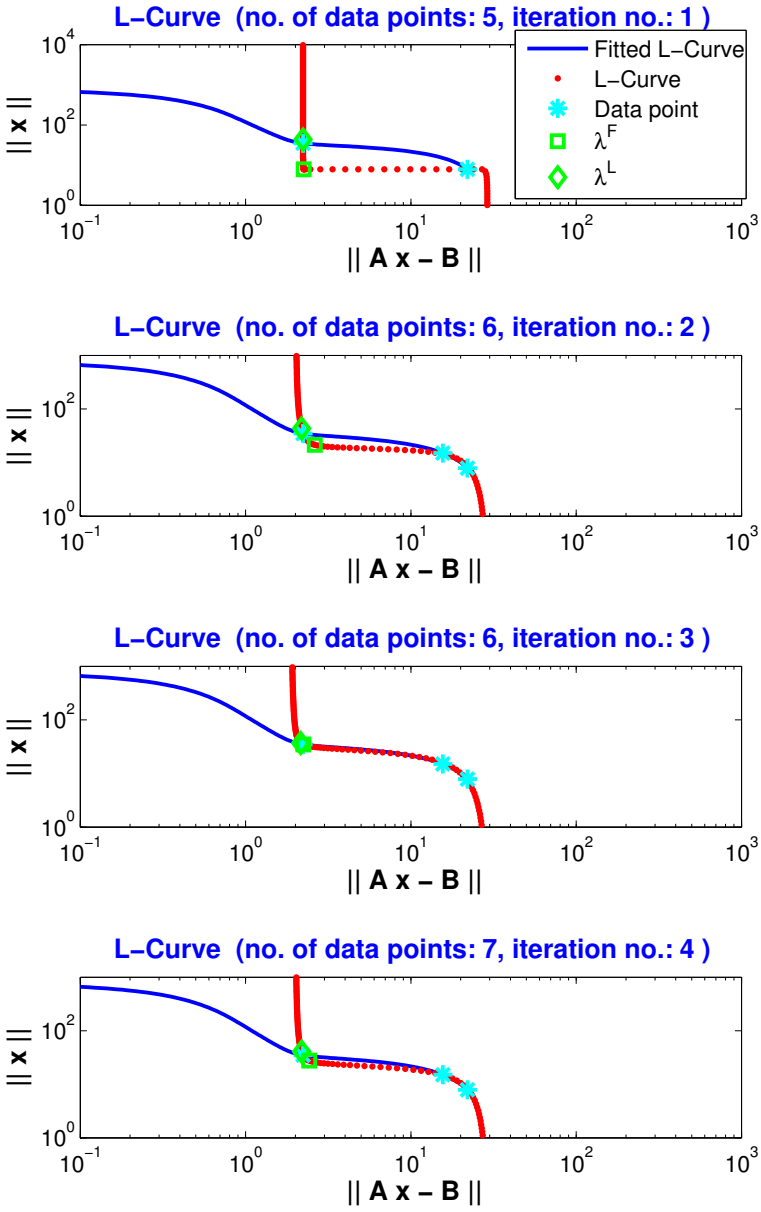

Figure 7.4: *The L-Curve, the data points, different regularization parameters and the fitted L-Curve for the* `Blur problem` [5] *for each iteration.*

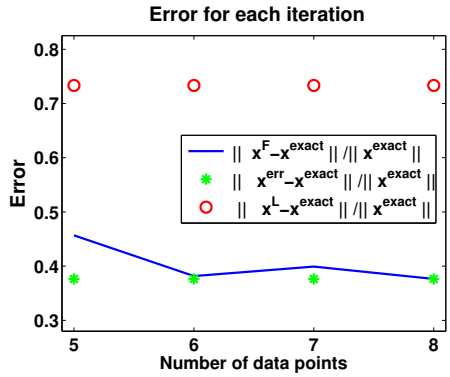

Figure 7.5: *The normalized error as a function of the number of iterations when solving the* `Blur problem` [5]. *The circle is the normed error from* x^{exact} *found with* x^{L} *and the star is the normed error from* x^{exact} *found with* x^{err}.

7.2 The 'Math Problem'

Figure 7.6: *The Math Problem. A blurred and noisy picture.*

To test the method further, a problem where the SVD cannot be computed is created. The problem is shown in Figure 7.6. The problem is created as a grayscale picture and using convolution to add some Gaussian blurring. The Wiener filter from the Image Processing Toolbox in Matlab, has been applied to compute the data points. Eight data points were computed, before the FRe-PUF was applied to get a regularization parameter.

The regularization parameter is found using the FRe-PUF, with just one iteration. The two fitted curves can be seen in Figure 7.7, where we see that the Solution Norm Curve does not follow the assumption of having two slopes linesegments and therefore can give problems. For this particular problem, in does not give troubles and a reconstruction can be made. In Figure 7.8 the solution from this regularization parameter can be seen. Here we see the solution has been deblurred.

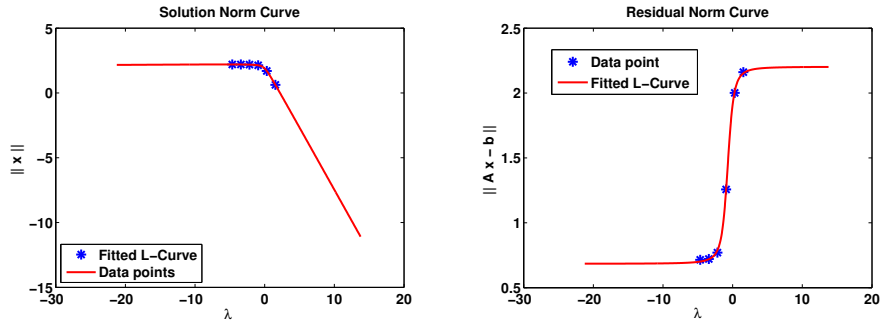

Figure 7.7: *The two fitted curves for the Math problem.*

Figure 7.8: *The solution for the Math Problem, found by FRe-PUF.*

Chapter 8

Conclusion

Throughout this thesis, we have presented how a specific type of problems, the large-scale linear inverse problems, can be solved using Tikhonov Regularization. The created method is targeted for large-scale problems, for which the regularization parameter cannot be found using a calculated analytical L-Curve. Data fitting has therefore been applied instead.

The method is to compute points on the well-known L-Curve and using data fitting to estimate an L-Curve. It is hereby possible to imitate the L-Curve Criterion. Through the testing on small problems with known solutions, the method has shown its justification as a method. Even on large-scale problems, the method has proved useful with reasonable results.

For this thesis, the focus is to test the idea. Consequently we have looked for robustness rather than speed for the different elements of the FRe-PUF.

Many things have been applied in order to improve the robustness of the method. Among the most significant results is the dependency on the initial interval. We have seen that the method is indeed dependent on having a good distribution of the data points. This dependency is mostly due to the simplification of the curve used for the data fitting. The curve has been seen to be very simple, which, for a good choice of initial interval, gives good solutions. However, for a less optimal initial interval, the curve is too simple, and leads to

a bad estimate of the L-Curve.

To further improve the method, we have tried to optimize the data fitting. An isolation of the linear parameters and further simplification of the parametric curve has been examined. We saw both improvements to be good enhancements for most cases, but examples can be made where they actually worsen the method. Especially the simplification of the curve should be applied with care, since the enhancement can worsen the method.

A way to improve the problem of the initial interval has been tried to overcome by computing new data points iteratively. When these new points are computed properly the method will be greatly improved. Different strategies have been tried for computing these new data points, but none of the strategies tried has proven reliable for all situations.

We have seen that the strategy of adding points works and gives good convergence for one large-scale problem.

8.1 Future Work

Most of the work in this thesis have been experimental and done by examination of the ideas. Though many things have been examined still many ideas have been left out. These ideas will be described here.

8.1.1 Constraints on the Minimization

We have seen the minimization is a problem with a long valley. Some reasonable initial parameters can be approximated, therefore an idea could be to bound these parameters, such that the minimizer cannot change the values much. Some of the parameters are known, for other parameters we know how they lie with respect to each other. Therefore the constraints can be, to make sure that the left bend point is to the left of the approximated right bend and similar to others.

In this thesis this knowledge has only been used as initial guess for the parameters, but making constraints on the variables can give a more robust data fitting.

In the same manner, the enhancement of the correlation between the slopes,

can be made as bounds. We have implemented the similarity as a strict equation, whereas implementing this knowledge as a constraint can give a better performance.

However none of these boundaries are likely to help preventing all the problems with a too simple curve.

8.1.2 Using Different Curves for the Data Fitting

The data fitting have been done using a simple parametric curve. Instead of this curve a more complex curve could be used. This could be a spline, of another parametric curve which takes the full shape of the Residual Norm Curve and Solution Norm Curve into consideration. We have seen how a specific problem can be hard to solve with the simple parametric curve, therefore a more complex one might be better.

However keeping the data fitting problem simple has shown to be good for the robustness, which makes it necessary to examine the improvements which a more complex curve can give. If it does not improve the method in general, another idea is to start by using the simple parametric curve for the first iteration. The next iteration can then be made with a more complex curve.

8.1.3 Number of Data Points

One thing which has only briefly been tried is to examine the number of points needed for the data fitting and also their distribution. We have made a quick examination of where new data points should be placed when adding them to the FRe-PUF.

This is a place where there can be made some progress and the robustness can be improved. By using statistics and the fitted curve, it should be easier to add a good data point. In this project we have seen that adding data points is not trivial, and therefore some more ideas and evaluation will be able to give convergence more often.

Yet another idea is to add the next data point interactively. The user will be able to see the progress and, maybe, be able to give better convergence than an automatic approach. At least the user will be better to detect if divergence has occured.

Bibliography

[1] C. E. Bruun and T. B. Nielsen, **Algorithms and Software for Large-Scale Geophysical Reconstructions;** IMM-M.Sc-2007-29, 2007.

[2] P. C. Hansen, **Computation of the Singular Value Expansion;** Computing 40, 1988, pp. 185-199.

[3] P. C. Hansen, **Deconvolution and Regularization with Toeplitz Matrices;** Numerical Algorithms 29, 2002, pp. 323-378.

[4] P. C. Hansen, **Discrete Inverse Problems - Insight and Algorithm;** Course note for DTU course: *02906 Discrete Ill-Posed Problems,* 2005.

[5] P. C. Hansen, **Regularization Tools: A Matlab Package for Analysis and Solution of Discrete Ill-Posed Problems,** Numerical Algorithms 6, 1994.

[6] P. C. Hansen, **The L-Curve and its Use in the Numerical Treatment of Inverse Problems;** Invited chapter in P. Johnston (Ed.), *Computational Inverse Problems in Electrocardiology,* WIT Press, Southampton, 2001, pp. 119-142.

[7] P. C. Hansen, **The Truncated SVD as a Method for Regularization;** BIT 27, 1987, pp. 534-553.

[8] J. B. Jørgensen, K. Madsen, H. B. Nielsen and M. Rojas, **Introduction to Optimization and Data Fitting,** Course note for DTU course: *02611 Optimization and Data Fitting,* 2006, pp. 222.

[9] M. Pedersen, **Functional Analysis in Applied Mathematics and Engeneering;** CRC Press, 2000.

[10] S. Leach, **Singular Value Decomposition - A Primer;** Unpublished Manuscript, Brown University,
url = "http://www.citeseer.ist.psu.edu/651803.html", 2000