Discrete Inverse Problems

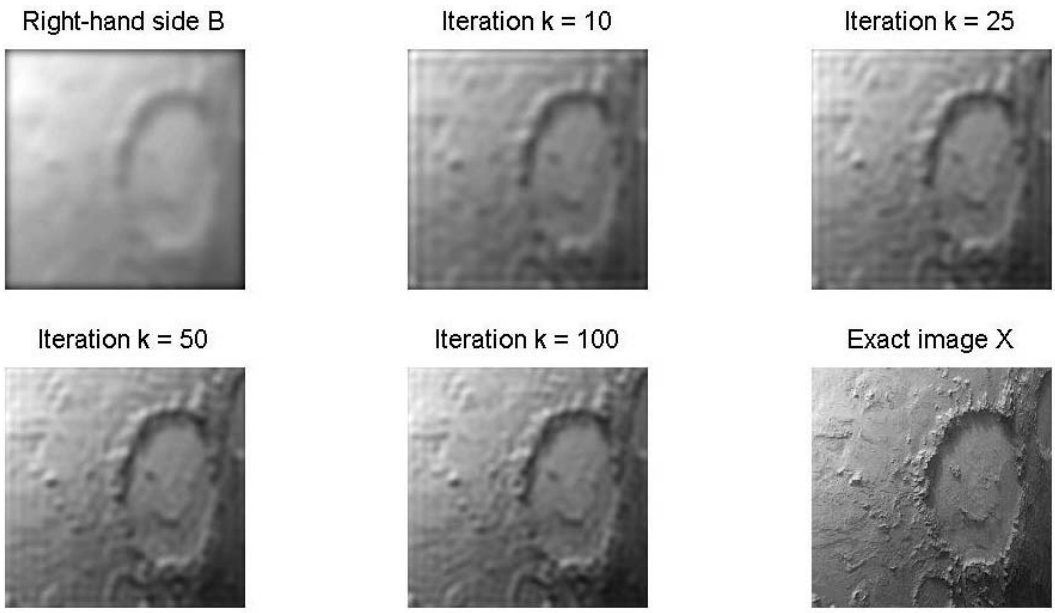
Insight and Algorithms

A tutorial with Matlab exercises

Per Christian Hansen

Informatics and Mathematical Modelling
Building 321, Technical University of Denmark
DK-2800 Lyngby, Denmark

November 21, 2005



Contents

Abstract

There are many books devoted to the subject of inverse and ill-posed problems – so why yet another book? My experience from teaching this subject is that there is a need for a text book which covers the basic subjects and which focuses on the computational aspects. Moreover, I believe that practical computational experience is important, and therefore such a tutorial should include a number of exercises to give the reader hands-on experience with the difficulties and challenges associated with the treatment of inverse problems.

The present book is therefore intended as a quite gentle introduction to a field characterized by advanced mathematics and sophisticated numerical methods. The book does not pretend to tell the whole story, to give all the details, or to survey all the important methods and techniques. The aim is to provide the reader with enough background in mathematics to understand the basic difficulties associated with linear inverse problems, how they affect the behavior of these problems in the face of measurement and rounding errors, and how we can design practical algorithms for computing regularized/stabilized solutions to these problem. Provided with this insight, the reader will be able to start reading the more advanced literature on the subject – indeed, anyone who wants to work in the area of linear inverse problems is advised to also consult some of the many well-written books on the subject, such as [1], [2], [3], [11], [13], [14], [15].

The focus of the book is on linear inverse problems in the form of Fredholm integral equations of the first kind. The presentation starts with a summary of the most important properties of linear inverse problems in the continuous setting. Then we briefly discuss discretization methods, and describe how many of the properties of the integral equation carry directly over to the discretized system – in the form of a linear (perhaps overdetermined) system of equations. The next chapter is devoted to simple regularization methods for computing regularized solutions in the form of filtered spectral expansions; this is an important class of methods which clearly illustrates the basic ideas of regularization. Since no regularization algorithm is complete without a method for choosing the regularization parameter, we also include a discussion of some state-of-the-art parameter choice methods. We conclude with a chapter on iterative methods for large-scale problems, and a chapter with a few real-world problems.

At the end of each section we give a number of exercises, most of them involving numerical experiments with the Matlab package Regularization Tools [6], and which further illustrate the concepts and methods discussed in the corresponding section. The package is available from Netlib at `http://www.netlib.org/numeralgo/na4`.

Acknowledgements. This tutorial grew out of a series of lectures on linear inverse problems given at the Fifth Winter School in Computational Mathematics in Geilo, Norway, February 2005. The atmosphere at the winter school was very positive, and I enjoyed the chance to teach one of my favorite subjects to a dedicated audience.

List of Symbols

A	coefficient matrix
A_k	TSVD matrix
$A_k^\dagger$	pseudoinverse of A_k
b	right-hand side
B_k	lower bidiagonal matrix
c	curvature of L-curve
e	noise component in right-hand side
f	solution function (integral equation)
f_i	filter factor (TSVD or Tikhonov)
F	filter matrix = diag(f_i)
g	right-hand side function (integral equation)
$G(\cdot)$	GCV function
h	grid spacing
I_p	identity matrix of order p
k	truncation parameter (TSVD)
K	kernel function (integral equation)
$\mathcal{K}_k$	Krylov subspace
ℓ_i	eigenvalue of kernel
L	regularization matrix
L_1, L_2	discrete 1. and 2. order derivatives
m, n	matrix dimensions: $A \in \mathbb{R}^{m\times n}$
s, t	independent variables (integral equation)
u_i, v_i	left and right singular functions or vectors
U, V	left and right singular matrices
w_k	basis vectors of projection method also CGLS and Lanczos vectors
W_k	matrix of basis vectors
x	"naive" solution
x_k, x_λ	TSVD and Tikhonov solutions
$x^{[k]}$	iteration vector (Landweber, ART)
$x^{(k)}$	solution computed via projection also CGLS solution
$y^{(k)}$	solution to projected problem

Continued on next page ...

α	relative decay of SVD coefficients
α_k, β_k	from bidiagonalization algorithm
$\bar{\alpha}_k$, $\bar{\beta}_k$	from CGLS algorithm
γ_k	constant i TSVD perturbation bound
$\Gamma(\cdot)$	Gamma function
δ	upper bound on solution norm
ΔA, Δb	matrix and right-hand side perturbations
ϵ	upper bound on residual norm
ϵ_k, ϵ_λ	TSVD and Tikhonov regularization errors
ζ_j	expansion coefficients
η	standard deviation for noise
κ_k, κ_λ	TSVD and Tikhonov condition numbers
λ	regularization parameter (Tikhonov)
μ_i	singular value of kernel
ξ, $\hat{\xi}$	solution norm squared, log of ditto
ρ, $\hat{\rho}$	residual norm squared, log of ditto
σ_i	singular value of matrix
Σ	diagonal matrix with singular values
τ, τ_k, τ_λ	constants in perturbation bounds
ϕ_i, ψ_i	basis functions
χ_i	"top hat" ⎍ function
ω_j	quadrature weights

$(\,\cdot\,,\,\cdot\,)$	inner product
$\|\cdot\|_2$, $\|\cdot\|_{\mathrm{F}}$	2-norm and Frobenius norm
$\mathrm{cond}(\,\cdot\,)$	condition number
$\mathrm{Cov}(\,\cdot\,)$	covariance matrix
$\mathcal{E}(\,\cdot\,)$	expected value
$\mathrm{span}\{\cdots\}$	subspace spanned by vectors
$\widetilde{\square}$	perturbed version of $\square$

Chapter 1

Introduction and Motivation

If you have acquired this book, perhaps you do not need a motivation for studying inverse problems. Still, it is preferable to start with a few examples of the use of linear inverse problems. One example is that of computing the magnetization inside the volcano Vesuvius (near Napoli in Italy) from measurements of the magnetic field above the volcano. Clearly, this is a safe way to monitor the internal activities. Figure 1.1 below shows a computer simulation of this situation; the left figure shows the measured data on the surface of the volcano (the heights are exaggerated), and the right figure shows a reconstruction of the internal magnetization. Another example is the computation of sharp images from blurred ones, using a mathematical model of the point spread function that describes the blurring process; see Fig. 1.2 next page for an example.

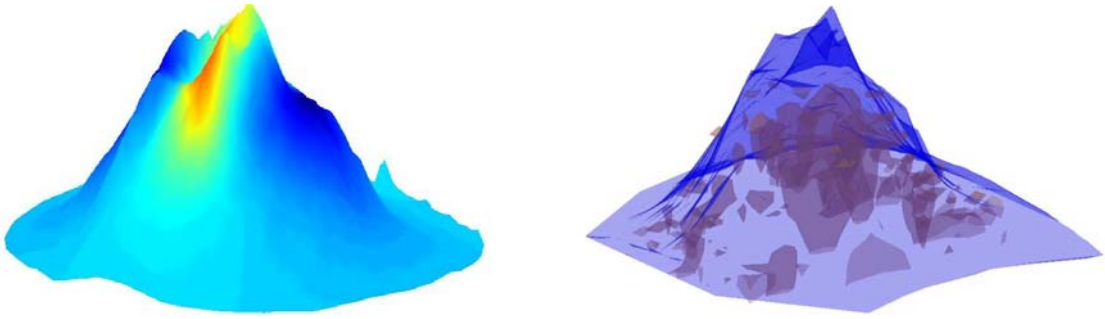


Figure 1.1: Left: simulated measurements of the magnetic field on the surface of Vesuvius. Right: a computed reconstruction of the magnetization inside the volcano. The heights are exaggerated.

Both are examples of a wide class of mathematical problems, referred to as *inverse problems*. These problems generally arise when we wish to computer information about internal or otherwise hidden data from outside (or otherwise accessible) measurements.

Inverse problems, in turn, belong to the class of *ill-posed problems*. The term was coined in the early 20th century by Hadamard who worked on problems in mathematical physics, and he believed that ill-posed problems did not model real-world problems (he was wrong). Hadamard defined a linear problem to be well-posed if it satisfies the following three requirements:

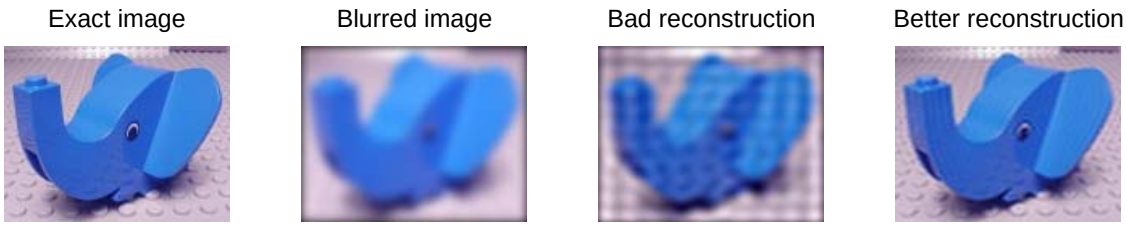


Figure 1.2: Example of image deblurring, i.e., reconstruction of a sharp image from a blurred one, using a mathematical model for the blurring proces.

1. Existence: the problem must have a solution.

2. Uniqueness: there must be only one solution to the problem.

3. Stability: the solution must depend continuously on the data.

If the problem violates one or more of these requirements, it is said to be ill posed.

Condition 1 seems to be trivial – and yet we shall demonstrate that we can write problems that do not have a solution. Consider, for example, the overdetermined system

$$\begin{pmatrix} 1 \\ 2 \end{pmatrix} x = \begin{pmatrix} 1 \\ 2.2 \end{pmatrix}.$$

This problem does not have a solution; there is no x such that $x = 1$ and $2x = 2.2$. Violations of the first criterion can often be fixed by a slight reformulation of the problem; for example, instead of the above system we can consider the associated least squares problem

$$\min_x \left\| \begin{pmatrix} 1 \\ 2 \end{pmatrix} x - \begin{pmatrix} 1 \\ 2.2 \end{pmatrix} \right\|_2^2 = \min\Big((x-1)^2 + (2x-2.2)^2 \Big)$$

which has the unique solution $x = 1.08$.

Condition 2 can be more critical; but again it can often be fixed by a reformulation of the problem – typically by adding additional requirements to the solution. If the requirements are carefully chosen the solution becomes unique. For example, the underdetermined problem

$$x_1 + x_2 = 1 \qquad \text{(the world's simplest ill-posed problem)}$$

has infinitely many solutions; but if we also require that $\|x\|_2^2 = x_1^2 + x_2^2$ is minimum, then there is a unique solution $x_1 = x_2 = 1/2$. (Note that Matlab does not compute minimum-norm solutions to underdetermined problem; instead Matlab returns a so-called "basic solution" with $x_1 = 1$ and $x_2 = 0$.)

Condition 3 is much harder to "fix" because a violation implies that arbitrarily small perturbations can produce arbitrarily large perturbations of the solution. At least, this is true for infinite-dimensional problems; for finite-dimensional problems the perturbation is always finite, but this is quite irrelevant if the perturbation is, say, of the order 10^{12}.

Again the key is to reformulate the problem such that the solution to the new problem is less sensitive to the perturbations. We say that we *stabilize* or *regularize* the problem, such

that the solution becomes more stable and regular. As an example, consider the least squares problem $\min_x \|A\,x - b\|_2$ with coefficient matrix and right-hand side given by

$$A = \begin{pmatrix} 0.16 & 0.10 \\ 0.17 & 0.11 \\ 2.02 & 1.29 \end{pmatrix}, \qquad b = A \begin{pmatrix} 1 \\ 1 \end{pmatrix} + \begin{pmatrix} 0.01 \\ -0.03 \\ 0.02 \end{pmatrix} = \begin{pmatrix} 0.27 \\ 0.25 \\ 3.33 \end{pmatrix}.$$

Here, we can consider the vector $(0.01\,,\,-0.03\,,\,0.02)^T$ a perturbation of the exact right-hand side $(0.26\,,\,0.28\,,\,3.31)^T$. There is no vector x such that $A\,x = b$, and the least squares solution is given by

$$x_{\mathrm{LS}} = \begin{pmatrix} 7.01 \\ -8.40 \end{pmatrix} \qquad \Rightarrow \qquad \|A\,x_{\mathrm{LS}} - b\|_2 = 0.022\ .$$

Two other "solutions" with a small residual are

$$x_{\mathrm{B}}^{(1)} = \begin{pmatrix} 1.65 \\ 0 \end{pmatrix} \qquad \Rightarrow \qquad \|A\,x_{\mathrm{B}}^{(1)} - b\|_2 = 0.031$$

$$x_{\mathrm{B}}^{(2)} = \begin{pmatrix} 0 \\ 2.58 \end{pmatrix} \qquad \Rightarrow \qquad \|A\,x_{\mathrm{B}}^{(2)} - b\|_2 = 0.036\ .$$

All three "solutions" x_{LS}, $x_{\mathrm{B}}^{(1)}$ and $x_{\mathrm{B}}^{(2)}$ have small residuals, yet they are far from the exact solution $(1\,,\,1)^T$!

The reason is that the matrix A is ill conditioned. When this is the case, it is well known from matrix computations that small perturbations of the right-hand side b can lead to large perturbations of the solution. It is also well known that a small residual does not imply that the perturbed solution is close to the exact solution.

Ill-conditioned problems are *effectively underdetermined.* For example, for the above problem we have

$$A \begin{pmatrix} -1.00 \\ 1.57 \end{pmatrix} = \begin{pmatrix} -0.0030 \\ 0.0027 \\ 0.0053 \end{pmatrix},$$

showing that the vector $(-1.00\,,\,1.57)^T$ is "almost" a null vector for A. Hence we can add a large component along this vector without changing the residual very much; the system behaves almost like an underdetermined system.

It turns out that we can modify the above problem such that the solution is more stable, i.e., less sensitive to perturbations. For example, we can enforce an upper bound δ on the norm of the solution; i.e, we solve the modified problem:

$$\min_x \|A\,x - b\|_2 \qquad \text{subject to} \qquad \|x\|_2 \le \delta\ .$$

The solution x_δ depends in a unique but nonlinear way on δ; for example

$$x_{0.1} = \begin{pmatrix} 0.08 \\ 0.05 \end{pmatrix}\ , \quad x_1 = \begin{pmatrix} 0.84 \\ 0.54 \end{pmatrix}\ , \quad x_{1.385} = \begin{pmatrix} 1.17 \\ 0.74 \end{pmatrix}\ , \quad x_{10} = \begin{pmatrix} 6.51 \\ -7.60 \end{pmatrix}\ .$$

The solution $x_{1.385}$ (for $\delta = 1.385$) is quite close to the exact solution. *By supplying the correct additional information we can compute a good approximate solution.* The main difficulty is how to choose the parameter δ when we have little knowledge about the exact solution.

Whenever we solve an inverse problem on a computer, we face difficulties similar to the above, because the associated computational problem is ill conditioned. The purpose of this book is:

1. To explain why ill-conditioned systems always arise when solving inverse problems.
2. To explain the fundamental "mechanisms" underlying the ill conditioning.
3. To explain how we can modify the problem in order to stabilize the solution.
4. To show how this can be done efficiently on a computer.

Regularization methods is at the heart of all this.

Chapter 2

Meet the First-Kind Fredholm Integral Equation

If you work in linear inverse problems arising from integral equations, you must make the first-kind Fredholm integral equation your friend. In particular, you must understand the "psyche" of this beast – and how it can play tricks on you, if you are not careful.

2.1 The Integral Equation

The Fredholm integral equation of the first kind takes the generic form

$$\int_0^1 K(s,t)\, f(t)\, dt = g(s), \qquad 0 \le s \le 1 \ . \tag{2.1}$$

Here, both the *kernel* K and the *right-hand side* g are known functions, while f is the unknown function. This equation establishes a linear relationship between the two functions f and g, and the kernel K describes the precise relationship between the two quantities. Thus, the function K describes the underlying model.

In multiple dimensions, this Eq. (2.1) takes the form

$$\int_{\Omega_{\mathbf{t}}} K(\mathbf{s},\mathbf{t})\, f(\mathbf{t})\, d\mathbf{t} = g(\mathbf{s}), \qquad \mathbf{s} \in \Omega_{\mathbf{s}} \ ,$$

where $\mathbf{s}$ and $\mathbf{t}$ are vectors in the two domains $\Omega_{\mathbf{s}}$ and $\Omega_{\mathbf{t}}$, respectively. In this presentation we will focus on the one-dimensional case.

If f and K are known then we can compute g by evaluating the integral; this is called the forward computation. The *inverse problem* consists of computing f given the right-hand side and the kernel.

An important special case of (2.1) is when the kernel is a function of the difference between s and t, i.e., $K(s,t) = h(s-t)$ where h is some function. This version of the integral equation is called *deconvolution*, and it takes the form

$$\int_0^1 h(s-t)\, f(t)\, dt = g(s), \qquad 0 \le s \le 1$$

(and similarly in more dimensions). Numerical regularization methods for deconvolution problems are described in [9].

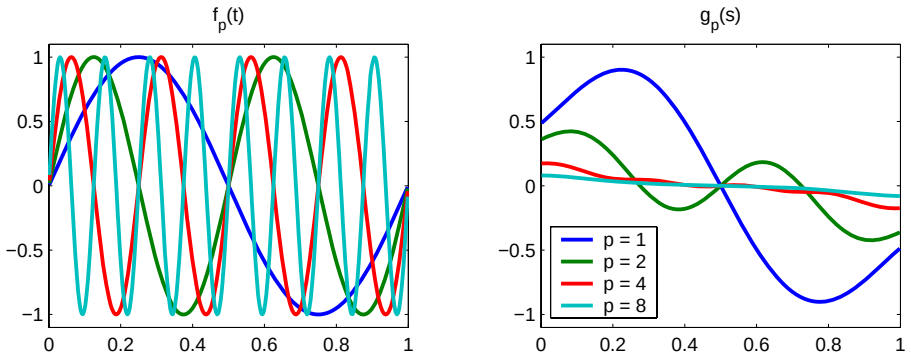


Figure 2.1: Illustration of the Riemann-Lebesgue lemma with the function $f_p(t) = \sin(2\pi\, p\, t)$. Clearly, the amplitude of g_p decreases as the frequency p increases.

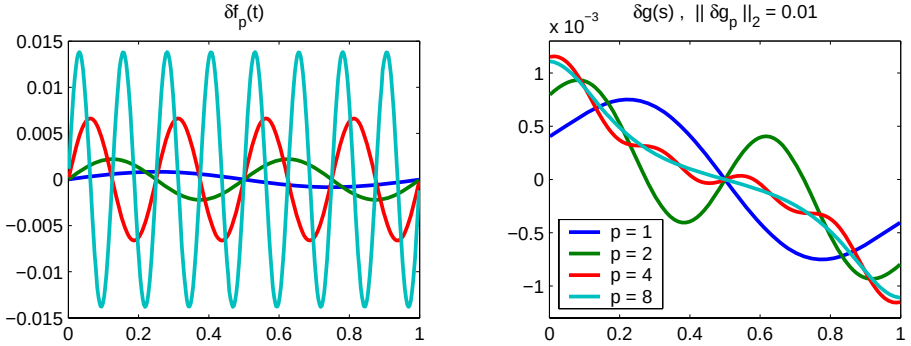


Figure 2.2: The consequence of the Riemann-Lebesgue lemma is that high frequencies are amplified in the inversion. Here the functions δg_p are normalized such that $\|\delta g_p\|_2 = 0.01$. Clearly, the amplitude of δf_p increases as the frequency p increases.

In the mapping from f to g, higher frequency components in f are generally damped more than components with lower frequency. Thus, the integration with K in (2.1) has a smoothing effect on the function f, such that g will appear smoother than f.

The *Riemann-Lebesgue lemma* is a precise mathematical statement of the above. If we define the function f_p by

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

then for "arbitrary" kernels K we have

$$g_p(s) = \int_0^1 K(s,t)\, f_p(t)\, dt \to 0 \qquad \text{for} \qquad p \to \infty.$$

That is, as the frequency of f increases – as measured by the integer p – the amplitude of g_p decreases; Figure 2.1 illustrates this.

In other words: higher frequencies are damped in the mapping of f to g, and therefore g will be smoother than f. The inverse problem, i.e., that of computing f from g, is therefore a process that amplifies high frequencies – and the higher the frequency, the more the

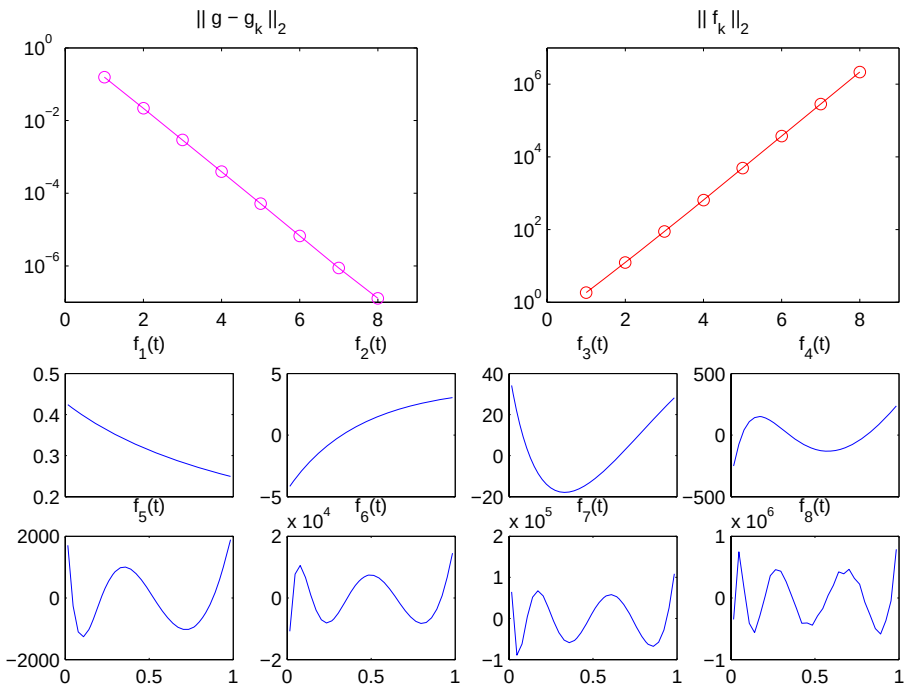


Figure 2.3: Top left: the norm of the error $g - g_k$ in the approximation for the right-hand side. Top right: the norm of the "approximate solution" f_k. Bottom: "approximate solutions" f_k for $k = 1, \ldots, 8$.

amplification. Figure 2.2 illustrates this; the functions δf_p and δg_p are related via the same integral equation as before, but now all four δg_p functions are scaled such that

$$\|\delta g_p\|_2 = \left(\int_0^1 \delta g_p(s)^2 ds \right)^{1/2} = 1.$$

The amplitude of δf_p increases as the frequency p increases.

Clearly, even a small random perturbation of g can lead to a very large perturbation of f if the perturbation has a high-frequency component. Think of the function δg_p in Fig. 2.2 as a perturbation of g, and the function δf_p as the corresponding perturbation of f. As a matter of fact, no matter how small the perturbation of g, the corresponding perturbation of f can be arbitrarily large (just set the frequency p of the perturbation high enough). This illustrates the fundamental problem of solving inverse problems, namely, that the solution may not depend continuously on the data.

Ursell has provided an example of a first-kind Fredholm integral equation that does not have a square integrable solution, i.e., there is no solution whose 2-norm is finite. His example takes the form

$$\int_0^1 \frac{1}{s+t+1} f(t)\, dt = 1, \qquad 0 \le s \le 1. \tag{2.2}$$

Hence, the kernel and the right-hand side are given by $K(s,t) = (s+t+1)^{-1}$ and $g(s) = 1$.

The kernel has a set of orthonormal eigenfunctions ϕ_i such that

$$\int_0^1 \frac{1}{s+t+1}\,\phi_i(t)\,dt = \ell_i\,\phi_i(s), \qquad i = 1, 2, \ldots,$$

where ℓ_i are the eigenvalues of K. Now let us expand the right-hand side $g(s) = 1$ in terms of these eigenfunctions:

$$g_k(s) = \sum_{i=1}^{k} (\phi_i, g)\,\phi_i(s),$$

and the top left plot in Fig. 2.3 shows that the error decreases as k increases; we have

$$\|g - g_k\|_2 \to 0 \qquad \text{for} \qquad k \to \infty.$$

Next we consider the integral equation $\int_0^1 (s+t+1)^{-1}\,f_k(t)\,dt = g_k(s)$, whose solution f_k is given by the expansion

$$f_k(t) = \sum_{i=1}^{k} \frac{(\phi_i, g)}{\ell_i}\,\phi_i(t).$$

Clearly $\|f_k\|_2$ is finite for all k; but the top right plot in Fig. 2.3 shows that the norm of f_k increases with k:

$$\|f_k\|_2 \to \infty \qquad \text{for} \qquad k \to \infty.$$

The bottom plots in Fig. 2.2 corroborate this: clearly, the amplitudes of the functions f_k increase with k, and the functions do not seem to converge to a square integrable solution.

Why bother about these (strange) issues associated with ill-posed problem? The fact is that Fredholm integral equations of the first kind are used to model a variety of real applications, such as:

- Medical imaging (CT brain scanning, etc.).
- Geophysical prospecting (search for oil, land-mines, etc.).
- Image deblurring (astronomy, CSI[1], etc.)
- Deconvolution of a measurement instrument's response.

We can only hope to compute useful solutions to these problems if we fully understand their *inherent* difficulties. Moreover, we must understand how these difficulties carry over to the discretized problems involved in a computer solution, such that we can deal with them (mathematically and numerically) in a satisfactory way. This is precisely what the rest of this note is about.

2.2 A Model Problem From Geophysics

It is convenient to have a simple model problem to illustrate our theory and algorithms. We will use a simplified problem from gravity surveying. An unknown mass distribution with density $f(t)$ is located at depth d below surface, from 0 to 1 on the t axis shown in Fig. 2.4. We assume there is no mass outside this source, which produces a gravity field everywhere. At

[1] Crime Scene Investigation.

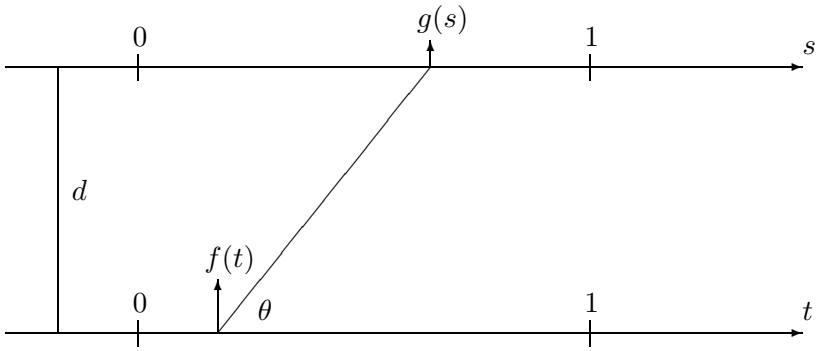


Figure 2.4: The geometry of the gravity surveying model problem.

the surface, along the s axis (see the figure) from 0 to 1, we measure the vertical component of the gravity field, which we refer to as $g(s)$.

The two functions f and g are (not surprisingly) related via a Fredholm integral equation of the first kind. The gravity field from an infinitesimally small part of $f(t)$, of length dt, on the t axis is identical to the field from a point mass at t of strength $f(t)\,dt$. Hence, the magnitude of the gravity field along s is $f(t)\,dt\,/\,r^2$, where $r = \sqrt{d^2 + (s-t)^2}$ is the distance between the "source point" at t and the "field point" at s. The direction of the gravity field is from the "field point" to the "source point," and therefore the measured value of $g(s)$ is

$$dg = \frac{\sin\theta}{r^2}\, f(t)\, dt \ ,$$

where θ is the angle shown in Fig. 2.4. Using that $\sin\theta = d/r$, we obtain

$$\frac{\sin\theta}{r^2}\, f(t)\, dt = \frac{d}{(d^2 + (s-t)^2)^{3/2}}\, f(t)\, dt \ .$$

The total value of $g(s)$ for any $0 \le s \le 1$ consists of contributions from all mass along the t axis (from 0 to 1), and it is therefore given by the integral

$$g(s) = \int_0^1 \frac{d}{(d^2 + (s-t)^2)^{3/2}}\, f(t)\, dt \ .$$

This is the forward problem which describes how we can compute the measurable data g given the source f.

The associated inverse problem of gravity surveying is obtained by swapping the ingredients of the forward problem, and writing it as

$$\int_0^1 \frac{d}{(d^2 + (s-t)^2)^{3/2}}\, f(t)\, dt = g(s), \qquad 0 \le s \le 1.$$

The kernel K, which represents the model, is thus given by

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

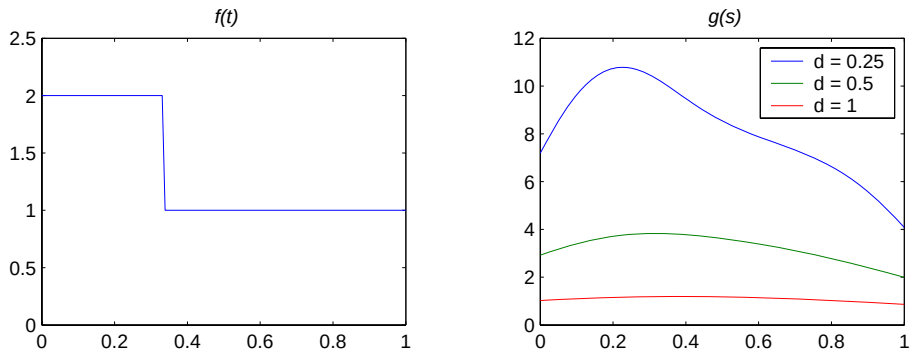


Figure 2.5: Illustration of the smoothing effect associated with the forward computation. The left figure shows the function f (the mass density distribution), and the right figure shows the measured signal for three different values of the depth d.

and the right-hand side g is what we are able to measure. From K and g we want to compute f, and this is the *inverse problem.*

Let us illustrate the smoothing effect of the forward computation that we have discussed in the previous section. Figure 2.5 shows an example of a forward computation with a given function f and three different values of the depth d. Obviously, the deeper the source, the weaker the signal. Moreover, we see that the observed signal g (the data) is much smoother than the source f, and in fact the discontinuity in f is not visible in g.

2.3 The Singular Value Expansion and the Picard Condition

The singular value expansion is a mathematical tool that gives us a "handle" on the discussion of the smoothing effect and the existence of solutions to first-kind Fredholm integral equations.

The kernel K in our generic problem is square integrable if the integral $\int_0^1 \int_0^1 K(s,t)^2 \, ds \, dt$ is finite. For any square integrable kernel K the singular value expansion (SVE) takes the form

$$K(s,t) \doteq \sum_{i=1}^{\infty} \mu_i \, u_i(s) \, v_i(t). \tag{2.3}$$

Note the dotted equation sign "$\doteq$" in (2.3); this is used to emphasize that the quantity on the right side (here, the singular value expansion) converges in the mean to the quantity on the left (here, the kernel K).

The functions u_i and v_i are called the left and right singular functions. They are orthonormal with respect to the usual inner product $(\phi, \psi) = \int_0^1 \phi(t) \, \psi(t) \, dt$, i.e.,

$$(u_i, u_j) = (v_i, v_j) = \delta_{ij}, \qquad i = 1, 2, \ldots$$

The quantities μ_i are called the singular values, and they form a non-increasing sequence:

$$\mu_1 \geq \mu_2 \geq \mu_2 \geq \ldots \geq 0.$$

If there is only a finite number of nonzero singular values, then the kernel is called degenerate. The singular values and functions satisfy a number of relations, and the most important is

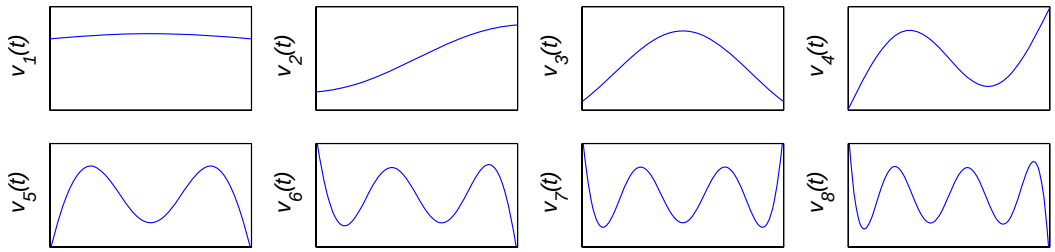


Figure 2.6: The number of oscillations, or zero-crossings, in u_i an v_i increases with i (as the corresponding singular values decay).

the "fundamental relation"

$$\int_0^1 K(s,t)\, v_i(t)\, dt \doteq \mu_i\, u_i(s)\ , \qquad i = 1, 2, \ldots$$

We note that it is rare that we can determine the SVE analytically; later we shall demonstrate how we can compute it numerically.

The singular values μ_i always decay to zero, and it turns out that the "smoother" the kernel K the faster the decay. Specifically, if the derivatives of K of order $0, \ldots, q$ exist and are continuous, then the singular values μ_i decay approximately as $\mathcal{O}(i^{-q-1/2})$.

The singular functions resemble spectral bases in the sense that the smaller the μ_i, the more oscillations (or zero-crossings) in the corresponding singular functions u_i and v_i. Figure 2.6 illustrates this.

The left and right singular functions form bases of the function space $L_2([0,1])$ of square integrable function in the interval $[0,1]$. Hence, we can expand both f and g in terms of these functions:

$$f(t) \doteq \sum_{i=1}^{\infty} (v_i, f)\, v_i(t) \qquad \text{and} \qquad g(s) \doteq \sum_{i=1}^{\infty} (u_i, g)\, u_i(s).$$

If we insert the expansion for f into the integral equation and make use of the fundamental relation, then we see that g can also be written as

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

Since v_i is mapped to $\mu_i\, u_i(s)$, this explains why the higher frequencies are damped more than lower frequencies in the forward problem, i.e., why the integration with K has a smoothing effect.

If we insert the two expansions for f and g into the integral equation and make use of the fundamental relation again, then it is easy to obtain the relation

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

By equating each of the terms in these expansions, we see that the expansion coefficients for the solution are $(v_i, f) = (u_i, g)\,/\,\mu_i$ for $i = 1, 2, \ldots$, and this leads to the following expression

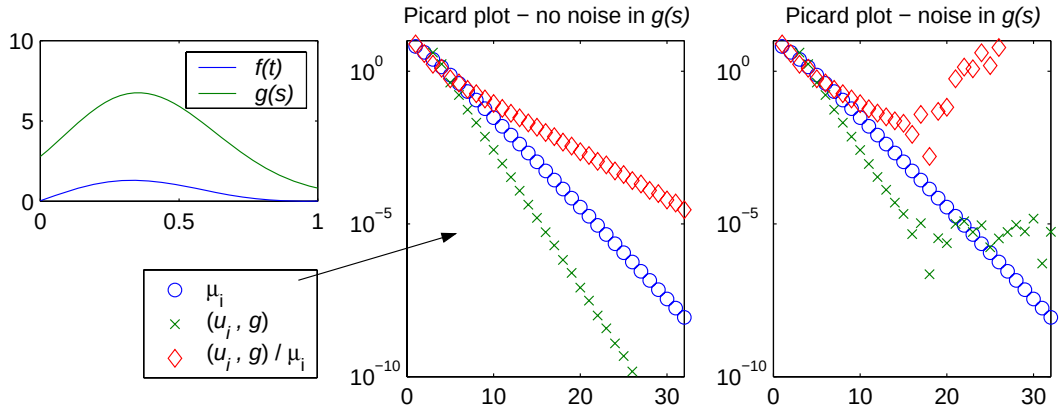


Figure 2.7: Illustration of the Picard condition. The left figure shows the functions f and g used in the example. The middle figure shows the singular values μ_i and the SVE expansion coefficients (u_i, g) and $(u_i, g)\,/\,\mu_i$ (for g and f, respectively) for a noise-free problem. The right figure shows how these coefficients change when we add noise to the right-hand side.

for the solution:

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

While this relation is not suited for numerical computations, it sheds important light on the existence of a solution to the integral equation. More precisely, we see that for a square integrable solution f to exist, the 2-norm $\|f\|_2$ must satisfy

$$\|f\|_2^2 = \int_0^1 f(t)^2 \, dt = \sum_{i=1}^{\infty} (v_i, f)^2 = \sum_{i=1}^{\infty} \left(\frac{(u_i, g)}{\mu_i} \right)^2 < \infty \ . \tag{2.4}$$

(This is equivalent to the condition that g must belong to the range of the kernel K.) The above relation is called the *Picard condition*, and it says that the right-hand side coefficients (u_i, g) must decay to zero somewhat faster than the singular values μ_i.

The trouble with first-kind Fredholm integral equations is that, even if the exact data satisfies the Picard condition, the measured and noisy data g usually violates the condition. Figure 2.7 illustrates this. The middle figure shows the singular values μ_i and the expansion coefficients (u_i, g) and $(u_i, g)\,/\,\mu_i$ for the ideal situation with no noise, and here the Picard condition is clearly satisfied. The right figure shows the behavior of these quantities when we add noise to the right-hand side; now the right-hand side coefficient (u_i, g) level off at the noise level, and the Picard condition is violated. As a consequence, the solution coefficients $(u_i, g)\,/\,\mu_i$ increase, and the norm of the "solution" becomes unbounded.

The violation of the Picard condition is the simple explanation of the instability of linear inverse problems in the form of first-kind Fredholm integral equations. At the same time, the insight we gain from a study of the quantities associated with the SVE gives a hint how to deal with noisy problems that violate the Picard condition; simply put, we want to filter the noisy SVE coefficients.

2.4 Exercises

2.4.1 An Analytic SVE

We can occasionally calculate the SVE analytically. Consider the first-kind Fredholm integral equation $\int_0^1 K(s,t)\, f(t)\, dt = g(s)$, $0 \le s \le 1$ with the kernel

$$K(s,t) = \begin{cases} s(t-1), & s < t \\ t(s-1), & s \ge t. \end{cases} \tag{2.5}$$

This equation is associated with the computation of the second derivative of a function: given a function g, the solution f is the second derivative of g, i.e., $f(t) = g''(t)$ for $0 \le t \le 1$. An alternative expression for the kernel is given by:

$$K(s,t) = -\frac{2}{\pi^2} \sum_{i=1}^{\infty} \frac{\sin(i\,\pi\,s)\, \sin(i\,\pi\,t)}{i^2} \ .$$

Use this relation to derive the following expressions

$$\mu_i = \frac{1}{(i\pi)^2}, \quad u_i(s) = \sqrt{2}\sin(i\pi s), \quad v_i(t) = -\sqrt{2}\sin(i\pi t), \qquad i = 1, 2, \ldots$$

for the singular values and singular functions.

2.4.2 An Analytic SVE of a Degenerate Kernel

The purpose of this exercise is to illustrate how the singular value expansion (SVE) can give valuable insight into the problem. We consider the following simple integral equation

$$\int_{-1}^{1} (s + 2t)\, f(t)\, dt = g(s) \ , \quad -1 \le s \le 1, \tag{2.6}$$

i.e., the kernel is given by

$$K(s,t) = s + 2t \ .$$

This kernel is chosen solely for its simplicity. Show (e.g., by insertion) that the SVE of the kernel K is given by

$$\mu_1 = 4/\sqrt{3}, \quad \mu_2 = 2/\sqrt{3}, \quad \mu_3 = \mu_4 = \ldots = 0$$

and

$$u_1(s) = 1/\sqrt{2}, \quad u_2(s) = \sqrt{3/2}\, s, \quad v_1(t) = \sqrt{3/2}\, t, \quad v_2(t) = 1/\sqrt{2}.$$

Show that this integral equation only has a solution if the right-hand side is a linear function, i.e., it has the form $g(s) = \alpha\, s + \beta$ where α and β are constants. Hint: evaluate $\int_{-1}^{1} K(s,t)\, f(t)\, dt$ for an arbitrary function f.

Chapter 3

Getting to Business: Discretizations of Linear Inverse Problems

After our encounter with some of the fundamental theoretical aspects of the first-kind Fredholm integral equation, it is now time to study how we can turn this beast into something that can be represented on a computer. We will discuss basic aspects of discretization methods, and we will introduce the matrix-version of the SVE, namely, the singular value decomposition. We finish with a brief look at noisy discrete problems.

3.1 Quadrature and Expansion Methods

In order to handle continuous problems, such as integral equations, on a computer (which is designed to work with numbers) we must replace the problem of computing the unknown function f in (2.1) with a discrete and finite-dimensional problem that we can solve on the computer. Since the underlying integral equation is assumed to be linear, we want to arrive at a system of linear algebraic equations. There are two basically different approaches to do this.

Quadrature Methods. These methods compute approximations of the form

$$\tilde{f}_j = \tilde{f}(t_j), \qquad j = 1, 2, \ldots, n$$

to the solution f at the abscissas $t_1, t_2, \ldots, t_n$. Notice the tilde: we compute sampled values of some function $\tilde{f}$ that approximates the exact solution f, but we do not obtain samples of f itself (unless it happens to be perfectly represented by the simple function underlying the quadrature rule).

Expansions Methods. These methods compute an approximation of the form

$$\tilde{f}(t) = \sum_{j=1}^{n} \alpha_j \, \phi_j(t),$$

where $\phi_1(t), \ldots, \phi_n(t)$ are expansion or basis functions chosen by the user. They should be chosen in such a way that they provide a good description of the solution – but no matter how good they are, we still compute an approximation $\tilde{f}$ living in the span of these expansion functions.

Quadrature methods take their basis in the general quadrature rule of the form

$$\int_0^1 \varphi(t)\,dt = \sum_{j=1}^{n} \omega_j\,\varphi(t_j) + R_n \ ,$$

where φ is the function whose integral we want to evaluate, R_n is the quadrature error, $t_1, t_2, \ldots, t_n$ are the abscissas for the quadrature rule, and $\omega_1, \omega_2, \ldots, \omega_n$ are the corresponding weights. For example, for the midpoint rule we have

$$t_j = \frac{j - \frac{1}{2}}{n}, \qquad \omega_j = \frac{1}{n}, \qquad j = 1, 2, \ldots, n.$$

The crucial step in the derivation of a quadrature method is to apply this rule *formally* to the integral, pretending that we know the values $f(t_j)$. Notice that the result is still a function of s:

$$\Psi(s) = \int_0^1 K(s,t)\,f(t)\,dt = \sum_{j=1}^{n} \omega_j\,K(s,t_j)\,f(t_j) + R_n(s) \ ;$$

here the error term R_n is a function of s. We now enforce the *collocation* requirement that Ψ must be equal to the right-hand side g at n selected points:

$$\Psi(s_i) = g(s_i) \ , \qquad i = 1, \ldots, n \ ,$$

where $g(s_i)$ are sampled or measured values of the function g. Finally we must, of course, neglect the unknown error term $R_n(s)$, and in doing so we introduce an error in the computed solution. Therefore we must replace the exact values $f(t_j)$ with approximate values, denoted by $\tilde{f}_j$:

$$\sum_{j=1}^{n} \omega_j\,K(s_i,t_j)\,\tilde{f}_j = g(s_i), \quad i = 1, \ldots, n \ . \tag{3.1}$$

We note in passing that we could have used $m > n$ collocation points, in which case we would obtain an overdetermined system. In this presentation, for simplicity we always use $m = n$.

The relation in (3.1) is just a linear system of equations in the unknowns

$$\tilde{f}_j = \tilde{f}(t_j), \qquad j = 1, 2, \ldots, n.$$

To see this, we can write the n equations in (3.1) in the form

$$\begin{pmatrix} \omega_1 K(s_1,t_1) & \omega_2 K(s_1,t_2) & \cdots & \omega_n K(s_1,t_n) \\ \omega_1 K(s_2,t_1) & \omega_2 K(s_2,t_2) & \cdots & \omega_n K(s_2,t_n) \\ \vdots & \vdots & & \vdots \\ \omega_1 K(s_n,t_1) & \omega_2 K(s_n,t_2) & \cdots & \omega_n K(s_n,t_n) \end{pmatrix} \begin{pmatrix} \tilde{f}_1 \\ \tilde{f}_2 \\ \vdots \\ \tilde{f}_n \end{pmatrix} = \begin{pmatrix} g(s_1) \\ g(s_2) \\ \vdots \\ g(s_n) \end{pmatrix}$$

or simply $A\,x = b$, where A is an $n \times n$ matrix. The elements of the matrix A, the right-hand side b and the solution vector x are given by

$$\left.\begin{aligned} a_{ij} &= \omega_j\,K(s_i,t_j) \\ x_j &= \tilde{f}(t_j) \\ b_i &= g(s_i) \end{aligned}\right\} \qquad i,j = 1, \ldots, n \ .$$

Expansion methods come in several flavors, and we will focus on the Galerkin method in which we must choose two sets of functions ϕ_i and ψ_j for the solution and the right-hand side, respectively. Then we can write

$$\begin{aligned} f(t) &= \tilde{f}(t) + R_f(t), \qquad & \tilde{f} &\in \text{span}\{\phi_1, \ldots, \phi_n\} \\ g(s) &= \tilde{g}(s) + R_g(s), \qquad & \tilde{g} &\in \text{span}\{\psi_1, \ldots, \psi_n\} \ , \end{aligned}$$

where R_f and R_g are the expansion errors for f and g due to the use of a finite set of basis vectors. To be more specific, we write the approximate solution $\tilde{f}$ in the form

$$\tilde{f}(t) = \sum_{j=1}^{n} \zeta_j \, \phi_j(t) \ ,$$

and we want to determine the expansion coefficients ζ_j such that $\tilde{f}$ is an approximate solution to the integral equation. We therefore introduce the function

$$\Phi(s) = \int_0^1 K(s,t) \, \tilde{f}(t) \, dt = \sum_{j=1}^{n} \zeta_j \int_0^1 K(s,t) \, \phi_j(t) \, dt$$

and, similarly to f and g, we write this function in the form

$$\Phi(s) = \widetilde{\Phi}(s) + R_\Phi(s), \qquad \widetilde{\Phi} \in \text{span}\{\psi_1, \ldots, \psi_n\} \ ,$$

How do we choose the functions ϕ_i and ψ_j? The basis functions ϕ_i for the solution should preferably reflect the knowledge we may have about the solution f (such as smoothness, monotonicity, periodicity, etc.). Similarly, the basis functions ψ_j should capture the main information in the right-hand side. Sometimes, for convenience, the two sets of functions are chosen to be identical.

The key step in the Galerkin method is to recognize that, in general, the function Φ is not identical to g, and neither does Φ lie in the subspace $\text{span}\{\psi_1, \ldots, \psi_n\}$ in which $\tilde{g}$ lies. Now define $\widetilde{\Phi}$ and $\tilde{g}$ as the orthogonal projections of Φ and g on the subspace $\text{span}\{\psi_1, \ldots, \psi_n\}$ (this implicitly determines the coefficients β_i, but we do not need them explicitly). The Galerkin approach is then to determine the coefficients α_j such that the two projections $\widetilde{\Phi}$ and $\tilde{g}$ are identical:

$$\begin{aligned} \widetilde{\Phi}(s) &= \tilde{g}(s) & &\Leftrightarrow \\ \Phi(s) - R_\Phi(s) &= g(s) - R_g(s) & &\Leftrightarrow \\ \Phi(s) - g(s) &= R_\Phi(s) - R_g(s). \end{aligned}$$

Here, we recognize the function $\Phi(s) - g(s)$ as the residual. Since both functions $R_\Phi(s)$ and $R_g(s)$ are orthogonal to the subspace $\text{span}\{\psi_1, \ldots, \psi_n\}$, the Galerkin condition can be stated as the requirement that the residual $\Phi(s) - g(s)$ is orthogonal to each of the functions $\psi_1, \ldots, \psi_n$.

In order to arrive at a system of equations for computing the unknowns $\zeta_1, \ldots, \zeta_n$, we now enforce the orthogonality of the residual and the ψ_i-functions: That is, we require that

$$(\psi_i \, , \, \Phi - g) = 0 \qquad \text{for} \qquad i = 1, \ldots, n.$$

This is equivalent to requiring that

$$(\psi_i\,,\,g) = \left(\psi_i\,,\,\int_0^1 K(s,t)\,\tilde{f}(t)\,dt\right), \qquad i = 1,\ldots,n.$$

Inserting the expansion for $\tilde{f}$, we obtain the relation

$$(\psi_i\,,\,g) = \sum_{j=1}^{n} \zeta_j \left(\psi_i\,,\,\int_0^1 K(s,t)\,\phi_j(t)\,dt\right), \qquad i = 1,\ldots,n$$

which, once again, is just an $n \times n$ system of linear algebraic equations $A\,x = b$ whose solution coefficients are the unknown expansion coefficients, i.e., $x_i = \zeta_i$ for $i = 1,\ldots,n$, and the elements of A and b are given by

$$a_{ij} = \int_0^1 \int_0^1 \psi_i(s)\,K(s,t)\,\phi_j(t)\,ds\,dt \tag{3.2}$$

$$b_i = \int_0^1 \psi_i(s)\,g(s)\,ds\ . \tag{3.3}$$

If these integrals are too complicated to be expressed in closed form, they must be evaluated numerically (by means of a suited quadrature formula). Notice that if K is symmetric and $\psi_i = \phi_i$, then the matrix A is symmetric.

To illustrate the expansion method, let us consider a very simple choice of orthonormal basis functions, namely, the "top hat" ⊓⊔ functions (or scaled indicator functions) on an equidistant grid with interspacing $h = 1/n$:

$$\chi_i(t) = \begin{cases} h^{-1/2}, & t \in [\,(i-1)h\,,\,ih\,] \\ 0, & \text{elsewhere} \end{cases} \qquad i = 1,2,\ldots,n. \tag{3.4}$$

These functions are clearly orthogonal (because their supports do not overlap), and they are scaled to have unit 2-norm (verify this). Hence, by choosing

$$\phi_i(t) = \chi_i(t) \qquad \text{and} \qquad \psi_i(s) = \chi_i(s) \qquad \text{for} \qquad i = 1,2,\ldots,n$$

we obtain two sets of orthonormal basis functions for the Galerkin method. The matrix and right-hand side elements are then given by

$$a_{ij} = h^{-1} \int_{(i-1)h}^{ih} \int_{(j-1)h}^{jh} K(s,t)\,ds\,dt$$

$$b_i = h^{-1/2} \int_{(i-1)h}^{ih} g(s)\,ds\ .$$

Again, it may be necessary to use quadrature rules to evaluate these integrals.

3.2 The Singular Value Decomposition

From the discussion in the previous chapter it should be clear that the SVE is a very powerful tool to analyze first-kind Fredholm integral equations. In the discrete setting, with

finite-dimensional matrices, there is a similarly powerful tool known as the *singular value decomposition*[1] (SVD).

The SVD is defined for any rectangular matrix. However, in this presentation we assume for simplicity of the exposition that the matrix is either square or has more rows that columns. Then, for any matrix $A \in \mathbb{R}^{m\times n}$ with $m \geq n$, the SVD takes the form

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

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

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

Two important matrix norms are easily expressed in terms of the singular values:

$$\begin{aligned}
\|A\|_{\mathrm{F}} &\equiv \left(\sum_{i=1}^{m}\sum_{j=1}^{n} a_{ij}^2\right)^{1/2} = \big(\mathrm{trace}(A^T A)\big)^{1/2} = \big(\mathrm{trace}(V\,\Sigma^2 V^T)\big)^{1/2} \\
&= \big(\mathrm{trace}(\Sigma^2)\big)^{1/2} = \big(\sigma_1^2 + \sigma_2^2 + \cdots + \sigma_n^2\big)^{1/2} \\
\|A\|_2 &\equiv \max_{\|x\|_2=1} \|A\,x\|_2 = \sigma_1.
\end{aligned}$$

(The derivation of the latter expression is omitted here.) The matrices $U \in \mathbb{R}^{m\times n}$ and $V \in \mathbb{R}^{n\times n}$ consist of left and right *singular vectors*

$$U = (u_1, \ldots, u_n)\ , \qquad V = (v_1, \ldots, v_n)$$

and both matrices have orthonormal columns:

$$U^T U = V^T V = I.$$

It is straightforward to show that the inverse of A (if it exists) is given by

$$A^{-1} = V\,\Sigma^{-1} U^T.$$

Thus we have $\|A^{-1}\|_2 = \sigma_n^{-1}$ and it follows immediately that the 2-norm condition number of A is given by

$$\mathrm{cond}(A) = \|A\|_2\,\|A^{-1}\|_2 = \sigma_1/\sigma_n.$$

The same expression holds if A is rectangular and has full rank.

Software for computing the SVD is included in all serious program libraries, and Table 3.1 summarizes some of the most important implementations. The complexity of all these SVD algorithms is $\mathcal{O}(m\,n^2)$ flops, as long as $m \geq n$. There is also software, based on Krylov subspace methods, for computing selected singular values of large sparse or structured matrices; for example, Matlab's `svds` function belongs in this class.

[1] While the SVD is now standard tool in signal processing and many other fields of applied mathematics, and which most students will encounter during their studies, it is interesting to note that the SVD was developed much later that the SVE.

Table 3.1: Some available software for computing the SVD.

Software package	Subroutine
ACM TOMS	`HYBSVD`
EISPACK	`SVD`
GNU Sci. Lib.	`gsl_linalg_SV_decomp`
IMSL	`LSVRR`
LAPACK	`_GESVD`
LINPACK	`_SVDC`
NAG	`F02WEF`
Numerical Recipes	`SVDCMP`
Matlab	`svd`

The singular values and vectors obey a number of relations which are similar to the SVE. Perhaps the most important relations are:

$$\left.\begin{array}{rclcrcl} A\,v_i &=& \sigma_i\,u_i & & \|A\,v_i\|_2 &=& \sigma_i \\ A^T u_i &=& \sigma_i\,v_i & & \|A^T u_i\|_2 &=& \sigma_i \end{array}\right\} \qquad i = 1, \ldots, n.$$

We can use some of these relations to derive an expression of for the solution $x = A^{-1}b$. First write x and b in the form

$$x = \sum_{i=1}^{n} (v_i^T x)\, v_i, \qquad b = \sum_{i=1}^{n} (u_i^T b)\, u_i$$

and use the expression for x, together with the SVD, to obtain

$$A\,x = \sum_{i=1}^{n} \sigma_i\, (v_i^T x)\, u_i \ .$$

By equating the expressions for $A\,x$ and b, and comparing the coefficients in the expansions, we immediately see that the solution is given by

$$x = A^{-1}b = \sum_{i=1}^{n} \frac{u_i^T b}{\sigma_i}\, v_i \ .$$

This expression can also be derived via the SVD-relation for A^{-1}, but the above derivation parallels that of the SVE. For the same reasons as in the previous chapter (small singular values σ_i and noisy coefficients $u_i^T b$) we do not want to compute this solution to discretizations of ill-posed problems.

The close relationship between the SVD and the SVE is reflected in the fact that we can use the SVD – together with the Galerkin discretization technique – to compute approximations to the SVE. Specifically, if we compute the matrix A according to (3.2), then the singular values σ_i of A are approximations to the singular values μ_i of K. More precisely, if we define the positive quantity Δ_n such that

$$\Delta_n^2 = \|K\|_2^2 - \|A\|_F^2, \qquad \|K\|_2^2 \equiv \int_0^1 \int_0^1 |K(s,t)|^2\, ds\, dt,$$

and if we use $\sigma_i^{(n)}$ to denote the singular values of the $n \times n$ matrix A, then it can be shown that

$$0 \leq \mu_i - \sigma_i^{(n)} \leq \Delta_n, \qquad i = 1, \ldots, n$$

and

$$\sigma_i^{(n)} \leq \sigma_i^{(n+1)} \leq \mu_i, \qquad i = 1, \ldots, n.$$

That is, the singular values of A are increasingly better approximations to those of K, and the error is bounded by the quantity Δ_n.

We can also compute approximations to the left and right singular functions via the SVD. Define the functions

$$\begin{aligned} \tilde{u}_j(s) &= \sum_{i=1}^{n} u_{ij}\, \psi_i(s)\ , \qquad j = 1, \ldots, n \\ \tilde{v}_j(t) &= \sum_{i=1}^{n} v_{ij}\, \phi_i(t)\ , \qquad j = 1, \ldots, n\ . \end{aligned}$$

Then these functions converge to the singular functions, i.e., $\tilde{u}_j(s) \to u_j(s)$ and $\tilde{v}_j(t) \to v_j(t)$, as $n \to \infty$. For example, it can be shown that

$$\max\left\{\|u_1 - \tilde{u}_1\|_2\,,\ \|v_1 - \tilde{v}_1\|_2\right\} \leq \left(\frac{2\,\Delta_n}{\mu_1 - \mu_2}\right)^{1/2}.$$

Note that this bound depends on the separation between the first and the second singular values. Similar, but more complicated, bounds exist for the other singular functions.

A nice, and important, consequence of this property of the singular functions is that the SVD-coefficients $u_i^T b$ are approximations to the corresponding inner products (u_i, g) in the SVE basis. To see this, we first note that the inner products $(\tilde{u}_j, \tilde{g})$ become increasingly better approximations to (u_i, g). We now insert the expressions for $\tilde{u}_j$ and $\tilde{g}$ into the definition of their inner product:

$$\begin{aligned} (\tilde{u}_j, \tilde{g}) &= \int_0^1 \sum_{i=1}^{n} u_{ij}\, \psi_i(s) \sum_{k=1}^{n} b_k\, \psi_k(s)\, ds \\ &= \sum_{i=1}^{n} u_{ij} \sum_{k=1}^{n} b_k\, (\psi_i, \psi_k) = \sum_{i=1}^{n} u_{ij}\, b_i = u_j^T b. \end{aligned}$$

Hence we can study all the important SVE relations by computing the SVD of the matrix A, and in particular we can use the numerical quantities to investigate whether the Picard conditions seems to be satisfied.

To illustrate these aspects, we consider a discretization of the gravity surveying model problem (from Chapter 2) by means of the Galerkin method with the "top hat" functions defined in (3.4). Figure 3.1 shows the singular values for a discretization with $n = 64$. Clearly, the singular values decay to zero with no gap anywhere in the spectrum. For $i \geq 55$ they tend to level off, which is due to the influence of the finite-precision arithmetic (these tiny singular values are smaller than the machine precision times the largest singular value σ_1, and they cannot be trusted). The condition number of A is of the order 10^{18} which, for all practical purposes, is infinite.

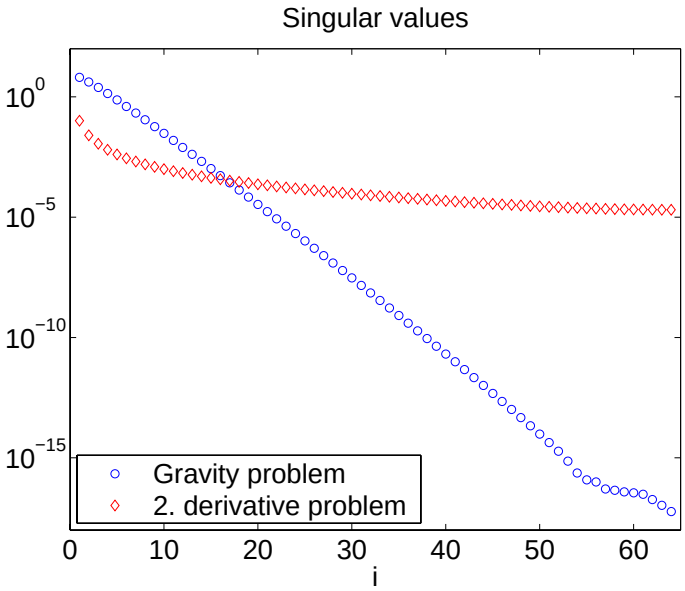


Figure 3.1: The singular values σ_i of two 64×64 matrixes A obtained from discretization of the gravity surveying problem (blue circles) and the second derivative problem from Exercise 2.4.1 (red diamonds).

Figure 3.1 also shows the singular values of the matrix obtained by discretization of the second derivative problem in (2.5) in Exercise 2.4.1. In accordance with the theory mentioned in the previous chapter, these singular values decay slower because the kernel K for this problem is less smooth than the kernel for the gravity problem. The kernel K in (2.5) is not differentiable at $s = t$.

Figure 3.2 shows the first nine left singular vectors u_i; the right singular functions are identical, because the matrix is symmetric. We see that the singular vectors have more oscillations as the index i increases, i.e., as the corresponding σ_i decrease (for this problem, the number of sign changes in the ith singular vector is $i-1$). We can safely conclude that the singular functions behave in the same manner. We note, however, that while all 64 singular vectors are orthonormal, by construction, the last 10 vectors do not carry reliable information about the singular functions (because the corresponding singular values are so tiny).

Finally let us investigate the behavior of the SVD coefficients $u_i^T b$ (of the right-hand side) and $u_i^T b/\sigma_i$ (of the solution). A plot of these coefficients, together with the singular values, is often referred to as a *Picard plot*, and Fig. 3.3 shows such a plot for the discretized gravity surveying problem for a noise-free right-hand side. Clearly, the coefficients $|u_i^T b|$ decay faster than the singular values – until they level off for $i \geq 35$ at a plateau determined by the machine precision, and these tiny coefficients are less than the machine precision times $\|b\|_2$. The solution coefficients $u_i^T b/\sigma_i$ also decay for $i < 35$, but for $i \geq 35$ they start to increase again due to the inaccurate values of $u_i^T b$.

This illustrates that even without errors in the data, we cannot always expect to compute a meaningful solution to the discretized inverse problem. The influence of rounding errors in the computation of the SVD is enough to destroy the computed solution. We note that the same is true if the solution $x = A^{-1}b$ is computed via Gaussian elimination, only the

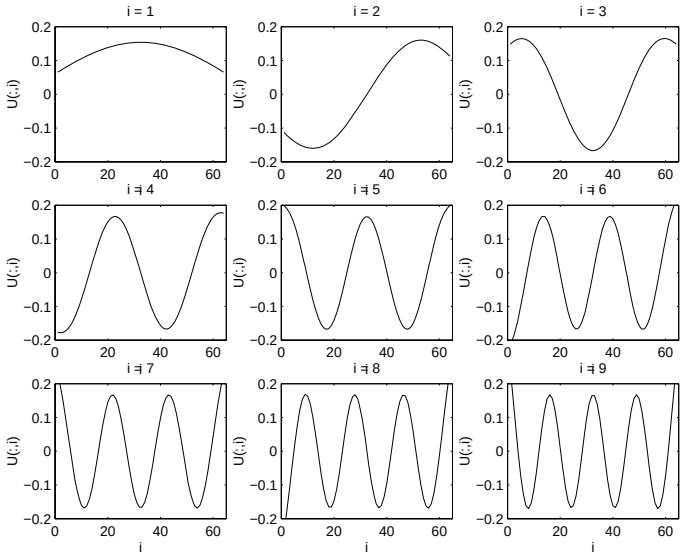


Figure 3.2: The first 9 left singular vectors u_i of the same matrix as in Fig. 3.1.

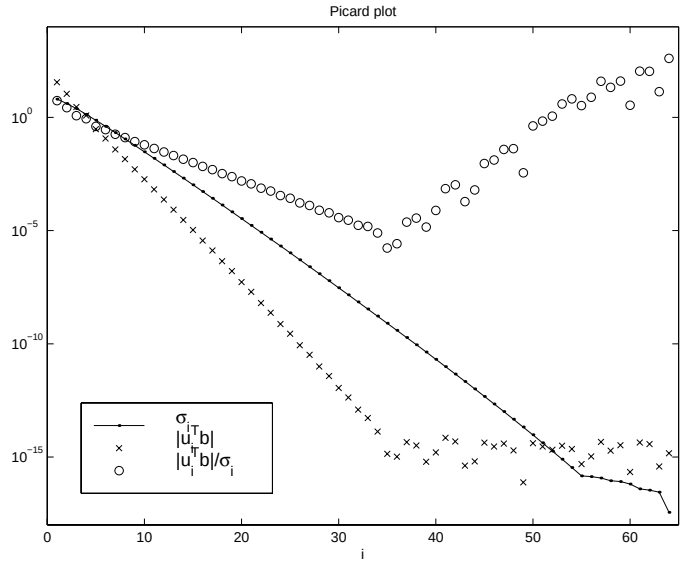


Figure 3.3: The Picard plot for the discretized gravity surveying problem with no noise in the right-hand side.

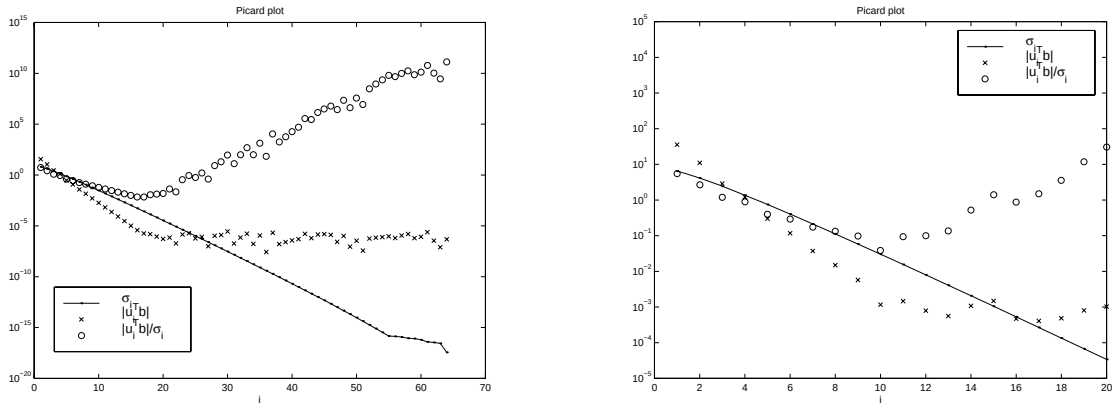


Figure 3.4: The Picard plots for the discretized gravity surveying problem with a noisy right-hand side. The two figures correspond to two different noise levels. The larger the noise, the fewer SVD coefficients $u_i^T b$ remain above the noise level.

influence of rounding errors leads to a different, but still wrong, solution.

Figure 3.4 shows two Picard plots for the gravity problem with two noisy right-hand sides, obtained by adding Gaussian white noise with zero mean to the vector $A\,x$. The standard deviation of the noise is larger in the right plot. Again, we see an initial decay of the SVD coefficients $|u_i^T b|$, until they level off at a plateau determined by the noise. The solution coefficients also decrease initially, but for larger values of the index i they start to increase again. The computed solutions are completely dominated by the SVD coefficients corresponding to the smallest singular values.

This kind of analysis of the SVD coefficients, together with an understanding of their relation to the SVE coefficients, lead to the introduction of a discrete version of the Picard condition. While the solution of a finite-dimensional problem can never become unbounded, its norm can become so large that it is useless in practise. Since the SVD coefficients are so closely related to their corresponding SVE coefficients, it is relevant and necessary to introduce the following condition for the discretized problem.

The Discrete Picard Condition. Let τ denote the level at which the computed singular values σ_i level off due to rounding errors. Then the discrete Picard condition is satisfied if, for all singular values greater than τ, the corresponding coefficients $|u_i^T b|$, on average, decay faster than the σ_i.

We emphasize that it is the *decay* of the σ_i and $|u_i^T b|$ that matters here – not the size of these quantities. A visual inspection of the Picard plot is often enough to verify if the discrete Picard condition is satisfied. One should always ignore the part of the Picard plot that corresponds to tiny singular values at the machine precision level, and if noise is present in the right-hand side then one should also ignore all those coefficients for which $|u_i^T b|$ level off at some noise plateau.

To further illustrate the power of the analysis that can be carried out by means of the Picard plot, we return to Ursell's test problem (2.2). Figure 3.5 shows the Picard plot for a discretization of this problem (computed with $n = 16$). Clearly, the discrete Picard condition is not satisfied – the coefficients $|u_i^T b|$ decay slower than the singular values σ_i for all singular values greater than the machine precision level. This is a very strong indication that the

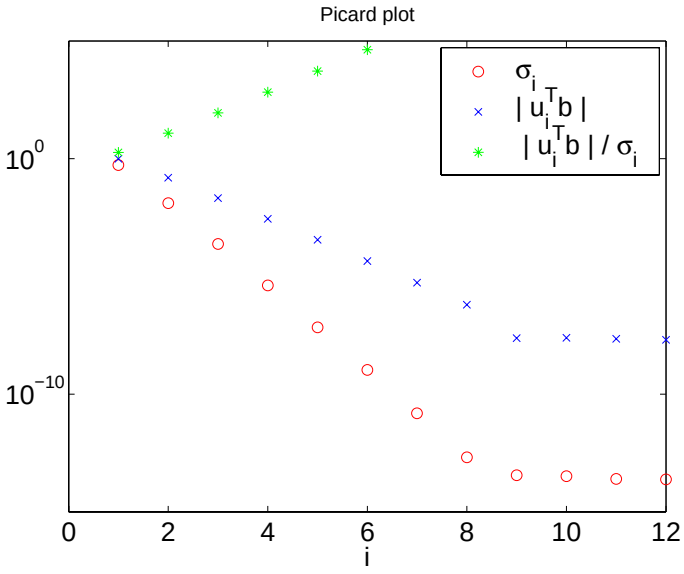


Figure 3.5: The Picard plots for the discretized version of Ursell's model problem (2.2). Clearly, the Picard condition is not satisfied.

underlying continuous problem does not satisfy the Picard condition. And this is, indeed, the case for this problem.

3.3 A Closer Look at Problems with Noisy Data

At this stage it is useful to take a closer look at discretized problems with noisy data. Of course, one can also study how the solution to the underlying integral equation changes when the right-hand side is perturbed, similar to our analysis in §2.1. However, it is easier to carry out the analysis for the discretized problem $A\,x = b$, and this analysis is still relevant for a large class of problems with noisy measured data.

We restrict ourselves to the case where the errors are confined to the right-hand side b. This is clearly an over-simplification; often the matrix A is influenced by model errors or quadrature errors in the evaluation of its coefficients. However, these errors tend to be more systematic in nature than those in the right-hand side, and the analysis is more complicated. But our analysis here still illustrates well the basic problems with the practical solution of ill-posed problems.

Assume therefore that there exists an exact solution x^{exact}, and that there is a corresponding exact right-hand side given by $b^{\text{exact}} = A\,x^{\text{exact}}$. Note that this requirement is not necessarily satisfied if A, b^{exact} and x^{exact} are obtained by discretization of K, g and f, respectively, in the Fredholm integral equation (because of the quadrature and truncation errors); but it is a very convenient assumption for our analysis. Thus, our model for the right-hand side is

$$b = b^{\text{exact}} + e, \qquad \text{where} \qquad b^{\text{exact}} = A\,x^{\text{exact}}$$

and where the vector e represents the noise in the data. This noise typically arises from measurement interference and other errors in the recorded signal, as well as noisy and inaccuracies

in the measuring device.

Throughout most of this presentation, we will assume that the errors are Gaussian white noise, i.e., that the elements of e are uncorrelated with those of b^{exact}, and that they all come from the same Gaussian distribution with zero mean and standard deviation η (we use the non-standard notation η here, instead of the standard symbol σ, to avoid any confusion with the singular values). This is equivalent to saying that the covariance matrix for e is a scaled identity,

$$\text{Cov}(e) \equiv \mathcal{E}(e\, e^T) = \eta^2\, I,$$

where $\mathcal{E}(\cdot)$ denotes the expected value, and η^2 is the variance. In particular, it follows that $\mathcal{E}(e_i^2) = \eta^2$. Since b^{exact} and e are assumed uncorrelated, we have $\mathcal{E}(b) = \mathcal{E}(b^{\text{exact}})$, and it follows that the covariance for the right-hand side b is

$$\text{Cov}(b) \equiv \mathcal{E}\Big((b - \mathcal{E}(b))\,(b - \mathcal{E}(b))^T\Big) = \mathcal{E}(e\, e^T) = \eta^2\, I.$$

Then it follows from standard results in multivariate statistics that the covariance matrix for the solution

$$x = A^{-1}b = A^{-1}(b^{\text{exact}} + e) = x^{\text{exact}} + A^{-1}e$$

is given by

$$\text{Cov}(x) = A^{-1}\text{Cov}(b)\, A^{-T} = \eta^2\, (A^T A)^{-1}.$$

The norm of this matrix is

$$\|\text{Cov}(x)\|_2 = \eta^2 \,/\, \sigma_n^2.$$

Clearly, if A is very ill conditioned, $\text{cond}(A) \gg 1$, then this covariance matrix is likely to have very large elements, indicating that $x = A^{-1}b$ is very sensitive to data errors.

Now let us consider the influence of the noise on the SVD coefficients, i.e., the elements of the vector $U^T b = U^T b^{\text{exact}} + U^T e$ which are given by

$$u_i^T b = u_i^T (b^{\text{exact}} + e) = u_i^T b^{\text{exact}} + u_i^T e, \qquad i = 1, \ldots, n.$$

Using again elementary relations from statistics, we obtain

$$\text{Cov}(U^T e) = U^T \text{Cov}(e)\, U = \eta^2\, I,$$

showing that the transformed noise vector $U^T e$ also behaves as white noise, and therefore $\mathcal{E}((u_i^T e)^2) = \eta^2$. It can be shown that

$$\mathcal{E}(|e_i|) = \mathcal{E}(|u_i^T e|) = \nu_n\, \eta\ , \qquad \text{with} \qquad \nu_n = \frac{\sqrt{2}\,\Gamma(\frac{n+1}{2})}{\Gamma(\frac{n}{2})}.$$

The factor ν_n is close to one, and approaches one as n increases.

We assume now that the exact right-hand side b^{exact} satisfies the discrete Picard condition, i.e., the coefficients $|u_i^T b^{\text{exact}}|$ decay faster than the singular values. We also make the reasonable assumption that the norm of the perturbation e is smaller than that of b^{exact} (otherwise all information from b^{exact} is lost in b).

From these assumptions it follows that some of the coefficients $u_i^T b^{\text{exact}}$ must be larger in magnitude than the "noise coefficients" $u_i^T e$, while others may be smaller. In other words, the noisy coefficients satisfy

$$u_i^T b = u_i^T b^{\text{exact}} + u_i^T e \approx \begin{cases} u_i^T b^{\text{exact}}, & |u_i^T b^{\text{exact}}| > |u_i^T e| \\ u_i^T e, & |u_i^T b^{\text{exact}}| < |u_i^T e|. \end{cases} \tag{3.5}$$

In other words, there are two different kinds of coefficients:

- There are "reliable" coefficients , characterized by $|u_i^T b| \gg \eta$, that mainly carry information about the exact data b^{exact}. Due to the discrete Picard condition, these coefficients correspond to the larger singular values σ_i.
- There are "noisy" SVD components, namely, those for which $|u_i^T b^{\text{exact}}|$ is smaller than η and for which, therefore, $|u_i^T b| \approx \eta$. These coefficients correspond to the smaller singular values σ_i, also due to the discrete Picard condition.

We have thus explained the overall behavior of the Picard plot, as observed in the figures shown in this Chapter.

3.4 Exercises

3.4.1 SVD Analysis of the Problem with Degenerate Kernel

This little exercise illustrates the use of the SVD analysis technique, applied to the problem with a degenerate kernel from Exercise 2.4.2.

The midpoint quadrature rule plus collocation in the quadrature abscissas leads to a discretized problem with a matrix A whose elements are given by

$$a_{ij} = h\left((i + 2j - \tfrac{3}{2})h - 3\right), \qquad \text{with} \qquad h = 2/n.$$

Show that the quadrature abscissas are $t_j = -1 - \frac{1}{2}h + jh$ for $j = 1, \ldots, n$ and derive the above equation for a_{ij}. Compute the SVD numerically to verify that the rank of A is 2.

Since, for this matrix, $A = u_1\,\sigma_1\,v_1^T + u_2\,\sigma_2\,v_2^T$, it follows that $A\,x$ is always a linear combination of u_1 and u_2 for any x. Show this. Then justify (e.g., by plotting the singular vectors u_1 and u_2) that linear combinations of these vectors represent samples of a linear function, in accordance with the results from Exercise 2.4.2.

3.4.2 SVD Analysis of the Gravity Surveying Problem

The purpose of this exercise is to illustrate how a simple inverse problem is discretized by means of the midpoint quadrature rule. The exercise also illustrates that the coefficient matrix for the resulting system of linear equations is very ill conditioned, and that the solution is highly sensitive to errors.

Derive the formula for the matrix elements a_{ij} for a problem with n data points and also n unknowns. Use $s_i = t_i$ for the collocation points, and theck that the matrix A becomes symmetric. Write a small Matlab function that returns the coefficient matrix, given the model parameter d and the discretization parameter n.

Then use the matrix A to compute right-hand sides $b = A\,x$ for two different exact solutions x – one corresponding to a smooth solution, and one corresponding to a solution with one or more discontinuities. Make up your own x vectors. Notice that the right-hand side is always smooth. Study experimentally how the right-hand side varies with the depth d.

Compute the condition number of A, and examine how it varies with n for a fixed value of the depth d. Then keep n fixed, and study how the condition number varies with d; try to explain the observed behavior.

Finally, try to solve the problem by computing the "naive" solution $x = A^{-1}b$. First solve the problems with a noise-free right-hand side; then try again with right-hand sides in which a small amount of Gaussian white noise has been added.

3.4.3 Convergence of the SVE Approximations

This exercise illustrates the convergence properties of the approximate SVE quantities, obtained from the discretization of the second derivative test problem in (2.5) for which the SVE is known analytically.

Familiarize yourself with the function `deriv2` from Regularization Tools, which implements the Galerkin method with the orthonormal "top-hat" ⎍ basis functions. Then compute the SVD for increasing values of n; for example, use $n = 8$, 16, 32, 64, etc. Compare the computed singular values with the exact ones, and discuss their convergence as n increases.

Compare the approximations $\tilde{u}_1(s)$ and $\tilde{v}_1(t)$ to the principal left and right singular function with the exact functions; don't forget the scaling $h^{-1/2}$. Discuss your choice of comparison of the functions.

3.4.4 SVD Analysis of a 1-D Image Reconstruction Problem

The purpose of this exercise is to illustrate how the SVE can be used to analyze the smoothing effects of a first-kind Fredholm integral equation. In this example, we use the 1-D image reconstruction test problem, which is implemented in Regularization Tools as function `shaw`. The kernel in this problem is given by

$$K(s,t) = \Big(\cos(s) + \cos(t)\Big)\, \frac{\sin\big(\,\pi(\sin(s) + \sin(t))\,\big)}{\pi(\sin(s) + \sin(t))}, \qquad -\pi/2 \le s, t \le \pi/2,$$

while the solution is

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

This integral equation models a situation where light passes through an infinitely long slit, and the function f is the light intensity as a function of the incidence angle t. The problem is discretized by means of the midpoint quadrature rule to produce A and x^{exact}, after which the exact right-hand side is computed as $b^{\text{exact}} = A\, x^{\text{exact}}$.

Choose $n = 24$ and generate the problem. Then compute the SVD of A, and plot and inspect some of the left and right singular vectors. What can be said about the number of sign changes?

Use the function `picard` to inspect the singular values σ_i and the SVD coefficients $u_i^T b^{\text{exact}}$ of the exact solution b^{exact}, as well as the corresponding solution coefficients $u_i^T b^{\text{exact}}/\sigma_i$. Is the Picard condition satisfied?

Add a very small amount of noise e to the right-hand side b^{exact}, with $\|e\|_2/\|b^{\text{exact}}\|_2 = 10^{-10}$. Inspect the singular values and SVD coefficients again. What happens to the SVD coefficients corresponding to the small singular values?

Recall that the undesired solution $x = A^{-1}b$ can be written as

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

Compute the partial sums

$$x_k = \sum_{i=1}^{k} \frac{u_i^T b}{\sigma_i}\, v_i$$

for $k = 1, 2, \ldots$ and inspect the vectors x_k. Explain the behavior of these vectors.

3.4.5 SVD Analysis of a Problem in Potential Theory

This exercise illustrate the use of the SVD in the analysis of a linear problem which, for a certain parameter, has a non-unique solution. The problem comes from 2-D potential theory, where Green's function is given by $G(r,q) = \ln(\|r-q\|_2)$ with $r, q \in \mathbb{R}^2$. A special feature of these 2-D problems is the existence of certain "critical contours" Γ^* on which there exists a nonzero charge distribution $\rho(r)$ that produces a zero field.

The field outside the contour Γ at the point $\mathbf{q}$ is given by

$$\phi(\mathbf{q}) = \int_\Gamma G(\mathbf{r},\mathbf{q})\,\rho(\mathbf{r})\,d\omega$$

where $\mathbf{r}$ is a point on Γ, and ω is the arch length along Γ. We are looking for a critical contour Γ^* such that

$$\int_{\Gamma^*} G(\mathbf{r},\mathbf{q})\,\rho(\mathbf{r})\,d\omega = 0.$$

Here, we restrict ourselves to line segments of length ℓ, and we assume that we discretize the problem at n equidistant points on the line segment. Then the corresponding matrix A has entries for $i, j = 1, \ldots, n$ given by

$$a_{ij} = (t_{j+1} - s_i)\,\ln(|t_{j+1} - s_i|) - (t_j - s_i)\,\ln(|t_j - s_i|) - (t_{j+1} - t_j),$$

where

$$s_i = (i - 0.5)\,\ell/n, \qquad i = 1, \ldots, n$$

and

$$t_j = (j-1)\,\ell/n, \qquad j = 1, \ldots, n+1.$$

We are merely interested in solutions to the homogeneous problem $A\,x = 0$, and therefore we do not need a particular right-hand side. All that is needed is the SVD of the matrix A.

Write a Matlab script or function that computes the matrix A, given values of n and ℓ, using the above equations. Then choose a small value of n, e.g., $n = 16$, and for a range of lengths ℓ compute the singular values of A and find the critical length ℓ^* for which the homogeneous problem has a solution. The critical length should be slightly larger than 4.

At the critical length, compute the solution to the homogeneous problem, using the SVD of the matrix A.

Chapter 4

Computational Aspects: Regularization Methods

After the careful analysis that we carried out in the previous two chapters, we now have a good understanding of the difficulties associated with discrete ill-posed problems and why the naive solution is pretty useless. In this chapter we use this insight to device different regularization methods for computing approximate solutions that are less sensitive to perturbations than the naive solution.

These methods are called regularization methods because they enforce regularity on the computed solution – typically in the form of a requirement that the solution is smooth, in some sense. And by enforcing this regularity, or smoothness, we suppress some of the unwanted noise components, leading to more stable approximate solutions.

From this stage we relax the condition that the coefficient matrix A is square to the situation where it is also allowed to have more rows than columns, i.e., from now on

$$A \in \mathbb{R}^{m \times n} \qquad \text{with} \qquad m \geq n.$$

When $m > n$ it is natural to consider the least squares problem $\min_x \|A\,x - b\|_2$. In this situation the phrase "naive solution" means the least squares solution, and we note that this solution has precisely the same SVD expansion as when A is square. Hence, there is no need to distinguish between the cases $m = n$ and $m > n$ throughout.

4.1 Truncated SVD

From the analysis in the two previous chapters, it is clear that the extremely large errors in the naive solution come from the noisy SVD components associated with the smaller singular values. In particular, the results in (3.5) show that the naive solution is dominated by SVD coefficients of the form $u_i^T b/\sigma_i \approx u_i^T e/\sigma_i$ (where e is the perturbation of b) corresponding to the smaller singular values; see Fig. 3.4.

The good news is that our analysis also reveals that some of the SVD coefficients are rather trustworthy, namely, the coefficients of the form $u_i^T b/\sigma_i \approx u_i^T b^{\text{exact}}/\sigma_i$ (in which $b^{\text{exact}} = A\,x^{\text{exact}}$ is the exact right-hand side) corresponding to the larger singular values; again, see Fig. 3.4. This gives us hope that we can actually recover some of the information about the solution to the problem.

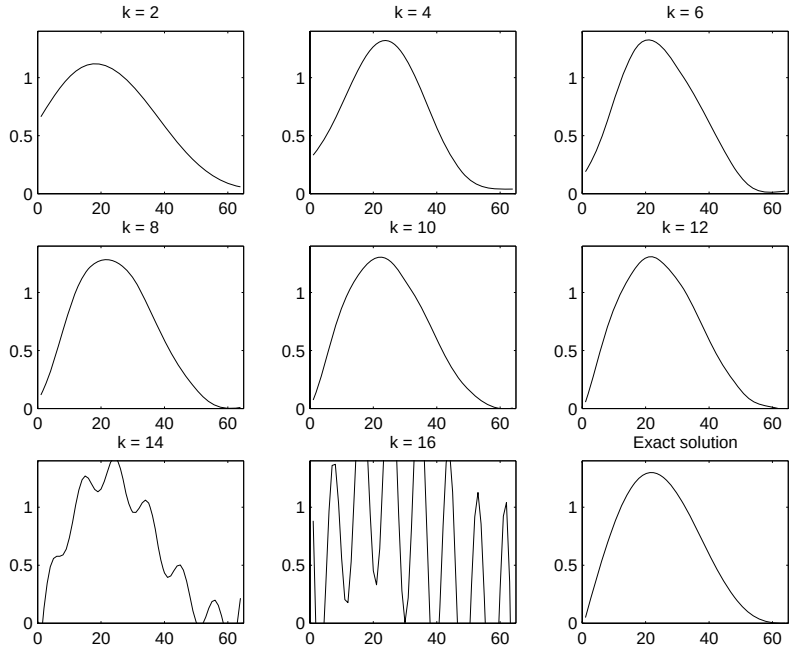


Figure 4.1: Truncated SVD (TSVD) solutions x_k to the gravity surveying problem, for 8 different values of k. The exact solution is shown in the bottom right corner. A good approximation is achieved for $k = 12$, while the noise is clearly visible in x_k for $k = 14$ and $k = 16$.

Another piece of good news is that we can assume that the exact right-hand side satisfies the discrete Picard condition (otherwise there is no point in trying to solve the discrete ill-posed problem). As a consequence, the SVD components of the exact solution with largest magnitude are precisely those coefficients that are approximated reasonably well, because for the small indices i we have $u_i^T b/\sigma_i \approx u_i^T b^{\text{exact}}/\sigma_i = v_i^T x^{\text{exact}}$.

These considerations immediately lead to a "brute force" method for computing regularized approximate solutions: simply chop off those SVD components that are dominated by the noise. Hence, we define the *truncated SVD* (TSVD) solution x_k as the solution obtained by retaining the first k components of the naive solution:

$$x_k \equiv \sum_{i=1}^{k} \frac{u_i^T b}{\sigma_i} v_i \ . \tag{4.1}$$

The *truncation parameter* k should be chosen such that all the noise-dominated SVD coefficients are discarded. A suitable value of k can often be found from an inspection of the Picard plot; for example, for the two noise problems in Fig. 3.4 we would choose $k = 16$ and $k = 9$, respectively.

There is an alternative formulation of the TSVD method which is important. While the definition of x_k in (4.1) takes its bases in a specific formula for computing the regularized solution, we can also define a regularized solution as the solution to a modified and better conditioned problem. Let us introduce the TSVD matrix A_k, which is the rank-k matrix

defined as

$$A_k = \begin{pmatrix} | & & | \\ u_1 & \cdots & u_k \\ | & & | \end{pmatrix} \begin{pmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_k \end{pmatrix} \begin{pmatrix} | & & | \\ v_1 & \cdots & v_k \\ | & & | \end{pmatrix}^T = \sum_{i=1}^{k} u_i \, \sigma_i \, v_i^T . \qquad (4.2)$$

It can be shown that the condition number of this matrix is $\mathrm{cond}(A_k) = \sigma_1/\sigma_k$, and it is typically much smaller than the condition number $\mathrm{cond}(A) = \sigma_1/\sigma_n$ of the original matrix A.

Hence, it seems like a good idea to replace the original and ill-conditioned problem $A\,x = b$ with the better conditioned least-squares problem $\min \|A_k\,x - b\|_2$. The least-squares formulation is needed, because we can no longer expect to find an x such that $A_k\,x = b$. However, since A_k is rank deficient (as long as $k < n$), there is not a unique solution to this least-squares problem; it is easy to show that the general solution has the form

$$x = \sum_{i=1}^{k} \frac{u_i^T b}{\sigma_i} \, v_i + \sum_{i=k+1}^{n} \zeta_i \, v_i \ , \qquad \zeta_i = \text{arbitrary.} \qquad (4.3)$$

To define a unique solution, we must supplement the least-squares problem with an additional constraint on the solution x. A natural constraint, in many applications, is that we seek the solution with minimum 2-norm:

$$\min \|x\|_2 \qquad \text{subject to} \qquad \|A_k\,x - b\|_2 = \min . \qquad (4.4)$$

It is easy to show that the solution to this constrained problem is precisely the TSVD solution x_k in (4.1). Hence, Eq. (4.4) is an alternative definition of the TSVD solution, in which the regularity requirement on x – in the form of the minimization of its norm $\|x\|_2$ – is explicit.

From the above definition of x_k it follows that we can also write the TSVD solution in the form

$$x_k = \begin{pmatrix} | & & | \\ v_1 & \cdots & v_k \\ | & & | \end{pmatrix} \begin{pmatrix} \sigma_1^{-1} & & \\ & \ddots & \\ & & \sigma_k^{-1} \end{pmatrix} \begin{pmatrix} | & & | \\ u_1 & \cdots & u_k \\ | & & | \end{pmatrix}^T b \ ,$$

showing that there exists a matrix[1] $A_k^\dagger = \sum_{i=1}^{k} v_i \, \sigma_i^{-1} u_i^T$ such that we can write $x_k = A_k^\dagger b$. As a consequence, it follows that if e is Gaussian white noise with $\mathrm{Cov}(e) = \eta^2 I$, then the covariance for the TSVD solution is

$$\mathrm{Cov}(x_k) = A_k^\dagger \, \mathrm{Cov}(b) \, (A_k^\dagger)^T = \eta^2 \sum_{i=1}^{k} \sigma_i^{-2} \, v_i \, v_i^T \ .$$

The norm of this matrix is

$$\|\mathrm{Cov}(x_k)\|_2 = \eta^2/\sigma_k^2$$

and since σ_k is always greater – and often much greater – than σ_n we conclude that the elements of $\mathrm{Cov}(x_k)$ are generally smaller – and often much smaller – than those of the covariance matrix $\mathrm{Cov}(x)$ for the naive solution.

[1]The matrix $A_k^\dagger$ is the Moore-Penrose generalized inverse, or pseudo-inverse, of the TSVD matrix A_k defined in (4.2).

The price we pay for this reduction of the variance in x_k, compared to the naive solution $x = A^{-1}b$, is bias. While $A^{-1}b$ is unbiased, i.e., the expected value is the exact solution, $\mathcal{E}(A^{-1}b) = x^{\text{exact}}$, the TSVD solution x_k has a nonzero bias:

$$\mathcal{E}(x_k) = \sum_{i=1}^{k} (v_i^T x^{\text{exact}})\, v_i = x^{\text{exact}} - \sum_{i=k+1}^{n} (v_i^T x^{\text{exact}})\, v_i \ .$$

However, due the the discrete Picard condition, the norm of the bias term,

$$\left\| \sum_{i=k+1}^{n} (v_i^T x^{\text{exact}})\, v_i \right\|_2 = \left(\sum_{i=k+1}^{n} (v_i^T x^{\text{exact}})^2 \right)^{1/2} ,$$

is typically small, compared to $\|x^{\text{exact}}\|_2 = \left(\sum_{i=1}^{n} (v_i^T x^{\text{exact}})^2\right)^{1/2}$.

A somewhat common, but unfortunate, mistake in connection with TSVD is to base the criterion for choosing the truncation parameter k on the size of the singular values σ_i. As long as the noise is restricted to the right-hand side, the truncation parameter k must define the break-point between the retained and discarded (filtered) SVD coefficients, and therefore the choice of k is determined by the behavior of the noisy coefficients $u_i^T b$. The confusion seems to arise because the numerical rank of a noisy matrix is related to the the singular values as well as the norm of the perturbation matrix; see, e.g., [7] for details about these issues.

4.2 Tikhonov Regularization

The advantage of the TSVD method is that it is intuitive, and it is easy to compute TSVD solutions x_k for different truncation parameters, once the SVD has been computed. The disadvantage is that it explicitly requires the computation of the SVD – or, at least, the principal k singular values and vectors. This computational task can be too overwhelming for large-scale problems, and therefore there is a need for other regularization methods that are better suited for large computational problems.

Probably the most successful regularization method of all times is *Tikhonov regularization*. This method has been invented several times, in several incarnations; but Tikhonov's name is correctly associated with the method because of his ground-breaking work in connection with this method. Similarly with the formulation of TSVD in (4.4), Tikhonov's method explicitly incorporates the regularity requirement in the formulation of the problem. Specifically, the Tikhonov solution x_λ is defined as the solution to the problem

$$\min_x \left\{ \|A\,x - b\|_2^2 + \lambda^2 \|x\|_2^2 \right\} . \tag{4.5}$$

Here, the *regularization parameter* λ is a positive parameter that controls the weighting between the two ingredients of the criterion function.

- The first term $\|A\,x - b\|_2^2$ measures the goodness-of-fit, i.e., how well the solution x predicts the given data b. Obviously, if this term is too large, then x cannot be considered as a good solution because it does not "solve the problem." On the other hand, intuitively we should not make the residual smaller than the average size of the errors in b; similar to data-fitting problems, we do not want to fit the noise in the data.

- The second term, the solution norm $\|x\|_2^2$, measures the regularity of the solution. The incorporation of this term is based on our knowledge that the naive solution is dominated by high-frequency components with large amplitudes, and the hope is therefore that if we control the norm of x then we can suppress (some of) the large noise components.

Hence, if we can find a good balance between these to terms, via a suitable value of λ, then the hope is that we achieve a regularized solution that approximates the exact solution.

While it may not be immediately clear from the formulation in (4.5), this is a linear least-squares problem in x. But if we use the fact that

$$\left\| \begin{pmatrix} y \\ z \end{pmatrix} \right\|_2^2 = \begin{pmatrix} y \\ z \end{pmatrix}^T \begin{pmatrix} y \\ z \end{pmatrix} = y^T y + z^T z = \|y\|_2^2 + \|z\|_2^2 \ ,$$

then it follows immediately that the Tikhonov problem can be reformulated as

$$\min \left\| \begin{pmatrix} A \\ \lambda\, I \end{pmatrix} x - \begin{pmatrix} b \\ 0 \end{pmatrix} \right\|_2 , \tag{4.6}$$

which is clearly a linear least-squares problem. The normal equations for this problem take the form

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

or simply

$$(A^T A + \lambda^2\, I)\, x = A^T b \qquad \Leftrightarrow \qquad x_\lambda = (A^T A + \lambda^2\, I)^{-1}\, A^T b \ ,$$

the latter of which is also frequently found in the literature. However, for reasons of both computational efficiency and numerical stability, algorithms for computing Tikhonov solutions should be based on the formulation in (4.6).

We can use the SVD to obtain more insight into the Tikhonov solution x_λ. When we insert the SVD of A into the normal equations, and use that $I = V\, V^T$, then we obtain

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

which leads to the following expression

$$x_\lambda = V\, (\Sigma^2 + \lambda^2\, I)^{-1} \Sigma\, U^T b.$$

If the insert the singular values and vectors, we get

$$x_\lambda = \sum_{i=1}^{n} f_i\, \frac{u_i^T b}{\sigma_i}\, v_i \ , \tag{4.7}$$

where we have introduced the *filter factors* f_i for $i = 1, \ldots, n$, which satisfy

$$f_i = \frac{\sigma_i^2}{\sigma_i^2 + \lambda^2} \approx \begin{cases} 1 \ , & \sigma_i \gg \lambda \\ \sigma_i^2/\lambda^2 \ , & \sigma_i \ll \lambda. \end{cases}$$

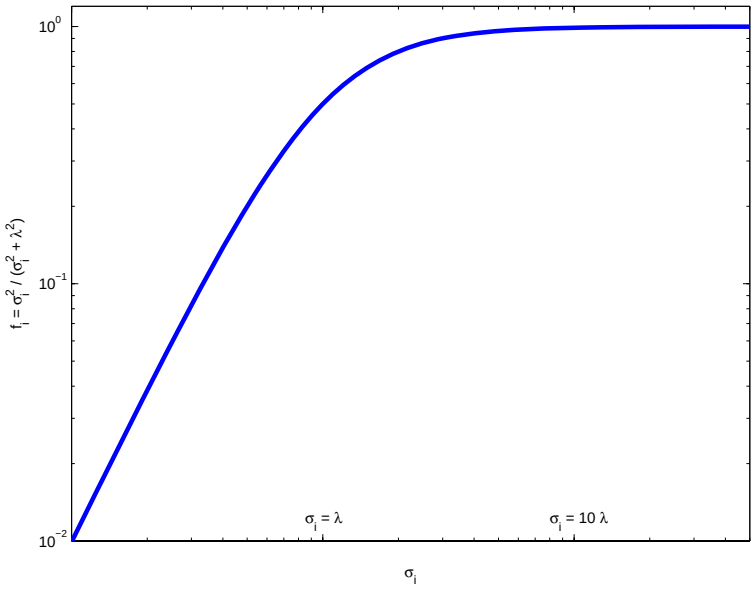


Figure 4.2: The filter factors $f_i = \sigma_i^2/(\sigma_i^2 + \lambda^2)$ for Tikhonov regularization.

The behavior of these filter factors is illustrated in Fig. 4.2. We see that for singular values σ_i greater than the parameter λ, the filter factors are close to one and the corresponding SVD components contribute to x_λ with almost full strength. On the other hand, for singular values much smaller than λ the filter factors are small, and therefore these SVD components are damped or filtered. We note that in the latter case, the filter factors f_i are proportional to σ_i^2 and therefore they decay fast enough to "kill" the increasing factors $u_i^T b/\sigma_i$.

The conclusion of this SVD analysis is that the Tikhonov solution is a filtered solution, in much the same manner as the TSVD solution. The transition in the filter factors f_i sets in for singular values whose size is comparable to λ, and in this way we can use λ to control the filtering (similar to the use of the truncation parameter k in TSVD); see the example in Fig. 4.3 and compare with the TSVD solutions in Fig. 4.1. However, the transition between retained and filtered SVD components is smoother, and the filtering can be achieved without explicit computation of the SVD – all that is needed is to solve the least-squares problem in (4.6).

Again, there are alternative formulations of the problem which can give insight and/or be used to device algorithms for certain classes of problems. For example, we can specify an upper bound δ for the norm of the solution, and solve the constrained least squares problem

$$\min \|A\,x - b\|_2^2 \qquad \text{subject to} \qquad \|x\|_2^2 \leq \delta^2.$$

The constraint is active whenever δ is smaller than the norm $\|A^{-1}b\|_2$ of the naive solution, which should always be the case if we want to regularize the solution. From optimization theory, we know that we can incorporate the constraint via a Lagrange multiplier γ, and the problem becomes

$$\min_x \|A\,x - b\|_2^2 + \gamma\,(\|x\|_2^2 - \delta^2)\ .$$

This problem is clearly equivalent to the Tikhonov problem. Similarly, we can minimize the norm of the solution (again, to suppress high-frequency components with large amplitude),

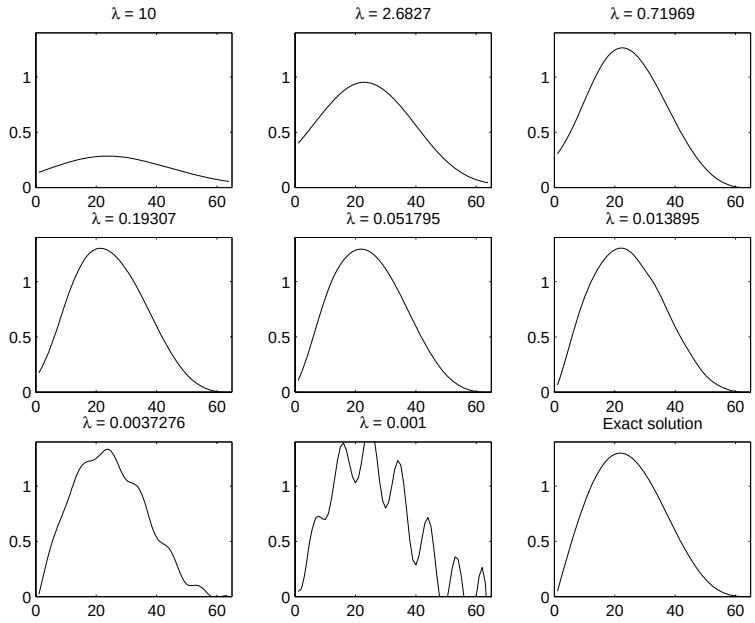


Figure 4.3: Tikhonov solutions x_λ to the gravity surveying problem, for eight different values of the regularization parameter λ. The exact solution is shown in the bottom right figure. The values $\lambda = 0.0037$ and $\lambda = 0.001$ are clearly too small, producing solutions with too many noisy SVD components.

subject to the constraint that the residual norm is smaller than some upper bound ϵ:

$$\min_x \|x\|_2^2 \qquad \text{subject to} \qquad \|A\,x - b\|_2^2 \leq \epsilon^2.$$

Again, the Lagrange multiplier formulation of this constrained problem leads to the Tikhonov formulation.

Analogous with the TSVD solution, let us take a brief look at the statistical aspects of the Tikhonov solution. Via the relation $x_\lambda = (A^T A + \lambda^2 I)^{-1} A^T b$, and assuming again that $\mathrm{Cov}(e) = \eta^2 I$, we can show that the covariance matrix for the Tikhonov solution is

$$\mathrm{Cov}(x_\lambda) = \eta^2 \sum_{i=1}^{n} f_i^2\, \sigma_i^{-2}\, v_i\, v_i^T,$$

and its norm is bounded as

$$\|\mathrm{Cov}(x_\lambda)\|_2 \leq \frac{\eta^2}{(2\,\lambda)^2}\ .$$

Obviously, if λ is chosen somewhat larger than the smallest singular values σ_n, then we obtain a reduction in the variance for the Tikhonov solution, compared to the naive solution. Again, we achieve this at the expense of introducing a bias in the solution:

$$\mathcal{E}(x_\lambda) = \sum_{i=1}^{n} f_i\, (v_i^T x^{\mathrm{exact}})\, v_i = x^{\mathrm{exact}} - \sum_{i=1}^{n} (1 - f_i)\, (v_i^T x^{\mathrm{exact}})\, v_i\ .$$

Since $1 - f_i = \lambda^2/(\sigma_i^2 + \lambda^2)$, and since we assume that the discrete Picard condition is satisfied, we can visually expect that the bias term is small compared to $\|x^{\mathrm{exact}}\|_2$.

4.3 Perturbation Theory

Regularization methods are designed to filter the influence from the noise, such that the solution is less dominated by the inverted noise – provided we can choose the regularization parameter (k or λ) properly. The question is then: how sensitive is the regularized solution to the perturbations, and how does the sensitivity depend on the regularization parameter?

In order to study this sensitivity, throughout this section we will be studying two related problem, namely,

$$A\,x = b \qquad \text{and} \qquad \widetilde{A}\,\widetilde{x} = \widetilde{b}$$

where $\widetilde{A}$ and $\widetilde{b}$ are perturbed versions of A and b,

$$\widetilde{A} = A + \Delta A \qquad \text{and} \qquad \widetilde{b} = b + \Delta b,$$

i.e., the matrix ΔA is the perturbation of A, and the vector Δb is the perturbation of the right-hand side b. Think of the two problems $A\,x = b$ and $\widetilde{A}\,\widetilde{x} = \widetilde{b}$ as two different noisy realizations of the underlying problem with exact data. The vectors x and $\widetilde{x}$ are the solutions to the two problems, and we are interested in bounding the difference $\widetilde{x} - x$. More precisely, we are interested in upper bounds for the relative norm-wise difference $\|\widetilde{x} - x\|_2 / \|x\|_2$, which is independent on the scaling of A, b and x.

Naive solutions. The perturbation bounds for the naive solutions $x = A^{-1}b$ and $\widetilde{x} = \widetilde{A}^{-1}\widetilde{b}$ (when A is square and invertible) can be found in many text books on numerical analysis, and it take the following form. If the perturbation matrix ΔA satisfies $\|\Delta A\|_2 < \sigma_n$ then

$$\frac{\|\widetilde{x} - x\|_2}{\|x\|_2} \leq \frac{\mathrm{cond}(A)}{1 - \tau} \left(\frac{\|\Delta b\|_2}{\|b\|_2} + \frac{\|\Delta A\|_2}{\|A\|_2} \right)$$

where

$$\tau = \|\Delta A\|_2 \, \|A^{-1}\|_2 = \frac{\|\Delta A\|_2}{\sigma_n} \; .$$

The requirement $\tau < 1$ is necessary for ensuring that the perturbed matrix $\widetilde{A}$ stays nonsingular. The perturbation is governed by A's condition number $\mathrm{cond}(A) = \sigma_1/\sigma_n$.

TSVD solutions. Let x_k and $\widetilde{x}_k$ denote the TSVD solutions for the same truncation parameter k, and assume that the perturbation matrix ΔA satisfies $\|\Delta A\|_2 < \sigma_k - \sigma_{k+1}$; then

$$\frac{\|\widetilde{x}_k - x_k\|_2}{\|x_k\|_2} \leq \frac{\kappa_k}{1 - \tau_k} \left(\frac{\|\Delta b\|_2}{\|b_k\|_2} + \frac{\|\Delta A\|_2}{\|A\|_2} + \frac{\tau_k}{1 - \tau_k - \gamma_k} \frac{\|b - b_k\|_2}{\|b_k\|_2} \right)$$

where $b_k = A\,x_k$ and

$$\kappa_k = \mathrm{cond}(A_k) = \frac{\sigma_1}{\sigma_k} = \frac{\|A\|_2}{\sigma_k} \; , \qquad \tau_k = \frac{\|\Delta A\|_2}{\sigma_k} \; , \qquad \gamma_k = \frac{\sigma_{k+1}}{\sigma_k} \; .$$

This results shows that the condition number for the TSVD solution is $\kappa_k = \sigma_1/\sigma_k$, which can be much smaller than the condition number for A. Notice that we obtain the classic result for the naive solution by setting $k = n$ (in which case $b_k = b$ and the last term vanishes).

Tikhonov solutions. Let x_λ and $\widetilde{x}_\lambda$ denote the Tikhonov solutions for the same regularization parameter λ, and assume that the perturbation matrix ΔA satisfies $\|\Delta A\|_2 < \lambda$; then

$$\frac{\|\widetilde{x}_\lambda - x_\lambda\|_2}{\|x_\lambda\|_2} \leq \frac{\kappa_\lambda}{1 - \tau_\lambda} \left(\frac{\|\Delta b\|_2}{\|b_\lambda\|_2} + 2\,\frac{\|\Delta A\|_2}{\|A\|_2} + \tau_\lambda \frac{\|b - b_\lambda\|_2}{\|b_\lambda\|_2} \right)$$

where $b_\lambda = A\,x_\lambda$ and

$$\kappa_\lambda = \frac{\sigma_1}{\lambda} = \frac{\|A\|_2}{\lambda}\ , \qquad\qquad \tau_\lambda = \frac{\|\Delta A\|_2}{\lambda}\ .$$

This results shows that the condition number for the Tikhonov solution is $\kappa_\lambda = \sigma_1/\lambda$, and – similar to the TSVD condition number – it can be much smaller than the condition number for A.

The perturbation results for the TSVD and Tikhonov solutions are, to no big surprise, quite similar. In both cases, the condition number is governed by the choice of the regularization parameter, k or λ, and the more filtering the smaller the condition number. And in both cases, the size of the residual, $b - b_k$ or $b - b_\lambda$ enters the perturbation bound. Also, in both cases, we assume an upper bound on the norm of the perturbation matrix E; this bound ensures that the SVD components of the perturbed and the unperturbed problems are related.

The main difference between the two perturbation bounds is the factor $\omega_k = \sigma_k/\sigma_{k+1}$, which measures the gap between the smallest retained singular value and the largest discarded one, and which only appears in the TSVD bound. If ω_k is close to one, then η_k is small, and only a very small perturbation of the matrix is allowed. Hence, one should refrain from truncating in the middle of a cluster[2] of singular values. The same difficulty does not apply to the Tikhonov solution; if one chooses a value of λ inside a cluster of singular values, then all these SVD components are included in the Tikhonov solution because all the associated filter factors $f_i = \sigma_i^2/(\sigma_i^2 + \lambda^2)$ are of the same size.

In addition to studying the errors in the solutions, as we have done in the above results, it may be of interest to study the "prediction errors," i.e., the sensitivity of the vectors $A\,x_k$ and $A\,x_\lambda$. Under the same assumption as before, we have for the TSVD solution

$$\frac{\|\widetilde{A}\,\widetilde{x}_k - A\,x_k\|_2}{\|b\|_2} \le \frac{\tau_k}{1 - \tau_k - \gamma_k} + \|\Delta b\|_2$$

The interesting result here is that the predictions errors do not depend on the condition number – a result which is well known for the naive solution.

We finish this section with a small example that illustrates the sensitivity of the Tikhonov solution x_λ to perturbations of b, as a function of the regularization parameter λ. The matrix, the unperturbed right-hand side and the unperturbed solution are are

$$A = \begin{pmatrix} 0.41 & 1.00 \\ -0.15 & 0.06 \end{pmatrix}, \qquad b = \begin{pmatrix} 1.41 \\ -0.09 \end{pmatrix}, \qquad x = \begin{pmatrix} 1.00 \\ 1.00 \end{pmatrix}.$$

We generated 25 random perturbations $\widetilde{b} = b + \Delta b$ with the perturbation scaled such that $\|\Delta b\|_2/\|b\|_2 = 0.15$, and for each perturbed problem we computed the Tikhonov solutions corresponding to four values of λ. The results are shown in Fig. 4.4, where the black cross indicates the unperturbed solution, while the red dots represent the perturbed solutions. Also included in the plots are the ellipsoidal level curves for the Tikhonov (least squares) problem for the particular λ.

We see that as λ increases, and more regularization (or filtering) is imposed on the problem, the less sensitive the Tikhonov solution is to the perturbations. As λ increases, the

[2]Here, a *cluster* is a set of singular values which are very close and well separated from the others.

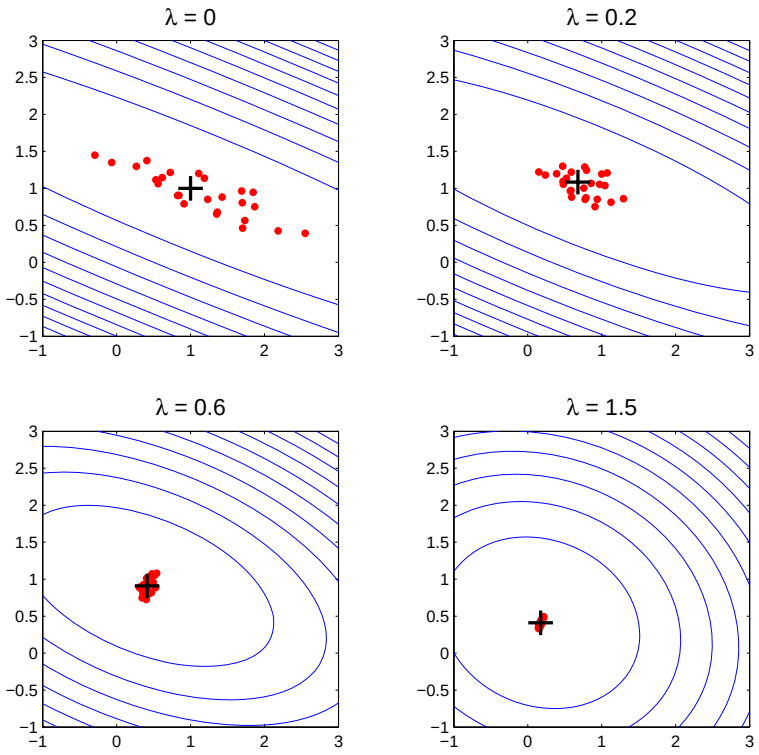


Figure 4.4: Illustration of the sensitivity of the Tikhonov solutions to perturbations of the right-hand side, for a small 2×2 test problem. For each of 25 random perturbations we computed x_λ for four different values of λ, as shown in the top of the plots. The ellipsoidal level curves for the associated Tikhonov problem are shown in blue.

condition number becomes smaller and the level curves become less elliptic. Note that for small values where the ellipsoids are more elongated, the perturbation is mainly in the direction of the longer semi-axis. This direction is defined by the right singular vector v_2 corresponding to the smaller singular value σ_2, and so the observed results agree with the theory, namely, that the largest perturbations occur in the directions corresponding to the smallest singular values.

4.4 The Role of the Discrete Picard Condition

We have already explained that satisfaction of the discrete Picard condition is crucial for the existence of a meaningful solution to discrete ill-posed problems. In this section we take another look at this condition and the role it plays in the computation of regularized solutions.

Throughout this section we will study a model problem in which the distribution of singular values is "arbitrary" (still assuming that they decay gradually to zero), and where the SVD coefficients of the exact right-hand side are given by the model

$$u_i^T b^{\text{exact}} = \sigma_i^\alpha \ , \quad i = 1, \ldots, n \tag{4.8}$$

in which $\alpha \geq 0$ is a model parameter that controls the decay of these coefficients, relative to the decay of the singular values.

- The larger the α, the faster the decay of the SVD coefficients.
- For $\alpha > 1$ the coefficients decay *faster* than the singular values, and the discrete Picard condition is satisfied.

While this is, indeed, a crude model it reflects the overall behavior often found in real problems.

Recall from the previous sections that we can always write the TSVD solutions as $x_k = A_k^\dagger b$, where $A_k^\dagger = \sum_{i=1}^k v_i \, \sigma_i^{-1} u_i^T$. Inserting $b = b^{\text{exact}} + e$, we obtain

$$x_k = A_k^\dagger \, b^{\text{exact}} + A_k^\dagger \, e \ .$$

Similarly, we can always write the Tikhonov solution as $x_\lambda = (A^T A + \lambda^2 I)^{-1} A^T b$, and therefore we have

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

In both methods, we want to choose the regularization parameter such that the second term, the inverted and filtered noise, is sufficiently suppressed. At the same time, we do not want to impose too much filtering, because we want the first term, either $A_k^\dagger \, b^{\text{exact}}$ or $(A^T A + \lambda^2 I)^{-1} A^T \, b^{\text{exact}}$, to be as close to x^{exact} as possible. This is at the heart of the parameter-choice methods for regularization algorithms, and we return to this important issue in the next chapter.

Our aim here is to take a closer look at the differences between the exact solution x^{exact} and the first terms $A_k^\dagger \, b^{\text{exact}}$ and $(A^T A + \lambda^2 I)^{-1} A^T \, b^{\text{exact}}$. These differences are called the *regularization errors* and they are given by

$$\begin{aligned} \varepsilon_k &\equiv x^{\text{exact}} - A_k^\dagger \, b^{\text{exact}} = \sum_{i=k+1}^{n} (v_i^T x^{\text{exact}}) \, v_i \\ \varepsilon_\lambda &\equiv x^{\text{exact}} - (A^T A + \lambda^2 I)^{-1} A^T \, b^{\text{exact}} = \sum_{i=1}^{n} \frac{\lambda^2}{\sigma_i^2 + \lambda^2} \, (v_i^T x^{\text{exact}}) \, v_i. \end{aligned}$$

Note that the regularization errors defined here are identical to the bias terms, as discussed previously.

Assume now that the exact right-hand side satisfies the model (4.8). Then the regularization errors (or bias terms) for TSVD and Tikhonov are bounded as follows.

$$\frac{1}{\sqrt{n}} \, \frac{\|\varepsilon_k\|_2}{\|x^{\text{exact}}\|_2} \ \leq \ \begin{cases} 1 & 0 \leq \alpha < 1 \\ \left(\frac{\sigma_k}{\sigma_1}\right)^{\alpha - 1} & 1 \leq \alpha \end{cases}$$

$$\frac{1}{\sqrt{n}} \, \frac{\|\varepsilon_\lambda\|_2}{\|x^{\text{exact}}\|_2} \ \leq \ \begin{cases} 1 & 0 \leq \alpha < 1 \\ \left(\frac{\lambda}{\sigma_1}\right)^{\alpha - 1} & 1 \leq \alpha < 3 \\ \left(\frac{\lambda}{\sigma_1}\right)^{2} & 3 \leq \alpha \end{cases}$$

We conclude that if the discrete Picard condition is not satisfied, i.e., if $\alpha \leq 1$, then the upper bounds are 1 and there is no guarantee that the regularization errors (or bias terms) are small.

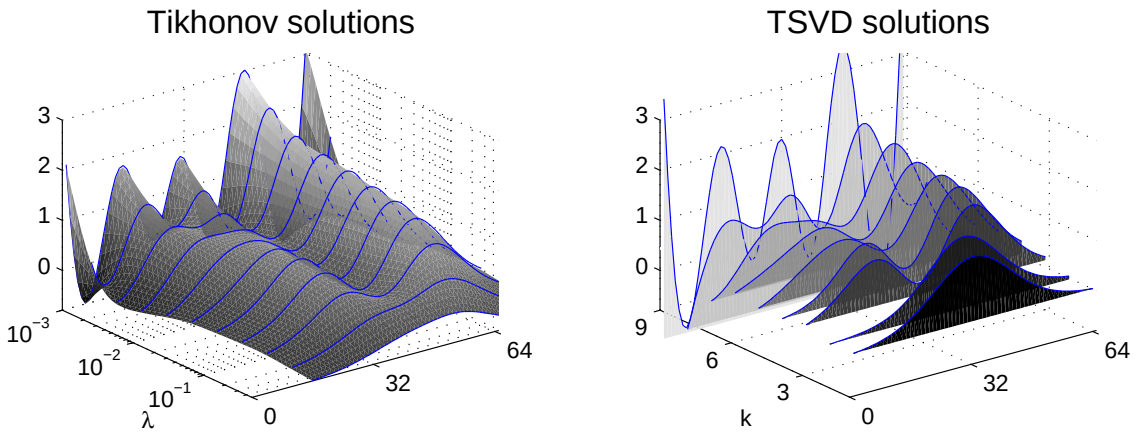


Figure 4.5: Tikhonov and TSVD solutions to the same problem. The left figure shows Tikhonov solutions x_λ, varying continuously with λ between 10^{-3} and 1. The right figure shows nine TSVD solutions x_k for $k = 1, \ldots, 9$.

In other words there is no guarantee that the terms $A_k^\dagger\, b^{\text{exact}}$ or $(A^T A + \lambda^2 I)^{-1} A^T\, b^{\text{exact}}$ will approximate x^{exact}. On the other hand, if the discrete Picard condition is satisfied, then the faster the decay the better these terms approximate x^{exact}. This clearly illustrates the importance of the discrete Picard condition.

TSVD and Tikhonov regularization are clearly related, due to the similarity of the filter factors for the two methods. Figure 4.5 illustrates this: for each value of k there exists a λ such that $x_k \approx x_\lambda$. It is therefore interesting to study the similarity of the TSVD and Tikhonov solutions – more precisely, we want to study the similarity of the noise-free regularized terms $A_k^\dagger\, b^{\text{exact}}$ or $(A^T A + \lambda^2 I)^{-1} A^T\, b^{\text{exact}}$. Let

$$\varepsilon_{k,\lambda} \equiv A_k^\dagger\, b^{\text{exact}} - (A^T A + \lambda^2 I)^{-1} A^T\, b^{\text{exact}} \ ,$$

and assume again that b^{exact} satisfies the model (4.8). Then, for a fixed value of k,

$$\frac{1}{\sqrt{n}} \min_\lambda \frac{\|\varepsilon_{k,\lambda}\|_2}{\|x^{\text{exact}}\|_2} \leq \begin{cases} \left(\frac{\sigma_{k+1}}{\sigma_k}\right)^{\alpha-1} & 0 \leq \alpha < 1 \\ \left(\frac{\sigma_k}{\sigma_1}\right)^{\alpha-1} & 1 \leq \alpha < 3 \\ \left(\frac{\sigma_k}{\sigma_1}\right)^{2} & 3 \leq \alpha \end{cases}$$

and the minimum is attained for $\lambda = (\sigma_k^3\, \sigma_{k+1})^{1/4}$. Again, the importance of the discrete Picard condition is obvious: α must be greater than one in order to ensure that the TSVD and Tikhonov solutions produce similar results. The situation is, however, different for the predicted vectors; for a fixed k:

$$\frac{1}{\sqrt{n}} \min_\lambda \frac{\|A\, \varepsilon_{k,\lambda}\|_2}{\|b^{\text{exact}}\|_2} \leq \begin{cases} \left(\frac{\sigma_k}{\sigma_1}\right)^{\alpha} & 0 \leq \alpha < 2 \\ \left(\frac{\sigma_k}{\sigma_1}\right)^{2} & 2 \leq \alpha. \end{cases}$$

This minimum is attained for $\lambda = (\sigma_k\,\sigma_{k+1})^{1/2}$. We see that the predicted vectors can always be expected to be similar, even if the discrete Picard condition is not satisfied. This underlines the danger of using the residual norm as a means for evaluating the quality of the regularized solution.

4.5 The L-Curve

In the previous sections we have used the SVD extensively to define and study the behavior of two important regularization methods. We continue our use of the SVD as an analysis tool, and we take a closer look at the solution norm and the corresponding residual norm. These norms play a central role in the practical treatment of discrete ill-posed problems, because they can always be computed no matter which regularization method is used – they do not require the computation of the SVD or any other decomposition of the matrix.

The norm of the TSVD solution and associated residual vary monotonically with k, as can be seen from the following expressions

$$\|x_k\|_2^2 \;=\; \sum_{i=1}^{k} \left(\frac{u_i^T b}{\sigma_i^2}\right) \;\leq\; \|x_{k+1}\|_2^2 \tag{4.9}$$

$$\|A\,x_k - b\|_2^2 \;=\; \sum_{i=k+1}^{m} (u_i^T b)^2 \;\geq\; \|A\,x_{k+1} - b\|_2^2 \tag{4.10}$$

Similarly, the norm of the Tikhonov solution and its residual are given by:

$$\|x_\lambda\|_2^2 \;=\; \sum_{i=1}^{n} \left(f_i\,\frac{u_i^T b}{\sigma_i}\right)^2 \tag{4.11}$$

$$\|A\,x_\lambda - b\|_2^2 \;=\; \sum_{i=1}^{n} \left((1-f_i)\,u_i^T b\right)^2, \tag{4.12}$$

where $f_i = \sigma_i^2/(\sigma_i^2 + \lambda^2)$ are the filter factors. To see that these norms vary continuously with λ, we introduce the notation

$$\xi = \|x_\lambda\|_2^2 \qquad \text{and} \qquad \rho = \|A\,x_\lambda - b\|_2^2\;.$$

Then we can show that

$$\xi' \equiv \frac{d\xi}{d\lambda} = -\frac{4}{\lambda}\sum_{i=1}^{n}(1-f_i)\,f_i^2\,\frac{(u_i^T b)^2}{\sigma_i^2}$$

$$\rho' \equiv \frac{d\rho}{d\lambda} = \frac{4}{\lambda}\sum_{i=1}^{n}(1-f_i)^2\,f_i\,(u_i^T b)^2 = -\lambda^2\,\xi'\;.$$

We see that $\xi' < 0$ and $\rho' > 0$ for all λ, confirming the monotonicity of the norms as functions of λ. Also, from the relation $\rho' = -\lambda^2\,\xi'$ we obtain $d\xi/d\rho = -\lambda^{-2}$, showing that that the squared solution norm $\|x_\lambda\|_2^2$ is a monotonically decreasing function of the squared residual norm $\|A\,x_\lambda - b\|_2^2$. With a little manipulation one can show that the same is true for the norms themselves.

There is also a simple relationship between the second derivatives of ξ and ρ. Specifically, we have

$$\rho'' \equiv \frac{d^2\rho}{d\lambda^2} = \frac{d}{d\lambda}\,\left(-\lambda^2\,\xi'\right) = -2\,\lambda\,\xi' - \lambda^2\,\xi'' \;,$$

in which $\xi'' \equiv d^2\xi/d\lambda^2$. Consider now the curve obtained by plotting ξ versus ρ, with λ as the parameter. The curvature of this curve (see any book on analytical geometry for its definition) is given by

$$c = \frac{\rho'\,\xi'' - \rho''\,\xi'}{((\rho')^2 + (\xi')^2)^{3/2}} = \frac{2\,\lambda\,(\xi')^2}{((\rho')^2 + (\xi')^2)^{3/2}}$$

where we have inserted the above relation for ρ''. Hence, $c > 0$ for all λ and therefore the curve (ρ, ξ) is convex.

The curve is of interest because it shows how the regularized solution changes as the regularization parameter λ changes. We can say more about this curve; for example, it is easy to see that

$$0 \le \|x_\lambda\|_2 \le \|A^{-1}b\|_2 \qquad \text{and} \qquad 0 \le \|A\,x_\lambda - b\|_2 \le \|b\|_2 \;.$$

Also, since any point (ρ, ξ) on the curve is a solution to the problem

$$\rho = \min\,\|A\,x - b\|_2^2 \qquad \text{subject to} \qquad \|x\|_2^2 \le \xi$$

(see §4.2), it follows that the curve defines a boundary between to regions of the first quadrant: it is impossible choose a vector x such that the point $(\,\|A\,x-b\|_2^2\,,\,\|x\|_2^2\,)$ lies below the curve; only points on or above the curve can be attained. The same is, of course, true for the curve $(\,\|A\,x - b\|_2\,,\,\|x\|_2\,)$. But when either of these curves is plotted in linear scale, it is difficult to inspect its features because of the large range of values for the two norms.

As shown in [10], the features become more pronounced (and easier to inspect) when the curve is plotted in double-logarithmic scale. The associated curve

$$(\,\tfrac{1}{2}\log\rho\,,\,\tfrac{1}{2}\log\xi\,) = (\,\log\|A\,x_\lambda - b\|_2\,,\,\log\|x_\lambda\|_2\,)$$

is referred to as the *L-curve* for Tikhonov regularization, and it plays a central role in the analysis of discrete ill-posed problems. Figure 4.6 shows an example of a typical L-curve.

We have already seen that the Tikhonov solution behaves quantitatively different for small and large values of λ. The same is true for the L-curve. When λ is large, then x_λ is dominated by SVD coefficients whose main contribution is from the exact right-hand side b^{exact} (the solution is over-smoothed). A careful analysis (see [8]) shows that for large values of λ we have that

$$\|x_\lambda\|_2 \approx \|x^{\text{exact}}\|_2 \text{ (a constant)} \qquad \text{and} \qquad \|A\,x_\lambda - b\|_2 \text{ increases with } \lambda.$$

For small values of λ the Tikhonov solution is dominated by the perturbation errors coming from the inverted noise (the solution is under-smoothed), and we have that

$$\|x_\lambda\|_2 \text{ increases with } \lambda^{-1} \qquad \text{and} \qquad \|A\,x_\lambda - b\|_2 \approx \|e\|_2 \text{ (a constant)}.$$

The conclusion is that the L-curve has two distinctly different parts.

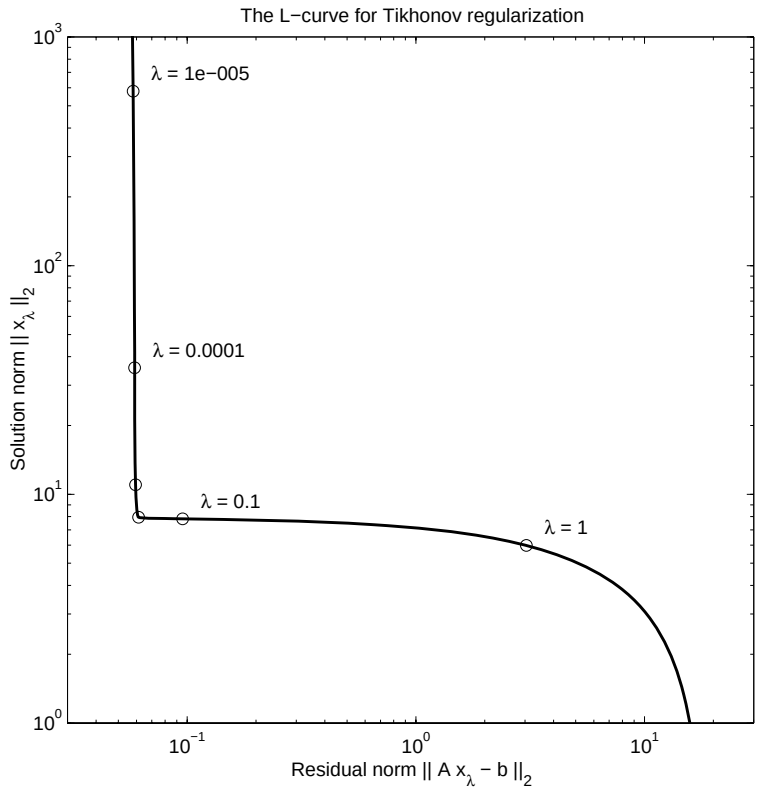


Figure 4.6: The L-curve for Tikhonov regularization: a log-log plot of the solution norm $\|x_\lambda\|_2$ versus the residual norm $\|A\,x_\lambda - b\|_2$ with λ as the parameter. Notice the vertical and horizontal parts, corresponding to solutions that are under- and over-smoothed, respectively.

- A part that is approximately horizontal, where the Tikhonov solution x_λ is dominated by the regularization error.
- A part that is approximately vertical, where x_λ is dominated by the perturbation error, i.e., by the inverted noise.
- The "corner" that separates these two parts is located roughly at

$$(\log \|e\|_2 \, , \, \log \|x^{\text{exact}}\|_2) \; .$$

Towards the right, for $\lambda \to \infty$, the L-curve starts to bend down as the increasing amount of regularization forces the solution norm towards zero. Likewise, towards the top left and above the vertical part, the L-curve will eventually become less steep as $\lambda \to 0$ and $\|x_\lambda\|_2 \to \|A^{-1}b\|_2$ (this part is not visible in Fig. 4.6).

This curve is found in many books and papers (perhaps starting with the pioneering book by Lawson and Hanson [12]), and a careful analysis of many of its properties can be found in [5]. It can also be used to determine[3] a suitable value of the regularization parameter; we return to this aspect in the next chapter.

[3]This use of the L-curve was not mentioned in [12].

4.6 Other Norms – Tikhonov Regularization in General Form

The presentation so far has focused on the 2-norm as a means for controlling (or suppressing) the error in the regularized solution. The norm $\|x\|_2$ enters explicitly in the Tikhonov formulation (4.5), and it is also part of the alternative formulation (4.4) of the TSVD method.

While the norm $\|x\|_2$ is indeed useful in some applications, it is not always the best choice. In order to introduce a more general formulation, let us return to the continuous formulation from Chapter 2, where the residual norm takes the form

$$R(f) = \left\| \int_0^1 K(s,t)\, f(t)\, dt - g(s) \right\|_2 .$$

In this setting, we can introduce a *smoothing norm* $S(f)$ that measures the "size" of the solution f. Common choices of $S(f)$ is the family given by

$$S(f) = \|f^{(p)}\|_2 = \left(\int_0^1 (f^{(p)}(t))^2\, dt \right)^{1/2} , \qquad p = 0, 1, 2$$

where $f^{(p)}$ denotes the pth derivative of f. Then we can write the Tikhonov regularization problem for f in the form

$$\min \left\{ R(f)^2 + \lambda^2\, S(f)^2 \right\}$$

where λ plays the same role as before. Clearly, the Tikhonov formulation in (4.5) is merely a discrete version of this general Tikhonov problem with $S(f) = \|f\|_2$.

Returning to the discrete setting, we can also formulate a general version of the Tikhonov problem by replacing the norm $\|x\|_2$ with a discretization of the smoothing norm $\omega(f)$. The general form of the discrete smoothing norm takes the form $\|L\, x\|_2$, where L is a discrete approximation of a derivative operator, and the associated Tikhonov regularization problem in *general form* thus takes the form

$$\min \left\{ \|A\, x - b\|_2^2 + \lambda^2\, \|L\, x\|_2^2 \right\} . \tag{4.13}$$

Similar to the standard-form problem (obtained for $L = I$), the general-form Tikhonov solution $x_{L,\lambda}$ is computed by recognizing that (4.13) is a linear least-squares problem of the form

$$\min \left\| \begin{pmatrix} A \\ \lambda\, L \end{pmatrix} x - \begin{pmatrix} b \\ 0 \end{pmatrix} \right\|_2 . \tag{4.14}$$

Some common choices of L are the two rectangular matrices

$$L_1 = \begin{pmatrix} -1 & 1 & & \\ & \ddots & \ddots & \\ & & -1 & 1 \end{pmatrix} \in \mathbb{R}^{(n-1)\times n}$$

$$L_2 = \begin{pmatrix} 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \end{pmatrix} \in \mathbb{R}^{(n-2)\times n}$$

which represent the first and second derivative operators, respectively. Thus, the discrete smoothing norm $\|L\, x\|_2$, with L given by either I, L_1 or L_2, represents the continuous smoothing norm $S(f) = \|f^{(p)}\|_2$ with $p = 0$, 1 and 2, respectively. Note the underlying assumption

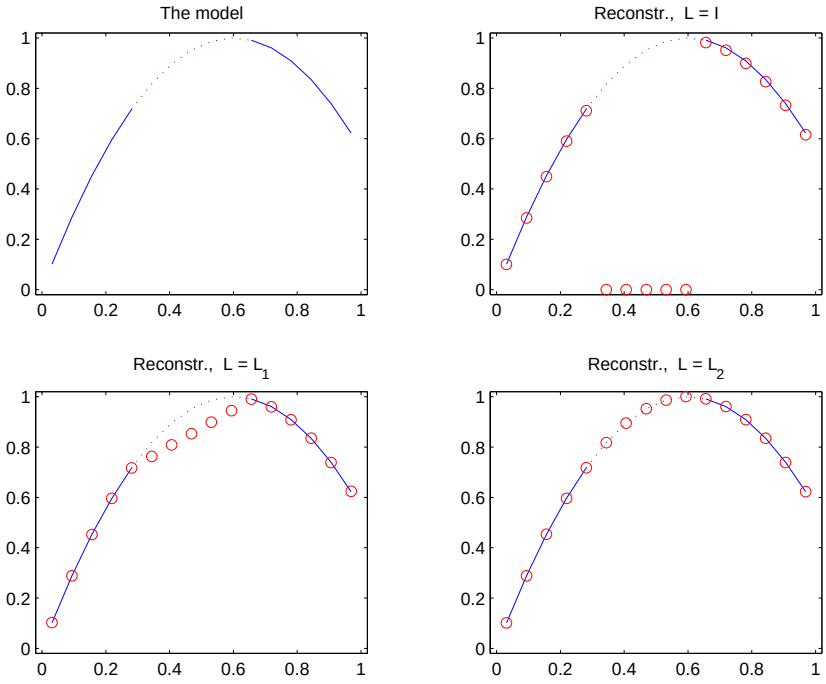


Figure 4.7: Illustration of the missing data problem. We show Tikhonov solutions for three choices of the discrete smoothing norm $\|L\,x\|_2$.

that the solution f is represented on a regular grid, and that a scaling factor (related to the grid size) is absorbed in the parameter λ.

To illustrate the improved performance of the general-form formulation, consider a simple ill-posed problem with missing data. Specifically, let x be given as samples of a function, and let the right-hand side be given by a subset of these samples, e.g.,

$$b = A\,x\ , \qquad A = \begin{pmatrix} I_{\text{left}} & 0 & 0 \\ 0 & 0 & I_{\text{right}} \end{pmatrix}\ ,$$

where I_{left} and I_{right} are two identity matrices. Figure 4.7 shows the solution x (consisting of samples of the sine function), as well as three reconstructions obtained with the three discrete smoothing norms $\|x\|_2$, $\|L_1\,x\|_2$ and $\|L_2\,x\|_2$. Clearly, the first choice is bad: the missing data are set to zero, in order to minimize the 2-norm of the solution. The choice $\|L_1\,x\|_2$ produces a linear interpolation in the interval with missing data, while the choice $\|L_2\,x\|_2$ produces a quadratic interpolation here.

We can also incorporate the general smoothing norm $\|L\,x\|_2$ in the TSVD setting, if we use the formulation in (4.4). The method is then referred to as *modified TSVD* (MTSVD), and it takes the form

$$\min \|L\,x\|_2 \qquad \text{subject to} \qquad \|A_k\,x - b\|_2 = \min, \tag{4.15}$$

where, again, $A_k = \sum_{i=1}^{k} u_i \sigma_i v_i^T$ is the TSVD matrix. To derive an algorithm for computing the MTSVD solution, recall that the general solution to the rank-deficient problem $\|A_k\,x - b\|_2$ has the form (4.3), and we must determined the coefficients $\zeta_{k+1}, \ldots, \zeta_n$ such that $\|L\,x\|_2$ is

minimized. This is achieved when we write the MTSVD solution as

$$x_{L,k} = x_k - V_k^o \, z_k \ ,$$

where x_k is the TSVD solution, where we have introduced the matrix $V_k^o = (v_{k+1}, \ldots, v_n)$, and where z_k is the least-squares solution to the problem

$$\min \| (L \, V_k^o) \, z - (L \, x_k) \|_2 \ .$$

We finish this discussion of general-form regularization methods by noting that the matrix L is allowed to have a nontrivial null space

$$\mathcal{N}(L) \equiv \{ x \in \mathbb{R}^n : L \, x = 0 \} \ .$$

For example, the null spaces of the two matrices L_1 and L_2 defined above are given by

$$\begin{aligned} \mathcal{N}(L_1) &= \text{span}\{ \, (1, 1, \ldots, 1)^T \, \} \\ \mathcal{N}(L_2) &= \text{span}\{ \, (1, 1, \ldots, 1)^T \, , \, (1, 2, \ldots, n)^T \, \} \ . \end{aligned}$$

These null spaces reflect the null spaces of the corresponding smoothing norms $\| f' \|_2$ and $\| f'' \|_2$, because f' is zero when f is any constant function, and f'' is zero if f is any constant or linear function.

Clearly, any component of the regularized solution $x_{L,\lambda}$ or $x_{L,k}$ in L's null space is unaffected by the smoothing norm $\| L \, x \|_2$:

$$y \in \mathcal{N}(L) \quad \Rightarrow \quad \| L \, y \|_2 = 0 \quad \text{and} \quad \| A \, y - b \|_2^2 + \lambda^2 \, \| L \, y \|_2^2 = \| A \, y - b \|_2^2 \ .$$

If the null-space component $y \in \mathcal{N}(L)$ shares components along the right singular vectors v_i associated with the small singular values of A, then we are in trouble because these components will not be filtered. Fortunately this is very unlikely – the null space of L is usually spanned by very smooth vectors that are well represented by the principal singular vectors v_1, v_2, etc. Hence, these are precisely the components that least need any filtering.

4.7 Exercises

4.7.1 SVD Analysis and TSVD Solutions

This exercise illustrates the use of the SVD as a tool for analysis of a discretization of the inverse heat equation problem, and it also illustrates the computation and use of TSVD solutions. The underlying ill-posed problem is a Volterra integral equation of the first kind, of the form

$$\int_0^s k(s - t) \, f(t) \, dt = g(s) \ , \qquad 0 \leq s \leq 1$$

with a convolution kernel k given by

$$k(\tau) = \frac{\tau^{-3/2}}{2 \, \kappa \, \sqrt{\pi}} \, \exp\left(-\frac{1}{4 \, \kappa^2 \, \tau^2} \right) \ .$$

Discretization of this problem by means of the midpoint rule leads to an ill-conditioned lower triangular matrix. This discretization is implemented in the Matlab function `heat` in the

Regularization Tools package, where a solution `x` and a corresponding right-hand side `b` is also computed.

For $n = 64$, use `[U,s,V] = csvd(A)` to compute the SVD of A, and then use `picard` to plot the singular values and the absolute values of the SVD coefficients of the right-hand side and the solution.

Assume that the perturbation e is Gaussian white noise with standard deviation η. How many SVD components can we expect to capture in a regularized TSVD solution when $\eta = 10^{-8}$? And when $\eta = 10^{-3}$?

Add the latter perturbation to the right-hand side, and use the function `tsvd` to compute TSVD solutions x_k for various choices of the truncation parameter k. Check whether your prediction about the optimal k holds.

4.7.2 A Closer Look at the Filter Factors

The goal is to derive Taylor expansions of the filter factors for the Tikhonov regularization methods. Specifically, determine the two different $O(\cdot)$-parts in the expression

$$f_i = \begin{cases} 1 - O(\cdot) & \sigma_i \gg \lambda \\ \sigma_i^2/\lambda^2 + O(\cdot) & \sigma_i \ll \lambda \ . \end{cases}$$

Hint: for $\sigma_i \gg \lambda$ rewrite the filter factor as

$$f_i = \frac{\sigma_i^2}{\sigma_i^2 + \lambda^2} = \frac{1}{1 + (\lambda/\sigma_i)^2}$$

and use the Taylor expansion $(1+\epsilon)^{-1} = 1 - \epsilon + O(\epsilon^2)$. Use the same basic technique for the case $\sigma_i \ll \lambda$.

4.7.3 Tikhonov Solutions via SVD

The purpose of this exercise is to illustrate Tikhonov regularization of the second derivative test problem. Use the `deriv2` function to generate the test problem, and use a small value of n, say, $n = 32$. Then use `[U,s,V] = csvd(A)` to compute the SVD of A, and inspect the singular values.

Add a small amount of noise to the right-hand side, e.g., `1e-3*randn(size(b))`. This noise is certainly not visible when plotting the right-hand side vector, but very significant with respect to the regularization. For a number of different regularization parameters λ in the range 10^{-4} to 1, compute the corresponding filter factors f_i by means of `fil_fac`, as well as the corresponding Tikhonov solution x_λ by means of `tikhonov`. For each λ, plot both the filter factors and the solution, and comment on your results. Use a logarithmic distribution of λ-values, as generated by Matlab's `logspace` function.

4.7.4 From Over-smoothing to Under-smoothing

The purpose of this exercise is to illustrate how the Tikhonov solution x_λ, its norm $\|x_\lambda\|_2$, and the residual norm $\|A\,x_\lambda - b\|_2$ change as λ goes from large values (over-smoothing) to small values (under-smoothing). Use the `shaw` test problem with $n = 32$, and Gaussian white noise with standard deviation $\eta = 10^{-3}$.

Use `lambda = logspace(1,-5,20)` to generate 20 logarithmically distributed values of λ from 10^{-5} to 10. Then compute the SVD of A and use `X = tikhonov(U,s,V,b,lambda)` to compute the 20 corresponding Tikhnonov solutions x_λ, stored as columns of the matrix `X`. Then inspect the columns of the matrix `X` (e.g., by means of `mesh` or `surf`) in order to study the progression of the regularized solution x_λ as λ varies from over-smoothing to under-smoothing.

For each value of λ, compute the solution norm $\|x_\lambda\|_2$ and the residual norm $\|A\,x_\lambda - b\|_2$, and plot the corresponding L-curve. Explain how the behavior of the L-curve is related to the behavior of the regularized solutions.

For the given λ-values in `lambda` (or perhaps more values in the same interval) compute the error norm $\|x^{\text{exact}} - x_\lambda\|_2$ and plot it versus λ. Determine the optimum value of λ – the one that leads to the smallest error – and locate the position of the corresponding solution on the L-curve. Is it near the "corner"?

4.7.5 The L-Curve

This exercise illustrates the typical behavior of the L-curve for a discrete ill-posed problem, using the second derivative test problem. Generate the test problem `deriv2` with $n = 64$, and add Gaussion white noise scaled such that $\|e\|_2 = 10^{-2}\,\|b^{\text{exact}}\|_2$. Then use `l_curve` from to plot the L-curves corresponding to the three different right-hand sides b^{exact}, e, and $b = b^{\text{exact}} + e$. Try to "zoom in" on the area near the corner.

Switch to lin-lin scale (e.g., by means of Axes properties ... in the Tools menu), and notice the different appearance of the curve. Again, try to zoom in on the corner.

Switch back to log-log scale and add a horizontal line at $\|x^{\text{exact}}\|_2$, the norm of the exact solution, and a vertical line at $\|e\|_2$, the norm of the perturbation. Relate the positions of these lines to the different parts of the L-curve.

Find a Tikhonov regularization parameter λ that approximately minimizes the error $\|x^{\text{exact}} - x_\lambda\|_2$ between the exact solution x^{exact} and the regularized solution x_λ. Add the point $(\,\|b - A\,x_\lambda\|_2\,,\,\|x_\lambda\|_2\,)$ to the L-curve (it must lie on the L-curve corresponding to b). Is it near the corner?

4.7.6 Limitations of TSVD and Tikhonov Solutions

This exercise illustrates one of the limitations of TSVD and Tikhonov solutions, namely, that they are not so well suited for computing regularized solutions when the exact solution is discontinuous. We use the model problem `wing` from Regularization Tools, whose solution has two discontinuities; see the manual for details. Since we are mainly interested in the approximation properties of the TSVD and Tikhonov solutions, we do not add any noise in this exercise.

Generate the model problem using `wing`, then plot the exact solution and notice its form. Compute TSVD and Tikhonov solutions for various regularization parameters. Monitor the solutions and try to find the "best" value of k and λ. Notice how difficult it is to reconstruct the discontinuities.

4.7.7 MTSVD Solutions

This exercise illustrates that the choice $L = I$ is not always a good choice in regularization problems. The exercise also illustrates the use of the MTSVD method. The test problem here

is the inverse Laplace transformation, which is implemented in REGULARIZATION TOOLS as the function `ilaplace`. The integral equation takes the form

$$\int_0^\infty \exp(-s\,t)\,f(t)\,dt = g(s)\ , \qquad 0 \le s \le \infty$$

with right-hand side g and solution f given by

$$g(s) = \frac{1}{s} - \frac{1}{s+1/2}\ , \qquad f(t) = 1 - \exp(-t/2)\ .$$

Use `[A,b,x] = ilaplace(n,2);` where "2" selects the above right-hand side and solution. We use two L-matrices, the identity matrix and the approximation L_1 of the first derivative operator.

Generate the test problem and add Gaussian white noise with standard deviation $\eta = 10^{-4}$ to the right-hand side. Plot the singular values and the SVD coefficients by means of the function `picard`. According to the plot, the problem looks innocent – nothing here reveals that the choice $L = I$ is bad.

Now compute TSVD solutions for $k = 1, 2, \ldots$ until the noise starts to dominate x_k, and plot the solutions. Notice that none of them are able to reconstruct the exact solution very well in the rightmost part of the plot.

Finally, compute the MTSVD solutions by means of the function `mtsvd`. Use a matrix L that approximates the first derivative operator; this matrix can be generated by means of the function `get_l`. Use $k = 2, 3, \ldots$ until the noise starts to dominate. The MTSVD solutions should be able to yield better approximations to the exact solution.

4.7.8 Tikhonov Regularization in General Form

The purpose of this exercise is to illustrate the use of Tikhonov regularization with a regularization term of the form $\|L\,x\|_2^2$, where L is an approximation to the second derivative operator. We use the same test problem as in Exercise 4.7.3, the second derivative problem, with the same noise level.

First compute the matrix $L = L_2$ by means of the command `L = get_l(n,2)`. What is the null space of this matrix?

Then solve the general-form Tikhonov problem[4] in its least-squares formulation (4.14) by means of Matlab's backslash, for several values of λ in the range 10^{-3} to 10 (use `logspace` to generate the λ-values). Plot the results and comment.

Why does this choice of L yield such good results, even for very large values of λ?

[4]There is an alternative way to compute general-form Tikhonov solutions that involves the generalized SVD. This is implemented in REGULARIZATION TOOLS, but the underlying theory is not covered here.

Chapter 5

Getting Serious: Choosing the Regularization Parameter

At this stage we have at our disposal several regularization methods, based on filtering of the SVD components – and we have also briefly discussed how to extend these methods to more general smoothing norms. The main thing that we are still missing is a reliable technique for choosing the regularization parameter, either k (for TSVD) or λ (for Tikhonov). Specifically, what we would like is an efficient, robust and reliable method for computing the regularization parameter from the given data.

Unfortunately, such a method has yet to be found – at the present time, no-one has devised a general-purpose parameter-choice algorithms which will always produce a good k or λ, such that the regularized solution is a good approximation to the unattainable exact solution. What we currently have at our disposal is a collection of methods which – under certain assumptions – tend to work well; but all of them can – and will – occasionally fail to produce good results.

We start with the simplest method, the so-called discrepancy principle, and then more on to two more robust methods, namely the L-curve criterion and generalized cross validation.

5.1 The Simple (but Dangerous) Discrepancy Principle

Let us start with a more careful analysis of the behavior of the regularized solution, as the regularization parameter changes. To simplify the presentation we restrict our analysis to the TSVD method, and we have already seen that as k increases, the characterization of the solution goes from over-regularization to under-regularization.

In order to study this in more detail, recall our assumption that the discrete Picard condition is satisfied. Hence the noisy right-hand side's SVD coefficients $u_i^T b$ will decay, on the average, until they level off at a plateau determined by the noise's standard deviation η. See also Eq. (3.5). Let k_η denote the index that marks the transition between decaying and flat coefficients. Due to the discrete Picard condition, the coefficients $u_i^T b/\sigma_i$ will also decay, on the average, for all $i < k_\eta$.

We now use this behavior in the analysis of the solution norm, using the relation (4.9) to

obtain rough approximations. We must distinguish between the two cases $k < k_\eta$ and $k > k_\eta$:

$$k < k_\eta : \qquad \|x_k\|_2^2 \approx \sum_{i=1}^{k} \left(\frac{u_i^T b^{\text{exact}}}{\sigma_i} \right)^2$$

$$k > k_\eta : \qquad \|x_k\|_2^2 \approx \sum_{i=1}^{k_\eta} \left(\frac{u_i^T b^{\text{exact}}}{\sigma_i} \right)^2 + \sum_{i=k_\eta+1}^{k} \left(\frac{\eta}{\sigma_i} \right)^2 \approx \|x^{\text{exact}}\|_2^2 + \eta^2 \sum_{i=k_\eta+1}^{k} \left(\frac{1}{\sigma_i} \right)^2 .$$

We see that for $k < k_\eta$ (over-regularization), the solution norm increases slowly with k, while for $k > k_\eta$ (under-regularization) it increases in a much more dramatic way. The index $k = k_\eta$ marks the transition between the two types of behavior.

In a similar fashion we can analyze the residual norm, now using the expression (4.9) to obtain rough approximations, and again distinguishing between the two cases for k:

$$k < k_\eta : \qquad \|b - A\,x_k\|_2^2 \approx \sum_{i=k+1}^{k_\eta} (u_i^T b)^2 + (n - k_\eta)\eta^2 \approx \sum_{i=k+1}^{k_\eta} (u_i^T b^{\text{exact}})^2$$

$$k > k_\eta : \qquad \|b - A\,x_k\|_2^2 \approx (n - k)\eta^2.$$

We conclude that for $k < k_\eta$ (over-regularization), the residual norm decreases steadily with k, while for $k > k_\eta$ (under-regularization) it decreases much more slowly. Again the transition between the two types of behavior occurs at $k = k_\tau$ when the regularization and perturbation errors are balanced.

Figure 5.1 illustrates the quality of the above approximations to $\|x_k\|_2$ and $\|b - A\,x_k\|_2$, for the `shaw` test problem (see Exercise 3.4.4) with $n = 20$ and $\eta = 10^{-6}$.

A good value of the truncation parameter is obviously $k = k_\eta$, for which we extract all the SVD coefficients for which the "signal" dominates, i.e., for which $|u_i^T b^{\text{exact}}| > \eta$. If we want to compute an approximation to this k_η then we should locate the transition point in the solution norm and the residual norm. Unfortunately, the automatic determination of such a transition point is not an easy task – many factors (such as problem scaling and the actual decay rates of the singular values and the coefficients $u_i^T b$) must be taken into account.

A related criterion which is much simpler to implement is to choose k such that the residual norm is equal to $(n - k_\eta)^{1/2}\eta$ because, ideally, this happens exactly at $k = k_\eta$. There are two drawbacks, however: k_η appears explicitly in the formula, and even if we happen to have an estimate of $(n - k_\eta)^{1/2}\eta$ then this criterion for choosing is very sensitive to the quality of the criterion. Instead one uses a slightly larger value which is easier to obtain, namely, the expected value of the perturbation norm $\|e\|_2 = n^{1/2}\eta$. This is the *discrepancy principle:* and it says that we should choose the truncation parameter k such that the residual norm equals the "discrepancy" in the data, as measured by $\|e\|_2$. It is rarely the case that the relation can be satisfied exactly, so instead we choose the *largest* k such that $\|A\,x_k - b\|_2 \geq \|e\|_2$:

$$\text{choose } k = k_{\text{DP}} \text{ such that } \|A\,x_k - b\|_2 \geq \|e\|_2 > \|A\,x_{k+1} - b\|_2 \tag{5.1}$$

The same strategy applies, of course, to the Tikhonov method with x_k replaced by x_λ, and here equality can be obtained:

$$\text{choose } \lambda = \lambda_{\text{DP}} \text{ such that } \|A\,x_\lambda - b\|_2 = \|e\|_2 \tag{5.2}$$

In both cases there is a unique solution because the residual norm varies monotonically with k and λ.

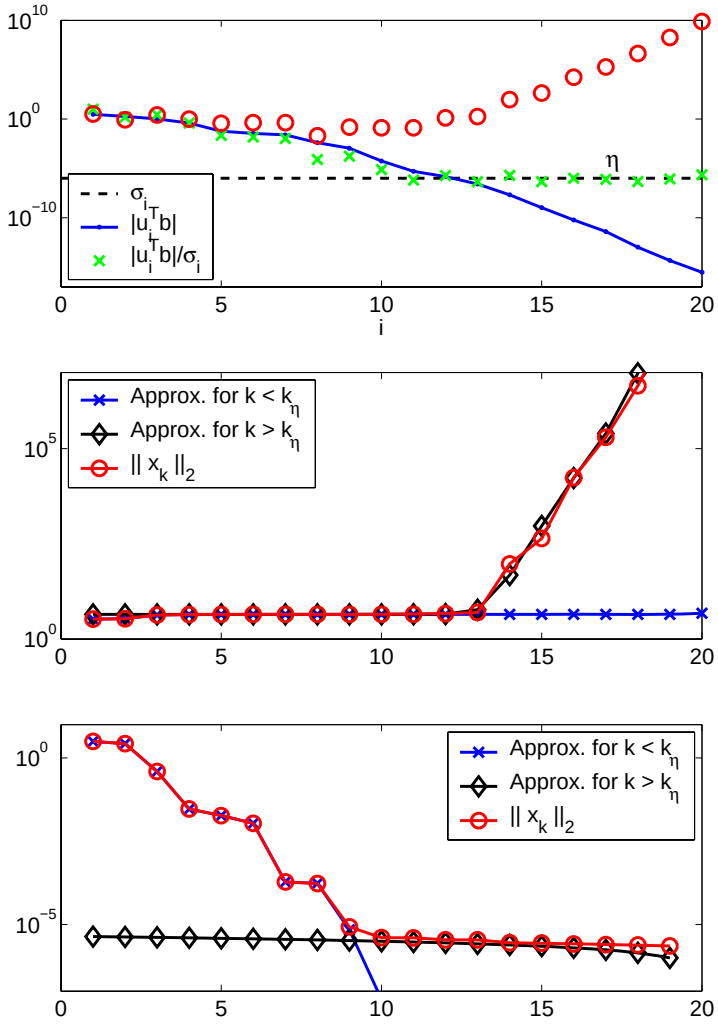


Figure 5.1: Top: Picard plot for the `shaw` test problem with $n = 20$ and $\eta = 10^{-6}$. Middle: the TSVD solution norm $\|x_k\|_2$ and the two approximations valid for $k < k_\eta$ and $k > k_\eta$. Bottom: the corresponding residual norm $\|b - A\,x_k\|_2$ and its approximations.

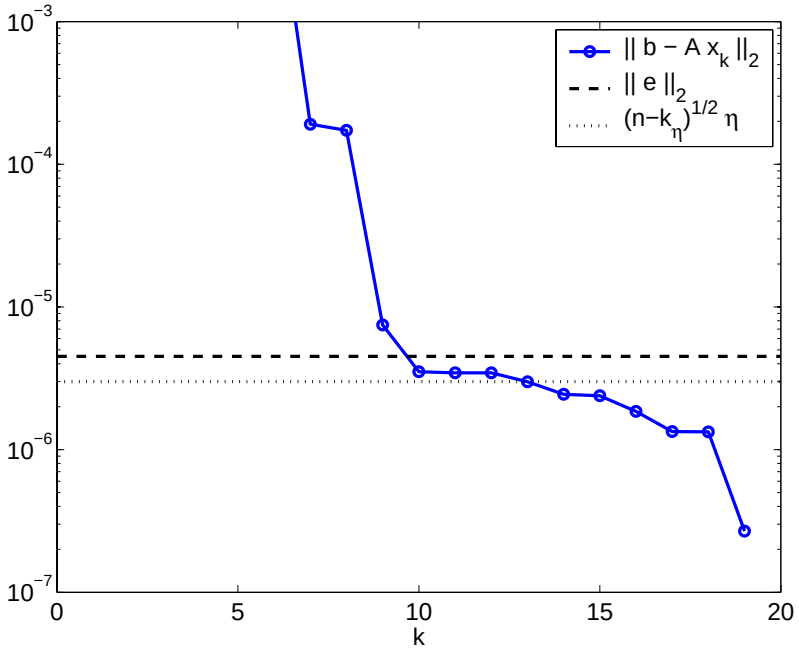


Figure 5.2: Illustration of the discrepancy principle. The choice $\|b - A\,x_k\|_2 \approx (n - k_\eta)^{1/2}\eta$ leads to a too large value of the truncation parameter k, while the more conservative choice $\|b - A\,x_k\|_2 \approx \|e\|_2$ leads to a better value of k.

The main advantage of this parameter-choice method is its simplicity – it only requires an efficient root finder. But it also has a big disadvantage: normally we do not know $\|e\|_2$ (or η) exactly, and therefore we must use an estimate of this norm. But unfortunately the quality of the computed regularization parameter k_{DP} or λ_{DP} is very sensitive to the accuracy of the estimate for $\|e\|_2$. In particular, a too small estimate can lead to dramatic under-regularization (because k is chosen too large, and λ is too small).

5.2 The Intuitive (but Non-Converging) L-Curve Criterion

Let us take a closer look at the error in the regularized solution. This time we want to consider both the Tikhonov solution and the TSVD solution; therefore we define the diagonal matrix F consisting of the filter factors,

$$F = \begin{pmatrix} f_1 & & \\ & \ddots & \\ & & f_n \end{pmatrix},$$

where $f_i = \sigma_i^2/(\sigma_i^2 + \lambda^2)$ for Tikhonov, and $f_i = 1$ or 0 for TSVD. Then we can write the regularized solution λ or x_k as

$$x^{\mathrm{reg}} = V\,F\,\Sigma^{-1}\,U^T b, \tag{5.3}$$

and we remind ourselves that the given right-hand side consists of an exact "signal" plus additive noise: $b = A\,x^{\mathrm{exact}} + e$. It follows that the error in the regularized solution is given

by

$$\begin{aligned} x^{\text{reg}} - x^{\text{exact}} &= V\,F\,\Sigma^{-1}\,U^T b - x^{\text{exact}} \\ &= V\,F\,\Sigma^{-1}\,U^T A\,x^{\text{exact}} + V\,F\,\Sigma^{-1}\,U^T e - x^{\text{exact}} \\ &= (V\,F\,\Sigma^{-1}\,U^T U\,\Sigma\,V^T - I)\,x^{\text{exact}} + V\,F\,\Sigma^{-1}\,U^T e \\ &= V\,(F-I)\,V^T x^{\text{exact}} + V\,F\,\Sigma^{-1}\,U^T e. \end{aligned}$$

The first term in the above expression is the *regularization error*, coming from the introduction of the filtering:

$$\Delta x_{\text{bias}} = V\,(F-I)\,V^T x^{\text{exact}} = \sum_{i=1}^{n} (f_i - 1)\ (v_i^T x^{\text{exact}})\,v_i,$$

and we recognize this as (minus) the bias term introduced in the previous chapter. It expresses the deviation of the expected value of x_λ from the exact solution.

The second error term is the *perturbation error*, coming from inverting and filtering the noise component in the data:

$$\Delta x_{\text{pert}} = V\,F\,\Sigma^{-1}\,U^T e.$$

The main purpose of the filtering/regularization, in the presence of the filter matrix F, is to prevent this perturbation error to blow up in magnitude and spoil the solution.

Both terms are always present in the regularized solution, but their size depend on the regularization parameter. If λ is very small then all f_i are close to one; hence F is close to I and the error term is small – but the perturbation error will be large because we apply very little filtering. On the other hand, if λ is very large then many filter factors will be small and thus the perturbation error will also be small – but then F is not close to the identity matrix, and the regularization error (the bias) will be large. We can therefore say that the goal of choosing λ is to balance the size of two error terms Δx_{bias} and Δx_{pert}.

The overall behavior of the L-curve from §4.5 is closely related to the size of these two error terms. Therefore one can learn a lot about the solution's dependence on the regularization error by studying the L-curve. Here, for simplicity, we consider the TSVD method.

- When k is smaller than k_η, then the regularization error dominates in x_k, and thus the overall behavior of its norm is largely determined by how many (or few) SVD components are effectively included. The fewer the components, the smaller the solution norm $\|x_k\|_2$ and the larger the residual norm $\|A\,x_k - b\|_2$. Using the analysis from the previous subsection (see also Fig. 5.1), we conclude that for TSVD these norms depend roughly on k as follows:
  - the solution norm $\|x_k\|_2$ is almost a constant given by $\|x^{\text{exact}}\|_2$ – except when $k \to 0$ and $\|x_k\|_2$ gets smaller.
  - the residual norm $\|A\,x_k - b\|_2$ increases as $k \to 0$, until it reaches its maximum value $\|b\|_2$ for $k = 0$.
- When k is larger than k_η, then the perturbation error will dominate x_k and the overall behavior of its norm is determined by this component. Now the norms depend in the following way on k:
  - the solution norm $\|x_k\|_2$ increases (sometimes dramatically) as k increases, because more and more – and increasingly larger – error components are included.

– the residual norm $\|A\,x_k - b\|_2$, on the other hand, stays almost constant at the value $\|e\|_2$ (except when $k \approx n$).

From this analysis it follows that the L-curve must have two distinctly different parts – one part where it is quite flat (when the regularization error dominates), and one part that is more vertical (when the perturbation error dominates). The log-log scale for the L-curve emphasizes the different characteristics of these two parts.

One important observation here is that the L-curve must also include a part with the transition between the vertical and horizontal parts. Very often this transition takes place in a small region, and then the L-curve will have a distinct corner between the two parts. Moreover, this transition region is associated with those values of λ or k for which the dominating error component changes between the perturbation error Δx_{bias} and the regularization error Δx_{pert}.

The key idea in the *L-curve criterion* is therefore to choose a value of λ that corresponds to the L-curve's corner, in the hope that this value will provide a good balance of the regularization end perturbation errors. We emphasize that this line of arguments is entirely heuristic; but it is supported with some analysis. It is for this reason that we allow ourselves to label the L-curve criterion as "intuitive."

We still need an operational definition of the corner of the L-curve, such that we can compute it automatically. For Tikhonov regularization, a natural definition of the corner is the point on the L-curve with maximum curvature. We emphasize that this curvature must be measured in the log-log system, i.e., we must compute the curvature $\hat{c}$ of the curve $(\log, \log)$.

At this stage it is most convenient to consider the L-curve for Tikhonov regularization. Following the analysis in §4.5, we introduce the quantities

$$\hat{\xi} = \log \|x_\lambda\|_2^2 \qquad \text{and} \qquad \hat{\rho} = \log \|A\,x_\lambda - b\|_2^2$$

such that the L-curve is given by $(\,\frac{1}{2}\hat{\rho}\,,\,\frac{1}{2}\hat{\xi}\,)$. If $'$ denotes differentiation with respect to λ, then it follows that

$$\hat{\xi}' = \frac{\xi'}{\xi} \qquad \text{and} \qquad \hat{\rho}' = \frac{\rho'}{\rho}.$$

When we use these relations, then it follows immediately that the second derivatives of $\hat{\imath}$ and $\hat{\rho}$ are given by

$$\hat{\xi}'' = \frac{\xi''\xi - (\xi')^2}{\xi^2}\,, \qquad \hat{\rho}'' = \frac{\rho''\rho - (\rho')^2}{\rho^2}.$$

When we insert these relations into the definition of the curvature

$$\hat{c} = 2\,\frac{\hat{\rho}'\,\hat{\xi}'' - \hat{\rho}''\,\hat{\xi}'}{((\hat{\rho}')^2 + (\hat{\xi}')^2)^{3/2}},$$

it turns out that we can express the curvature of the log-log L-curve entirely in terms of the quantities ξ, ρ and ξ':

$$\hat{c} = 2\,\frac{\xi\,\rho}{\xi'}\,\frac{\lambda^2\xi'\rho + 2\,\lambda\,\xi\,\rho + \lambda^4\xi\,\xi'}{(\lambda^2\xi^2 + \rho^2)^{3/2}}\,.$$

The two quantities $\xi = \|x_\lambda\|_2^2$ and $\rho = \|A\,x_\lambda - b\|_2^2$ are often easy to compute, so it remains to demonstrate how to compute ξ' efficiently. Using the SVD it is straightforward to verify that this quantity is given by

$$\xi' = \frac{4}{\lambda}\,x_\lambda^T z_\lambda,$$

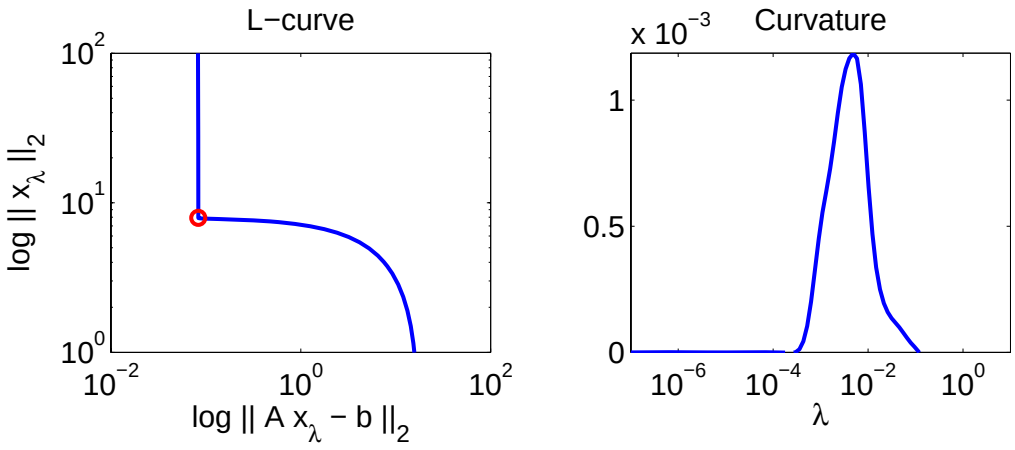


Figure 5.3: An L-curve and the corresponding curvature $\hat{c}$ as a function of λ. The corner, which corresponds to the point with maximum curvature, is marked by the red circle; it occurs for $\lambda_{\mathrm{L}} = 4.86 \cdot 10^{-3}$.

where the vector z_λ is given by

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

Recalling the formulas in §4.2 for the Tikhonov solution, we see immediately that z_λ is the solution to the Tikhonov least-squares problem

$$\min_z \left\| \begin{pmatrix} A \\ \lambda\, I \end{pmatrix} z - \begin{pmatrix} b - A\, x_\lambda \\ 0 \end{pmatrix} \right\|_2 .$$

Therefore, this vector is easy to compute by solving a new Tikhonov problem whose right-hand side is the residual vector $A\, x_\lambda - b$.

The conclusion of this analysis is that for each value of λ, we can compute the curvature $\hat{c}$ with little overhead, and thus we can use our favorite optimization routine to find the value λ_{L} that maximizes $\hat{c}$:

$$\text{choose } \lambda = \lambda_{\mathrm{L}} \text{ such that the curvature } \hat{c} \text{ is maximum.} \tag{5.4}$$

Figure 5.3 shows an L-curve and the corresponding curvature $\hat{c}$ as a function of λ.

The above analysis is not immediately valid for the TSVD method, because this method produces a finite (small) set of solutions for $k = 1, 2, \ldots$, and thus the corresponding "L-curve" really consists of a finite set of points. However, the analysis and the arguments do carry over to this case, and it still often makes good sense to locate the TSVD solution at the overall corner of the discrete L-curve:

$$\text{choose } k = k_{\mathrm{L}} \text{ at the } \textit{overall corner} \text{ of the discrete L-curve.} \tag{5.5}$$

How to actually compute this corner is not as straightforward as it may seem, because a discrete L-curve can have many small local corners; we refer to [HaJeRo] for a recent algorithm.

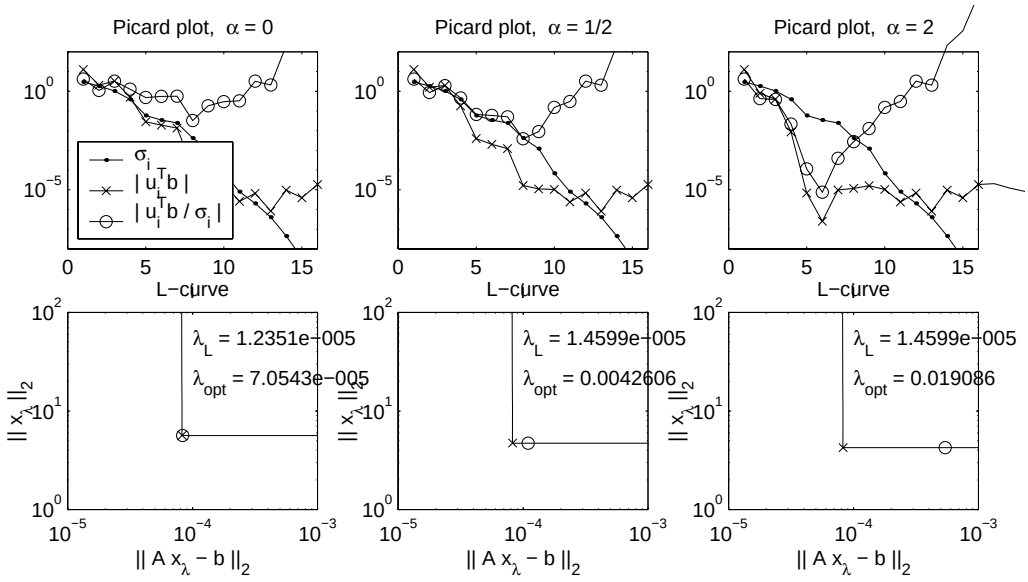


Figure 5.4: Picard plots and L-curves for (smooth) exact solutions generated as $x^{\text{exact},\alpha} = (A^T A)^{\alpha/2}\, x^{\text{exact}}$, where x^{exact} is from the `shaw` model problem, and α controls the smoothness (the larger the α, the faster the decay of the coefficients $|v_i^T x^{\text{exact},\alpha}|$.

We conclude this discussion of the L-curve criterion with a few words of caution. First of all, the criterion is essentially a (good) heuristic and there is no guarantee that it will always produce a good regularization parameter, i.e., that λ_{L} produces a solution in which the regularization and perturbation errors are balanced.

For example, the L-curve criterion is likely to fail when the SVD components $v_i^T x^{\text{exact}}$ of the exact solution decay fast to zero (in which case x^{exact} will appear very smooth, because it is dominated by the first few – and smooth – right singular vectors v_i). The L-curve's corner is typically located at the point where the solution norm starts to increase dramatically; but for very smooth solutions this happens when we have included too many noisy components, and hence λ_{L} will lead to an under-regularized solution; see Fig. 5.4

Finally, if we could let the noise go to zero, then it has been shown that the regularization parameter λ_{L} tends to diverge from the optimal one. That is, $x_{\lambda_{\text{L}}}$ diverges from x^{exact} as $\|e\|_2 \to 0$ (the parameter-choice method is non-converging in an undesired way). Fortunately, this is a rare situation in practise; usually we work with fixed data and thus a fixed noise level.

5.3 The Statistician's Choice – Generalized Cross Validation

The third method discussed here represents a somewhat different approach to choosing the regularization parameter, in that the goal here is to find the value of λ or k such that $A\, x_\lambda$ or $A\, x_k$ predicts the *exact* data b^{exact} as good as possible.

It is easiest to carry out the derivation and analysis for TSVD, for which the filter matrix is $F = \text{diag}(1, \ldots, 1, 0, \ldots, 0)$. Now consider the difference between b^{exact} and the predictor

$A\,x_k$:

$$\begin{aligned} b^{\text{exact}} - A\,x_k &= b^{\text{exact}} - A\,V\,F\,\Sigma^{-1}U^T(b^{\text{exact}} + e) \\ &= U\,U^T b^{\text{exact}} - U\begin{pmatrix} I_k & 0 \\ 0 & 0 \end{pmatrix} U^T b^{\text{exact}} - U\begin{pmatrix} I_k & 0 \\ 0 & 0 \end{pmatrix} U^T b \\ &= U\begin{pmatrix} 0 & 0 \\ 0 & I_{n-k} \end{pmatrix} U^T b^{\text{exact}} - U\begin{pmatrix} I_k & 0 \\ 0 & 0 \end{pmatrix} U^T e, \end{aligned}$$

and therefore, using again the SVD analysis, the norm of the prediction error satisfies

$$\|b^{\text{exact}} - A\,x_k\|_2^2 = \sum_{i=1}^{k}(u_i^T e)^2 + \sum_{i=k+1}^{n}(u_i^T b^{\text{exact}})^2 \approx k\,\eta^2 + \sum_{i=k+1}^{n}(u_i^T b^{\text{exact}})^2.$$

Once gain, let us split the analysis in two cases, depending on k:

$$\begin{aligned} k < k_\eta: &\qquad \|b^{\text{exact}} - A\,x_k\|_2^2 \approx k\,\eta^2 + \sum_{i=k+1}^{k_\eta}(u_i^T b^{\text{exact}})^2 \\ k > k_\eta: &\qquad \|b - A\,x_k\|_2^2 \approx k\,\eta^2. \end{aligned}$$

Our analysis shows that for $k < k_\eta$ the norm of the prediction error is dominated by the second term and is therefore decreasing with k, while for $k > k_\eta$ the norm increases slowly with k. The minimum arises at the transition, i.e., for $k \approx k_\eta$. Hence it makes good sense to search for the regularization parameter that minimizes the prediction error.

Of course, the prediction error in the form of $b^{\text{exact}} - A\,x_k$ is not available to use, and instead we must find another function whose minimum (approximately) coincides with the minimum of $\|b^{\text{exact}} - A\,x_k\|_2$. Such a function can be found by means of *generalized cross validation* (GCV) which leads to minimization of the function $\|b - A\,x^{\text{reg}}\|_2/\text{trace}(I - F)$. Specifically, for the Tikhonov and TSVD solutions we obtain

$$\text{choose } k = k_{\text{GCV}} \text{ as the minimizer of the function } G(k) = \frac{\|b - A\,x_k\|_2}{n - k} \tag{5.6}$$

and

$$\text{choose } \lambda = \lambda_{\text{GCV}} \text{ as the minimizer of the function } G(\lambda) = \frac{\|b - A\,x_\lambda\|_2}{n - \sum_{i=1}^{n} f_i}\,. \tag{5.7}$$

Experience shows that this parameter-choice method is quite robust and accurate, as long as the noise is white (because this assumption underlies the derivation of the GCV function). See Fig. 5.5 for an example of the GCV function $G(\lambda)$ for Tikhonov regularization. This figure also shows the typical behavior of GCV when it fails, namely, that the computed value λ_{GCV} is much too small.

5.4 Comparison of the methods

See Fig. 5.6

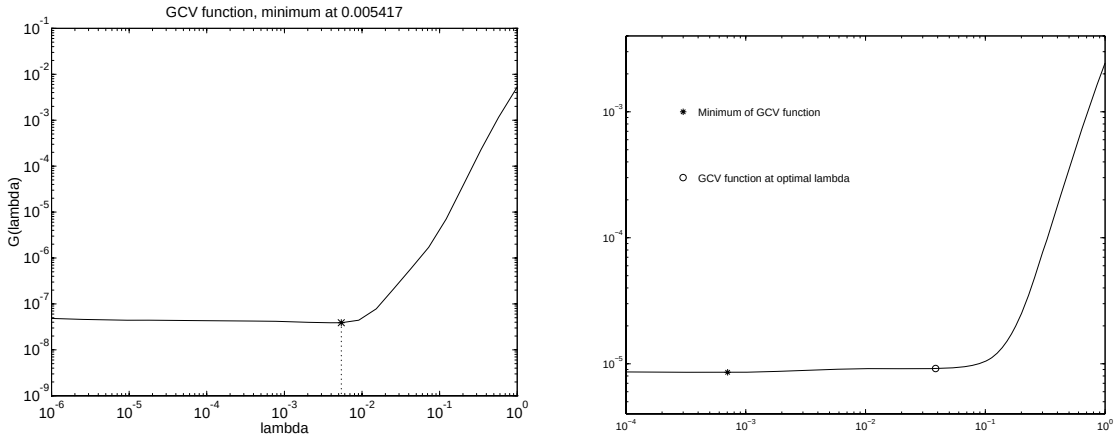


Figure 5.5: The GCV function $G(\lambda)$ for Tikhonov regularization. Left: the minimum is correctly located at $\lambda_{\text{GCV}} = 0.0054$. Right: the minimum is located at $\lambda_{\text{GCV}} \approx 7 \cdot 10^{-4}$, which is much smaller than the optimal $\lambda \approx 4 \cdot 10^{-2}$.

5.5 Exercises

5.5.1 The Discrepancy Principle

The purpose of this exercise is to illustrate the sensitivity of the discrepancy principle to variations of the estimate of the error norm. First generate the `shaw` test problem, and add Gaussian noise with zero mean and standard deviation η to the right-hand side.

Use the discrepancy principle to compute the Tikhonov solution. The discrepancy principle is implemented in the Matlab function `discrep`. This function may, occasionally, have convergence problems – if this happens, then change η slightly and try again.

As the "right-hand side" $\|e\|_2$ in Eq. (5.2), try to use both the norm estimate $\sqrt{n}\,\eta$ and the actual 2-norm of the perturbation vector e (perhaps try with different perturbations). Is there a significant difference in the results?

Use the discrepancy principle again, this time with values of $\sqrt{n}\,\eta$ and/or $\|e\|_2$ that are slightly too large and slightly too small; this simulates a situation where only a rough estimate is available. For example, scale $\sqrt{n}\,\eta$ and/or $\|e\|_2$ up and down by 5 %. How sensitive is the regularized solution to overestimates and underestimates of the noise level?

5.5.2 The GCV and L-curve Methods

This exercise illustrates the use of the GCV and L-curve methods for choosing the regularization parameter, and we compare these methods experimentally. As part of this comparison we investigate how robust – or reliable – the methods are, i.e., how often the produce a regularization parameter close to the optimal one. We use the same test problem as before.

Plot the GCV function by means of the Matlab function `gcv` for, say, 5 or 10 different perturbations, and note the general behavior of the GCV function. Is the minimum always at the transition region between the flat part and the more vertical part?

Use the L-curve to compute the regularization parameter, by means of the Matlab function `l_curve`, for the same 5 or 10 perturbations as above. Does the regularization parameter

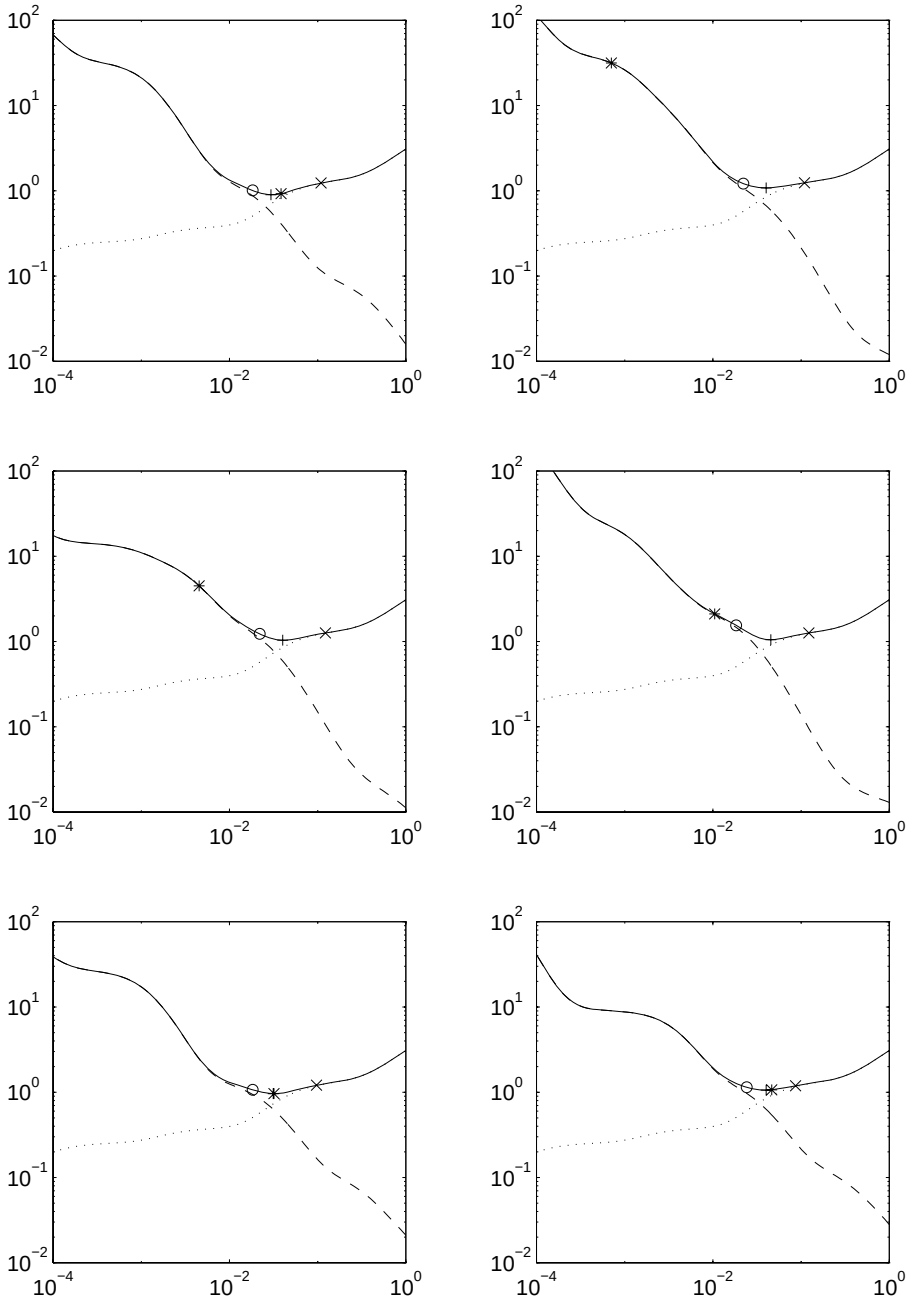


Figure 5.6: Norms of the perturbation errors (dashed lines), regularization errors (dotted lines), and total errors (solid lines) for six different perturbations e of the right-hand side. We also indicate the total errors corresponding to regularization parameters found by GCV method (asterisks) and the L-curve criterion (circles). FORGET THE CROSSES!

computed by means of `l_curve` always correspond to a solution near the corner?

For a particular perturbation of the right-hand side, compute the regularization error and the perturbation error for Tikhonov's method, for a range of regularization parameters λ. Plot the norms of the two error components components Δx_{bias} and Δx_{pert}, and relate the behavior of this plot to the two regularization parameters found by the GCV method and the L-curve criterion

5.5.3 Sensitivity Analysis

The purpose of this exercise is to investigate how sensitive the regularized solutions are to perturbations, once a regularization parameter has been computed by means of GCV or the L-curve criterion.

Generate your favorite test problem, add a particular perturbation e (neither too large nor too small), and compute the regularization parameters λ_{GCV} and λ_{L} for Tikhonov regularization by means of the GCV and L-curve methods. Make sure that these parameters are reasonable – otherwise choose another perturbation and try again.

Now keep the two regularization parameters λ_{GCV} and λ_{L} fixed, add a series of perturbations e with the same statistics to b^{exact}, and compute the Tikhonov solutions and the associated errors. How sensitive are the solutions to the variations in e?

Chapter 6

Towards Real-World Problems: Large-Scale Methods

We could stop our presentation at this stage! We have at our disposal several regularization methods, as well as several methods for choosing the regularization parameter from the given data. By combining these methods we have a good chance for solving a discrete ill-posed problem. It is important to develop and study these methods, because they give us a good understanding of the fundamental mechanisms of regularization.

Unfortunately, the methods we have discussed so far are designed for smaller problems where it is feasible to compute the SVD or compute Tikhonov solutions – via the least squares formulation – for many values of the regularization parameter.

Many real problems, however, are large. Typically, in these large-scale problems it is infeasible to have the coefficient matrix A explicitly available; instead we have software (a "black-box" function) that computes matrix-vector products with A and A^T. An even if the matrix A is explicitly available, it may be so big that it is infeasible (with respect to computing time and/or storage) to compute the SVD or solve the Tikhonov problem many times. Hence, there is a need for large-scale regularization methods that satisfy the following two requirements.

1. The main "building blocks" must be matrix-vector multiplications, such that there is not need to form and store the matrix A explicitly.

2. These methods should allow the user to select the regularization parameter – via one of the parameter-choice methods – without solving the problem from scratch for each new parameter.

One method that immediately comes to mind is to use an *iterative method* for solving the Tikhonov least squares problem (4.6). Iterative methods are ideal for large-scale problems because they only require matrix-vector products – as desired here. The disadvantage of this approach is mainly that for each new regularization parameter λ one needs to re-run the iterative method from the beginning.

As we shall demonstrate in this chapter, it is possible to combine the ideas from iterative methods with ideas from projection methods, leading to methods that satisfy both of the above-mentioned criteria.

6.1 The Classical (and Slow) Iterative Methods

Iterative method always start with a user-specified starting vector $x^{[0]}$ (often the zero vector) and produce a sequence of iterates $x^{[1]}$, $x^{[2]}$, ... that converge to some solution. We start with a brief discussion of some classical iterative methods, where the number of iterations plays the role of the regularization parameter. This is quite handy: as more and more iterations are performed, more and more SVD components are included in the solution $x^{[k]}$, where k denotes the iteration number.

As a consequence of this behavior, the iterates $x^{[k]}$ resemble regularized (filtered) solutions, and for $k = 1, 2, \ldots$ they tend to be better and better approximations to the exact solution x^{exact}. However, as yet more and more SVD components are included in $x^{[k]}$, the iterate will start to diverge again from x^{exact} and instead converge to the naive solution. This kind of convergence is called *semi-convergence*, and if we can stop the iterations at the right time then, in principle, we have a large-scale regularization method.

One of the best known methods that exhibits semi-convergence is known as *Landweber iteration* (many scientists have independently discovered this simple method, including Richardson, Fridman, Picard and Cimino). In its basic form the method takes the following simple form

$$x^{[k+1]} = x^{[k]} + \omega\, A^T(b - A\, x^{[k]}), \qquad k = 1, 2, \ldots \tag{6.1}$$

where ω is a real number that must satisfy $0 < \omega < 2\,\|A^T A\|_2^{-1} = 2/\sigma_1^2$. That's it ... all that is needed in each iteration is to compute the residual vector $r^{[k]} = b - A\, x^{[k]}$ and then multiply with A^T and ω; this correction is then added to the iterate $x^{[k]}$ to obtain the next iterate.

The iterate $x^{[k]}$ has a simple expression as a filtered SVD solution, similar to the TSVD and Tikhonov solutions, cf. (5.3). Specifically, we can write the kth iterate as

$$x^{[k]} = V\, F^{[k]}\, \Sigma^{-1} U^T b$$

where the elements of the diagonal matrix $F^{[k]} = \text{diag}(f_1^{[k]}, \ldots, f_n^{[k]})$ are the filter factors for $x^{[k]}$, which are given by

$$f_i^{[k]} = 1 - (1 - \omega\, \sigma_i^2)^k, \qquad i = 1, 2, \ldots, n. \tag{6.2}$$

(The verification of this result is left as Exercise 6.5.1.) Figure 6.1 shows how these filters "evolve" as the number of iterations k increases. We see that as k increases, the filters are shifted to the left and more SVD components are effectively included in the iterate $x^{[k]}$.

If we, quite arbitrarily, define the breakpoint in the filter factors as the value $\sigma_{\text{break}}^{[k]}$ of the singular values for which $f_i^{[k]} = 0.5$, then it is easy to show that

$$\sigma_{\text{break}}^{[k]} = \sqrt{\frac{1 - (\frac{1}{2})^{\frac{1}{k}}}{\omega}}.$$

These values are indicated in Fig. 6.1 by the circles. To see how this breakpoint varies with k, let us consider the ratio $\sigma_{\text{break}}^{[k]}/\sigma_{\text{break}}^{[2k]}$ between the breakpoint at k and $2k$ iterations:

$$\frac{\sigma_{\text{break}}^{[k]}}{\sigma_{\text{break}}^{(2k)}} = \sqrt{\frac{1 - (\frac{1}{2})^{\frac{1}{k}}}{1 - (\frac{1}{2})^{\frac{1}{2k}}}} = \sqrt{2^{\frac{1}{2k}}\, \frac{2^{\frac{1}{2k}} - 1}{2^{\frac{1}{k}} - 1}} = \sqrt{2^{\frac{1}{2k}}\, \frac{2^{\frac{1}{2k}} - 1}{(2^{\frac{1}{2k}} + 1)(2^{\frac{1}{2k}} - 1)}}$$

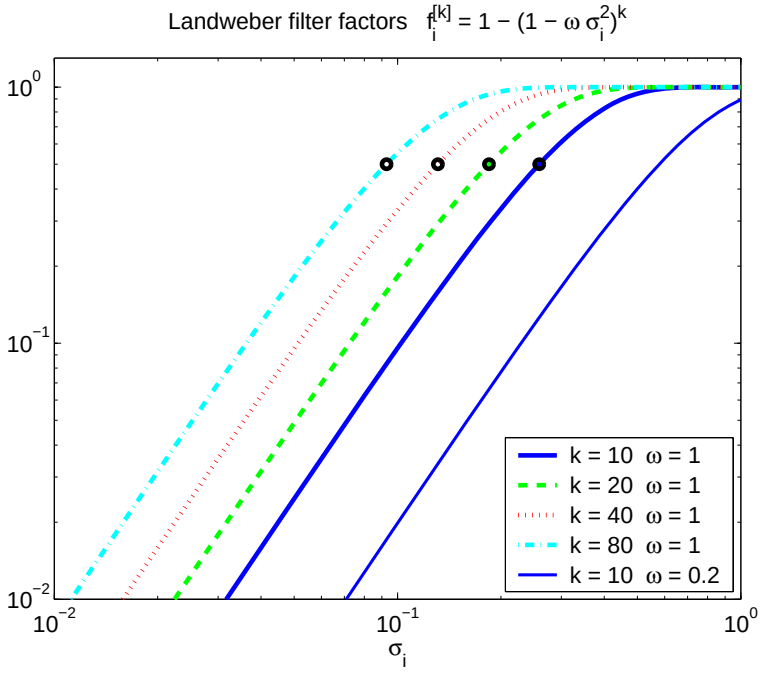


Figure 6.1: The Landweber filter factors $f_i^{[k]} = 1 - (1 - \omega\,\sigma_i^2)^k$ are functions of the number of iterations k and the parameter ω. The circles indicate the breakpoints $\sigma_{\text{break}}^{[k]}$ in the filter factors where they attain the value 0.5.

$$= \sqrt{\frac{2^{\frac{1}{2k}}}{2^{\frac{1}{2k}}+1}} = \sqrt{\frac{1}{1+(\frac{1}{2})^{\frac{1}{2k}}}} \to \sqrt{\frac{1}{2}} \quad \text{for} \quad k \to \infty.$$

Hence, as k increases, the breakpoint tends to be reduced by a factor $\sqrt{2} \approx 1.4$ each time the number of iterations k is doubled. The consequence is that the semi-convergence towards the exact solution is quite slow.

The Landweber iteration can be augmented to take the form

$$x^{[k+1]} = \mathcal{P}\left(x^{[k]} + \omega\,A^T(b - A\,x^{[k]})\right), \tag{6.3}$$

where the operator $\mathcal{P}$ represents so-called "hard constraint" by which we can incorporate additional a priori knowledge about the regularized solution. Exercise 6.5.3 illustrates this for the case where $\mathcal{P}$ represents a nonnegativity constraint.

Another classical method that is worth mentioning here is Kaczmarz's method, also known in computerized tomography as the algebraic reconstruction technique (ART). In this method, each iteration consists of a "sweep" through the rows $a_i^T = A(i,:)$ of A, in which the current solution vector $x^{[k]}$ is updated as:

$$x^{[k]} \leftarrow x^{[k]} + \frac{b_i - a_i^T x^{[k]}}{\|a_i\|_2^2}\,a_i, \qquad i = 1, 2, \ldots, m.$$

Here b_i is the ith component of the right-hand side b. The row norms $\|a_i\|_2$ are, of course, precomputed once. It can be shown that this method is mathematically equivalent to applying

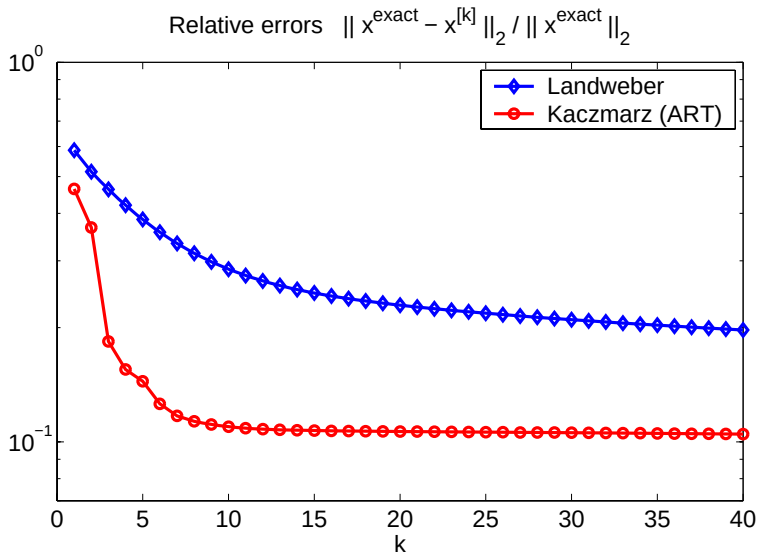


Figure 6.2: Example of the convergence histories for the Landweber and Kaczmarz (ART) methods applied to the `shaw` test problem of size $n = 128$. Both methods converge slowly, but the initial converge of ART (during the first, say, 5 iterations) is quite fast.

Gauss-Seidel iterations to the problem $x = A^T y$, $A\,A^T y = b$. This method converges quite fast in the first few iterations, after which the convergence becomes very slow. It was precisely this feature that made it a favorite method in the early days of computerized tomography, where only a few iterations were carried out.

Figure 6.2 shows an example of the error histories, i.e., the relative errors $\|x^{\text{exact}} - x^{[k]}\|_2 / \|x^{\text{exact}}\|_2$ as functions of k, for the Landweber and Kaczmarz (ART) methods. The Landweber parameter was chosen as $\omega = \|A\|_{\text{F}}^{-2}$ (using the Frobenius norm as an estimate of σ_1). Both methods are seen to exhibit semi-convergence: they actually converge (during the first many iterations) towards the exact solution x^{exact}. Moreover, both methods converge very slowly. During the first few iterations the convergence of ART is quite good; but the method essentially stagnates after about 10 iterations. Clearly, if better accuracy is required then there is a need for faster iterative methods.

6.2 Projection Methods

Before we discuss more modern – and faster – iterative methods, we sidetrack a bit into the area of projection methods, partly because they are interesting in their own right, and partly because they provide a means for understanding the behavior of the iterative methods based on Krylov subspaces.

As we have already seen in the analysis of the discrete ill-posed problems, it is impossible to compute a perfect reconstruction of x^{exact}. Our SVD analysis informs us that the best we can do is to compute an approximation to x^{exact} that effectively lies in a (low-dimensional) subspace of $\mathbb{R}^n$. For TSVD, this subspace is spanned by the first k right singular vectors $v_1, \ldots, v_k$ – which happens to mainly capture the low-frequency information in x^{exact}, because these singular vectors have few sign changes. For Tikhonov's method, and for a suitably

chosen value of λ, all the significant SVD components lie approximately in the same (low-dimensional) subspace.

For large-scale problems it is infeasible – or impossible – to compute the SVD of A. But we still know from the SVD analysis that the regularized solution should be dominated by low-frequency components. Hence, we might be able to provide, a priori, a set of basis vectors $w_1, \ldots, w_k$ that have the same overall features as the singular vectors, namely, being dominated by low-frequency components.

As an example, as out basis vectors $w_1, \ldots, w_k$ we could chose the orthonormal basis vectors of the discrete cosine transform (DCT), which are given by

$$w_1 = \sqrt{1/n}\,(1, 1, \ldots, 1)^T$$

and for $i = 2, 3, \ldots$

$$w_i = \sqrt{2/n}\,\left(\cos\left(\frac{i\,\pi}{2n}\right), \cos\left(\frac{3\,i\,\pi}{2n}\right), \ldots, \cos\left(\frac{(2n-1)\,i\,\pi}{2n}\right)\right)^T.$$

Some of these vectors are shown in Fig. 6.3; we see that they are scaled samples of cosine functions with increasing frequency. We emphasize that these basis vectors are not needed explicitly; instead all computations that involve these vectors are performed by means of efficient numerical algorithms based on the FFT. For example, if $W_k = (w_1, \ldots, w_n)$ denotes the $n \times k$ matrix consisting of the first k DCT basis vectors, and if we need to compute the vector $y = W_k^T x$ (of length k), then we can do this in Matlab by means of the `dct` function:

```
y = dct(x); y = y(1:k);
```

i.e., we first compute the DCT of x and then we extract the first k elements. Similarly, if we need to compute $x = W_k\,y = \sum_{i=1}^{k} w_i\,y_i$, then we will use the `idct` function which implements the inverse DCT:

```
x = idct([y;zeros(n-k,1)]);
```
(6.4)

Obviously, the same techniques apply to many other sets of basis functions associated with fast transforms, such as wavelets.

Let us now assume that we have chosen a suitable basis for a low-dimensional subspace, spanned by the columns of the matrix $W_k = (w_1, \ldots, w_k) \in \mathbb{R}^{n\times k}$, and that we want a solution expressed in this basis that fits the right-hand side b. We can formulate this as constrained least squares problem

$$\min_x \|A\,x - b\|_2 \qquad \text{s.t.} \qquad x \in \mathcal{W}_k = \text{span}\{w_1, \ldots, w_k\}. \tag{6.5}$$

We can also express the constraint as the requirement that $x = W_k\,y$, where $y \in \mathbb{R}^k$ is the new unknown. This leads to a regularized solution of the form

$$x^{(k)} = W_k\,y_k, \qquad y^{(k)} = \text{argmin}_y\,\|(A\,W_k)\,y - b\|_2. \tag{6.6}$$

We refer to the least squares problem $\|(A\,W_k)\,y - b\|_2$ as the *projected problem*, because it is obtained by projecting the original problem onto the subspace $\text{span}(w_1, \ldots, w_n)$. If k is not too large, then we can explicitly compute the matrix $A\,W_k \in \mathbb{R}^{n\times k}$ and then solve the projected problem, i.e., the least squares problem, for y.

We note in passing that if $W_k = (v_1, \ldots, v_k)$, i.e., if the basis vectors are the first k right singular vectors of A, then the projected problem takes the form

$$\|U\,\Sigma\,V^T W_k\,y - b\|_2 = \left\| U\,\Sigma \begin{pmatrix} I_k \\ 0 \end{pmatrix} y - b \right\|_2 = \left\| \begin{pmatrix} | & & | \\ u_1 & \cdots & u_k \\ | & & | \end{pmatrix} \begin{pmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_k \end{pmatrix} y - b \right\|_2 .$$

It follows immediately that the elements of $y^{(k)}$ are $y_i^{(k)} = u_i^T b/\sigma_i$, and therefore the projected solution $x^{(k)} = W_k\,y^{(k)}$ is the well-known TSVD solution. Hence, TSVD is actually a projection method based on the singular subspace $\mathrm{span}(v_1, \ldots, v_n)$.

Assume once again that the columns of W are the DCT basis vectors. To compute the matrix $\widehat{A}_k = A\,W_k$ we note that it consists of the first k columns of the matrix $A\,W = (W^T A^T)^T$. Hence, the ith row $(\hat{a}_k)_i^T$ of $\widehat{A}_k$ consists of the first k components of the DCT of the ith row of A, and therefor the Matlab code for computing the ith row `Ahat(i,:)` takes the form

```
z = dct(A(i,:)'); Ahat(i,:)  = y(1:k);
```

Once the solution y to the projected problem has been computed, we compute x by means of (6.4).

Figure 6.3 illustrates the use of a projection method with the DCT basis, applied to the `shaw` test problem of size $n = 128$ and with a small noise level of $\|e\|_2/\|b^{\mathrm{exact}}\|_2 \approx 4 \cdot 10^{-5}$. We show the solutions computed by projecting the original problem onto the basis vectors $w_1, \ldots, w_k$ for $k = 1, 2, \ldots, 10$. These solutions are clearly different from the TSVD solutions, in that the inherit the characteristics of the DCT basis vectors. We see that the projection onto a low-dimensional indeed has a regularizing effect on the solution, and that $k = 9$ seems to be the optimal dimension of the projection subspace.

If the solution is known a priori to have certain properties, then we may be able to take advantage of this knowledge in choosing the basis functions. For example, if we know that the exact solution is periodic, then we should choose periodic basis functions. As an illustration of this feature of the projection method, consider the `phillips` test problem from Regularization Tools whose exact solution is symmetric, i.e., $x_i^{\mathrm{exact}} = x_{n-i+1}^{\mathrm{exact}}$ for $i = 1, 2, \ldots, n/2$. Figure 6.4 shows projected solutions with two different bases, namely, the full DCT basis $w_1, w_2, w_3, \ldots$ and the basis of odd-numbered DCT vectors $w_1, w_3, w_5, \ldots$, the latter consisting purely of symmetric vectors. As shown in the bottom right plot of Fig. 6.4, the exact solution's coefficients to the even-numbered basis vectors are all zero. Hence, we can safely exclude these components in the projected solution to the noisy problem, and thus obtain good approximate solution for a smaller value of k using the odd basis than using the full basis.

6.3 Regularizing Krylov-Subspace Iterations

The advantage of the projection approach described above, with a fixed set of basis vectors, is that the operations involving these basis vectors can often be performed fast. At the same time, this is also the main disadvantage of this approach; the basis vectors are not adapted to the particular problem.

On the other hand, the "optimal" set of basis vectors for a particular matrix A – namely, the singular vectors – are out of computational reach for large-scale problems.

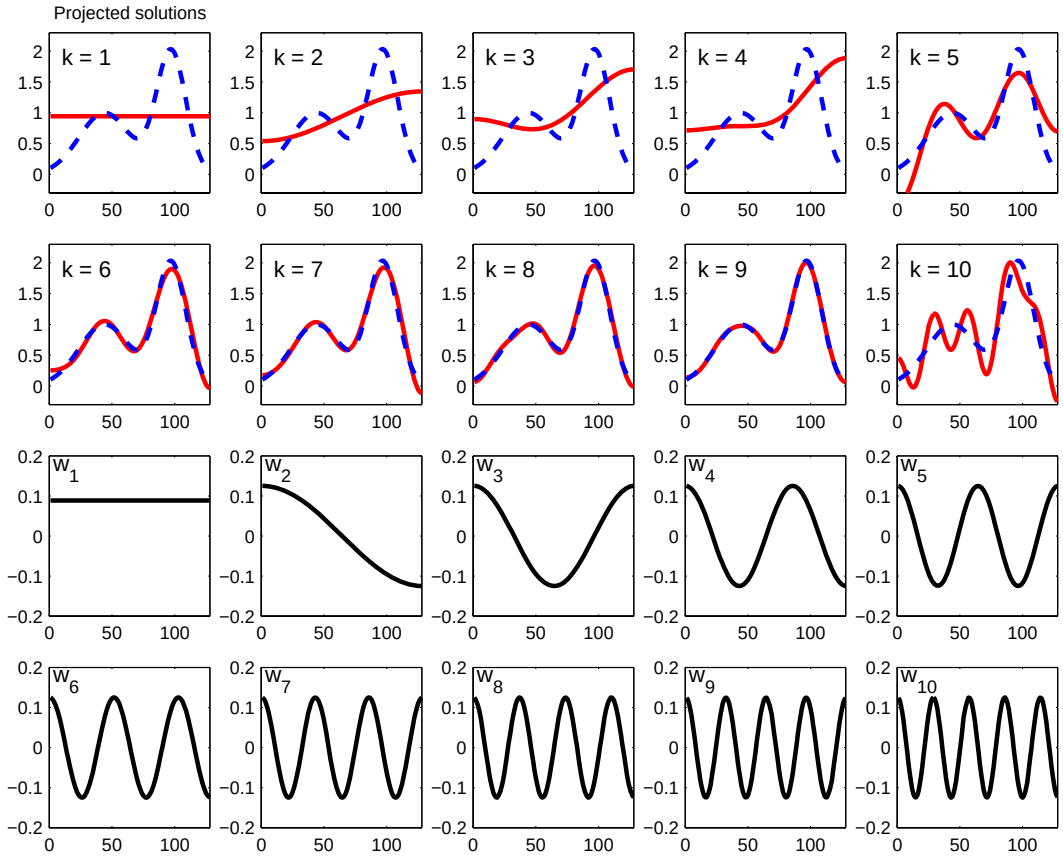


Figure 6.3: Example of the use of a projection method based on the DCT basis. The figures in the two bottom rows show the DCT basis vectors $w_1, \ldots, w_{10}$, which are sampled cosines. The figures in the two top rows show the exact solution x^{exact} (dashed line) and the projected solution $x^{(k)}$ computed by projecting the original problem onto the basis vectors $w_1, \ldots, w_k$ for $k = 1, 2, \ldots, 10$.

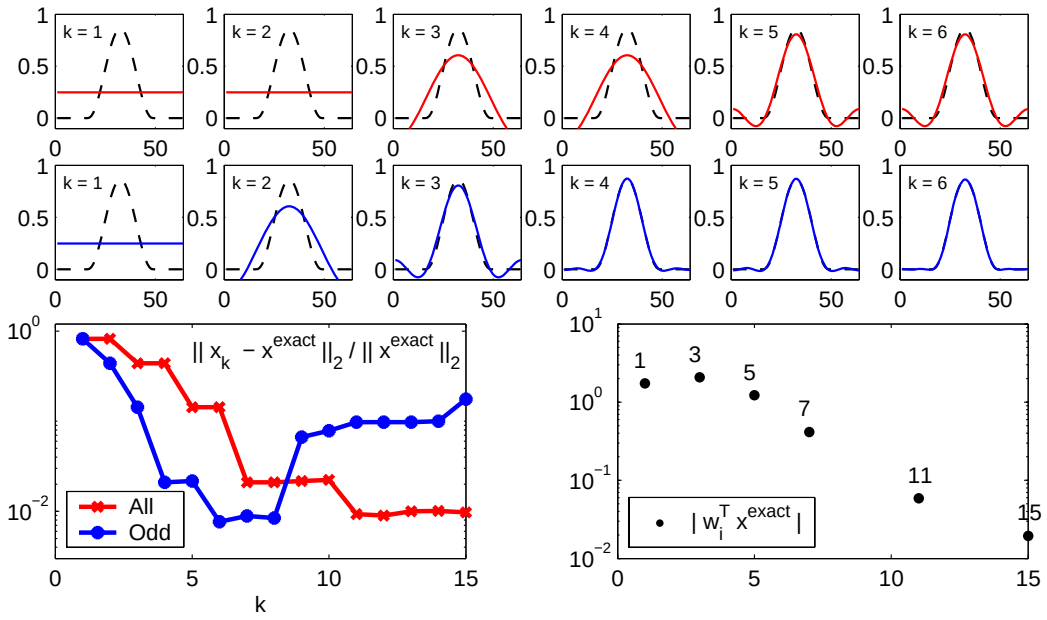


Figure 6.4: The `phillips` test problem has a symmetric solution with $x_i^{\text{exact}} = x_{n-i+1}^{\text{exact}}$ for $i = 1, 2, \ldots, n/2$. The top row shows solutions $x^{(k)}$ computed with a projection method using the full DCT basis $w_1, w_2, w_3, \ldots$, while the second row shows solutions using the basis of odd-numbered DCT vectors $w_1, w_3, w_5, \ldots$; clearly, a smaller number of basis vectors in needed in the second case to obtain a good approximate solution. The bottom right plot of the relative errors in $x^{(k)}$ confirms this. The bottom right plot shows the coefficients $w_i^T x^{\text{exact}}$ of the exact solution in the DCT basis, showing that only (some) odd-numbered basis vectors contribute to x^{exact}.

To rescue comes the *Krylov subspace* associated with A and b, defined as the span of powers of A^TA applied to A^Tb:

$$\mathcal{K}_k \equiv \text{span}\{A^Tb, A^TA\,A^Tb, (A^TA)^2A^Tb, \ldots, (A^TA)^{k-1}A^Tb\}. \tag{6.7}$$

The dimension of this subspace is at most k. The vectors in the definition of $\mathcal{K}_k$ are the iteration vectors of the power method for computing the largest eigenpair of a matrix, and they become increasing richer in the direction of the principal eigenvector of A^TA, which is really the principal right singular vector v_1. Therefore these vectors are not immediately useful as basis vectors for a projection method.

However, the subspace $\mathcal{K}_k$ itself carries important information about the problem, and is well suited to play the role of the subspace $\mathcal{W}_k$ in a projection method (6.5) for discrete il-posed problems. We just need a better representation of its basis, and we can obtain such a basis by orthonormalizing the vectors in (6.7), starting from A^Tb:

$$\begin{aligned}
&w_1 \leftarrow A^Tb; \quad w_1 = w_1/\|w_1\|_2 \\
&w_2 \leftarrow A^TA\,w_1; \quad w_2 \leftarrow w_2 - w_1^Tw_2\,w_1; \quad w_2 \leftarrow w_2/\|w_2\|_2 \\
&w_3 \leftarrow A^TA\,w_2; \quad w_3 \leftarrow w_3 - w_1^Tw_3\,w_1; \quad w_3 \leftarrow w_3 - w_2^Tw_3\,w_2; \quad w_3 \leftarrow w_3/\|w_3\|_2
\end{aligned}$$

and so forth. When we assume that the discrete Picard condition is satisfied, then the first basis vector w_1 is already rich in the direction of v_1. Due to the orthogonalization, this direction is purged from the second basis vector and therefore – still due to the discrete Picard condition – w_2 is rich is the next few directions v_2, v_3 etc. The argument continues for all the basis vectors w_i, and hence we obtain an orthonormal basis which tends to be rich in the directions of the principal right singular vectors $(v_1, v_2, \ldots)$.

Figure 6.5 illustrates this. We see that the vectors A^Tb, $A^TA\,A^Tb$, $(A^TA)^2A^Tb$, ... which define the Krylov subspace become increasingly dominated by v_1. However, the alternative and orthonormal basis vectors w_1, w_2, ... for the same subspace behave quite differently: each vector w_i is dominated by a few singular vectors v_j with index $j \approx i$. This illustrates that the Krylov subspace $\mathcal{K}_k$ indeed carries important information about the principal k left singular vectors. Hence, the Krylov subspace is a good choice for use in a projection method – and we emphasize that this fact hinges on the special property of the right-hand side for ill-posed problem, namely, that it satisfies the discrete Picard condition.

So how do we compute the basis vectors w_i efficiently? The fact is that the well-known Lanczos tridiagonalization process, applied to the matrix A^TA with starting vector A^Tb, produces precisely these vectors as the Lanczos vectors! Specifically, after k iterations the Lanczos algorithm has produced the matrix $W_k = (w_1, \ldots, w_k)$ and a symmetric tridiagonal matrix T_k such that

$$A^TA\,W_k = W_k\,T_k \qquad \text{for} \qquad k = 1, 2, \ldots \tag{6.8}$$

Strictly speaking, this is only true if the computations are done in infinite precision; but we refrain from discussing the finite-precision aspects here. When applied to a symmetric positive definite matrix, (here: A^TA) it can be shown that the Lanczos algorithm produces a tridiagonal matrix T_k whose larger eigenvalues converge to the larger eigenvalues of the matrix (here: the squared singular values of A). Therefore this algorithm is typically used for computing some of the large eigenvalues of a large sparse or structured matrix.

But there is more! The Lanczos process is closely related to the classical method of conjugate gradients (CG) for solving a system of linear equations with a symmetric positive

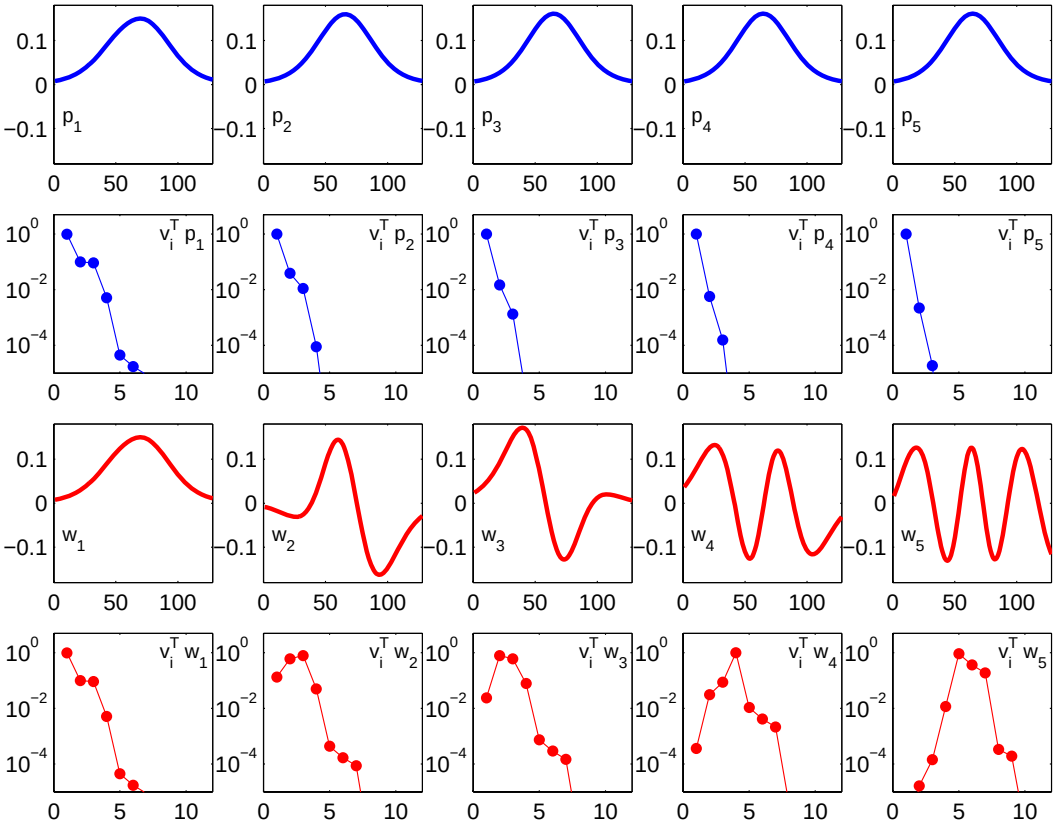


Figure 6.5: Illustration of the vectors associated with the Krylov subspace defined as $\mathcal{K}_k = \text{span}\{A^T b, A^T A\, A^T b, (A^T A)^2 A^T b, \ldots, (A^T A)^{k-1} A^T b\}$. The top row shows the first five normalized vectors $p_i = (A^T A)^{i-1} A^T b / \|(A^T A)^{i-1} A^T b\|_2$ that define the Krylov subspace, and the second row shows their expansion in the SVD basis; the vectors become increasingly dominated by v_1. The third row shows the first five orthonormal basis vectors w_i for the same subspace, and the fourth row shows these vectors' expansion in the SVD basis. Clearly, w_1 is dominated by v_1, while both w_2 and w_3 are dominated by v_2 and v_3. Then w_4 is dominated by v_4, and w_5 is dominated by v_5, v_6 and v_7. The overall behavior is that each vector w_i is dominated by a few singular vectors v_j with index $j \approx i$. We emphasize that this behavior of the Krylov subspace hinges on the problem satisfying the discrete Picard condition.

definite coefficient matrix. Here, this system is the normal equations $A^T A\,x = A^T b$ associated with the un-regularized least squares problem $\min_x \|A\,x - b\|_2$, and one can show that the solution $x^{(k)}$ obtained after applying k steps of the CG algorithm to the normal equations $A^T A\,x = A^T b$ is precisely the solution to the projected problem (6.5) with $\mathcal{W}_k = \mathcal{K}_k$:

$$x^{(k)} = \text{argmin}_x \|A\,x - b\|_2 \qquad \text{s.t.} \qquad x \in \mathcal{K}_k.$$

The CG algorithm computes this solution without the need to store all the basis vectors $w_1, w_2, \ldots$ explicitly; only a few auxiliary vectors are needed. Moreover, all that is needed during the iterations – in addition to vector-operations – are matrix-vector multiplications with A and A^T.

The most stable way to implement the CG algorithm for the normal equations is known as the *CGLS* algorithm. This algorithm can be found in the original CG paper by Hestenes and Stiefel [**?**], and it takes the following form:

$x^{(0)} = \text{starting vector}$
$r^{(0)} = b - A\,x^{(0)}$
$d^{(0)} = A^T r^{(0)}$
for $k = 1, 2, \ldots$
  $\bar{\alpha}_k = \|A^T r^{(k-1)}\|_2^2 / \|A\,d^{(k-1)}\|_2^2$
  $x^{(k)} = x^{(k-1)} + \bar{\alpha}_k\,d^{(k-1)}$
  $r^{(k)} = r_{(k-1)} - \bar{\alpha}_k\,A\,d^{(k-1)}$
  $\bar{\beta}_k = \|A^T r^{(k)}\|_2^2 / \|A^T r^{(k-1)}\|_2^2$
  $d^{(k)} = A^T r^{(k)} + \bar{\beta}_k\,d^{(k-1)}$
end

The starting vector $x^{(0)}$ is often the zero vector. Note that the algorithm requires one multiplication with A and one multiplication with A^T per iteration (to compute $A\,d^{(k-1)}$ and $A^T r^{(k)}$, respectively). Except for a normalization and perhaps a sign change, the "search vectors" $d^{(k)}$ of the CG algorithm are equal to the orthonormal basis vectors w_k for the Krylov subspace $\mathcal{K}_k$.

Since CGLS is an iterative method, we could also use the notation $x^{[k]}$ with brackets for the iteration vectors; instead we have chosen to use $x^{(k)}$ with parentheses to emphasize the connection to the projection methods.

To illustrate the regularizing effects of the CGLS algorithm, we applied this algorithm to the same test problem as in the previous section, namely, the `shaw` problem with $n = 128$. Figure Fig. 6.6 shows the CGLS iterates $x^{(k)}$ for $k = 1, 2, \ldots, 12$. We see that the Krylov subspace $\mathcal{K}_k$ is a good basis for this problem, and the best approximation to x^{exact} is achieved after 10 iterations.

To further illustrate the regularizing properties of the Krylov subspace $\mathcal{K}_k$ we applied the CGLS algorithm to the `phillips` test problem (which was also used in the previous section). Due to the symmetries in both the matrix and the exact solution, many of the SVD coefficients $v_i^T x^{\text{exact}}$ of the exact solution are zero. The Picard plot in the top part of Fig. 6.7 reveals that about every second SVD component $v_i^T x^{\text{exact}}$ is zero. Similarly, the Picard plot for the noisy problem (shown in the bottom half of Fig. 6.7) shows that the corresponding reconstructed coefficients $u_i^T b/\sigma_i$ are quite small, of the order 10^{-3}.

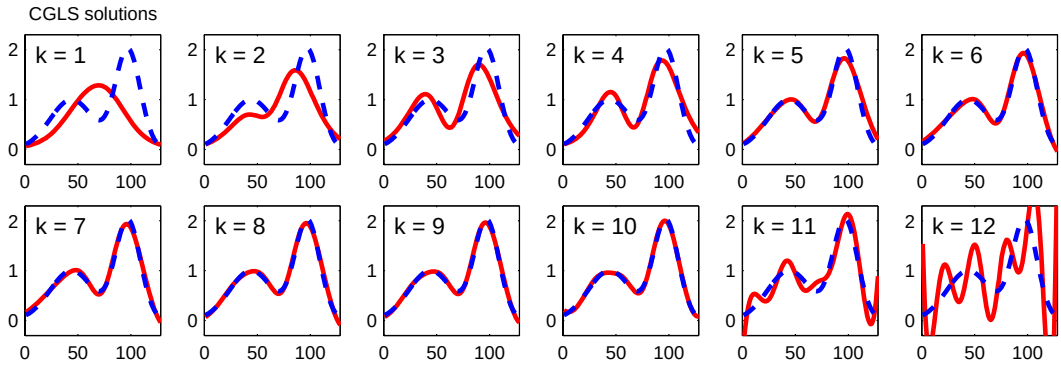


Figure 6.6: Red solid curves: the first 10 CGLS solutions $x^{(k)}$ for $k = 1, \ldots, 12$, for the same test problem as in Fig. 6.3. Dashed blue curves: the exact solution. A good approximation to x^{exact} is achieved after 10 iterations.

For this reason, the TSVD method with truncation parameter k – which insists on including *all* SVD components from 1 to k – includes about 50% components in x_k that are not necessary because they are so small. This is reflected in the error histories for the TSVD solutions: about every second increment of k (when a small SVD component is included) leaves the error unchanged, for both the noise-free and the noisy problem.

The Krylov subspace, on the other hand, is constructed from the starting vector $A^T b$ which has the SVD expansion

$$A^T b = A^T (A\, x^{\text{exact}}) + e = \sum_{i=1}^{n} \sigma_i^2 \, (v_i^T x^{\text{exact}}) \, v_i + \sum_{i=1}^{n} \sigma_i \, (u_i^T e) \, v_i .$$

Hence, 50% of the SVD coefficients of this vector are solely due to the noise components, and therefore these components are small (they are zero for the ideal noise-free problem). Consequently, the Krylov spaces $\mathcal{K}_k$ are dominated by precisely those singular vectors that contribute most the the solution – only the needed SVD directions are well represented in each Krylov subspace. For this reason, we can expect that the CGLS solution $x^{(k)}$ is a better approximation to x^{exact} than the corresponding TSVD solution with the same value of k (because the latter includes small and unnecessary components).

All these observations are confirmed by the results in Fig. 6.7, which shows both the TSVD and the CGLS solutions for a noise-free and a noisy problem. For the noise-free problem, we see that the error decreases much more rapidly for the CGLS solutions than the TSVD solutions. The same overall behavior is observed for the noisy problem; but the decreases of the error in the CGLS solution is slower here, because some unnecessary SVD components are "picked up" in the Krylov subspace – due to the noise – for $k = 5$, 6 and 7.

We emphasize that for a realistic large-scale problem, the situation is typically more complicated than we have illustrated here with small test problem. However, the overall behavior is the same; the Krylov subspace $\mathcal{K}_k$ tends to be a good subspace for projecting the solution, and the subspace tends to "pick up" the most important SVD directions first.

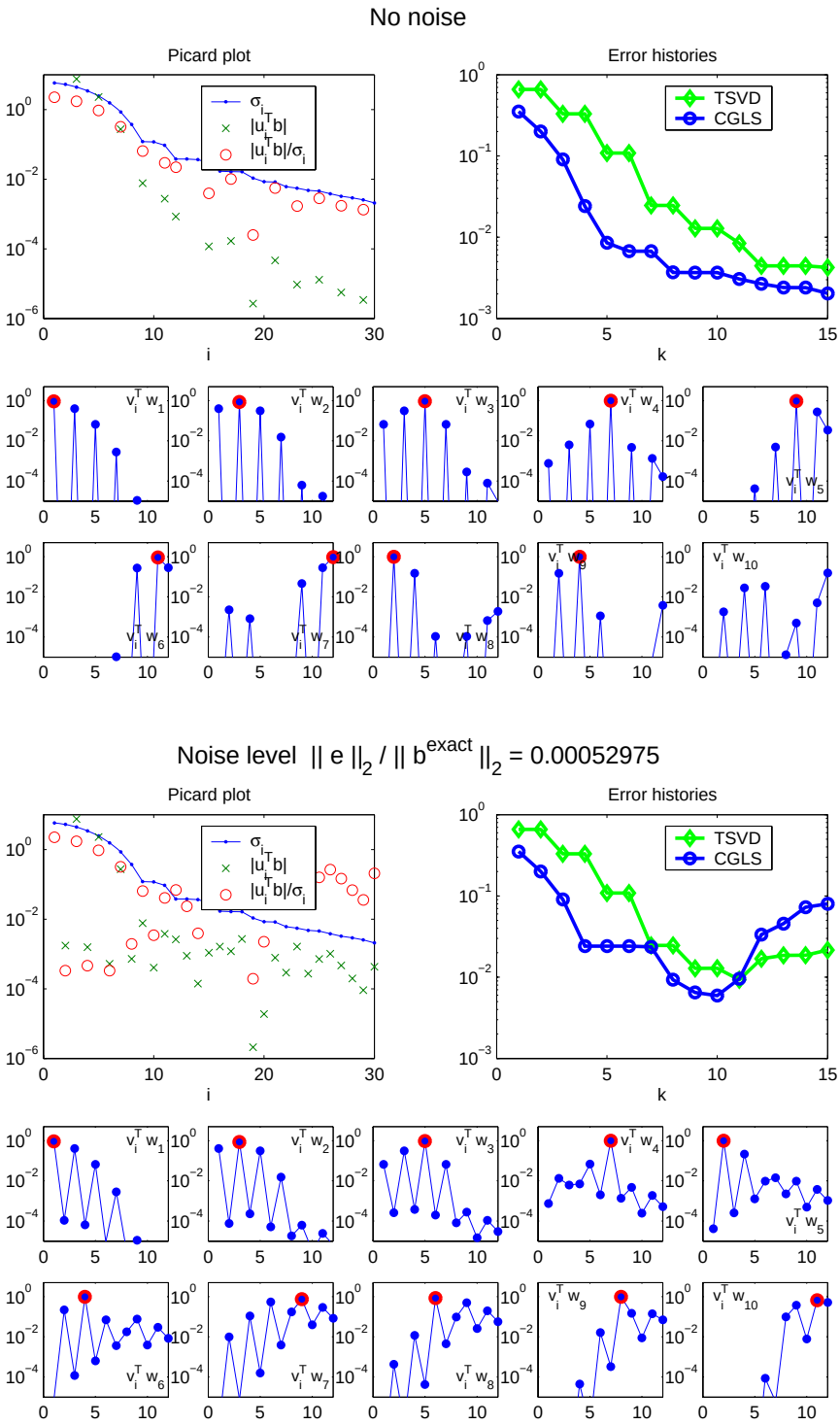


Figure 6.7: Insight into the behavior of the CGLS method applied to a discrete ill-posed problem that satisfies the discrete Picard condition, with no noise and a small noise level. For both situations we show the Picard plot, the error histories for TSVD and CGLS, and the coefficients of the Krylov/CGLS basis vectors w_k in the SVD basis – the largest value in absolute values is marked by the red circle.

6.4 Projection + Regularization = Best of Both Worlds

In principle the Krylov subspace $\mathcal{K}_k$ should always provide a good basis for a projection method – but in practise, this is not always the case. The main reason is that the Krylov subspace is generated on the basis of the given, noisy right-hand side b (and no the exact data b^{exact}), and for this reason the Krylov subspace tends to include both the desired basis vectors and, due to the noise, some basis vectors that are not so important for the regularized solution. The fact that we always work with finite-precision arithmetic has essentially the same effect.

In summary, we cannot be sure that all the important basis vectors have emerged in the Krylov subspace before some undesired basis vectors appear. The result is that the projected solution may not be a good regularized solution, because it can include very noisy components in the directions of the undesired vectors.

The solution to this dilemma is to combine the Krylov method with a regularization method applied to the low-dimensional projected problem. In the setting of Section 6.2, we apply regularization to the least squares problem in Eq. (6.6). In the important case of Tikhonov regularization, the regularized projected problem then takes the form:

$$x_\lambda^{(k)} = W_k \, y_\lambda^{(k)}, \qquad y_\lambda^{(k)} = \text{argmin}_y \, \left\{ \|(A\,W_k)\,y - b\|_2^2 + \lambda^2 \, \|y\|_2^2 \right\}. \tag{6.9}$$

It is interesting to note that if the matrix W_k has orthonormal columns – which is always recommended for reasons of numerical stability – then the solution $x_\lambda^{(k)}$ to the Tikhonov-regularized projected problem in (6.9) is identical to the solution obtained by applying the same projection method to the Tikhonov problem (4.6), i.e.,

$$x_\lambda^{(k)} = \text{argmin} \left\| \begin{pmatrix} A \\ \lambda I \end{pmatrix} x - \begin{pmatrix} b \\ 0 \end{pmatrix} \right\|_2 \qquad \text{s.t.} \qquad x \in \mathcal{W}_k = \text{span}\{w_1, w_2, \ldots, w_k\}.$$

To see this, simply note that if we insert $x = W_k \, y$ then the problem becomes

$$\begin{aligned} \min_y \left\| \begin{pmatrix} A \\ \lambda I \end{pmatrix} W_k \, y - \begin{pmatrix} b \\ 0 \end{pmatrix} \right\|_2 &= \min_y \left\| \begin{pmatrix} A\,W_k \\ \lambda W_k \end{pmatrix} y - \begin{pmatrix} b \\ 0 \end{pmatrix} \right\|_2 \\ &= \min_y \left\{ \|(A\,W_k)\,y - b\|_2^2 + \lambda^2 \, \|W_k \, y\|_2^2 \right\}, \end{aligned}$$

which is equivalent to (6.9) because $\|W_k \, y\|_2 = \|y\|_2$ when $W_k^T W_K = I_k$ (prove this). Hence, there is no distinction here between the two approaches "first-regularized-then-project" and "first-project-then-regularize."

The typical situation is that the columns of the matrix W_k are the orthonormal basis vectors for the Krylov subspace $\mathcal{K}_k$. There are two ways to generate these vectors.

- We can normalize the "search vectors" $d^{(k)}$ of the CGLS algorithm: $w_k = d^{(k)} / \|d^{(k)}\|_2$.
- We can apply Lanczos tridiagonalization to the matrix $A^T A$, cf. Eq. (6.8).

The latter approach is probably the most elegant, especially if we implement it via a bidiagonalization of A, thus avoiding to work with $A^T A$. The underlying algorithm is known as Golub-Kahan bidiagonalization [**?**], and when started with the right-hand side b it takes the following simple form, where the quantities α_k and β_k are chosen such that the corresponding vectors w_k and z_k are normalized:

$w_0 = 0$
$\beta_1 z_1 = b$
for $k = 1, 2, \ldots$
$\quad \alpha_k w_k = A^T z_k - \beta_k w_{k-1}$
$\quad \beta_k z_{k+1} = A v_k - \alpha_k z_k$
end

After k iterations, this remarkably simple algorithm has produced two matrices $W_k \in \mathbb{R}^{n\times k}$ and $Z_k \in \mathbb{R}^{m\times(k+1)}$ with orthonormal columns,

$$W_k = (w_1, w_2, \ldots, w_k), \qquad Z_{k+1} = (z_1, z_2, \ldots, z_{k+1}),$$

and a lower bidiagonal matrix $B_k \in \mathbb{R}^{(k+1)\times k}$,

$$B_k = \begin{pmatrix} \alpha_1 & & & \\ \beta_2 & \alpha_2 & & \\ & \beta_3 & \ddots & \\ & & \ddots & \alpha_k \\ & & & \beta_{k+1} \end{pmatrix},$$

such that

$$A W_k = Z_{k+1} B_k. \tag{6.10}$$

The columns of W_k are the desired basis vectors for the Krylov subspace $\mathcal{K}_k$, and when we insert the relation (6.10) into the projected problem (6.6) or (6.9), then the least-squares residual can be reformulated as follows:

$$\begin{aligned} \|(A W_k) y - b\|_2 &= \|Z_{k+1} B_k y - b\|_2 \\ &= \|Z_{k+1}^T (Z_{k+1} B_k y - b)\|_2 \\ &= \|B_k y - Z_{k+1}^T b\|_2 \\ &= \|B_k y - \beta_1 e_1\|_2. \end{aligned}$$

Here, $e_1 = (1, 0, \ldots, 0)^T \in \mathbb{R}^{k+1}$. We have used that $z_1 = b/\beta_1$ and that a multiplication with Z_{k+1}^T leaves the 2-norm unchanged. Hence, via the bidiagonalization algorithm, we arrive at the alternative formulation of (6.9):

$$x_\lambda^{(k)} = W_k y_\lambda^{(k)}, \qquad y_\lambda^{(k)} = \operatorname{argmin}_y \left\{ \|B_k y - \beta_1 e_1\|_2^2 + \lambda^2 \|y\|_2^2 \right\}. \tag{6.11}$$

The most important property of the projected Tikhonov problem $\|B_k y - \beta_1 e_1\|_2^2 + \lambda^2 \|y\|_2^2$ is that it can be solved in $O(k)$ operations, due to the bidiagonal structure of B_k. See [?] for details.

Another important feature of (6.11) is that the solution norm $\|x_\lambda^{(k)}\|_2$ and the residual norm and $\|A x_\lambda^{(k)} - b\|_2$ can be obtained directly from the Tikhonov solution $y_\lambda^{(k)}$ to the projected problem. Specifically, we have

$$\|x_\lambda^{(k)}\|_2 = \|W_k y_\lambda^{(k)}\|_2 = \|W_k^T W_k y_\lambda^{(k)}\|_2 = \|y_\lambda^{(k)}\|_2$$

and

$$\|A\, x_\lambda^{(k)} - b\|_2 = \|B_k\, y_\lambda^{(k)} - \beta_1 e_1\|_2.$$

The key observation here is that it is very easy to solve the regularized projected problem many times, for different values of λ, and that the solution and residual norms associated with $x_\lambda^{(k)}$ can be computed directly from $y_\lambda^{(k)}$. Hence, we can afford to use a parameter-choice method based on these norms, such as the discrepancy principle or the L-curve criterion, by applying it to the regularized projected problem. The back-transformation, via the multiplication with W_k, in only done once.

A complete algorithm based on these ideas – occasionally referred to as a "hybrid method" – would take the following overall form. Run the bidiagonalization algorithm for $k = 1, 2, 3, \ldots$ and store the coefficients α_k and β_k. For each 10th iteration, say, form the projected problem and apply Tikhonov regularization with a parameter-choice rule to this problem, to arrive at the parameter $\lambda^{(k)}$. Once the regularization parameter tends to "settle down" and stay almost the same over several value of k, this is a sign that the dimension of the Krylov subspace is large enough to have captured all relevant information, and that the appropriate amount of regularization has been determined. Then run the bidiagonalization algorithm again, this time applied to the Tikhonov problem and using the regularization parameter determined above. The precise details of this approach depends, of course, on the particular problem.

6.5 Exercises

6.5.1 Verification of the Landweber Filter Factors

Using the SVD of A, verify by insertion in (6.1) that the Landweber iterate $x^{[k]}$ is given by $x^{[k]} = V\, F^{[k]}\, \Sigma^{-1} U^T b$ with filter factors $f_i^{[k]} = 1 - (1 - \omega\, \sigma_i^2)^k$. Also show that for $\omega\, \sigma_i^2 \ll 1$ we have $f_i^{[k]} \approx k\, \omega\, \sigma_i^2$ (i.e., same decay as the Tikhonov filter factors).

6.5.2 Convergence of Three Iterative Methods

Jeres opgave her er bl.a. at hjælpe mig med at finde et godt test-problem; "shaw" er allerede brugt i teksten, derfor skal der bruges et andet i denne exercise!

This exercise illustrates and compares the convergence of three iterative regularization methods: Landweber iteration, ART (Kaczmarz's method) and CGLS. For each method we compute the error history and the filter factors.

First implement Landweber's method and ART with starting vector $x^{[k]} = 0$. Similar to the iterative methods in Regularization Tools the output should be a matrix whose columns are the iterates for $k = 1, 2, \ldots, k_{\max}$, where $k_{\max}$ is an input parameter.

Then generate the `xxx` test problem of size $n = 128$, with no noise in the right-hand side, and perform 100 iterations with all three methods. Use $\omega = \|A\|_{\mathrm{F}}^{-1}$ in the Landweber iterations. Plot the error histories, i.e., $\|x^{\mathrm{exact}} - x^{[k]}\|_2 / \|x^{\mathrm{exact}}\|_2$ versus k. Which method is fastest, and which is slowest?

For each iterative method and each iterate $x^{[k]}$, compute the corresponding filter factors $f_i^{[k]}$, $i = 1, \ldots, n$ and plot them. For Landweber's method we can use Eq. (6.2); but for ART and CGLS we must use a different approach. Use the SVD and show that if we apply these

methods to a noise-free problem, i.e., $b = b^{\text{exact}} = A\,x^{\text{exact}}$, then the iterates are given by

$$x^{[k]} = V\,F^{[k]}\,V^T x^{\text{exact}} \qquad \text{with} \qquad F^{[k]} = \text{diag}(f_1^{[k]}, \ldots, f_n^{(n)}).$$

Then use this expression to show that – in the absence of rounding errors – the filter factors can be computed as

$$f_i^{[k]} = \frac{v_i^T x^{[k]}}{v_i^T x^{\text{exact}}}, \qquad i = 1, 2, \ldots, n.$$

Finally, describe the behavior of these filter factors as k increases.

6.5.3 Landweber-Iteration for Nonnegative Solutions

The purpose of this exercise is to illustrate the use of the augmented Landweber iteration (6.3) where the operator $\mathcal{P}$ represents a nonnegativity constraint. In other words, if x and $y = \mathcal{P}(x)$ are vectors, then

$$y_i = \begin{cases} x_i, & x_i \geq 0 \\ 0, & x_i < 0. \end{cases}$$

The test problem to be used here is a 1-D version of the coherent imaging problem described in G. D. de Villiers, B. McNally and E. R. Pike, *Positive solutions to linear inverse problems*, Inverse Problems, 15 (1999), pp. 615–635. This is a first-kind Fredholm integral equation with kernel K and solution f given by

$$K(s,t) = \frac{\sin(6\,(s-t))}{\pi\,(s-t)}.$$

Both integration intervals are $[-1, 1]$, and the solution is

$$f(t) = \begin{cases} 1, & -0.75 < t < -0.5 \\ 1, & 0.5 < t < 0.65 \\ 0, & \text{elsewhere} \end{cases}$$

Use a simple method to discretize the problem and obtain A and x^{exact}, and then construct the right-hand side as $b^{\text{exact}} = A\,x^{\text{exact}}$.

Write a Matlab program that implements the discretization of the test problem, and compute a discretization with $n = 100$. In this exercise we focus in the reconstruction properties, and therefore we do not add noise to the problem. Run Landweber and CGLS iterations (remember the constraint on ω), compute the error histories, and monitor the solutions (they are not particularly good approximations to x^{exact}).

Modify your Landweber-implementation to incorporate the above-mentioned operator $\mathcal{P}$. Then run the augmented Landweber iterations, and again compute the error histories and monitor the solutions. Comment on the usefulness of this method for this particular problem.

6.5.4 Illustration of the CGLS Algorithm

The purpose of this exercise is to illustrate the use regularizing properties of the CGLS algorithm. This algorithm is implemented in Regularization Tools; use `help cgls` to obtain more information about its use.

The model problem is a small image deblurring problem which is generated by means of the function `blur` from the same package. Both the exact image and the blurred image are $N \times N$, and they are represented by N^2-vectors by stacking their columns. Use `X = reshape(x,N,N)` to get from the "stacked" representation of the image to the image itself. To display[1] the images, you can use the function `imagesc` together with the commands `axis image` and `colormap gray`.

Choose N as large as your patience allows, and generate the test problem. Plot the sharp and the blurred $N \times N$ images. At this stage, do not add noise to the problem. Perform a number of CGLS iterations – note that the CGLS iterates are returned as columns of the output matrix. Plot some of the iterates and notice how the regularized image gets sharper as the number of iterations increases.

Now add noise e to the blurred image, and repeat the CGLS computations. Choose the noise level such that $\|e\|_2/\|b\|_2 \approx 0.1$. You should notice that after a certain number of steps, the noise starts to dominate. This illustrates that the number of iterations plays the role of the regularization parameter for the CGLS algorithm.

6.5.5 The GCV and L-Curve Methods for Regularizing Iterations

This exercise illustrates the use of the GCV and L-curve methods applied to an iterative regularization method. We use the `blur` test problem and the CGLS method. Generate the `blur` test problem with $N = 32$, `band` $= 3$, and `sigma` $= 1.2$. Add noise to the right-hand side using the same noise level as before, run a number of CGLS iterations, and make sure that `cgls` also returns the residual and solution norms.

For iterative methods, it is difficult to compute the denominator of the GCV function, but it can sometimes be approximated by $n - k$, where k is the number of iterations – this is what we do here. Plot the approximate GCV function

$$\widetilde{G}(k) = \frac{\|b - A\,x^{(k)}\|_2}{n - k} \ , \qquad k = 1, 2, \ldots$$

and determine its minimum and thus the GCV-based number of iterations. Does it work for this problem? Now plot the discrete L-curve for the CGLS solutions, and try to locate the corner. Does the L-curve approach work for this problem?

[1] If the Image Processing Toolbox is available, use `imshow(X,[])` to display image `X`.

Chapter 7

Regularization Methods at Work: Two Case Studies

7.1 Adventures in 2D – Image Deblurring

7.2 Digging Deeper – 3D Gravity Surveying

Bibliography

[1] R. C. Aster, B. Borchers and C. H. Thurber, *Parameter Estimation and Inverse Problems*, Elsevier Academic Press, Amsterdam, 2005.

[2] H. W. Engl, M. Hanke and A. Neubauer, *Regularization of Inverse Problems*, Kluwer, Dordrecht, 1996.

[3] C. W. Groetsch, *Inverse Problems in the Mathematical Sciences*, Vieweg, Wiesbaden, 1993.

[4] P. C. Hansen, *Numerical tools for analysis and solution of Fredholm integral equations of the first kind*, Inverse Problems, 8 (1992), pp. 849–872.

[5] P. C. Hansen, *Analysis of discrete ill-posed problems by means of the L-curve*, SIAM Review, 34 (1992), pp. 561–580.

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

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

[8] 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.

[9] P. C. Hansen, *Deconvolution and regularization with Toeplitz matrices*, Numerical Algorithms, 29 (2002), pp. 323–378.

[10] P. C. Hansen and D. P. O'Leary, *The use of the L-curve in the regularization of discrete ill-posed problems*, SIAM J. Sci. Comput., 14 (1993), pp. 1487–1503.

[11] J. Kaipio and E. Somersalo, *Statistical and Computational Inverse Problems*, Springer, New York, 2005.

[12] C. L. Lawson and R. J. Hanson, *Solving Least Squares Problems*, Prentica-Hall, Englewood Cliffs, N.J., 1974; reprinted by SIAM, Philadelphia, 1995.

[13] A. Tarantola, *Inverse Problem Theory and Methods for Model Parameter Estimation*, SIAM, Philadelphia, 2005.

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

[15] G. M. Wing and J. D. Zahrt, *A Primer on Integral Equations of the First Kind*, SIAM, Philadelphia, 1991.

Index