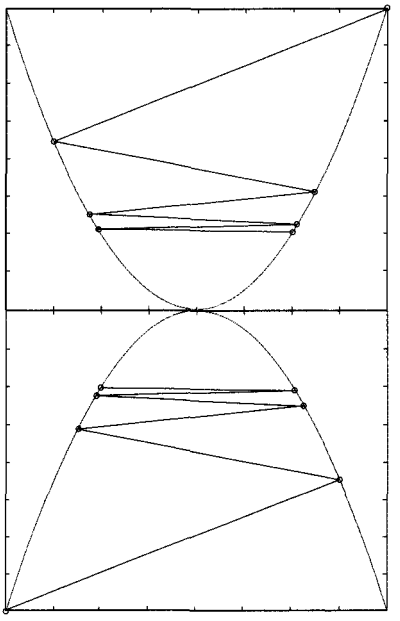
Computational Methods for Inverse Problems

Curtis R. Vogel

siam

FRONTIERS IN APPLIED MATHEMATICS

Computational Methods for Inverse Problems



This page intentionally left blank

FRONTIERS IN APPLIED MATHEMATICS

The SIAM series on Frontiers in Applied Mathematics publishes monographs dealing with creative work in a substantive field involving applied mathematics or scientific computation. All works focus on emerging or rapidly developing research areas that report on new techniques to solve mainstream problems in science or engineering.

The goal of the series is to promote, through short, inexpensive, expertly written monographs, cutting edge research poised to have a substantial impact on the solutions of problems that advance science and technology. The volumes encompass a broad spectrum of topics important to the applied mathematical areas of education, government, and industry.

EDITORIAL BOARD

BOOKS PUBLISHED IN FRONTIERS IN APPLIED MATHEMATICS

Vogel, Curtis R., *Computational Methods for Inverse Problems*

Lewis, F. L.; Campos, J.; and Selmic, R., *Neuro-Fuzzy Control of Industrial Systems with Actuator Nonlinearities*

Bao, Gang; Cowsar, Lawrence; and Masters, Wen, editors, *Mathematical Modeling in Optical Science*

Banks, H. T.; Buksas, M. W.; and Lin, T., *Electromagnetic Material Interrogation Using Conductive Interfaces and Acoustic Wavefronts*

Oostveen, Job, *Strongly Stabilizable Distributed Parameter Systems*

Griewank, Andreas, *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*

Kelley, C. T., *Iterative Methods for Optimization*

Greenbaum, Anne, *Iterative Methods for Solving Linear Systems*

Kelley, C. T., *Iterative Methods for Linear and Nonlinear Equations*

Bank, Randolph E., *PLTMG: A Software Package for Solving Elliptic Partial Differential Equations. Users' Guide 7.0*

Moré, Jorge J. and Wright, Stephen J., *Optimization Software Guide*

Rüde, Ulrich, *Mathematical and Computational Techniques for Multilevel Adaptive Methods*

Cook, L. Pamela, *Transonic Aerodynamics: Problems in Asymptotic Theory*

Banks, H. T., *Control and Estimation in Distributed Parameter Systems*

Van Loan, Charles, *Computational Frameworks for the Fast Fourier Transform*

Van Huffel, Sabine and Vandewalle, Joos, *The Total Least Squares Problem: Computational Aspects and Analysis*

Castillo, José E., *Mathematical Aspects of Numerical Grid Generation*

Bank, R. E., *PLTMG: A Software Package for Solving Elliptic Partial Differential Equations. Users' Guide 6.0*

McCormick, Stephen F., *Multilevel Adaptive Methods for Partial Differential Equations*

Grossman, Robert, *Symbolic Computation: Applications to Scientific Computing*

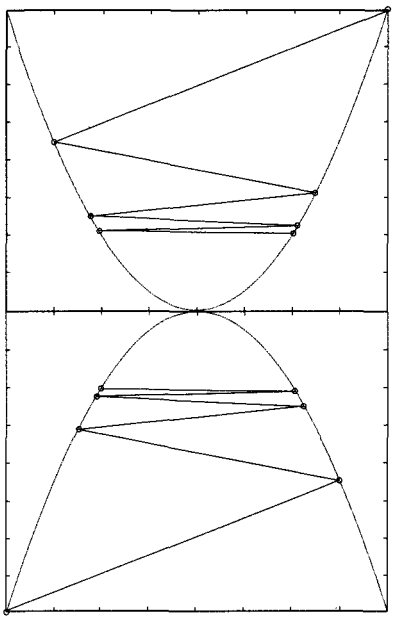
Coleman, Thomas F. and Van Loan, Charles, *Handbook for Matrix Computations*

McCormick, Stephen F., *Multigrid Methods*

Buckmaster, John D., *The Mathematics of Combustion*

Ewing, Richard E., *The Mathematics of Reservoir Simulation*

Computational Methods for Inverse Problems



Curtis R. Vogel
Montana State University
Bozeman, Montana

siam
Society for Industrial and Applied Mathematics
Philadelphia

10 9 8 7 6 5 4 3 2 1

Library of Congress Cataloging-in-Publication Data

Vogel, Curtis R.
Computational methods for inverse problems / Curtis R. Vogel.
p. cm.— (Frontiers in applied mathematics)
Includes bibliographical references and index.
ISBN 0-89871-507-5
I. Inverse problems (Differential equations)—Numerical solutions. I. Title.
II. Series.

QA377 .V575 2002
515'.35 — dc21

2002022386

To my father,

Fred Nickey Vogel

This page intentionally left blank

Contents

Foreword

Inverse problems are ubiquitous in science and engineering and have rightfully received a great deal of attention by applied mathematicians, statisticians, and engineers. Since most inverse problems cannot be solved analytically, computational methods play a fundamental role. The present volume is a research level introduction to a large class of techniques developed over the past several decades to treat inverse problems primarily formulated in the context of convolution-type Fredholm integral equations Kf=d which must be inverted. Discretization (desirable for solution on digital computers) leads to an undesirable ill-posedness in matrix equation inversions. Motivated by an image reconstruction example, the author treats both deterministic and statistical aspects of computational methods (a wide range of numerical optimization techniques are included) with a strong focus on regularization. Statistical aspects are treated in terms of model uncertainty (in K) and measurement error (noisy data d).

It is not surprising that there is a large mathematical literature on inverse problem methods. What might be surprising is that this literature is significantly divided along deterministic/nondeterministic lines. Methods abound in the statistics literature, where generally the models are assumed quite simple (and often even analytically known!) and the emphasis is on treating statistical aspects of fitting models to data. On the other hand, the applied mathematical literature has a plethora of increasingly complex parameterized models (nonlinear ordinary differential equations, partial differential equations, and delay equations) which are treated theoretically and computationally in a deterministic framework with little or no attention to inherent uncertainty in either the modeled mechanisms or the data used to validate the models. The present monograph is a successful attempt to treat certain probabilistic aspects of a class of inverse problems. It is a research monograph and as such is not meant to be a comprehensive treatment of statistical methods in inverse problems. For example, it does not treat models with random parameters in complex systems, mixed or random effects, mixing distributions, etc. (e.g., see M. Davidian and D. Giltinan, Nonlinear Models for Repeated Measurement Data, Monographs on Statistics and Applied Probability 62 (1998), Chapman & Hall/CRC, Boca Raton, FL), or statistical methods (e.g., ANOVA) associated with model validation. It is, however, a most welcome addition and just the first of what the editors hope will be several volumes treating randomness and uncertainty in computational aspects of inverse or parameter estimation problems.

H. T. Banks
Center for Research in Scientific Computation
North Carolina State University
Raleigh, N. C.

This page intentionally left blank

Preface

The field of inverse problems has experienced explosive growth in the last few decades. This is due in part to the importance of applications, like biomedical and seismic imaging, that require the practical solution of inverse problems. It is also due to the recent development of powerful computers and fast, reliable numerical methods with which to carry out the solution process. This monograph will provide the reader with a working understanding of these numerical methods. The intended audience includes graduate students and researchers in applied mathematics, engineering, and the physical sciences who may encounter inverse problems in their work.

Inverse problems typically involve the estimation of certain quantities based on indirect measurements of these quantities. For example, seismic exploration yields measurements of vibrations recorded on the earth's surface. These measurements are only indirectly related to the subsurface geological formations that are to be determined. The estimation process is often ill-posed in the sense that noise in the data may give rise to significant errors in the estimate. Techniques known as regularization methods have been developed to deal with this ill-posedness.

The first four chapters contain background material related to inverse problems, regularization, and numerical solution techniques. Chapter 1 provides an informal overview of a variety of regularization methods. Chapter 2 is guided by the philosophy that sensible numerical solutions to discretized problems follow from a thorough understanding of the underlying continuous problems. This chapter contains relevant functional analysis and infinite-dimensional optimization theory. Chapter 3 contains a review of relevant numerical optimization methods. Chapter 4 presents statistics material that pertains to inverse problems. This includes topics like maximum likelihood estimation and Bayesian estimation.

The remaining five chapters address more specialized topics. Emphasis is placed on the two-dimensional image reconstruction problem, which is introduced in Chapter 5. While this problem is quite simple to formulate and to visualize, its solution draws on a variety of fairly sophisticated tools, including mathematical modeling, statistical estimation theory, Fourier transform methods, and large-scale optimization and numerical linear algebra. This problem also provides motivation and a test case for more specialized techniques like total variation regularization (see Chapter 8) and nonnegativity constraints (Chapter 9).

Chapter 6 contains a brief introduction to parameter (i.e., coefficient) identification for differential equations. This topic serves to introduce nonlinear optimization techniques like the Gauss–Newton and Levenberg–Marquardt methods for nonlinear least squares. It also provides motivation for adjoint, or costate, methods for the efficient computation of gradients and higher order derivatives.

Chapter 7 covers the important topic of regularization parameter selection from a statistical perspective. This chapter includes practical implementations as well as a theoretical analysis of several of the more popular regularization parameter selection methods.

Several web-based resources are available (@http://www.siam.org/books/fr23) to make this monograph somewhat interactive. One of these resources is a collection of MATLAB* m-files used to generate many of the examples and figures. This enables readers to conduct their own computational experiments to gain insight and build intuition. It also provides templates for the implementation of regularization methods and numerical solution techniques for other inverse problems. Moreover, it provides some realistic test problems to be used to further develop and test various numerical methods.

Also available on the web are a list of errata and comments and a collection of solutions to some of the exercises.

I would like to express special gratitude to my colleagues and close friends Martin Hanke, Jim Nagy, Bob Plemmons, and Brian Thelen. These individuals have strongly influenced my research career, and they have made many helpful comments and corrections to preliminary versions of the manuscript. I also owe thanks to a number of other colleagues who reviewed parts of the manuscript. These include John Bardsley, Warren Esty, Luc Gilles, Eldad Haber, Per Christian Hansen, Thomas Scofield, and Lisa Stanley. Finally, I would like to thank Tom Banks and the editorial staff at SIAM for their patience and assistance.

Curt Vogel
Bozeman, Montana

*MATLAB is a registered trademark of The MathWorks, Inc., 3 Apple Hill Drive, Natick, MA 01760, USA, tel. 508-647-7000, fax 508-647-7001, info@mathworks.com, http://www.mathworks.com.

Chapter 1
Introduction

Inverse problems arise in a variety of important applications in science and industry. These range from biomedical and geophysical imaging to groundwater flow modeling. See, for example, [6, 7, 35, 70, 87, 90, 107, 108] and the references therein. In these applications the goal is to estimate some unknown attributes of interest, given measurements that are only indirectly related to these attributes. For instance, in medical computerized tomography, one wishes to image structures within the body from measurements of X-rays that have passed through the body. In groundwater flow modeling, one estimates material parameters of an aquifer from measurements of pressure of a fluid that immerses the aquifer. Unfortunately, a small amount of noise in the data can lead to enormous errors in the estimates. This instability phenomenon is called ill-posedness. Mathematical techniques known as regularization methods have been developed to deal with ill-posedness. This chapter introduces the reader to the concepts ill-posedness and regularization. Precise definitions are given in the next chapter.

1.1 An Illustrative Example

Consider the Fredholm first kind integral equation of convolution type in one space dimension:

$$g(x) = \int_0^1 k(x - x')\, f(x')\, dx' \stackrel{\text{def}}{=} (\mathcal{K}f)(x), \qquad 0 < x < 1. \tag{1.1}$$

This is a one-dimensional version of a model that occurs in two-dimensional optical imaging and is discussed in more detail in Chapter 5. In this application, f represents light source intensity as a function of spatial position, and g represents image intensity. The kernel k characterizes blurring effects that occur during image formation. A kernel that models the long-time average effects of atmospheric turbulence on light propagation is the Gaussian [7, 99]. Its one-dimensional version is

$$k(x) = C\, \exp(-x^2/2\gamma^2), \tag{1.2}$$

where C and γ are positive parameters.

The direct problem, or forward problem, associated with the model equation (1.1) is the following: Given the source f and the kernel k, determine the blurred image g. Figure 1.1 shows the blurred image corresponding to a piecewise smooth source. Since k is a smooth

function, the accurate approximation of $g = \mathcal{K}f$ using standard numerical quadrature is straightforward.

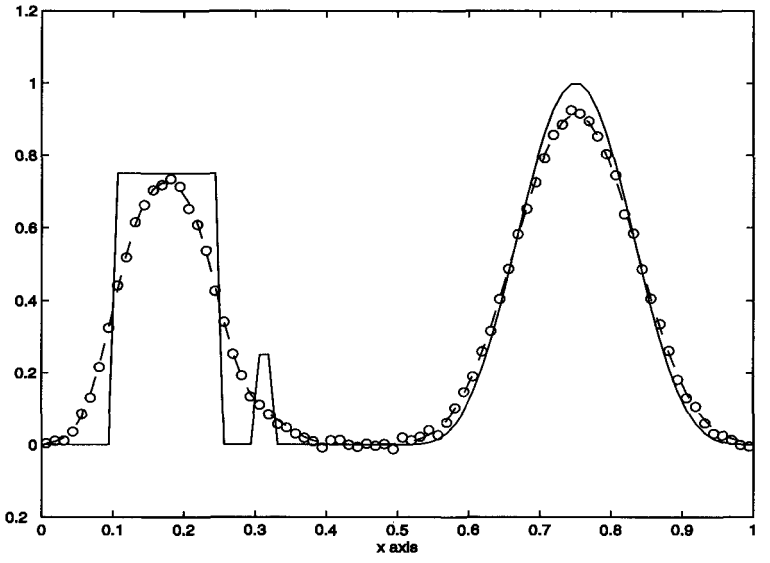


Figure 1.1. *One-dimensional image data. The source function f is represented by the solid line, the blurred image $g = \mathcal{K}f$ is represented by the dashed line, and the discrete noisy data $\mathbf{d}$ is represented by circles. The data were generated according to* (1.1)–(1.4) *with parameters $\gamma = 0.05$ and $C = 1/(\gamma\sqrt{2\pi})$. Midpoint quadrature was used to approximate integrals.*

An associated inverse problem of practical interest is as follows: Given the kernel k and the blurred image g, determine the source f. At first glance, the approximate solution to this inverse problem seems straightforward. One may simply discretize equation (1.1), e.g., using collocation in the independent variable x and quadrature in x', to obtain a discrete linear system $K\mathbf{f} = \mathbf{d}$. For instance, if midpoint quadrature is applied, then K has entries

$$[K]_{ij} = h\ C\ \exp\left(-\frac{((i-j)h)^2}{2\gamma^2}\right), \qquad 1 \le i, j \le n, \tag{1.3}$$

where $h = 1/n$. If the matrix K is nonsingular, one may then compute the discrete approximation $K^{-1}\mathbf{d}$ to f. To obtain an accurate quadrature approximation, n must be relatively large. Unfortunately, the matrix K becomes increasingly ill-conditioned as n becomes large, so errors in $\mathbf{d}$ may be greatly amplified. Certain errors, like those due to quadrature, can be controlled. Others, like the noise in the image recording device, cannot be controlled in a practical setting. Consequently, this straightforward solution approach is likely to fail.

1.2 Regularization by Filtering

Despite ill-conditioning, one can extract some useful information from the discrete linear system $K\mathbf{f} = \mathbf{d}$. To simplify the presentation, consider a discrete data model

$$\mathbf{d} = K\mathbf{f}_{\text{true}} + \eta \tag{1.4}$$

with

$$\delta \stackrel{\text{def}}{=} ||\eta|| > 0. \tag{1.5}$$

Here $||\cdot||$ denotes standard Euclidean norm, $\mathbf{f}_{\text{true}}$ represents the true discretized source, and η represents error in the data. The parameter δ is called the error level. For further simplicity, assume K is an invertible, real-valued matrix. It then has a singular value decomposition (SVD) [46],

$$K = U\ \text{diag}(s_i)\ V^T, \tag{1.6}$$

with strictly positive decreasing singular values s_i. The SVD and its connection with inverse problems are discussed in more detail in the next chapter; cf. Definition 2.15, and see [58]. At this point we require the following facts: The column vectors $\mathbf{v}_i$ of V, which are called right singular vectors, and the column vectors $\mathbf{u}_i$ of U, which are the left singular vectors, satisfy

$$\mathbf{u}_i^T\mathbf{u}_j = \delta_{ij}, \qquad \mathbf{v}_i^T\mathbf{v}_j = \delta_{ij}, \tag{1.7}$$

$$K\mathbf{v}_i = s_i\mathbf{u}_i, \qquad K^T\mathbf{u}_i = s_i\mathbf{v}_i. \tag{1.8}$$

Here δ_{ij} denotes the Kronecker delta (equation (2.2)), and $U^T = U^{-1}$ and $V^T = V^{-1}$. Note that if K is symmetric and positive definite, then the singular values s_i are the eigenvalues of K, and $U = V$ has columns consisting of orthonormalized eigenvectors. The singular values and vectors for our discretized one-dimensional imaging problem are represented graphically in Figure 1.2.

Using properties (1.7)–(1.8),

$$K^{-1}\mathbf{d} \;=\; V\,\text{diag}(s_i^{-1})\,U^T\mathbf{d} \;=\; \mathbf{f}_{\text{true}} + \sum_{i=1}^{n} s_i^{-1}(\mathbf{u}_i^T\eta)\mathbf{v}_i. \tag{1.9}$$

Instability arises due to division by small singular values. One way to overcome this instability is to modify the s_i^{-1}'s in (1.9), e.g., by multiplying them by a regularizing filter function $w_\alpha(s_i^2)$ for which the product $w_\alpha(s^2)s^{-1} \to 0$ as $s \to 0$. This filters out singular components of $K^{-1}\mathbf{d}$ corresponding to small singular values and yields an approximation to $\mathbf{f}_{\text{true}}$ with a representation

$$\begin{aligned}\mathbf{f}_\alpha &= V\,\text{diag}(w_\alpha(s_i^2)s_i^{-1})\,U^T\mathbf{d} \\ &= \sum_{i=1}^{n} w_\alpha(s_i^2)s_i^{-1}(\mathbf{u}_i^T\mathbf{d})\mathbf{v}_i.\end{aligned} \tag{1.10}$$

To obtain some degree of accuracy, one must retain singular components corresponding to large singular values. This is done by taking $w_\alpha(s^2) \approx 1$ for large values of s^2. An example of such a filter function is

$$w_\alpha(s^2) = \begin{cases} 1 & \text{if} \quad s^2 > \alpha, \\ 0 & \text{if} \quad s^2 \le \alpha. \end{cases} \tag{1.11}$$

The approximation (1.10) then takes the form

$$\mathbf{f}_\alpha = \sum_{s_i^2 > \alpha} s_i^{-1}(\mathbf{u}_i^T\mathbf{d})\mathbf{v}_i \tag{1.12}$$

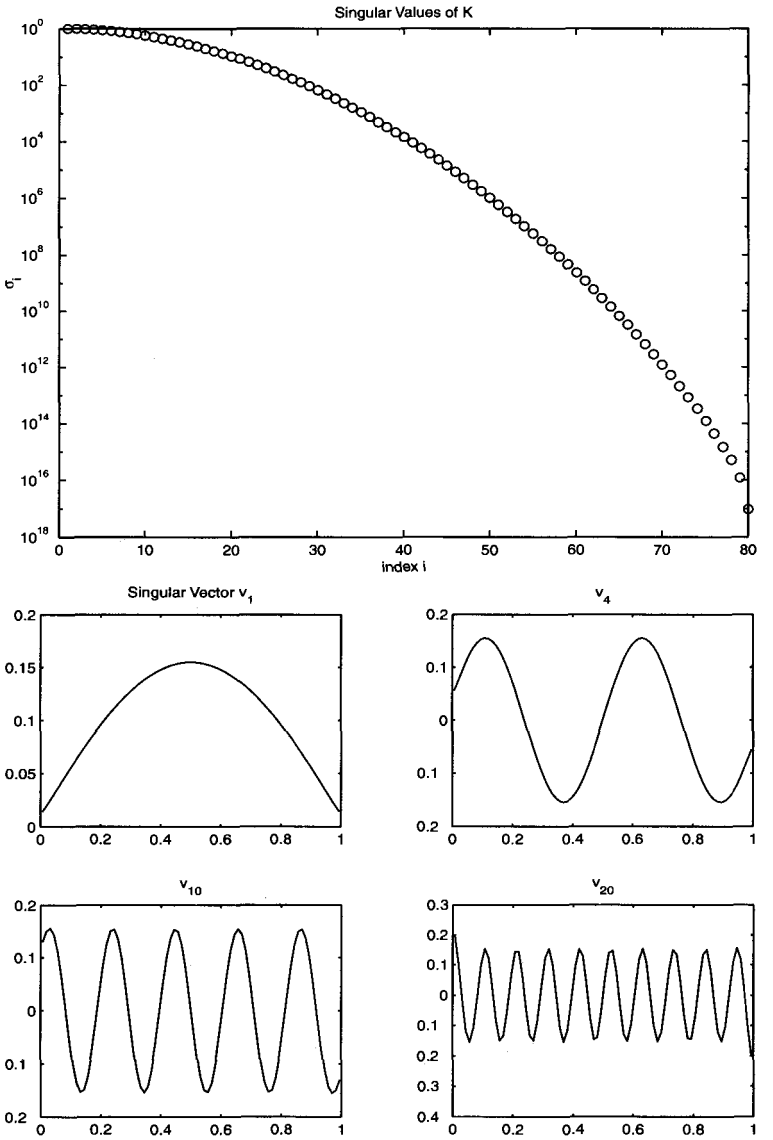


Figure 1.2. *SVD of the matrix K having ijth entry $k(x_i - x_j)\,h$, where $h = 1/n$, with $n = 80$ and $x_i = (i + 1/2)h$, $i = 1, \ldots, n$. The top plot shows the distribution of the singular values s_i of K. The subplots below it show a few of the corresponding singular vectors $\mathbf{v}_i$. (K is symmetric, so the left and right singular vectors are the same.) The components $[\mathbf{v}_i]_j$ are plotted against the x_j's. At the middle left is the singular vector $\mathbf{v}_1$ corresponding to the largest singular value s_1 of K. At the middle right is the singular vector $\mathbf{v}_4$ corresponding to the fourth largest singular value s_4. The bottom left and bottom right subplots show, respectively, the singular vectors corresponding to the* 10th *and* 20th *largest singular values.*

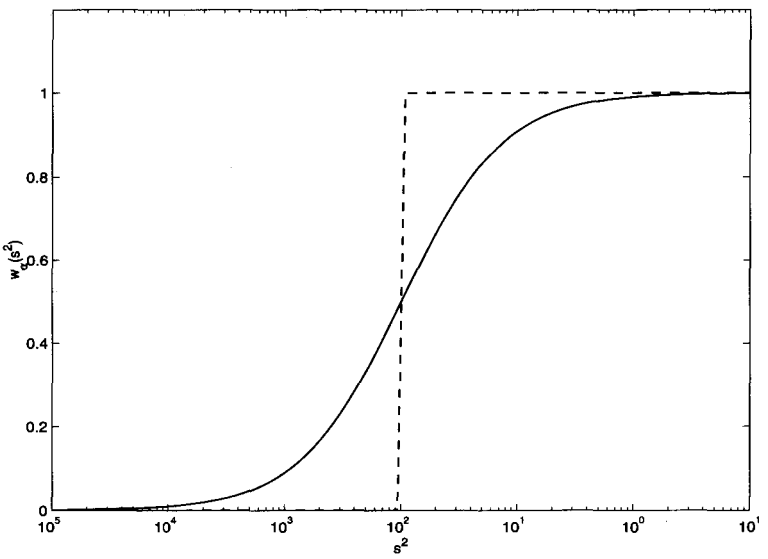


Figure 1.3. *Semilog plots of filter functions w_α corresponding to TSVD regularization (dashed line) and Tikhonov regularization (solid line) as functions of squared singular values s^2. The value of the regularization parameter is $\alpha = 10^{-2}$.*

and is known as the truncated SVD (TSVD) solution to $K\mathbf{f} = \mathbf{d}$. Another example is the Tikhonov filter function:

$$w_\alpha(s^2) = \frac{s^2}{s^2 + \alpha}. \tag{1.13}$$

The corresponding regularized approximation (1.10) can be expressed as

$$\mathbf{f}_\alpha = \sum_{i=1}^{n} \frac{s_i(\mathbf{u}_i^T\mathbf{d})}{s_i^2 + \alpha}\mathbf{v}_i \tag{1.14}$$

$$= (K^TK + \alpha I)^{-1}K^T\mathbf{d}. \tag{1.15}$$

Equation (1.15) is a consequence of (1.6)–(1.8). See Exercise 1.3. This yields a technique known as Tikhonov(–Phillips) regularization [106, 93].

The α in (1.11) and (1.13) is called a regularization parameter. In (1.11) this parameter determines the cut-off, or threshold, level for the TSVD filter. From the plots of the filter functions in Figure 1.3, it can be seen that the regularization parameter for Tikhonov regularization in (1.13) plays a similar role. Figure 1.4 illustrates how the TSVD solution $\mathbf{f}_\alpha$ varies with α. Similar behavior can be observed when Tikhonov regularization is applied. When α is very small, filtering of the noise is inadequate and $\mathbf{f}_\alpha$ is highly oscillatory. On the other hand, when α is large, the noise components are filtered out. Unfortunately, most components of the solution are also filtered out, and $\mathbf{f}_\alpha$ is overly smooth.

An obvious question arises: Can the regularization parameter be selected to guarantee convergence as the error level goes to zero? The answer to this question lies in the following analysis.

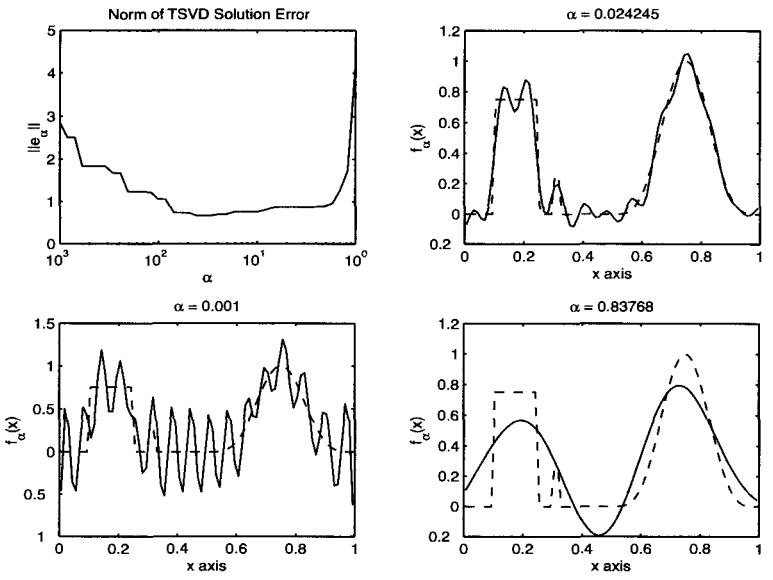


Figure 1.4. *TSVD regularized solutions. The upper left subplot shows the norm of the solution error,* $||\mathbf{f}_\alpha - \mathbf{f}_{\text{true}}||$, *versus the regularization parameter* α. *In the upper right subplot, the solid line represents the regularized solution* $\mathbf{f}_\alpha$ *for* $\alpha = 0.0242$. *This value of* α *minimizes the solution error norm. The lower left and lower right subplots, respectively, show* $\mathbf{f}_\alpha$ *for* $\alpha = 0.001$ *and* $\alpha = 0.50119$. *The dashed curves represent the true solution* $\mathbf{f}_{\text{true}}$.

1.2.1 A Deterministic Error Analysis

The right-hand side of (1.10) defines a linear regularization operator, which we denote by R_α. Hence $f_\alpha = R_\alpha \mathbf{d}$. From (1.4), the regularized solution error is given by

$$\begin{aligned} \mathbf{e}_\alpha &\stackrel{\text{def}}{=} \mathbf{f}_\alpha - \mathbf{f}_{\text{true}} \\ &= \mathbf{e}_\alpha^{\text{trunc}} + \mathbf{e}_\alpha^{\text{noise}}, \end{aligned} \tag{1.16}$$

where

$$\begin{aligned} \mathbf{e}_\alpha^{\text{trunc}} &\stackrel{\text{def}}{=} R_\alpha K \mathbf{f}_{\text{true}} - \mathbf{f}_{\text{true}} \\ &= \sum_{i=1}^{n} (w_\alpha(s_i^2) - 1)(\mathbf{v}_i^T \mathbf{f}_{\text{true}})\mathbf{v}_i \end{aligned} \tag{1.17}$$

and

$$\begin{aligned} \mathbf{e}_\alpha^{\text{noise}} &\stackrel{\text{def}}{=} R_\alpha \boldsymbol{\eta} \\ &= \sum_{i=1}^{n} w_\alpha(s_i^2) s_i^{-1} (\mathbf{u}_i^T \boldsymbol{\eta})\mathbf{v}_i. \end{aligned} \tag{1.18}$$

We call $\mathbf{e}_\alpha^{\text{trunc}}$ the solution truncation error due to regularization. It quantifies the loss of information due to the regularizing filter. The term $\mathbf{e}_\alpha^{\text{noise}}$ is called the noise amplification error. We will show that for both the TSVD filter (1.11) and the Tikhonov filter (1.13), the

regularization parameter α can be selected in a manner that guarantees that both these errors converge to zero as the error level $\delta \to 0$.

We first consider the truncation error. For both the TSVD filter (1.11) and the Tikhonov filter (1.13), for any fixed $s > 0$,

$$w_\alpha(s^2) \to 1 \quad \text{as} \quad \alpha \to 0, \tag{1.19}$$

and hence from (1.17),

$$\mathbf{e}_\alpha^{\text{trunc}} \to 0 \quad \text{whenever} \quad \alpha \to 0. \tag{1.20}$$

To deal with the noise amplification error, one can show (see Exercise 1.5) that both filter functions satisfy

$$w_\alpha(s^2)\, s^{-1} \leq \alpha^{-1/2}. \tag{1.21}$$

Consequently, from (1.18) and (1.5),

$$||\mathbf{e}_\alpha^{\text{noise}}|| \leq \alpha^{-1/2}\delta. \tag{1.22}$$

One can obtain $||\mathbf{e}_\alpha^{\text{noise}}|| \to 0$ as $\delta \to 0$ by choosing $\alpha = \delta^p$ with $p < 2$. If in addition $p > 0$, then (1.20) also holds. Since $\mathbf{e}_\alpha = \mathbf{e}_\alpha^{\text{trunc}} + \mathbf{e}_\alpha^{\text{noise}}$, the regularization parameter choice

$$\alpha = \delta^p, \quad 0 < p < 2, \tag{1.23}$$

guarantees that

$$\mathbf{e}_\alpha \to 0 \quad \text{as} \quad \delta \to 0 \tag{1.24}$$

when either TSVD or Tikhonov regularization is applied to data (1.4). A regularization method together with a parameter selection rule like (1.23) are called convergent if (1.24) holds.

1.2.2 Rates of Convergence

Consider the TSVD filter (1.11), and assume $\delta \geq s_n^2$, the square of the smallest singular value. If this assumption doesn't hold, then noise amplification is tolerable even without regularization. To obtain a convergence rate for the solution error, one needs bounds on the truncation error. Assume

$$\mathbf{f}_{\text{true}} = K^T\mathbf{z}, \qquad \mathbf{z} \in \mathbb{R}^n. \tag{1.25}$$

This is an example of a so-called source condition [35] or range condition. Since $|(K^T\mathbf{z})^T\mathbf{v}_i| = |\mathbf{z}^T K\mathbf{v}_i| = s_i|\mathbf{z}^T\mathbf{u}_i|$, one obtains

$$\begin{aligned} ||\mathbf{e}_\alpha^{\text{trunc}}||^2 &= \sum_{i=1}^{n} (w_\alpha(s_i^2) - 1)^2 s_i^2 |\mathbf{z}^T\mathbf{u}_i|^2 \\ &\leq \max_{1\leq i\leq n} (w_\alpha(s_i^2) - 1)^2 s_i^2 ||\mathbf{z}||^2 \\ &\leq \alpha||\mathbf{z}||^2. \end{aligned} \tag{1.26}$$

See Exercise 1.7. This bound is sharp in the sense that the inequality becomes an equality for certain combinations of α and the s_i's. See Exercise 1.8. Combining equation (1.26) with (1.22) gives

$$||\mathbf{e}_\alpha|| \leq \alpha^{1/2}||\mathbf{z}|| + \alpha^{-1/2}\delta. \tag{1.27}$$

The right-hand side is minimized by taking

$$\alpha = \frac{\delta}{||\mathbf{z}||}. \tag{1.28}$$

This yields

$$||\mathbf{e}_\alpha|| \leq 2\,||\mathbf{z}||^{1/2}\,\delta^{1/2}. \tag{1.29}$$

A regularization method together with a parameter choice rule for which $||\mathbf{e}_\alpha|| = \mathcal{O}(\sqrt{\delta})$ as $\delta \to 0$ is called order optimal given the information $\mathbf{f}_{\text{true}} \in \text{Range}(K^T)$. We have established that TSVD regularization with the parameter choice (1.29) is order optimal given $\mathbf{f}_{\text{true}} \in \text{Range}(K^T)$.

In a continuous setting, $f_{\text{true}} \in \text{Range}(K^T)$ is a condition on the smoothness of f_{true}. This conclusion carries over to the discrete case, although with less conciseness. From Figure 1.2, singular vectors corresponding to very small singular values are highly oscillatory. If $||\mathbf{z}||$ is not large, then either $\mathbf{f}_{\text{true}}$ is very small or the singular expansion of $\mathbf{f}_{\text{true}}$ is dominated by singular components corresponding to large singular values, and these components are smoothly varying.

Equation (1.28) is an a priori regularization parameter choice rule. It requires prior information about both the data noise level δ and the true solution, through assumption (1.25) and quantity $||\mathbf{z}||$. In practice, such prior information about the true solution is unlikely to be available.

1.2.3 A Posteriori Regularization Parameter Selection

A parameter choice rule is called a posteriori if the selection of the regularization parameter depends on the data but not on prior information about the solution. One such rule is the discrepancy principle due to Morozov [86], where in the case of either TSVD or Tikhonov regularization one selects the largest value of the regularization parameter α for which

$$||K\mathbf{f}_\alpha - \mathbf{d}|| \leq \delta. \tag{1.30}$$

It can be shown that both TSVD and Tikhonov regularization with the discrepancy principle parameter choice rule are convergent, and, assuming the source condition (1.25), both are order optimal [48, 35, 70].

What follows is a finite-dimensional version of the analysis for the discrepancy principle applied to Tikhonov regularization. To simplify notation, define the data discrepancy functional

$$D(\alpha) = ||K\mathbf{f}_\alpha - \mathbf{d}||.$$

We first establish conditions under which there exists a value of the regularization parameter for which $D(\alpha) = \delta$. From the representation (1.15) and the SVD (1.6)–(1.8),

$$\begin{aligned} D^2(\alpha) &= ||(I - K(K^TK + \alpha I)^{-1}K^T)\mathbf{d}||^2 \\ &= \sum_{i=1}^{n} \left(1 - \frac{s_i^2}{s_i^2 + \alpha}\right)^2 (\mathbf{u}_i^T\mathbf{d})^2. \end{aligned} \tag{1.31}$$

Thus $D(\alpha)$ is continuous and strictly increasing with $D(0) = 0$ and $D(\alpha) \to ||\mathbf{d}||$ as $\alpha \to \infty$. Thus $D(\alpha) = \delta$ has a unique solution provided that the data noise level is less than the data norm,

(1.32) $$\delta < ||\mathbf{d}||.$$

Assume that condition (1.32) holds, and let $\alpha(\delta)$ denote the unique solution to $D(\alpha) = \delta$. Then

(1.33) $$||\mathbf{f}_{\alpha(\delta)}|| \leq ||\mathbf{f}_{\text{true}}||.$$

To verify this,

$$\begin{aligned}\delta^2 + \alpha||\mathbf{f}_{\alpha(\delta)}||^2 &= ||K\mathbf{f}_{\alpha(\delta)} - \mathbf{d}||^2 + \alpha||\mathbf{f}_{\alpha(\delta)}||^2 \\ &\leq ||K\mathbf{f}_{\text{true}} - \mathbf{d}||^2 + \alpha||\mathbf{f}_{\text{true}}||^2 \\ &= \delta^2 + \alpha||\mathbf{f}_{\text{true}}||^2.\end{aligned}$$

The inequality follows from the variational representation (1.34) in section 1.3 (see Exercise 1.13), while the last equality follows from the data model (1.4)–(1.5).

Finally, we establish order optimality. Following the proof of Theorem 3.3 in [48],

$$\begin{aligned}||\mathbf{f}_{\alpha(\delta)} - \mathbf{f}_{\text{true}}||^2 &= ||\mathbf{f}_{\alpha(\delta)}||^2 - 2\mathbf{f}_{\alpha(\delta)}^T\mathbf{f}_{\text{true}} + ||\mathbf{f}_{\text{true}}||^2 \\ &\leq 2||\mathbf{f}_{\text{true}}||^2 - 2\mathbf{f}_{\alpha(\delta)}^T\mathbf{f}_{\text{true}}, \quad \text{by} \quad (1.33) \\ &= 2(\mathbf{f}_{\text{true}} - \mathbf{f}_{\alpha(\delta)})^T K^T\mathbf{z}, \quad \text{by} \quad (1.25) \\ &= 2(K\mathbf{f}_{\text{true}} - \mathbf{d} + \mathbf{d} - K\mathbf{f}_{\alpha(\delta)})^T\mathbf{z} \\ &\leq 4\delta\ ||\mathbf{z}||.\end{aligned}$$

The last inequality follows from the Cauchy–Schwarz inequality, the triangle inequality, and the fact that $D(\alpha(\delta)) = ||K\mathbf{f}_{\text{true}} - \mathbf{d}|| = \delta$.

1.3 Variational Regularization Methods

For very large ill-conditioned systems, it is often impractical to directly implement regularization by filtering, since the representation (1.10) requires the SVD of a large matrix. However, the Tikhonov solution (1.14) has an alternate variational representation,

(1.34) $$\mathbf{f}_\alpha = \arg\min_{\mathbf{f}\in\mathbb{R}^n} ||K\mathbf{f} - \mathbf{d}||^2 + \alpha||\mathbf{f}||^2,$$

which may be easier to compute. This representation may have other advantages. For instance, in optics the source intensity f is nonnegative. Nonnegativity can be imposed as a constraint in (1.34). Moreover, the least squares term $||K\mathbf{f} - \mathbf{d}||^2$ can be replaced by other fit-to-data functionals. See section 4.2 for specific examples. The term $||\mathbf{f}||^2$ in (1.34) is called a penalty functional. Other penalty functionals can be used to incorporate a priori information. An example is the discrete one-dimensional total variation

$$\begin{aligned}TV(\mathbf{f}) &= \sum_{i=1}^{n-1} |f_{i+1} - f_i| \\ &= \sum_{i=1}^{n-1} \left|\frac{f_{i+1} - f_i}{\Delta x}\right| \Delta x. \qquad (1.35)\end{aligned}$$

This penalizes highly oscillatory solutions while allowing jumps in the regularized solution. Note that for smooth f, the sum in (1.35) approximates the L^1 norm of the derivative, a nonquadratic function of f. Figure 1.5 illustrates that the reconstructions obtained with total variation can be qualitatively quite different from those obtained with methods like TSVD, (see Figure 1.4). Unfortunately, total variation reconstructions are much more difficult to compute. See Chapter 8 for further details.

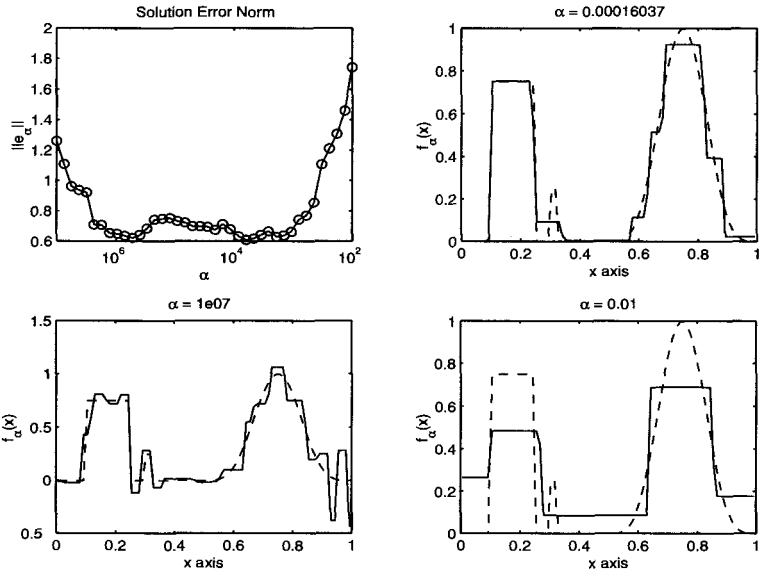


Figure 1.5. *One-dimensional total variation regularized solutions. The upper left subplot shows the norm of the solution error,* $\|\mathbf{f}_\alpha - \mathbf{f}_{\text{true}}\|$, *versus the regularization parameter* α. *In the upper right subplot, the solid line represents the regularized solution* $\mathbf{f}_\alpha$ *for* $\alpha = 1.604 \times 10^{-4}$. *This value of* α *minimizes the solution error norm. The lower left and lower right subplots, respectively, show* $\mathbf{f}_\alpha$ *for* $\alpha = 10^{-6}$ *and* $\alpha = 1.0$. *The dashed curves represent the true solution* $\mathbf{f}_{\text{true}}$.

1.4 Iterative Regularization Methods

We illustrate the concept of iterative regularization with a simple example. Consider the scaled least squares fit-to-data functional

$$J(\mathbf{f}) = \frac{1}{2}\|K\mathbf{f} - \mathbf{d}\|^2. \tag{1.36}$$

This has as its gradient $\text{grad}\, J(\mathbf{f}) = K^T(K\mathbf{f} - \mathbf{d})$. Consider the iteration

$$\mathbf{f}_{\nu+1} = \mathbf{f}_\nu - \tau\, \text{grad}\, J(\mathbf{f}_\nu), \qquad \nu = 0, 1, \ldots. \tag{1.37}$$

If at each iteration ν the scalar τ is chosen to minimize $\tilde{J}(\tau) = J(\mathbf{f}_\nu - \tau\, \text{grad}\, J(\mathbf{f}_\nu))$, then one obtains the steepest descent method. See section 3.1 for details. If one fixes τ with $0 < \tau < 1/\|K\|^2$, one obtains a method known as Landweber iteration [72]. With either choice of τ, if one takes the initial guess $\mathbf{f}_0 = \mathbf{0}$ and one assumes that K is invertible, one can show that the iterates $\mathbf{f}_\nu$ converge to $\mathbf{f}_* = K^{-1}\mathbf{d}$. This is not desirable if error is present in the data. From the plot of solution error norm versus iteration count ν shown in Figure 1.6, we see that the iteration count appears to play the role of a regularization parameter. Very small values of ν yield overly smooth approximate solutions. On the other hand, as ν becomes large, the reconstructions become highly oscillatory. This phenomenon is called semiconvergence.

To explain this phenomenon, one can show that when $\mathbf{f}_0 = \mathbf{0}$, $\mathbf{f}_\nu$ has the representation (1.10) with the filter function

$$w_\nu(s^2) = 1 - (1 - \tau s^2)^\nu. \tag{1.38}$$

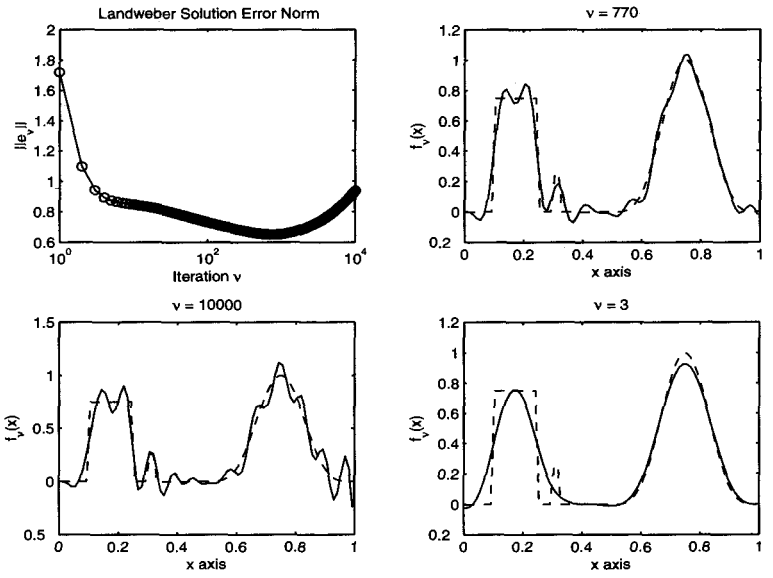


Figure 1.6. *Results for Landweber iteration. The upper left subplot shows the norm of the solution error,* $||\mathbf{f}_\nu - \mathbf{f}_{\text{true}}||$, *versus the iteration count* ν. *In the upper right subplot, the solid line represents the regularized solution* $\mathbf{f}_\nu$ *for* $\nu = 766$. *This value of* ν *minimizes the solution error norm. The lower left and lower right subplots, respectively, show* $\mathbf{f}_\nu$ *for* $\nu = 10^4$ *and* $\nu = 3$. *The dashed curves represent the true solution* $\mathbf{f}_{\text{true}}$.

See Exercise 1.15. The iteration count ν is indeed a regularization parameter. One can show [35, section 6.1] that the discrepancy principle is order optimal for Landweber iteration. Unfortunately, the method tends to require many iterations to generate accurate regularized solutions, thereby limiting its practical use.

Exercises

1.1. Verify that when midpoint quadrature is applied to the integral operator $\mathcal{K}$ in (1.1)–(1.2), one obtains matrix K in (1.3).

1.2. Use properties (1.7)–(1.8) to obtain (1.9).

1.3. Use the decomposition (1.6)–(1.8) to confirm the equality (1.14)–(1.15).

1.4. Confirm equations (1.16)–(1.18) using (1.10) and (1.4).

1.5. Verify that equation (1.21) is satisfied for both the TSVD filter function (1.11) and the Tikhonov filter function (1.13).

1.6. Using (1.18) and (1.5), show that equation (1.22) holds.

1.7. Confirm the inequality (1.26).

1.8. Show that the inequality (1.26) is sharp. To do this, give the vector $\mathbf{z}$ for which equality holds.

1.9. Show that the right-hand side of (1.27) is minimized with the choice (1.28). Then confirm (1.29).

1.10. Mimic the analysis of section 1.2.2 to show that TSVD with $\alpha \sim \delta^{2/3}$ is order optimal for $\mathbf{f}_{\text{true}} \in \text{Range}(K^T K)$.

1.11. Show that for the continuous operator (1.1) the source condition $f_{\text{true}} = K^* z$, $z \in L^2(0, 1)$, implies that f_{true} is smooth.

1.12. Confirm that the operator representation (1.15) is equivalent to the Tikhonov filter representation (1.10), (1.13). To do this, use properties of the SVD to verify that

$$(K^T K + \alpha I)^{-1} K^T \mathbf{d} = V \text{ diag}(s_i/(s_i^2 + \alpha))\, U^T \mathbf{d}.$$

1.13. Confirm that the variational representation (1.34) is equivalent to the Tikhonov filter representation (1.10), (1.13). Verify that

$$\begin{aligned} ||K\mathbf{f} - \mathbf{d}||^2 + \alpha||\mathbf{f}||^2 &= \mathbf{f}^T(K^T K + \alpha I)\mathbf{f} - 2\mathbf{f}^T K^T \mathbf{d} + ||\mathbf{d}||^2 \\ &= \tilde{\mathbf{f}}^T \text{diag}(s_i^2 + \alpha)\tilde{\mathbf{f}} - 2\tilde{\mathbf{f}}^T \text{diag}(s_i)\, \tilde{\mathbf{d}} + ||\mathbf{d}||^2, \end{aligned}$$

where $\tilde{\mathbf{f}} = V^T \mathbf{f}$ and $\tilde{\mathbf{d}} = U^T \mathbf{d}$. Then minimize with respect to $\tilde{\mathbf{f}}$ and take $\mathbf{f} = V\tilde{\mathbf{f}}$.

1.14. Numerically implement standard Tikhonov regularization for the test problem presented in Figure 1.1. The true solution is given by

$$f_{\text{true}}(x) = \begin{cases} 0.75, & 0.1 < x < 0.25, \\ 0.25, & 0.3 < x < 0.32, \\ \sin^4(2\pi x), & 0.5 < x < 1, \\ 0 & \text{otherwise.} \end{cases}$$

1.15. Confirm that Landweber iteration yields a representation (1.10) with filter function (1.38). To do this, show that $\mathbf{f}_\nu = \sum_{j=0}^{\nu-1} G^j \mathbf{b}$, where $G = I - \tau K^T K$ and $\mathbf{b} = \tau K^T \mathbf{d}$. Then apply the SVD.

1.16. Generate plots of the Landweber filter function $w_\nu(s^2)$ in equation (1.38). Let the independent variable s^2 range between 0 and 1. How does the behavior of $w_\nu(s^2)$ change as ν and τ vary?

Chapter 2

Analytical Tools

In the previous chapter we informally introduced concepts like ill-posedness and regularization. Our goal in this chapter is to rigorously define these concepts and to present some basic tools with which to analyze and solve ill-posed problems. We assume the reader has a solid background in linear algebra and has some exposure to real analysis. More advanced material from functional analysis is introduced here. This presentation is quite terse, with proofs either omitted or placed in the exercises. For a more systematic development of these topics, consult [71, 125, 126, 127, 128].

Notation

In this chapter, Ω denotes a simply connected, nonempty, open set in $\mathbb{R}^n$ that has a Lipschitz continuous boundary, denoted by $\partial\Omega$. A relevant example is the unit square in $\mathbb{R}^2$, $\Omega = \{(x_1, x_2) \mid 0 < x_i < 1,\ i = 1, 2\}$. $C^1(\Omega)$ denotes the space of functions $f : \Omega \to \mathbb{R}$ for which all the first partial derivatives $\frac{\partial f}{\partial x_i}$ exist and are continuous. $\mathcal{H}$ will denote a real Hilbert space with inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ and induced norm $\| \cdot \|_{\mathcal{H}}$. If the context is clear, the subscript $\mathcal{H}$ indicating the space is omitted. The symbol $\to$ denotes strong convergence, i.e., $f_n \to f$ means $\lim_{n\to\infty} \|f_n - f\| = 0$. The orthogonal complement of S, consisting of $f \in \mathcal{H}$ such that $\langle f, s \rangle = 0$ for all $s \in S$, is denoted by $S^{\perp}$. The sum of two sets S and T is the set $S + T = \{s + t | s \in S,\ t \in T\}$. For $s \in \mathcal{H}$, $s + T$ means $\{s\} + T$. The symbol $\mathbb{R}^n_+$ denotes the nonnegative orthant, consisting of vectors $\mathbf{x} = (x_1, \ldots, x_n)$ for which each $x_i \geq 0$. Little "o" notation is defined as usual:

$$f(\alpha) = o(g(\alpha)) \text{ as } \alpha \to \alpha_* \quad \text{if and only if} \quad \lim_{\alpha \to \alpha_*} \frac{f(\alpha)}{g(\alpha)} = 0. \tag{2.1}$$

The Kronecker delta is defined by

$$\delta_{ij} = \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{if } i \neq j. \end{cases} \tag{2.2}$$

Let $A : \mathcal{H}_1 \to \mathcal{H}_2$ be a (possibly nonlinear) operator. The range of A, $\{A(f) | f \in \mathcal{H}_1\}$, is denoted by Range$(A)$. A is continuous if $A(f_n) \to A(f_*)$ whenever $f_n \to f_*$. If A is linear, we adopt the notation Af for $A(f)$. Then the null space of A, $\{f \in \mathcal{H}_1 | Af = 0\}$, is denoted by Null$(A)$. A linear operator $A : \mathcal{H}_1 \to \mathcal{H}_2$ is bounded if and only if the induced

operator norm,

$$||A|| \stackrel{\text{def}}{=} \sup_{||f||_{\mathcal{H}_1}=1} ||Af||_{\mathcal{H}_2},$$

is finite. Bounded linear operators are continuous. The space of bounded linear operators from $\mathcal{H}_1$ into $\mathcal{H}_2$ is denoted by $\mathcal{L}(\mathcal{H}_1, \mathcal{H}_2)$. The adjoint of a bounded linear operator A is the operator $A^* \in \mathcal{L}(\mathcal{H}_2, \mathcal{H}_1)$ characterized by

$$\langle Af, g\rangle_{\mathcal{H}_2} = \langle f, A^*g\rangle_{\mathcal{H}_1} \quad \text{whenever} \quad f \in \mathcal{H}_1,\ g \in \mathcal{H}_2. \tag{2.3}$$

A is self-adjoint if $A = A^*$ (this requires $\mathcal{H}_1 = \mathcal{H}_2 = \mathcal{H}$). Then

$$\lambda_{\min}(A) \stackrel{\text{def}}{=} \inf_{||f||_{\mathcal{H}}=1} \langle Af, f\rangle_{\mathcal{H}} \tag{2.4}$$

and

$$\lambda_{\max}(A) \stackrel{\text{def}}{=} \sup_{||f||_{\mathcal{H}}=1} \langle Af, f\rangle_{\mathcal{H}} \tag{2.5}$$

are both finite real numbers. A is positive semidefinite if $\lambda_{\min}(A) \geq 0$. A is positive definite if $\langle Af, f\rangle_{\mathcal{H}} > 0$ whenever $f \neq 0$. A is called strongly positive, or coercive, if $\lambda_{\min}(A) > 0$. If A is self-adjoint, then $||A|| = \max(|\lambda_{\min}(A)|, |\lambda_{\max}(A)|)$. In general, $||A|| = \sqrt{\lambda_{\max}(A^*A)}$.

Example 2.1. $\mathbb{R}^n$ is a Hilbert space under the Euclidean inner product, $\langle \mathbf{f}, \mathbf{g}\rangle = \mathbf{f}^T\mathbf{g} = \sum_{j=1}^n f_j g_j$. The induced norm is the Euclidean norm, $||\mathbf{f}|| = \sqrt{\sum_{j=1}^n f_j^2}$. A linear operator A on $\mathbb{R}^n$ is always bounded and has a real $n \times n$ representation matrix $\overline{A}$ with respect to an orthonormal basis $\{\mathbf{e}_i\}_{i=1}^n$ with entries $[\overline{A}]_{ij} = \langle A\mathbf{e}_j, \mathbf{e}_i\rangle$. (Typically, the standard unit basis vectors $[\mathbf{e}_i]_j = \delta_{ij}$ are used, and the same symbol is used for A and for its representation matrix.) The adjoint A^* has the matrix transpose, $\overline{A}^T$, as its representation matrix, and A is self-adjoint if and only if $\overline{A}$ is symmetric. For self-adjoint A, $\lambda_{\min}(A)$ is the smallest eigenvalue of $\overline{A}$, and the right-hand side of (2.4) is attained for $f = \sum_{i=1}^n \overline{f}_i \mathbf{e}_i$, where $(\overline{f}_1, \ldots, \overline{f}_n)$ is a corresponding normalized eigenvector for $\overline{A}$. Similarly, the largest eigenvalue equals $\lambda_{\max}(\overline{A})$ and is also attained. Positive definiteness and strong positivity are equivalent in this finite-dimensional setting.

Example 2.2. Let $\mathbb{R}^{m\times n}$ denote the set of $m \times n$ real-valued matrices. This is a Hilbert space under the Frobenius inner product

$$\langle A, B\rangle_{Fro} = \sum_{i=1}^m \sum_{j=1}^n a_{ij} b_{ij}. \tag{2.6}$$

The induced norm is the usual Frobenius norm, $||A||_{Fro} = \sqrt{\sum_{i=1}^m \sum_{j=1}^n a_{ij}^2}$.

Example 2.3. $\ell^2(\mathbb{R})$ consists of real-valued infinite sequences $f = (f_1, f_2, \ldots)$ for which $\sum_{j=1}^\infty f_j^2 < \infty$. It is a Hilbert space under the inner product $\langle f, g\rangle_{\ell^2} = \sum_{j=1}^\infty f_j g_j$ and induced norm $||f||_{\ell^2} = \sqrt{\sum_{j=1}^\infty f_j^2}$. The diagonal operator

$$[Df]_j = d_j f_j, \qquad j = 1, 2, \ldots,$$

is bounded if and only if $B \stackrel{\text{def}}{=} \sup_j |d_j| < \infty$, in which case $||D|| = B$. D is self-adjoint.

Example 2.4. The space of real-valued, square integrable functions on Ω, denoted by $L^2(\Omega)$, is a Hilbert space under the inner product $\langle f, g\rangle_{L^2} = \int_\Omega f(x)g(x)\,dx$, and the induced norm $||f||_{L^2} = \sqrt{\int_\Omega f(x)^2\,dx}$. The Fredholm first kind integral operator

$$(Kf)(x) = \int_\Omega k(x, y)f(y)dy, \qquad x \in \Omega, \tag{2.7}$$

is bounded if $B \stackrel{\text{def}}{=} \int_\Omega \int_\Omega k(x, y)^2\,dx\,dy < \infty$, in which case $||K|| \leq \sqrt{B}$. The adjoint of K is given by

$$(K^*g)(y) = \int_\Omega k(x, y)g(x)dx, \qquad y \in \Omega, \tag{2.8}$$

and K is self-adjoint if and only if $k(x, y) = k(y, x)$.

Example 2.5. For functions $f, g \in C^1(\Omega)$, define the Sobolev H^1 inner product

$$\langle f, g\rangle_{H^1} = \int_\Omega \left(fg + \sum_{i=1}^n \frac{\partial f}{\partial x_i}\frac{\partial g}{\partial x_i} \right) dx_1 \ldots dx_n. \tag{2.9}$$

The Sobolev space $H^1(\Omega)$ is the closure of $C^1(\Omega)$ with respect to the norm induced by this inner product. For smooth f, define the negative Laplacian operator

$$Lf = -\sum_{i=1}^n \frac{\partial^2 f}{\partial x_i^2}. \tag{2.10}$$

This operator has an extension $L : H^1(\Omega) \to L^2(\Omega)$ with $||L|| = 1$. L is self-adjoint and positive semidefinite. See [5, Chapter 3] for details. Also, see Exercise 2.13.

Best Approximation in a Hilbert Space

Let $f \in \mathcal{H}$ and let $\mathcal{S}$ be a subspace of $\mathcal{H}$. s_* is called a best approximation to f from $\mathcal{S}$ if

$$s_* = \arg\min_{s\in\mathcal{S}} ||s - f||. \tag{2.11}$$

This means that $s_* \in \mathcal{S}$ and $||s_* - f|| \leq ||s - f||$ for all $s \in \mathcal{S}$. If it exists, the best approximation is unique. Existence is guaranteed provided that $\mathcal{S}$ is closed in $\mathcal{H}$. This is the case when $\mathcal{S}$ is a finite-dimensional subspace of $\mathcal{H}$. The following result provides a characterization of the best approximation.

Theorem 2.6. *If s_* is the best approximation to f from a subspace $\mathcal{S}$, then*

$$\langle s_* - f, s\rangle = 0 \quad \textit{whenever} \quad s \in \mathcal{S}. \tag{2.12}$$

Given a basis $\{\phi_1, \ldots, \phi_N\}$ for a (finite-dimensional) subspace $\mathcal{S}$, Theorem 2.6 provides a formula for computing the best approximation,

$$s_* = \sum_{j=1}^N \hat{\alpha}_j\phi_j, \tag{2.13}$$

where the coefficient vector $\hat{\alpha} = (\hat{\alpha}_1, \ldots, \hat{\alpha}_N)$ solves the linear system

$$G\alpha = \mathbf{b}, \tag{2.14}$$

with

$$[G]_{ij} = \langle \phi_j, \phi_i \rangle, \quad [\mathbf{b}]_i = \langle f, \phi_i \rangle. \tag{2.15}$$

The $N \times N$ matrix G is called the Gram matrix. The basis vectors ϕ_j are called orthonormal if

$$\langle \phi_j, \phi_i \rangle = \delta_{ij}.$$

In this case, the Gram matrix becomes the identity, and the best approximation has the representation

$$s_* = \sum_{j=1}^{N} \langle f, \phi_j \rangle \, \phi_j. \tag{2.16}$$

2.1 Ill-Posedness and Regularization

Definition 2.7. Let $K : \mathcal{H}_1 \to \mathcal{H}_2$. An operator equation

$$K(f) = g \tag{2.17}$$

is said to be well-posed provided

(i) for each $g \in \mathcal{H}_2$ there exists $f \in \mathcal{H}_1$, called a solution, for which (2.17) holds;

(ii) the solution f is unique; and

(iii) the solution is stable with respect to perturbations in g. This means that if $Kf_* = g_*$ and $Kf = g$, then $f \to f_*$ whenever $g \to g_*$.

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

If equation (2.17) is well-posed, then K has a well-defined, continuous inverse operator K^{-1}. In particular, $K^{-1}(K(f)) = f$ for any $f \in \mathcal{H}_1$, and $\text{Range}(K) = \mathcal{H}_2$. If K is a linear operator, equation (2.17) is well-posed if and only if properties (i) and (ii) hold or, equivalently, $\text{Null}(K) = \{0\}$ and $\text{Range}(K) = \mathcal{H}_2$. If K is a linear operator on $\mathbb{R}^n$, then (2.17) is well-posed if and only if either one of properties (i) and (ii) holds. The remaining properties are a consequence of the compactness of the unit ball in finite-dimensional spaces. See [35, 70].

Example 2.8. Consider the diagonal operator D on $\ell^2(\mathbb{R})$, defined in Example 2.3, with $d_j = 1/j$, $j = 1, 2, \ldots$. If a solution to $Df = g$ exists, it is unique, since D is linear and $\text{Null}(D) = \{0\}$. However, $g = (1, 1/2, 1/3, \ldots)$ lies in $\ell^2(\mathbb{R})$ but not in $\text{Range}(D)$; this illustrates that a solution need not exist, and hence the equation $Df = g$ is ill-posed. Stability is also lacking: Take $f_n \in \ell^2(\mathbb{R})$ to have jth component $[f_n]_j = \delta_{nj}$. Then $||Df_n|| = 1/n \to 0$ and $D0 = 0$, but f_n does not converge to zero.

Whether an operator equation is ill-posed may depend on topological considerations like the choice of norms. This is illustrated in the following example.

Example 2.9. Define $D : \mathcal{H}_1 \to \mathcal{H}_2$ with $\mathcal{H}_1 = \ell^2(\mathbb{R})$ as in Example 2.8. However, let $\mathcal{H}_2$ consist of infinite sequences g that satisfy

$$||g||^2_{\mathcal{H}_2} \stackrel{\text{def}}{=} \sum_{j=1}^{\infty} j^2 g_j^2 < \infty.$$

Then $||Df||_{\mathcal{H}_2} = ||f||_{\mathcal{H}_1}$, and the equation $Df = g$ is well-posed.

In practical applications one cannot simply define away ill-posedness by arbitrarily changing topologies. The choice of norms and other more general distance measures is addressed in sections 2.4.1 and 2.4.2.

2.1.1 Compact Operators, Singular Systems, and the SVD

Many ill-posed problems arising in applications involve compact operators.

Definition 2.10. A bounded linear operator $K : \mathcal{H}_1 \to \mathcal{H}_2$ is compact if and only if the image of any bounded set is a relatively compact set, i.e., if the closure of this image is a compact subset of $\mathcal{H}_2$.

Example 2.11. Any linear operator $K : \mathcal{H}_1 \to \mathcal{H}_2$ for which Range(K) is finite-dimensional is compact. In particular, matrix operators are compact.

Example 2.12. The diagonal operator D in Example 2.8 is a compact operator on $\ell^2(\mathbb{R})$.

Example 2.13. The Fredholm first kind integral operator (2.7) in Example 2.4 is a compact operator on $L^2(\Omega)$.

The following theorem establishes a connection between compactness and ill-posedness for linear operators on infinite-dimensional Hilbert spaces.

Theorem 2.14. *Let $K : \mathcal{H}_1 \to \mathcal{H}_2$ be a compact linear operator, and let $\mathcal{H}_1$ and $\mathcal{H}_2$ be infinite-dimensional. If* Range(K) *is infinite-dimensional, then the operator equation $Kf = g$ is ill-posed in the sense that conditions* (i) *and* (iii) *of Definition* 2.7 *are violated. In this case* Range(K) *is not closed. If* Range(K) *has finite dimension, then condition* (ii) *is violated.*

If K is compact, then K^*K is compact and self-adjoint. From the spectral theory for compact, self-adjoint linear operators [125], there exist positive eigenvalues and a corresponding set of orthonormal eigenvectors that form a basis for $\text{Null}(K^*K)^{\perp} = \text{Null}(K)^{\perp}$. From this eigendecomposition, one can construct a singular system, whose properties are given below. See Exercise 2.9.

Definition 2.15. A singular system for a compact linear operator $K : \mathcal{H}_1 \to \mathcal{H}_2$ is a countable set of triples $\{u_j, s_j, v_j\}_j$ with the following properties: (i) the right singular vectors v_j form an orthonormal basis for $\text{Null}(K)^{\perp}$; (ii) the left singular vectors u_j form an orthonormal basis for the closure of Range(K); (iii) the singular values s_j are positive real numbers and are in nonincreasing order, $s_1 \geq s_2 \geq \cdots > 0$; (iv) for each j

$$Kv_j = s_j u_j;$$

and (v) for each j

$$K^* u_j = s_j v_j.$$

If Range(K) is infinite-dimensional, one has the additional property

$$\lim_{j\to\infty} s_j = 0. \tag{2.18}$$

SVD of a Matrix

Let K be an $m \times n$ matrix. K can be viewed as a compact linear operator, mapping $\mathcal{H}_1 = \mathbb{R}^n$ into $\mathcal{H}_2 = \mathbb{R}^m$; see Examples 2.1 and 2.11. It has a singular system $\{\mathbf{u}_j, s_j, \mathbf{v}_j\}_{j=1}^r$, where r is the rank of K, i.e., the dimension of Range(K). Let U_r denote the $m \times r$ matrix whose jth column is $\mathbf{u}_j$, and let V_r denote the $n \times r$ matrix whose jth column is $\mathbf{v}_j$. If $r < n$, then Null(K) is a nontrivial subspace of $\mathbb{R}^n$, and one can construct an $n \times (n-r)$ matrix V_0 whose columns form an orthonormal basis for Null(K). Similarly, if $r < m$, then Range(K)$^\perp$ is a nontrivial subspace of $\mathbb{R}^m$, and one can construct an $m \times (m-r)$ matrix $U_\perp$ whose columns form an orthonormal basis for Range(K)$^\perp$. Then the SVD of K is the matrix decomposition

$$K = U\, D\, V^T, \tag{2.19}$$

where

$$U = [U_r\ U_\perp], \quad V = [V_r\ V_0], \quad D = \begin{bmatrix} \operatorname{diag}(s_1, \ldots, s_r) & 0_{r\times(n-r)} \\ 0_{(m-r)\times r} & 0_{(m-r)\times(n-r)} \end{bmatrix}.$$

2.1.2 Least Squares Solutions and the Pseudo-Inverse

Let K be a compact operator with a singular system $\{u_j, s_j, v_j\}$. Then K has a representation

$$Kf = \sum_j s_j \langle f, v_j\rangle_{\mathcal{H}_1} u_j. \tag{2.20}$$

Given $g \in$ Range(K), one can construct a vector

$$K^\dagger g = \sum_j \frac{\langle g, u_j\rangle_{\mathcal{H}_2}}{s_j} v_j \tag{2.21}$$

that lies in Null(K)$^\perp$ and for which $K(K^\dagger g) = g$. The series on the right-hand side of (2.21) converges even if $g \in$ Range(K) + Range(K)$^\perp$. Note that Range(K) is contained in, but not necessarily equal to, the closure of Range(K). The linear operator $K^\dagger$ defined in (2.21) is called the pseudo-inverse, or generalized inverse, of K. The domain of $K^\dagger$ is Range(K)+Range(K)$^\perp$, and Null(K)$^\perp$ is its range. In case K is a matrix, the decomposition (2.19) yields the representation

$$K^\dagger = V\, D^\dagger U^T,$$

where

$$[D^\dagger]_{ij} = \begin{cases} \frac{1}{s_j} & \text{if} \quad i = j \quad \text{and} \quad 1 \le i \le r, \\ 0 & \text{otherwise.} \end{cases}$$

What follows is an alternative characterization of $K^\dagger g$.

Definition 2.16. Let $K : \mathcal{H}_1 \to \mathcal{H}_2$ be bounded and linear. $f_{ls} \in \mathcal{H}_1$ is a least squares solution to $Kf = g$ if

$$||Kf_{ls} - g||_{\mathcal{H}_2} \leq ||Kf - g||_{\mathcal{H}_2} \quad \text{for each} \quad f \in \mathcal{H}_1. \tag{2.22}$$

A least squares solution need not exist. If a least squares solution f_{ls} does exist, the set of all least squares solutions is given by the affine subspace $f_{ls} + \text{Null}(K)$. The least squares minimum norm solution to $Kf = g$ is then given by

$$f_{lsmn} = \arg \min_{f \in f_{ls} + \text{Null}(K)} ||f||_{\mathcal{H}_1}. \tag{2.23}$$

Theorem 2.17. *If* $g \in \text{Range}(K) + \text{Range}(K)^{\perp}$, *then*

$$f_{lsmn} = K^{\dagger} g.$$

$K^{\dagger}$ *is bounded (and hence* f_{lsmn} *exists for any* $g \in \mathcal{H}_2$*) if and only if* $\text{Range}(K)$ *is closed.*

2.2 Regularization Theory

In Chapter 1 we presented several examples of regularization schemes that yielded operator approximations to the pseudo-inverse. Following are generalizations that can sometimes be applied even in a nonlinear setting. With regard to the operator equation (2.17), we assume there exists an operator R_* that assigns to each $g \in \text{Range}(K)$ a unique $R_*(g) \in \mathcal{H}_1$ for which $K(R_*(g)) = g$. In the linear case, one typically takes $R_* = K^{\dagger}$. We consider a family of regularization operators $R_\alpha : \mathcal{H}_2 \to \mathcal{H}_1$, where α is the regularization parameter, which lies in an index set I. See section 1.2 for some concrete examples.

Definition 2.18. $\{R_\alpha\}_{\alpha \in I}$ is a *regularization scheme that converges* to R_* if

(i) for each $\alpha \in I$, R_α is a continuous operator; and

(ii) given any $g \in \text{Range}(K)$, for any sequence $\{g_n\} \subset \mathcal{H}_2$ that converges to g, one can pick a sequence $\{\alpha_n\} \subset I$ such that

$$R_{\alpha_n}(g_n) \to R_*(g) \quad \text{as} \quad n \to \infty.$$

The regularization scheme is linear if each R_α is a (bounded) linear operator.

Of particular interest are linear regularization schemes with a filtered singular system representation,

$$R_\alpha(g) = \sum_j \frac{w_\alpha(s_j^2)}{s_j} \langle g, u_j \rangle_{\mathcal{H}_2} v_j, \tag{2.24}$$

which converge to $R_* = K^{\dagger}$. Examples of regularizing filters $w_\alpha(s^2)$ include the TSVD filter (1.11), the Tikhonov filter (1.13), and the Landweber filter (1.38). In the case of TSVD and Tikhonov regularization, the index set $I = (0, \infty)$. For Landweber iteration, the regularization parameter $\alpha = \nu$ is the iteration count, and the index set $I = \{0, 1, 2, \ldots\}$.

One can show from the representation (2.24) that

$$||R_\alpha|| = \sup_j \frac{w_\alpha(s_j^2)}{s_j}. \tag{2.25}$$

Suppose that $g \in \text{Range}(K)$ and that $g_n \in \mathcal{H}_2$, $\delta_n > 0$ satisfy

$$||g_n - g|| \leq \delta_n. \tag{2.26}$$

Then from the triangle inequality and (2.25),

$$||R_{\alpha_n} g_n - R_* g|| \leq ||R_{\alpha_n} g - R_* g|| + ||R_{\alpha_n}||\, \delta_n. \tag{2.27}$$

The following theorem establishes conditions that guarantee that one can select $\alpha = \alpha(\delta)$ so that both terms of the sum on the right-hand side of (2.27) converge to zero as $\delta_n \to 0$. Let α_* denote the limiting value of the regularization parameter that yields R_*, e.g., for TSVD and Tikhonov regularization, $\alpha_* = 0$, and for Landweber iteration, $\alpha_* = \infty$.

Theorem 2.19. *Assume that for each $\alpha \in I$, $\sup_{s>0} |w_\alpha(s^2) \,/\, s| < \infty$, and that for each $s > 0$,*

$$\lim_{\alpha \to \alpha_*} w_\alpha(s^2) = 1. \tag{2.28}$$

Also assume that there exists a function $\alpha = \alpha(\delta)$, mapping $\mathbb{R}_+$ into the index set I, such that

$$\lim_{\delta \to 0} \alpha(\delta) = \alpha_*, \tag{2.29}$$

$$\lim_{\delta \to 0} ||R_{\alpha(\delta)}||\, \delta = 0. \tag{2.30}$$

Then (2.24) *defines a regularization scheme that converges to $K^\dagger$.*

The following examples establish that TSVD, Tikhonov regularization, and Landweber iteration yield regularization schemes that converge to $K^\dagger$.

Example 2.20. For both the TSVD filter (1.11) and the Tikhonov filter (1.13), (2.28) holds with $\alpha_* = 0$, and $w_\alpha(s^2)/s \leq 1/\sqrt{\alpha}$ whenever $\alpha > 0$, $s > 0$. One can choose $\alpha = \delta$ to guarantee that both (2.29) and (2.30) hold.

Example 2.21. For the Landweber filter (1.38), (2.28) holds with $\alpha_* = \nu_* = \infty$, and

$$w_\nu(s^2)/s \leq \sqrt{\nu}/||K||. \tag{2.31}$$

See Exercise 2.16. Choose $\nu = [||K||^2/\delta]$, where $[x]$ denotes the greatest integer less than or equal to x, to guarantee (2.29) and (2.30).

2.3 Optimization Theory

Our goal here is to introduce tools to analyze and compute minimizers for the Tikhonov functional (1.34) and certain of its nonquadratic generalizations to be presented later. Let

$J : \mathcal{H} \to \mathbb{R}$ and let $\mathcal{C}$ be a subset of $\mathcal{H}$. We wish to compute a minimizer of J over $\mathcal{C}$, which we denote by

$$f_* = \arg\min_{f \in \mathcal{C}} J(f).$$

If $\mathcal{C} = \mathcal{H}$, the minimization problem is called unconstrained. Otherwise, it is called constrained. f_* is a local minimizer if there exists $\delta > 0$ for which

$$J(f_*) \leq J(f) \quad \text{whenever} \quad f \in \mathcal{C}, \ ||f - f_*||_{\mathcal{H}} < \delta.$$

The minimizer is called strict if $J(f_*) \leq J(f)$ can be replaced by $J(f_*) < J(f)$ whenever $f \neq f_*$.

We first present conditions that guarantee the existence and uniqueness of minimizers.

Definition 2.22. A sequence $\{f_n\}$ in a Hilbert space $\mathcal{H}$ converges weakly to f_*, denoted by $f_n \rightharpoonup f_*$, provided $\lim_{n\to\infty} \langle f_n - f_*, f \rangle_{\mathcal{H}} = 0$ for all $f \in \mathcal{H}$.

Strong convergence implies weak convergence. In finite-dimensional spaces, strong and weak convergence are equivalent. This is not the case in infinite-dimensional spaces, as the following example shows.

Example 2.23. Let $\{f_n\}$ be an infinite orthonormal set, e.g., $f_n(x) = \sin(2n\pi x)/\sqrt{2}$, $n = 1, 2, \ldots$, in $L^2(0, 1)$. Then f_n converges weakly to 0. The convergence is not strong, since $||f_n|| = 1$ for each n.

Definition 2.24. $J : \mathcal{H} \to \mathbb{R}$ is weakly lower semicontinuous if

$$J(f_*) \leq \liminf_{n\to\infty} J(f_n) \quad \text{whenever} \quad f_n \rightharpoonup f_*. \tag{2.32}$$

Example 2.25. The Hilbert space norm $J(f) = ||f||$ is weakly lower semicontinuous. Example 2.23 illustrates that the inequality (2.32) may be strict.

Definition 2.26. $J : \mathcal{C} \subset \mathcal{H} \to \mathbb{R}$ is a convex functional if

$$J(\tau f_1 + (1 - \tau) f_2) \leq \tau J(f_1) + (1 - \tau) J(f_2) \tag{2.33}$$

whenever $f_1, f_2 \in \mathcal{C}$ and $0 < \tau < 1$. J is strictly convex provided the inequality (2.33) is strict whenever $f_1 \neq f_2$.

Convex functionals are weakly lower semicontinuous [128].

Recall that a set $\mathcal{C}$ is convex provided $\tau f_1 + (1 - \tau) f_2 \in \mathcal{C}$ whenever $f_1, f_2 \in \mathcal{C}$ and $0 < \tau < 1$. $\mathcal{C}$ is closed if for any convergent sequence $\{f_n\} \subset \mathcal{C}$, $\lim_{n\to\infty} f_n \in \mathcal{C}$.

Example 2.27. The nonnegative orthant $\mathbb{R}^n_+$, consisting of vectors $\mathbf{f} = (f_1, \ldots, f_n) \in \mathbb{R}^n$ for which $f_i \geq 0$ for each i, is closed and convex.

Example 2.28. The set $\mathcal{C} = \{f \in L^2(\Omega) | f(x) \geq 0 \text{ for a.e. } x \in \Omega\}$ is closed and convex.

Definition 2.29. A functional $J : \mathcal{H} \to \mathbb{R}$ is coercive if

$$J(f_n) \to \infty \quad \text{whenever} \quad ||f_n||_{\mathcal{H}} \to \infty.$$

Theorem 2.30. *Assume that $J : \mathcal{H} \to \mathbb{R}$ is weakly lower semicontinuous and coercive and that $\mathcal{C}$ is a closed, convex subset of $\mathcal{H}$. Then J has a minimizer over $\mathcal{C}$. If J is also strictly convex, then the minimizer is unique.*

Proof. Let $\{f_n\}$ be a minimizing sequence for J in $\mathcal{C}$, i.e., each $f_n \in \mathcal{C}$ and $J(f_n) \to J_* \stackrel{\text{def}}{=} \inf_{f \in \mathcal{C}} J(f)$. Since J is coercive, $\{f_n\}$ must be bounded. Boundedness of the sequence in a Hilbert space implies the existence of a weakly convergent subsequence [127, 128], which we denote by $\{f_{n_j}\}$. Let f_* denote the weak limit of this subsequence. Since closed, convex sets in a Hilbert space are weakly closed [127, 128], $f_* \in \mathcal{C}$. By weak lower semicontinuity of J,

$$J(f_*) \leq \liminf J(f_{n_j}) = \lim J(f_n) = J_*,$$

and hence $J(f_*) = J_*$. Now assume J is strictly convex and $J(f_0) = J_*$ with $f_0 \neq f_*$. Taking $\tau = 1/2$ in (2.33) gives $J((f_0 + f_*)/2) < J_*$, a contradiction. □

We next look at characterizations of minimizers.

Definition 2.31. An operator $A : \mathcal{H}_1 \to \mathcal{H}_2$ is said to be Fréchet differentiable at $f \in \mathcal{H}_1$ if and only if there exists $A'(f) \in \mathcal{L}(\mathcal{H}_1, \mathcal{H}_2)$, called the Fréchet derivative of A at f, for which

$$A(f+h) = A(f) + A'(f)h + o(||h||_{\mathcal{H}_1}) \quad \text{as} \quad ||h||_{\mathcal{H}_1} \to 0. \tag{2.34}$$

Higher order Fréchet derivatives are defined recursively, e.g.,

$$A'(f+k) = A'(f) + A''(f)k + o(||k||_{\mathcal{H}_1})$$

defines the second Fréchet derivative, $A''(f) \in \mathcal{L}(\mathcal{H}_1, \mathcal{L}(\mathcal{H}_1, \mathcal{H}_2))$. The mapping $(h, k) \mapsto (A''(f)k)h$ is bounded and bilinear from $\mathcal{H}_1 \times \mathcal{H}_1$ into $\mathcal{H}_2$. This mapping can be shown to be symmetric, i.e.,

$$(A''(f)k)h = (A''(f)h)k. \tag{2.35}$$

Definition 2.32. Suppose that $J : \mathcal{H} \to \mathbb{R}$ is Fréchet differentiable at f. By the Riesz representation theorem [127, 128] there exists $\operatorname{grad} J(f) \in \mathcal{H}$, called the gradient of J at f, for which

$$J'(f)h = \langle \operatorname{grad} J(f), h \rangle_{\mathcal{H}}. \tag{2.36}$$

Example 2.33. If $J : \mathbb{R}^n \to \mathbb{R}$ has continuous partial derivatives, then

$$\operatorname{grad} J = \left(\frac{\partial J}{\partial f_1}, \ldots, \frac{\partial J}{\partial f_n} \right). \tag{2.37}$$

The following result provides a means for computing gradients.

Proposition 2.34. *If $J : \mathcal{H} \to \mathbb{R}$ is Fréchet differentiable at f, then for any $h \in \mathcal{H}$, the mapping $\tau \mapsto J(f + \tau h)$ from $\mathbb{R}$ into $\mathbb{R}$ is differentiable at $\tau = 0$ with*

$$\frac{d}{d\tau} J(f + \tau h)|_{\tau=0} = \langle \operatorname{grad} J(f), h \rangle_{\mathcal{H}}. \tag{2.38}$$

Remark 2.35. The left-hand side of (2.38) defines the Gateaux, or directional, derivative, or the first variation of J at f in the direction h. This is often denoted by $\delta J(f, h)$.

Example 2.36. Let $K : \mathcal{H}_1 \to \mathcal{H}_2$ be a bounded linear operator, fix $g \in \mathcal{H}_2$ and $\alpha > 0$, and consider the Tikhonov functional, scaled by a factor of $1/2$, $J(f) = (||Kf - g||^2_{\mathcal{H}_2} + \alpha||f||^2_{\mathcal{H}_1})/2$. For any $h \in \mathcal{H}_1$ and $\tau \in \mathbb{R}$,

$$\begin{aligned} J(f + \tau h) = \frac{1}{2}(||Kf - g||^2_{\mathcal{H}_2} + \alpha||f||^2_{\mathcal{H}_1}) + \frac{\tau}{2}(\langle Kf - g, Kh\rangle_{\mathcal{H}_2} \\ + \langle Kh, Kf - g\rangle_{\mathcal{H}_2} \quad + \quad \alpha\langle f, h\rangle_{\mathcal{H}_1} + \alpha\langle h, f\rangle_{\mathcal{H}_1}) \\ + \frac{\tau^2}{2}(||Kh||^2_{\mathcal{H}_2} + \alpha||h||^2_{\mathcal{H}_1}). \end{aligned}$$

Then

$$\begin{aligned} \frac{d}{d\tau} J(f + \tau h)|_{\tau=0} &= \langle Kf - g, Kh\rangle_{\mathcal{H}_2} + \alpha\langle f, h\rangle_{\mathcal{H}_1} \\ &= \langle (K^*K + \alpha I)f - K^*g, h\rangle_{\mathcal{H}_1}, \end{aligned}$$

and hence $\operatorname{grad} J(f) = (K^*K + \alpha I)f - K^*g$.

The following two theorems provide first order necessary conditions for a local minimizer. The first deals with the unconstrained case and the second with the constrained case.

Theorem 2.37. *Let $J : \mathcal{H} \to \mathbb{R}$ be Fréchet differentiable. If J has a local unconstrained minimizer at f_*, then* $\operatorname{grad} J(f_*) = 0$.

Proof. Let $g = \operatorname{grad} J(f_*)$. By (2.34) and (2.36),

$$J(f_* - \tau g) = J(f_*) - \tau||g||^2 + o(\tau) \quad \text{as} \quad \tau \to 0. \tag{2.39}$$

If $g \neq 0$, then the right-hand side can be made smaller than $J(f_*)$ by taking $\tau > 0$ and sufficiently small. □

Theorem 2.38. *Let C be a closed, convex subset of $\mathcal{H}$. If $f_* \in C$ is a local constrained minimizer for J and J is Fréchet differentiable at f_*, then*

$$\langle \operatorname{grad} J(f_*), f - f_*\rangle_{\mathcal{H}} \geq 0 \quad \textit{for each} \quad f \in C. \tag{2.40}$$

Proof. Let $f \in C$ be arbitrary but fixed. Since C is convex, $f_* + \tau(f - f_*) = \tau f + (1-\tau)f_* \in C$ whenever $0 \leq \tau \leq 1$. Then, since f_* is a constrained minimizer,

$$\lim_{\tau \to 0^+} \frac{J(f_* + \tau(f - f_*)) - J(f_*)}{\tau} \geq 0.$$

Equation (2.40) follows from Proposition 2.34. □

The following theorems characterize differentiable convex functionals.

Theorem 2.39. *Suppose J is Fréchet differentiable on a convex set C. Then J is convex if and only if*

$$\langle \operatorname{grad} J(f_1) - \operatorname{grad} J(f_2), f_1 - f_2\rangle \geq 0 \tag{2.41}$$

whenever $f_1, f_2 \in \mathcal{C}$. *J is strictly convex if and only if the inequality is strict whenever* $f_1 \neq f_2$.

Definition 2.40. If the second Fréchet derivative of J exists at f, then by (2.35) it has a representation

$$(J''(f)h)k = \langle \text{Hess } J(f)h, k\rangle, \tag{2.42}$$

where Hess $J(f)$ is a self-adjoint, bounded linear operator on $\mathcal{H}$ called the Hessian of J at f.

Example 2.41. If $J : \mathbb{R}^n \to \mathbb{R}$ has continuous second partial derivatives, then Hess $J(\mathbf{f})$ is an $n \times n$ matrix with entries

$$[\text{Hess } J(\mathbf{f})]_{ij} = \frac{\partial^2 J}{\partial f_i \partial f_j}, \qquad 1 \leq i, j \leq n. \tag{2.43}$$

Theorem 2.42. *Let J be twice Fréchet differentiable on a convex set $\mathcal{C}$. Then J is convex if and only if* Hess $J(f)$ *is positive semidefinite for all f in the interior of $\mathcal{C}$. If* Hess $J(f)$ *is positive definite, then J is strictly convex.*

If J is twice Fréchet differentiable at f, then

$$J(f+h) = J(f) + \langle \text{grad } J(f), h\rangle + \frac{1}{2}\langle \text{Hess } J(f)h, h\rangle + o(||h||^2) \tag{2.44}$$

as $||h||_{\mathcal{H}} \to 0$. This provides the basis for second order sufficient conditions for an unconstrained minimizer.

Theorem 2.43. *If J is twice Fréchet differentiable at f_*,* grad $J(f_*) = 0$, *and* Hess $J(f_*)$ *is strongly positive, then f_* is a strict local minimizer for J.*

2.4 Generalized Tikhonov Regularization

A generalized Tikhonov functional for problem (2.17) takes the form

$$T_\alpha(f; g) = \rho(K(f), g) + \alpha J(f). \tag{2.45}$$

Here $\alpha > 0$ is the regularization parameter, $J : \mathcal{H}_1 \to \mathbb{R}$ is called the penalty functional or regularization functional, and $\rho : \mathcal{H}_2 \times \mathcal{H}_2 \to \mathbb{R}$ is called the data discrepancy functional or fit-to-data functional.

2.4.1 Penalty Functionals

The purpose of the penalty functional is to induce stability and to allow the incorporation of a priori information about the desired solution f. In a statistical setting, the penalty functional is often called the prior. See section 4.3. The standard Tikhonov penalty functional on a Hilbert space $\mathcal{H}_1$ is simply

$$J(f) = \frac{1}{2}||f||^2_{\mathcal{H}_1}. \tag{2.46}$$

When $\mathcal{H}_1 = L^2(\Omega)$ (see Example 2.4), we refer to J as the L^2 penalty functional. Another example is the Sobolev H^1 penalty functional:

$$J_{H^1}(f) = \frac{1}{2}\int_\Omega \sum_{i=1}^{d} \left(\frac{\partial f}{\partial x_i}\right)^2. \tag{2.47}$$

Functionals like this penalize nonsmooth solutions. If f is twice continuously differentiable, one can integrate by parts to obtain

$$J_{H^1}(f) = \frac{1}{2}\int_\Omega (Lf)(x) f(x)\, dx + \frac{1}{2}\int_{\partial\Omega} \sum_{i=1}^{d} \frac{\partial f}{\partial x_i} n_i\, dS.$$

Here $\hat{n} = (n_1, \ldots, n_d)$ denotes the outward unit normal to the boundary $\partial\Omega$, and L denotes the negative Laplacian (see (2.10)). If one imposes so-called natural boundary conditions or homogeneous Neumann boundary conditions,

$$\sum_{i=1}^{d} \frac{\partial f}{\partial x_i} n_i = 0 \quad \text{in} \quad \partial\Omega,$$

then the boundary integral term disappears, and one can express

$$J_{H^1}(f) = \frac{1}{2}\langle Lf, f\rangle_{L^2(\Omega)}. \tag{2.48}$$

The condition that f is twice continuously differentiable can be relaxed [5]. Here L is referred to as the penalty operator. Penalty operators of more general diffusion type,

$$Lf = -\sum_{i=1}^{d} \frac{\partial}{\partial x_i}\left(\kappa \frac{\partial f}{\partial x_i}\right), \tag{2.49}$$

arise in the context of total variation regularization in Chapter 8. They also arise in other image processing applications [124].

A penalty functional not of form (2.48) is the negative entropy [35, p. 137]

$$J(f) = \langle f, \log f\rangle_{\mathcal{H}_1}, \tag{2.50}$$

defined on the set $\mathcal{C}$ given in either Example 2.27 or Example 2.28. For a discussion of the motivation for this penalty functional, see [49, p. 102].

2.4.2 Data Discrepancy Functionals

The purpose of the data discrepancy functional in (2.45) is to quantify how well the prediction $K(f)$ matches the observed data g. Perhaps the most familiar example is the squared Hilbert space norm,

$$\rho_{LS}(g_1, g_2) = \frac{1}{2}\|g_1 - g_2\|^2_{\mathcal{H}_2}, \qquad g_1, g_2 \in \mathcal{H}_2. \tag{2.51}$$

Less well-known is the Kullback–Leibler information divergence,

$$\rho_{KL}(g_1, g_2) = \langle g_1, \log(g_1/g_2)\rangle_{\mathcal{H}_2}, \qquad g_1, g_2 \in \mathcal{C}, \tag{2.52}$$

where $\mathcal{C}$ is given in Example 2.27 or Example 2.28. Closely related to this is the negative Poisson log likelihood functional,

$$\rho_{LHD}(g_1, g_2) = \langle g_1, 1\rangle_{\mathcal{H}_2} - \langle g_2, \log g_1\rangle_{\mathcal{H}_2}, \tag{2.53}$$

where 1 denotes the vector (or function) whose components (or pointwise values) equal 1. See section 4.2 for details.

2.4.3 Some Analysis

Recall from Definition 2.18 that a regularization scheme for problem (2.17) requires a family of continuous, convergent mappings $R_\alpha : \mathcal{H}_2 \to \mathcal{H}_1$. We now use previously developed tools to establish continuity for certain forms of Tikhonov regularization. More general forms require more abstract tools [35]. For now, take

$$T_\alpha(f; g) = \frac{1}{2}||Kf - g||^2_{\mathcal{H}_2} + \alpha\langle Lf, f\rangle_{\mathcal{H}_1} \tag{2.54}$$

and define

$$R_\alpha(g) = \arg\min_{f\in\mathcal{C}} T_\alpha(f; g), \qquad \alpha > 0.$$

Assume

A1. $\mathcal{C}$ is a closed, convex subset of $\mathcal{H}_1$;

A2. L is a self-adjoint, strongly positive linear operator on $\mathcal{H}_1$; this implies that there exists $c_0 > 0$ for which
$$\langle Lf, f\rangle_{\mathcal{H}_1} \geq c_0\, ||f||^2_{\mathcal{H}_1}; \tag{2.55}$$

A3. K is bounded and linear.

The following theorem establishes the existence and continuity of R_α.

Theorem 2.44. *Under the above assumptions, R_α exists and is continuous for any $\alpha > 0$.*

Proof. First we establish that the operator R_α is well defined. Fix $g \in \mathcal{H}_2$ and $\alpha > 0$. To simplify notation, let $T(f) = T_\alpha(f; g)$. T is convex and, hence, it is weakly lower semicontinuous. Assumption A2 implies that T is coercive, since $T(f) \geq \alpha c_0||f||^2_{\mathcal{H}_1}$. Theorem 2.30 guarantees that (2.54) has a unique minimizer.

To establish continuity, fix $g_0 \in \mathcal{H}_2$ and let $g_n \to g_0$. Set $f_0 = R_\alpha(g_0)$ and $f_n = R_\alpha(g_n)$. To simplify notation, take $T_0(f) = T_\alpha(f; g_0)$ and $T_n(f) = T_\alpha(f; g_n)$. From Theorem 2.38,

$$\begin{aligned} 0 &\geq \langle \operatorname{grad} T_n(f_n), f_n - f_0\rangle_{\mathcal{H}_1} - \langle \operatorname{grad} T_0(f_0), f_n - f_0\rangle_{\mathcal{H}_1} \\ &= \langle (K^*K + \alpha L)(f_n - f_0), f_n - f_0\rangle_{\mathcal{H}_1} + \langle K^*(g_n - g_0), f_n - f_0\rangle_{\mathcal{H}_1} \\ &\geq \alpha c_0||f_n - f_0||^2_{\mathcal{H}_1} - ||K^*(g_n - g_0)||_{\mathcal{H}_1}\, ||f_n - f_0||_{\mathcal{H}_1}. \end{aligned}$$

The last inequality follows from (2.55). Consequently,

$$||f_n - f_0||_{\mathcal{H}_1} \leq \frac{1}{\alpha c_0}||K^*(g_n - g_0)||_{\mathcal{H}_1}. \qquad \square$$

Exercises

2.1. Verify the statements made in Example 2.1 regarding the boundedness and self-adjointness of A and the characterization of $\lambda_{\min}(A)$.

2.2. Show that the operator D in Example 2.3 is bounded if and only if $B = \sup_j |d_j| < \infty$, and show that $||D|| = B$.

2.3. Show that the operator K in Example 2.4 is bounded if $B = \int_\Omega \int_\Omega k(x, y)^2 \, dx \, dy < \infty$, and show that $||K|| \leq \sqrt{B}$.

2.4. Show that the adjoint K^* of the operator K in Example 2.4 is given by equation (2.8).

2.5. Prove Theorem 2.6.

2.6. Use Theorem 2.6 to derive the representation (2.13)–(2.15) for the best approximation.

2.7. For the operator D in Example 2.8, show that $\lambda_{\min}(D) = 0$, but the right-hand side of (2.4) is not attained for any unit vector $f \in \ell^2(\mathbb{R})$. Also show that Range(D) is dense in $\ell^2(\mathbb{R})$.

2.8. Show that the diagonal operator D in Example 2.8 is a compact operator on $\ell^2(\mathbb{R})$.

2.9. Let K^*K have countably many positive eigenvalues λ_j and corresponding orthonormal eigenvectors v_j that span Null$(K)^\perp$. Use these to construct a singular system $\{u_j, s_j, v_j\}$ for K. *Hint:* Take $s_j = \sqrt{\lambda_j}$ and $u_j = Kv_j/s_j$.

2.10. Prove that a linear operator $K : \mathcal{H}_1 \to \mathcal{H}_2$ for which Range(K) is finite dimensional is compact.

2.11. Use the open mapping theorem to show that $K^\dagger$ is bounded if and only if K has closed range.

2.12. Prove that the Fredholm first kind integral operator (2.7) in Example 2.4 is a compact operator on $L^2(\Omega)$.

2.13. Let L denote the negative Laplacian operator (see (2.10)). Show for any $f \in C^1(\Omega)$ with $||f||_{H^1} \leq 1$ that $||Lf||_{L^2} \leq 1$. Show also that the restriction of L to $f \in C^1(\Omega)$ is self-adjoint and positive semidefinite.

2.14. Verify equation (2.25).

2.15. Prove Theorem 2.19.

2.16. Confirm the inequality (2.31) for Landweber iteration. See [35, p. 156].

2.17. Prove that strong convergence implies weak convergence. Also, prove that in finite dimensional spaces, strong and weak convergence are equivalent.

2.18. Prove that if $J : \mathcal{H} \to \mathbb{R}$ is convex, then it is weakly lower semicontinuous.

2.19. Prove that the set in Example 2.27 is convex and closed.

2.20. Prove that the set in Example 2.28 is convex and closed. *Hint:* To verify closure, suppose $f_* = \lim_{n\to\infty} f_n \notin C$. Define $S_- = \{x \in \Omega | f_*(x) < 0\}$ and take $u(x) = +1$ if $x \in S_-$ and zero otherwise. Then $0 > \langle f_*, u\rangle_{L^2} = \lim_{n\to 0} \langle f_n, u\rangle_{L^2}$. But each $\langle f_n, u\rangle_{L^2} > 0$.

2.21. Confirm equation (2.37) in Example 2.33.

2.22. Verify equation (2.35). See [125, p. 195].

2.23. In Example 2.36, use Definition 2.40 to show that Hess $J = K^*K + \alpha I$.

2.24. Prove Theorem 2.38.

2.25. Prove Theorem 2.39. See [80, Proposition 4, p. 116].

2.26. Prove Theorem 2.42. See [80, Proposition 5, p. 118].

2.27. Prove Theorem 2.43.

2.28. Show that the functional in equation (2.54) is convex.

This page intentionally left blank

Chapter 3
Numerical Optimization Tools

In the previous chapter we presented an abstract method, Tikhonov regularization, for solving ill-posed problems. The practical implementation of this method requires the minimization of a discretized version of the Tikhonov functional. In case this functional is nonquadratic, nonlinear iterative methods like the steepest descent method or variants of Newton's method can be used to compute regularized solutions. If the functional is quadratic, its minimizer satisfies a system of linear equations. If this system is large, iterative methods may be required to efficiently compute the minimizer. Our goal in this chapter is to present some basic tools to solve large-scale discretized problems. As in Chapter 2, the development here is quite terse. Motivational material and technical details can be found in [25, 32, 46, 91, 94, 102].

In this chapter we assume functionals J that map $\mathbb{R}^n$ into $\mathbb{R}$. When we say that J is smooth in the context of a particular method, we mean that it possesses derivatives of sufficiently high degree to implement the method. Unless otherwise stated, $\langle \cdot, \cdot \rangle$ denotes the Euclidean inner product, and $|| \cdot ||$ denotes the (induced) Euclidean norm; see Example 2.1. The SPD denotes *symmetric, positive definite* in the context of matrices. A matrix A is assumed to be $n \times n$ and to have real-valued components, unless otherwise specified.

Definition 3.1. Suppose that a sequence $\{\mathbf{f}_\nu\}$ converges to $\mathbf{f}_*$ as $\nu \to \infty$. The rate of convergence is linear if there exists a constant c, $0 \leq c < 1$, for which

$$||\mathbf{f}_{\nu+1} - \mathbf{f}_*|| \leq c||\mathbf{f}_\nu - \mathbf{f}_*||. \tag{3.1}$$

Convergence is superlinear if there exists a sequence $\{c_\nu\}$ of positive real numbers for which $\lim_{\nu\to\infty} c_\nu = 0$, and

$$||\mathbf{f}_{\nu+1} - \mathbf{f}_*|| \leq c_\nu||\mathbf{f}_\nu - \mathbf{f}_*||. \tag{3.2}$$

The rate is quadratic if for some constant $C > 0$,

$$||\mathbf{f}_{\nu+1} - \mathbf{f}_*|| \leq C||\mathbf{f}_\nu - \mathbf{f}_*||^2. \tag{3.3}$$

Quadratic convergence implies superlinear convergence, which in turn implies linear convergence. See Exercise 3.1.

3.1 The Steepest Descent Method

The proof of Theorem 2.37 motivates the following algorithm.

Algorithm 3.1.1. Steepest Descent Method.
To minimize a smooth functional $J(\mathbf{f})$,

$\nu := 0;$
$\mathbf{f}_0 :=$ initial guess;
begin steepest descent iterations
  $\mathbf{p}_\nu := -\text{grad } J(\mathbf{f}_\nu);$ *% compute negative gradient*
  $\tau_\nu := \arg\min_{\tau>0} J(\mathbf{f}_\nu + \tau\, \mathbf{p}_\nu);$ *% line search*
  $\mathbf{f}_{\nu+1} := \mathbf{f}_\nu + \tau_\nu \mathbf{p}_\nu;$ *% update approximate solution*
  $\nu := \nu + 1;$
end steepest descent iterations

One of the key components of Algorithm 3.1 is the line search.

Definition 3.2. The one-dimensional minimization problem

$$\min_{\tau>0} J(\mathbf{f}_\nu + \tau \mathbf{p}_\nu) \tag{3.4}$$

is called a line search. The vector $\mathbf{p}_\nu$ is called the search direction. $\mathbf{p}_\nu$ is called a descent direction for J at $\mathbf{f}_\nu$ if there exists $\delta > 0$ for which

$$J(\mathbf{f}_\nu + \tau \mathbf{p}_\nu) < J(\mathbf{f}_\nu) \quad \text{whenever} \quad 0 < \tau < \delta. \tag{3.5}$$

An obvious requirement for a successful line search is that the search direction $\mathbf{p}_\nu$ is a descent direction. If J is smooth, then this is guaranteed if

$$\langle \text{grad } J(\mathbf{f}_\nu), \mathbf{p}_\nu \rangle < 0. \tag{3.6}$$

See Exercise 3.2. Note that $\mathbf{p}_\nu = -\text{grad } J(\mathbf{f}_\nu)$ is a descent direction whenever grad $J(\mathbf{f}_\nu) \neq \mathbf{0}$. In practice, the line search subproblem need not be solved exactly. In section 3.4 we discuss both theoretical and computational issues related to inexact line searches.

Convergence rate analysis for the steepest descent method requires some preliminary definitions.

Definition 3.3. Let A be an SPD matrix. The energy inner product induced by A is given by

$$\langle \mathbf{f}_1, \mathbf{f}_2 \rangle_A \stackrel{\text{def}}{=} \langle A\mathbf{f}_1, \mathbf{f}_2 \rangle, \qquad \mathbf{f}_1, \mathbf{f}_2 \in \mathbb{R}^n.$$

The induced norm, $||\mathbf{f}||_A = \sqrt{\langle \mathbf{f}, \mathbf{f} \rangle_A}$, is called the energy norm induced by A.

The energy norm is equivalent to the Euclidean norm, with

$$\sqrt{\lambda_{\min}(A)}\, ||f|| \leq ||f||_A \leq \sqrt{\lambda_{\max}(A)}\, ||f|| \quad \text{whenever} \quad f \in \mathbb{R}^n. \tag{3.7}$$

See Exercise 3.4.

Definition 3.4. Let A be a (possibly nonsymmetric) $n \times n$ matrix. Then the condition number of A is defined to be

$$\text{cond}(A) = \frac{\max \sigma_0(A)}{\min \sigma_0(A)},$$

where $\sigma_0(A)$ denotes the set of nonzero singular values of A. If A is SPD, then $\text{cond}(A)$ reduces to the ratio of the largest to the smallest eigenvalue of A. The matrix A is called ill-conditioned if $\text{cond}(A)$ is large.

$J : \mathbb{R}^n \to \mathbb{R}$ is a quadratic functional if there exist $c \in \mathbb{R}$, $\mathbf{b} \in \mathbb{R}^n$, and a symmetric matrix A for which

$$J(\mathbf{f}) = c + \langle \mathbf{b}, \mathbf{f} \rangle + \frac{1}{2}\langle A\mathbf{f}, \mathbf{f} \rangle. \tag{3.8}$$

Note that $A = \text{Hess}\, J(\mathbf{f})$. A quadratic functional J is called positive if A is SPD. In this case, by Theorem 2.42, J is strictly convex and has $\mathbf{f}_* = -A^{-1}\mathbf{b}$ as its unique minimizer. Then the exact solution to the line search step in the steepest descent algorithm is

$$\tau_\nu = \frac{||\mathbf{g}_\nu||^2}{\langle A\mathbf{g}_\nu, \mathbf{g}_\nu \rangle}, \tag{3.9}$$

where $\mathbf{g}_\nu = \text{grad}\, J(\mathbf{f}_\nu) = \mathbf{b} + A\mathbf{f}_\nu$. See Exercise 3.3. We also obtain the following result [80, p. 152].

Theorem 3.5. *Assume that J is a positive quadratic functional with representation* (3.8). *Then for any initial guess $\mathbf{f}_0$, the steepest descent iterates $\mathbf{f}_{\nu+1} = \mathbf{f}_\nu - \tau_\nu \mathbf{g}_\nu$, cf.,* (3.9), *converge to $\mathbf{f}_*$ at a linear rate, with*

$$||\mathbf{f}_\nu - \mathbf{f}_*||_A \leq \left(\frac{\text{cond}(A) - 1}{\text{cond}(A) + 1}\right)^\nu ||\mathbf{f}_0 - \mathbf{f}_*||_A. \tag{3.10}$$

A similar result holds in the more general (nonquadratic) case, where the matrix A is replaced by the Hessian of J at $\mathbf{f}_*$. See [94, p. 62]. From these results, we see that if the Hessian is ill-conditioned, the converge rate for steepest descent can be very slow. A more rapidly convergent method is the conjugate gradient method.

3.2 The Conjugate Gradient Method

For the extremely large, highly structured linear systems to be described in Chapter 5, direct solution methods like the Cholesky factorization (see [46]) are not practical. On the other hand, the slow convergence rate of the steepest descent iteration for ill-conditioned systems limits its usefulness. In this case, an iterative technique known as the conjugate gradient (CG) method, together with an acceleration scheme known as preconditioning, provides a very efficient means of solving symmetric positive definite linear systems which are large and ill-conditioned. The CG method can also be adapted to solve nonquadratic optimization problems.

We begin with a version of CG for the minimization of positive quadratic functionals.

Algorithm 3.2.1. CG Method for Quadratic Minimization.
To minimize $J(\mathbf{f}) = c + \langle \mathbf{b}, \mathbf{f} \rangle + \frac{1}{2}\langle A\mathbf{f}, \mathbf{f} \rangle$, where A is SPD, or, equivalently, to solve $A\mathbf{f} = -\mathbf{b}$,

$\nu := 0$;
$\mathbf{f}_0 :=$ initial guess
$\mathbf{g}_0 := A\mathbf{f}_0 + \mathbf{b}$; *% initial gradient*
$\mathbf{p}_0 := -\mathbf{g}_0$; *% initial search direction*

$\delta_0 = ||\mathbf{g}_0||^2$;
begin CG iterations
$\mathbf{h}_\nu := A\mathbf{p}_\nu$;
$\tau_\nu := \delta_\nu / \langle \mathbf{p}_\nu, \mathbf{h}_\nu \rangle$; *% line search parameter*
$\mathbf{f}_{\nu+1} := \mathbf{f}_\nu + \tau_\nu \mathbf{p}_\nu$; *% update approximate solution*
$\mathbf{g}_{\nu+1} := \mathbf{g}_\nu + \tau_\nu \mathbf{h}_\nu$; *% update gradient*
$\delta_{\nu+1} = ||\mathbf{g}_{\nu+1}||^2$;
$\beta_\nu := \delta_{\nu+1} / \delta_\nu$;
$\mathbf{p}_{\nu+1} := -\mathbf{g}_{\nu+1} + \beta_\nu \mathbf{p}_\nu$; *% update search direction*
$\nu := \nu + 1$;
end CG iterations

We next provide a characterization of the CG iterates $\mathbf{f}_\nu$. See [5] or [102] for details.

Definition 3.6. The νth Krylov subspace generated by an SPD matrix A and a vector $\mathbf{v} \in \mathbb{R}^n$ is given by

$$\begin{aligned} \mathcal{S}_\nu(A, \mathbf{v}) &\stackrel{\text{def}}{=} span(\mathbf{v}, A\mathbf{v}, \ldots, A^{\nu-1}\mathbf{v}) \\ &= \{p(A)\mathbf{v} \mid p \in \Pi^{\nu-1}\}, \end{aligned}$$

where $\Pi^{\nu-1}$ denotes the set of polynomials of degree less than or equal to $\nu - 1$.

Theorem 3.7. *For $\nu = 1, 2, \ldots$, the CG iterates $\mathbf{f}_\nu$ satisfy*

$$\mathbf{f}_\nu = \arg \min_{\mathbf{f} \in \mathbf{f}_0 + \mathcal{S}_\nu(A, \mathbf{g}_0)} ||\mathbf{f} - \mathbf{f}_*||_A. \tag{3.11}$$

The corresponding iterative solution errors $\mathbf{e}_\nu = \mathbf{f}_\nu - \mathbf{f}_*$ satisfy

$$||\mathbf{e}_\nu||_A = \min_{q \in \pi_1^\nu} ||q(A)\mathbf{e}_0||_A, \tag{3.12}$$

where π_1^ν denotes the set of polynomials $q(t)$ of degree less than or equal to ν for which $q(0) = 1$.

Corollary 3.8. *If A is SPD, then for any $\mathbf{b} \in \mathbb{R}^n$ and any initial guess $\mathbf{f}_0 \in \mathbb{R}^n$, the CG algorithm will yield the exact solution to the system $A\mathbf{f} = -\mathbf{b}$ in at most n iterations.*

Remark 3.9. If $\mathbf{f}_0 = \mathbf{0}$, then CG generates a sequence of polynomial approximations $p_\nu(A)$ to $-A^{-1}$. If $\mathbf{b}$ is nondegenerate (i.e., the projections onto each of the eigenspaces are nonzero), then after at most n iterations, $p_\nu(\lambda_i) = -1/\lambda_i$ for each eigenvalue λ_i of A, and $p_\nu(A) = -A^{-1}$.

If A is a symmetric positive semidefinite matrix and $\mathbf{f}_0 = \mathbf{0}$, then CG will converge to the least squares minimum norm solution to $A\mathbf{f} = -\mathbf{b}$, $-A^\dagger \mathbf{b}$; see Definition 2.16. CG iteration can also be applied directly as a regularization method, with the iteration count ν playing the role of the regularization parameter. See [35] for details.

Using a particular scaled Chebyshev polynomial $p \in \pi_1^\nu$ in (3.12) [5], one can compute the iterative error bound

$$||\mathbf{e}_\nu||_A \leq 2 \left(\frac{\sqrt{\text{cond}(A)} - 1}{\sqrt{\text{cond}(A)} + 1} \right)^\nu ||\mathbf{e}_0||_A. \tag{3.13}$$

Comparing this with the steepest descent rate (3.10), we see that this bound is considerably smaller when cond(A) is large. One may obtain stronger convergence results if the eigenvalues of A are clustered away from zero [27], [65, Chapter 5]. By this we mean that the eigenvalues are all contained in a few small intervals, none of which contains the origin.

3.2.1 Preconditioning

The error bound (3.13) and the eigenvalue clustering results motivate the concept of preconditioning. By this we mean a transformation that yields a better eigenvalue distribution in the sense that the condition number is smaller or the eigenvalues are more clustered or both. Consider the quadratic functional (3.8) with SPD matrix A. We desire an SPD matrix M, called the preconditioner, for which $M^{-1}A$ has a better eigenvalue distribution than does A. In principle, we can compute a self-adjoint, positive square root of M, which we denote by $M^{1/2}$ (see Exercise 3.6), and transform (3.8):

$$\tilde{J}(\tilde{\mathbf{f}}) = c + \langle \tilde{\mathbf{b}}, \tilde{\mathbf{f}} \rangle + \frac{1}{2}\langle \tilde{A}\tilde{\mathbf{f}}, \tilde{\mathbf{f}} \rangle,$$

where

$$\tilde{A} = M^{-1/2} A M^{-1/2}, \quad \tilde{\mathbf{b}} = M^{-1/2}\mathbf{b}, \quad \tilde{\mathbf{b}} = M^{1/2}\mathbf{f}.$$

Note that $\tilde{A}$ is also SPD. Again, in principle one can apply the CG algorithm to compute a sequence of approximations $\tilde{f}_\nu$ to the minimizer of $\tilde{J}$. Then $M^{-1/2}\tilde{\mathbf{f}}_\nu$ approximates the minimizer of the original functional J. In practice, one can derive the following preconditioned CG algorithm (PCG) to compute $M^{-1/2}\tilde{\mathbf{f}}_\nu$ in a manner that requires M^{-1} instead of $M^{-1/2}$. See [5, p. 28] for details.

Algorithm 3.2.2. PCG Method.
To minimize $J(\mathbf{f}) = c + \langle \mathbf{b}, \mathbf{f} \rangle + \frac{1}{2}\langle A\mathbf{f}, \mathbf{f} \rangle$, where A is SPD, or, equivalently, to solve $A\mathbf{f} = -\mathbf{b}$, given an SPD preconditioning matrix M,

$\nu := 0;$
$\mathbf{f}_0 :=$ initial guess
$\mathbf{g}_0 := A\mathbf{f}_0 + \mathbf{b};$ *% initial gradient*
$\mathbf{z}_0 := M^{-1}\mathbf{g}_0;$ *% apply preconditioner*
$\mathbf{p}_0 := -\mathbf{z}_0;$ *% initial search direction*
$\delta_0 = \langle \mathbf{g}_0, \mathbf{z}_0 \rangle;$
begin PCG iterations
  $\mathbf{h}_\nu := A\mathbf{p}_\nu;$
  $\tau_\nu := \delta_\nu / \langle \mathbf{p}_\nu, \mathbf{h}_\nu \rangle;$ *% line search parameter*
  $\mathbf{f}_{\nu+1} := \mathbf{f}_\nu + \tau_\nu \mathbf{p}_\nu;$ *% update approximate solution*
  $\mathbf{g}_{\nu+1} := \mathbf{g}_\nu + \tau_\nu \mathbf{h}_\nu;$ *% update gradient*
  $\mathbf{z}_{\nu+1} := M^{-1}\mathbf{g}_{\nu+1};$ *% apply preconditioner*
  $\delta_{\nu+1} = \langle \mathbf{g}_{\nu+1}, \mathbf{z}_{\nu+1} \rangle;$
  $\beta_\nu := \delta_{\nu+1}/\delta_\nu;$
  $\mathbf{p}_{\nu+1} := -\mathbf{z}_{\nu+1} + \beta_\nu \mathbf{p}_\nu;$ *% update search direction*
  $\nu := \nu + 1;$
end PCG iterations

Remark 3.10. A comparison of the CG algorithm and the PCG algorithm reveals that both require one multiplication by the Hessian matrix A and two inner product computations at

each iteration. The PCG algorithm requires a preconditioning step, i.e., the solution of the linear system $M\mathbf{z} = \mathbf{g}_\nu$ to obtain $\mathbf{z}_\nu$, while CG does not. Hence PCG will significantly reduce computational cost if (i) one can cheaply solve linear systems $M\mathbf{z} = \mathbf{g}$ and (ii) the PCG convergence rate is significantly faster than the rate for CG.

3.2.2 Nonlinear CG Method

The CG method can be adapted to solve nonquadratic minimization problems. The following implementation is due to Fletcher and Reeves [37].

Algorithm 3.2.3. Nonlinear CG Method.
To minimize a (nonquadratic) functional $J(f)$,

$\nu := 0;$
$\mathbf{f}_0 :=$ initial guess
$\mathbf{g}_0 := \text{grad}\, J(\mathbf{f}_0);$ *% initial gradient*
$\mathbf{p}_0 := -\mathbf{g}_0;$ *% initial search direction*
$\delta_0 = ||\mathbf{g}_0||^2;$
begin CG iterations
$\tau_\nu := \arg\min_{\tau>0} J(\mathbf{f}_\nu + \tau\mathbf{p}_\nu);$ *% line search*
$\mathbf{f}_{\nu+1} := \mathbf{f}_\nu + \tau_\nu\mathbf{p}_{\nu+1};$ *% update approximate solution*
$\mathbf{g}_{\nu+1} := \text{grad}\, J(\mathbf{f}_{\nu+1});$ *% compute gradient*
$\delta_{\nu+1} := ||\mathbf{g}_{\nu+1}||^2;$
$\beta_\nu := \delta_{\nu+1}/\delta_\nu;$
$\mathbf{p}_{\nu+1} := -\mathbf{g}_{\nu+1} + \beta_\nu\mathbf{p}_\nu;$ *% update search direction*
$\nu := \nu + 1;$
end CG iterations

In a variant of Algorithm 3.2.2, due to Polak and Ribiere [94, p. 95], the computation of β_ν is replaced by

$$\beta_\nu = \frac{\langle \mathbf{g}_{\nu+1} - \mathbf{g}_\nu, \mathbf{g}_{\nu+1}\rangle}{\langle \mathbf{g}_{\nu+1} - \mathbf{g}_\nu, \mathbf{p}_\nu\rangle}. \tag{3.14}$$

See [94] for convergence analysis and implementation details.

3.3 Newton's Method

The second order expansion (2.44) motivates Newton's method. Let $\mathbf{f}_\nu$ be an estimate for the minimizer of J. Consider the quadratic approximation to $J(\mathbf{f}_\nu + \mathbf{s})$,

$$Q_\nu(\mathbf{s}) = J(\mathbf{f}_\nu) + \langle \text{grad}\, J(\mathbf{f}_\nu), \mathbf{s}\rangle + \frac{1}{2}\langle \text{Hess}\, J(\mathbf{f}_\nu)\mathbf{s}, \mathbf{s}\rangle. \tag{3.15}$$

If $\text{Hess}\, J(\mathbf{f}_\nu)$ is positive definite, then $Q_\nu(\mathbf{s})$ has a unique minimizer which satisfies

$$\text{grad}\, J(\mathbf{f}_\nu) + \text{Hess}\, J(\mathbf{f}_\nu)\mathbf{s} = \mathbf{0}. \tag{3.16}$$

Taking $\mathbf{f}_\nu + \mathbf{s}$ as the new estimate for the minimizer of J, we obtain the Newton iteration

$$\mathbf{f}_{\nu+1} = \mathbf{f}_\nu - [\text{Hess}\, J(\mathbf{f}_\nu)]^{-1}\text{grad}\, J(\mathbf{f}_\nu), \qquad \nu = 0, 1, \ldots. \tag{3.17}$$

Recall the definition of quadratic convergence in Definition 3.1.

Theorem 3.11. *Assume* $\text{grad}\, J(\mathbf{f}_*) = 0$, *the Hessian* $H_* = \text{Hess}\, J(\mathbf{f}_*)$ *is SPD, and the mapping* $\mathbf{f} \mapsto \text{Hess}\, J(\mathbf{f})$ *is Lipschitz continuous, i.e.,*

$$||\text{Hess}\, J(\mathbf{f}+\mathbf{h}) - \text{Hess}\, J(\mathbf{f})|| \le \gamma ||\mathbf{h}||,$$

where $\gamma > 0$ *is a* Lipschitz constant. *Define the Newton convergence constant*

$$c_* = \frac{\gamma}{\lambda_{\min}(H_*)}. \tag{3.18}$$

If $||\mathbf{f}_0 - \mathbf{f}_*|| < \frac{1}{2c_*}$, *then the Newton iterates* $\mathbf{f}_\nu$ *converge to* $\mathbf{f}_*$, *and the rate of convergence is quadratic with*

$$||\mathbf{f}_{\nu+1} - \mathbf{f}_*|| \le c_* ||\mathbf{f}_\nu - \mathbf{f}^*||^2. \tag{3.19}$$

Proof. See [32, p. 90]. The rate (3.19) follows from

$$\begin{aligned} \mathbf{f}_{\nu+1} - \mathbf{f}_* &= \mathbf{f}_\nu - H_\nu^{-1}(\mathbf{g}_\nu) - \mathbf{f}_* \\ &= H_\nu^{-1}[\mathbf{g}_* - \mathbf{g}_\nu - H_\nu(\mathbf{f}_* - \mathbf{f}_\nu)], \end{aligned} \tag{3.20}$$

where $\mathbf{g}_\nu = \text{grad}\, J(\mathbf{f}_\nu)$, $\mathbf{g}_* = \text{grad}\, J(\mathbf{f}_*)$, and $H_\nu = \text{Hess}\, J(\mathbf{f}_\nu)$. □

Equation (3.19) guarantees very rapid convergence near a minimizer $\mathbf{f}_*$. One can show (see Exercise 3.9) that near $\mathbf{f}_*$ the number of significant digits in the approximate solution will double at each iteration. In practice, Newton's method has several shortcomings. First, convergence is guaranteed only for $\mathbf{f}_0$ sufficiently close to a local minimizer $\mathbf{f}_*$. This difficulty can sometimes be alleviated with a globalization strategy. A globalization strategy can be viewed as an enhancement to increase the set of initial guesses $\mathbf{f}_0$ for which the method will converge. One approach is to incorporate a line search (see Definition 3.2). Another approach, called a trust region, is discussed in the next section.

Another possible shortcoming of Newton's method is that it may be quite expensive to compute Hessians $H = \text{Hess}\, J(\mathbf{f}_\nu)$ and solve linear systems $H\mathbf{s} = -\text{grad}\, J(_\nu)$ to obtain the Newton steps $\mathbf{s}$.

3.3.1 Trust Region Globalization of Newton's Method

In principle, the implementation of a trust region requires the solution $\mathbf{s}_\nu$ of a quadratic constrained minimization problem,

$$\min_{\mathbf{s}} Q_\nu(\mathbf{s}) \quad \text{subject to} \quad ||\mathbf{s}|| \le \Delta_\nu. \tag{3.21}$$

Here Q_ν is the quadratic approximation to J in (3.15), and Δ_ν is a positive scalar called the trust region radius, which is varied as the iteration proceeds. This yields iterates of the form

$$\mathbf{f}_{\nu+1} = \mathbf{f}_\nu - [\text{Hess}\, J(\mathbf{f}_\nu) + \gamma_\nu I]^{-1}\, \text{grad}\, J(\mathbf{f}_\nu), \tag{3.22}$$

where γ_ν is zero if the constraint in (3.21) is inactive at iteration ν (i.e., if $||\mathbf{s}_\nu|| < \Delta_\nu$), and γ_ν is a positive Lagrange multiplier otherwise. Trust region methods tend to be more robust for ill-conditioned problems than a line search. However, their implementation can be more problematic than a line search implementation. For instance, the exact solution of subproblem (3.21) can be very expensive when the number of unknowns is large. This has lead to several approximate solution schemes for this subproblem. One of the most effective for large-scale problems is the Steihaug–Toint algorithm [31, 104, 25], [91, p. 136]. CG iterations are applied to minimize $Q_\nu(\mathbf{s})$. These inner iterations are terminated when either (i) the CG iterates leave the trust region $||\mathbf{s}|| \le \Delta_\nu$, (ii) a CG residual stopping tolerance is reached, or (iii) negative curvature of Q_ν is detected.

3.3.2 The BFGS Method

An alternative to the Steihaug–Toint trust region–CG approach is to replace the true Hessian by an approximation derived from current and previous gradients of J. This yields a secant method [32, Chapter 9]. Perhaps the most popular secant method is the BFGS method, which was discovered by Broyden, Fletcher, Goldfarb, and Shanno. To implement BFGS, given an approximation H_ν to Hess $J(\mathbf{f}_\nu)$, one first computes $\mathbf{s}_\nu = \mathbf{f}_{\nu+1} - \mathbf{f}_\nu$ and $\mathbf{y}_\nu = \text{grad}\, J(\mathbf{f}_{\nu+1}) - \text{grad}\, J(\mathbf{f}_\nu)$. One then computes an approximation to Hess $J(\mathbf{f}_{\nu+1})$:

$$H_{\nu+1} = H_\nu - \frac{H_\nu \mathbf{s}_\nu \mathbf{s}_\nu^T H_\nu}{\langle H_\nu \mathbf{s}_\nu, \mathbf{s}_\nu \rangle} + \frac{\mathbf{y}_\nu \mathbf{y}_\nu^T}{\langle \mathbf{y}_\nu, \mathbf{s}_\nu \rangle}. \tag{3.23}$$

If H_ν is symmetric positive definite and the curvature condition

$$\langle \mathbf{y}_\nu, \mathbf{s}_\nu \rangle > 0 \tag{3.24}$$

holds, then $H_{\nu+1}$ will also be symmetric positive definite. BFGS can be combined with trust region globalization, but it is more commonly used with a line search.

Algorithm 3.3.1. BFGS Method with Line Search.
To minimize a (nonquadratic) functional $J(\mathbf{f})$,

$\nu := 0$;
$\mathbf{f}_0 :=$ initial guess for minimizer;
$H_0 :=$ initial guess for Hessian;
$\mathbf{g}_0 :=$ grad $J(\mathbf{f}_0)$; *% initial gradient*
begin quasi-Newton iterations
  $\mathbf{p}_{\nu+1} := -H_\nu^{-1}\mathbf{g}_\nu$; *% compute quasi-Newton step*
  $\tau_{\nu+1} := \arg\min_{\tau>0} J(\mathbf{f}_\nu + \tau \mathbf{p}_{\nu+1})$; *% line search*
  $\mathbf{f}_{\nu+1} := \mathbf{f}_\nu + \tau_{\nu+1}\mathbf{p}_{\nu+1}$; *% update approximate solution*
  $\mathbf{g}_{\nu+1} :=$ grad $J(\mathbf{f}_{\nu+1})$; *% new gradient*
  $H_{\nu+1} :=$ updated Hessian based on (3.23);
  $\nu := \nu + 1$;
end quasi-Newton iterations

BFGS has a limited memory variant [91, p. 224], which requires no explicit matrix storage for the approximate Hessian matrix. It is based on the recursion for the inverse,

$$H_{\nu+1}^{-1} = \left(I - \frac{\mathbf{s}_\nu \mathbf{y}_\nu^T}{\langle \mathbf{y}_\nu, \mathbf{s}_\nu \rangle}\right) H_\nu^{-1} \left(I - \frac{\mathbf{y}_\nu \mathbf{s}_\nu^T}{\langle \mathbf{y}_\nu, \mathbf{s}_\nu \rangle}\right) + \frac{\mathbf{s}_\nu \mathbf{s}_\nu^T}{\langle \mathbf{y}_\nu, \mathbf{s}_\nu \rangle}. \tag{3.25}$$

In standard implementations, the initial Hessian approximation H_0 is taken to be a scalar multiple of the identity. However, knowledge of the structure of the true Hessian may lead to a better choice. For instance, if one wishes to minimize a generalized Tikhonov functional of the form (2.45) and the (discretized) penalty, or regularization, functional has the form $\alpha\langle L\mathbf{f}, \mathbf{f}\rangle$, where L is SPD and $\alpha > 0$, then the choice $H_0 = \alpha L$ may be more appropriate than a scalar multiple of the identity [116].

3.4 Inexact Line Search

In this section we address the numerical solution of the line search minimization problem (3.4). For a more detailed discussion of line search methods, see [91, Chapter 3] or [32,

section 6.3]. To simplify notation, define

$$\phi(\tau) = J(\mathbf{f}_\nu + \tau \mathbf{p}_\nu), \tag{3.26}$$

where $\mathbf{f}_\nu$, $\mathbf{p}_\nu \in \mathbb{R}^n$ are fixed and $\mathbf{p}_\nu$ is a descent direction for J at $\mathbf{f}_\nu$, cf., (3.5). Problem (3.4) can then be reformulated as

$$\min_{\tau>0} \phi(\tau). \tag{3.27}$$

In a few instances, an exact solution to (3.26)–(3.27) can be easily computed, e.g., when J is quadratic, but this is typically not the case. Two important questions then arise:

(i) What constitutes an acceptable approximate solution to the line search minimization problem?

(ii) How does one efficiently compute such an acceptable approximate solution?

At first glance, it might appear that it suffices to find any value of $\overline{\tau}$ for which $\phi(\overline{\tau}) < \phi(0)$. If one sets $\mathbf{f}_{\nu+1} = \mathbf{f}_\nu + \overline{\tau}\mathbf{p}_\nu$, then one obtains a decrease in the cost functional, $J(\mathbf{f}_{\nu+1}) < J(\mathbf{f}_\nu)$. The following one-dimensional examples, taken from [32], illustrate why this is not sufficient.

Example 3.12. Let x play the role of f; let $J(x) = x^2$ with initial guess $x_0 = 2$; take the descent directions $p_\nu = -1$; and take step lengths $\tau_\nu = 2^{-\nu+1}$. This generates the sequence $x_\nu = 1 + 2^{-\nu}$. Clearly $J(x_{\nu+1}) < J(x_\nu)$, but the x_ν's converge to 1 instead of to the minimizer $x^* = 0$ of J. See the top plot in Figure 3.1.

In Example 3.12, the step length parameters τ_ν were too small to allow sufficient decrease in J for convergence. Simply requiring longer step lengths may not remedy this problem, as the following example shows.

Example 3.13. Again take $J(x) = x^2$ and $x_0 = 2$, but now take descent directions $p_\nu = (-1)^{\nu+1}$ and step lengths $\tau_\nu = 2 + 3 \times 2^{-\nu-1}$. Then $x_\nu = (-1)^\nu(1+2^{-\nu})$. Again $J(x_{\nu+1}) < J(x_\nu)$, but the x_ν's do not converge to $x^* = 0$. See the bottom plot in Figure 3.1.

The situation in Example 3.12 can be prevented by imposing the sufficient decrease condition,

$$\phi(\tau) \leq \phi(0) + c_1 \tau \phi'(0), \qquad \tau > 0, \tag{3.28}$$

where $0 < c_1 < 1$ is constant. The situation in Example 3.13 can be prevented by imposing the curvature condition,

$$\phi'(\tau) \geq c_2 \phi'(0), \qquad \tau > 0, \tag{3.29}$$

where $c_1 < c_2 < 1$. Conditions (3.28) and (3.29) are known collectively as the Wolfe conditions. If the step length parameter τ_ν satisfies both these conditions and, in addition, certain mild conditions of the cost function J and the search directions $\mathbf{p}_\nu$ are satisfied, then one can show that the sequence $\mathbf{f}_{\nu+1} = \mathbf{f}_\nu + \tau_\nu \mathbf{p}_\nu$ will converge to a critical point for J (i.e., a point where grad $J(\mathbf{f}) = \mathbf{0}$).

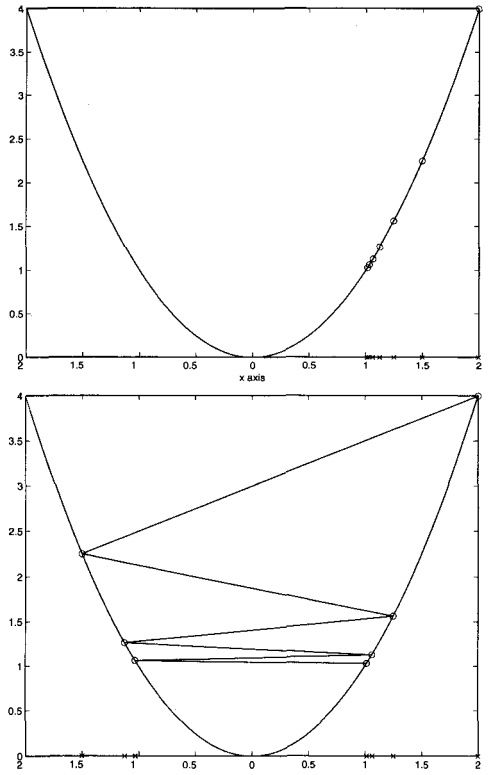


Figure 3.1. *Sequences that monotonically decrease the cost function but do not converge to a minimizer. In the top plot, the x's on the horizontal axis represent the sequence x_k described in Example* 3.12, *and the circles represent the points $(x_k, J(x_k))$. The bottom plot shows analogous information for Example* 3.13.

Inexact Line Search Algorithms

We sketch some basic ideas here. A more detailed development can be found in [91, section 3.4] or [36, section 2.6]. Line search algorithms typically have two stages. First comes a bracketing stage, in which one computes an initial interval $[a_0, b_0]$, $0 \leq a_0 < b_0$, guaranteed to contain points τ that are acceptable in the sense that they satisfy both Wolfe conditions. In the second stage, one finds an acceptable point within the initial interval.

To carry out the first stage, one might take an interval of the form $[0, 2^k]$, where k is a sufficiently large positive integer. This corresponds to simply doubling the length of the initial interval until it is guaranteed to contain acceptable points.

We now present a quadratic backtracking scheme to carry out the second stage. Assume that J (and hence ϕ) is smooth and that the initial interval $0 \leq \tau \leq \tau_0$ contains acceptable points, but τ_0 is not acceptable. The idea is to find the quadratic $q(\tau)$ that interpolates (i.e., matches) the data $\phi(0)$, $\phi'(0)$, and $\phi(\tau_0)$. One then minimizes $q(\tau)$ to obtain

$$\tau_1 = \frac{-\phi'(0)\,\tau_0^2}{2[\phi(\tau_0) - \phi(0) - \phi'(0)\tau_0]}. \tag{3.30}$$

If τ_1 is not acceptable, then one replaces τ_0 by τ_1 and this process until an acceptable point is

found. One can show that τ_k's generated by this approach are decreasing (see Exercise 3.10) and that one will eventually find an acceptable τ_k.

Exercises

3.1. Verify that equation (3.3) implies (3.2), which in turn implies (3.1).

3.2. Confirm that if J is continuously differentiable, then condition (3.6) guarantees that $\mathbf{p}_\nu$ is a descent direction.

3.3. Verify that the solution to the line search subproblem for the steepest descent method for quadratic functionals with positive definite Hessians is given by (3.9).

3.4. Confirm the inequalities (3.7), where A is an SPD matrix. $\lambda_{min}(A)$ and $\lambda_{max}(A)$, respectively, denote the smallest and largest eigenvalues of A. See Example 2.1.

3.5. Prove Corollary 3.8.

3.6. Let an SPD matrix M have an eigendecomposition $M = V\text{diag}(\lambda_i)\, V^T$, where $V^T V = I$. Show that $M^{1/2} = V\text{diag}(\sqrt{\lambda_i})\, V^T$ satisfies $\left(M^{1/2}\right)^2 = M$.

3.7. With $Q(\mathbf{s})$ given in (3.15) and under the assumption that Hess $J(\mathbf{f}_\nu)$ is positive definite, prove (3.16).

3.8. Verify the second equality in (3.20) and use it to obtain (3.19).

3.9. Show that if (3.19) holds, the number of significant digits in the solution, quantified by $\log_{10} ||\mathbf{f}_\nu - \mathbf{f}_*||$, will double at each iteration as $\mathbf{f}_\nu \to \mathbf{f}_*$.

3.10. Let $q(\tau)$ be the quadratic that satisfies $q(0) = \phi(0)$, $q'(0) = \phi'(0)$, and $q(\tau_0) = \phi(\tau_0)$. Show that $q'(\tau_1) = 0$ yields (3.30). Then show that if $\mathbf{p}_\nu$ is a descent direction (so $\phi'(0) < 0$) and the sufficient decrease condition (3.28) is not satisfied at $\tau_0 > 0$, then τ_1 minimizes q and $0 < \tau_1 < \tau_0$.

This page intentionally left blank

Chapter 4
Statistical Estimation Theory

Practical inverse problems involve measured data, which is inexact. Statistical models provide a rigorous, effective means with which to deal with measurement error. In addition, statistical estimation techniques can provide useful tools and insight into regularization. For example, the least squares fit-to-data functional (2.51) arises when maximum likelihood estimation is applied to a linear model with Gaussian statistics, and a variant of the Kullback–Leibler functional (2.52) arises from maximum likelihood estimation with Poisson statistics. The material presented here has direct applications in Chapter 7, which deals with regularization parameter selection methods. Also presented here is an iterative technique for maximum likelihood estimation known as the expectation maximation (EM) algorithm. This is used in Chapter 9 for nonnegatively constrained image deblurring.

4.1 Preliminary Definitions and Notation

We use the symbol Π to denote products, e.g., $\Pi_{i=1}^{n} i = 1 \times 2 \times \cdots \times n = n!$. The symbol $\emptyset$ denotes the empty set.

In statistical modeling the concepts of sample space, probability, and random variable play key roles. Intuitively, a sample space $\mathcal{S}$ is the set of all possible outcomes of an unpredictable experiment. For example, when tossing a coin, the possible outcomes are the occurrence of heads H and the occurrence of tails T. Then $\mathcal{S} = \{H, T\}$. Probability provides a means of quantifying how likely it is for an outcome to take place. If the coin is fair, then heads and tails are equally likely to occur, and one can assign a probability of $1/2$ to each of these two possible outcomes. Random variables assign numerical values to outcomes in the sample space. Once this has been done, one can systematically work with notions like average value, or mean, and variability. A rather terse mathematical development of these concepts is presented below. For a more detailed treatment of much of the material in this chapter, see, for example, [12].

It is customary in mathematical statistics to use capital letters to denote random variables and corresponding lowercase letters to denote values in the range of the random variables. If $X : \mathcal{S} \to \mathbb{R}$ is a random variable, then for any $x \in \mathbb{R}$, by $\{X \leq x\}$ we mean $\{s \in \mathcal{S} \mid X(s) \leq x\}$.

Definition 4.1. A probability space $(\mathcal{S}, \mathcal{B}, \mathcal{P})$ consists of a set $\mathcal{S}$ called the sample space, a collection $\mathcal{B}$ of subsets of $\mathcal{S}$ (see [12] for properties of $\mathcal{B}$), and a probability function

$\mathcal{P} : \mathcal{B} \to \mathbb{R}_+$ for which $\mathcal{P}(\emptyset) = 0$, $\mathcal{P}(\mathcal{S}) = 1$, and $\mathcal{P}(\cup_i S_i) = \sum_i \mathcal{P}(S_i)$ for any disjoint, countable collection of sets $S_i \in \mathcal{B}$. A random variable is a measurable function $X : \mathcal{S} \to \mathbb{R}$. Associated with the random variable X is its cumulative distribution function,

$$F_X(x) = \mathcal{P}\{X \leq x\}, \qquad x \in \mathbb{R}.$$

The cumulative distribution function is nondecreasing and right continuous and satisfies $\lim_{x\to-\infty} F_X(x) = 0$, $\lim_{x\to+\infty} F_X(x) = 1$.

Definition 4.2. A random variable X is called discrete if there exist countable sets $\{x_i\} \subset \mathbb{R}$ and $\{p_i\} \subset \mathbb{R}_+$ for which

$$p_i = \mathcal{P}\{X = x_i\} > 0 \quad \text{for each i, and} \quad \sum_i p_i = 1.$$

In this case, the probability mass function for X is the real-valued function with discrete support,

$$p_X(x) = \begin{cases} p_i & \text{if} \quad x = x_i, \qquad i = 1, 2, \ldots, \\ 0 & \text{otherwise.} \end{cases} \tag{4.1}$$

The x_i's are the points of discontinuity of the cumulative distribution function F_X, and one can represent

$$F_X(x) = \sum_{\{i \mid x_i \leq x\}} p_X(x_i) = \int_{-\infty}^{x} \left(\sum_i p_X(x_i)\delta(u - x_i) \right) du,$$

where $\delta(\cdot)$ denotes the Dirac delta.

Definition 4.3. X is called a continuous random variable if its cumulative distribution function F_X is absolutely continuous. In this case,

$$F_X(x) = \int_{-\infty}^{x} p_X(u)\, du,$$

and the derivative

$$p_X(x) = \frac{dF_X}{dx}$$

is called the probability density function for X.

Definition 4.4. The mean, or expected value, of a random variable X is given by the Riemann–Stieltjes integral

$$E(X) = \int_{-\infty}^{\infty} x\, dF_X(x).$$

If X is a continuous random variable, then $dF_X(x) = p_X(x)\, dx$, while in the discrete case, $dF_X(x) = p_X(x_i)\delta(x - x_i)$. In the latter case, $E(X) = \sum_i x_i p_X(x_i)$. The expectation operator E is a linear operator.

Example 4.5. To model the outcome of the toss of a fair coin, take $\mathcal{S} = \{T, H\}$, where T denotes the occurrence of tails and H denotes heads. Let $\mathcal{B} = \{\phi, \{H\}, \{T\}, \mathcal{S}\}$, and $\mathcal{P}(\phi) = 0$, $\mathcal{P}\{T\} = \mathcal{P}\{H\} = 1/2$, and $\mathcal{P}(\mathcal{S}) = 1$. Define $X : \mathcal{S} \to \mathbb{R}$ by $X(T) = 0$ and

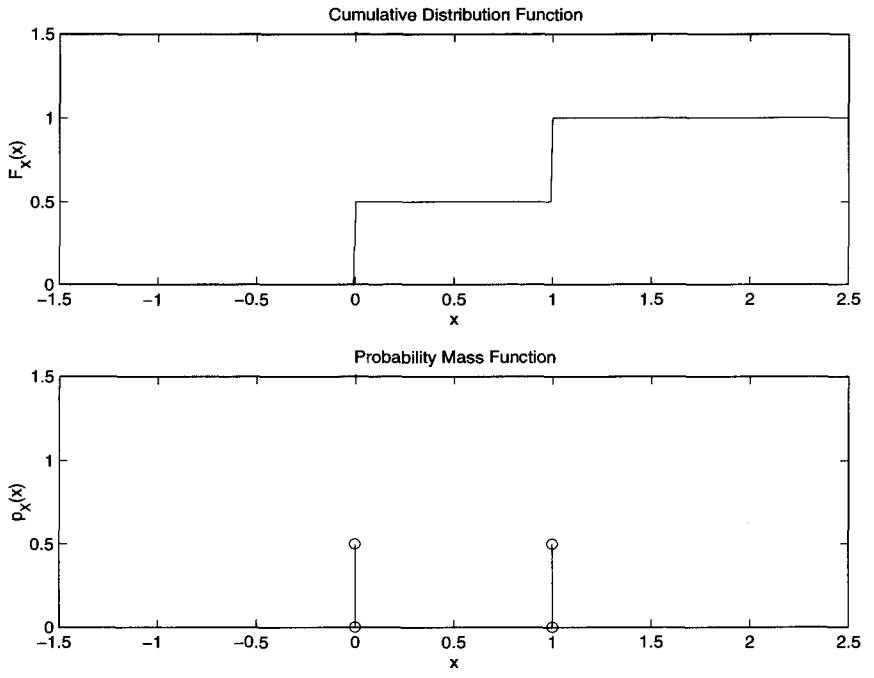


Figure 4.1. *The top plot shows the cumulative distribution function F_X for the discrete random variable described in Example* 4.5. *The bottom plot represents the corresponding probability mass function p_X.*

$X(H) = 1$. X is a discrete random variable whose probability mass function has the form (4.1) with $x_1 = 0, x_2 = 1$, and $p_1 = p_2 = 1/2$. The cumulative distribution function and the probability mass function for X are shown in Figure 4.1. Its expected value is $E(X) = 1/2$.

Example 4.6. When a fair dial is spun, it is equally likely to point in any direction. To model this situation, let the sample space S consist of all possible orientations of the dial. Let the random variable X denote the orientation angle, measured in radians clockwise from the vertical. Then X is uniformly distributed on the interval $[0, 2\pi]$. The cumulative distribution function for X is given by

$$F_X(x) = \begin{cases} 0, & x < 0, \\ \frac{x}{2\pi}, & 0 \le x < 2\pi, \\ 1, & x \ge 2\pi, \end{cases}$$

and the probability density function is given by

$$p_X(x) = \begin{cases} \frac{1}{2\pi}, & 0 < x < 2\pi, \\ 0 & \text{elsewhere.} \end{cases}$$

See Figure 4.2.

Definition 4.7. Two random variables X and Y are jointly distributed if they are both defined

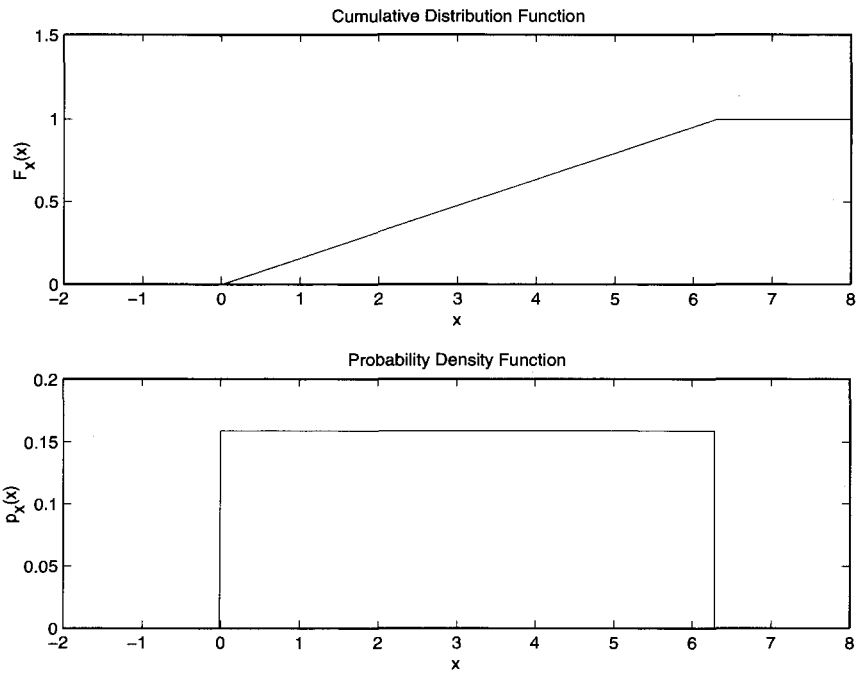


Figure 4.2. *The top plot shows the cumulative distribution function F_X for the continuous random variable described in Example* 4.6. *The bottom plot shows the corresponding probability density function p_X.*

on the same probability space $(\mathcal{S}, \mathcal{B}, \mathcal{P})$. Jointly distributed random variables X and Y are equal, denoted by $X = Y$, if $\mathcal{P}\{X = Y\} = 1$. X is distributed as Y, denoted by $X \sim Y$, if X and Y have the same cumulative distribution function.

Remark 4.8. Jointly distributed random variables with the same distribution need not be equal. For example, let X be as in Example 4.5, and define a jointly distributed random variable Y by $Y(T) = 1$ and $Y(H) = 0$. Then $X \sim Y$ but $X \neq Y$.

Definition 4.9. A random vector $\mathbf{X} = (X_1, \ldots, X_n)$ is a mapping from $\mathcal{S}$ into $\mathbb{R}^n$ for which all the components X_i are jointly distributed. The joint distribution function of $\mathbf{X}$ is given by

$$F_{\mathbf{X}}(\mathbf{x}) = \mathcal{P}\{X_1 \leq x_1, \ldots, X_n \leq x_n\}, \quad \mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{R}^n.$$

The components X_i are independent if the joint distribution function is the product of the cumulative distribution functions for the components, i.e., if $F_{\mathbf{X}}(\mathbf{x}) = \Pi_i F_{X_i}(x_i)$.

Definition 4.10. A random vector $\mathbf{X}$ is discrete if there exists countable sets $\{\mathbf{x}_i\} \subset \mathbb{R}^n$ and $\{p_i\} \subset \mathbb{R}_+$ for which

$$p_i = \mathcal{P}\{\mathbf{X} = \mathbf{x}_i\} > 0 \quad \text{for each i, and} \quad \sum_i p_i = 1.$$

The joint probability mass function for $\mathbf{X}$ is then given by

$$p_{\mathbf{X}}(\mathbf{x}) = \begin{cases} p_i & \text{if } \mathbf{x} = \mathbf{x}_i, \quad i = 1, 2, \ldots, \\ 0 & \text{otherwise,} \end{cases}$$

and

$$F_{\mathbf{X}}(\mathbf{x}) = \int_{-\infty}^{x_1} \cdots \int_{-\infty}^{x_n} \left(\sum_i p_i \delta(\mathbf{u} - \mathbf{x}) \right) du_1 \ldots du_n.$$

Definition 4.11. A random vector $\mathbf{X}$ is continuous with joint probability density function $p_{\mathbf{X}}$ if

$$F_{\mathbf{X}}(\mathbf{x}) = \int_{-\infty}^{x_1} \cdots \int_{-\infty}^{x_n} p_{\mathbf{X}}(\mathbf{u})\, du_1 \ldots du_n.$$

In either the discrete or the continuous case, if the components X_i are independent, then

$$p_{\mathbf{X}}(\mathbf{x}) = \Pi_{i=1}^n p_{X_i}(x_i), \tag{4.2}$$

where p_{X_i} denotes the probability density/mass function for X_i.

Definition 4.12. The mean or expected value of a random vector $\mathbf{X} = (X_1, \ldots, X_n)$ is the n-vector $E(\mathbf{X})$ with components

$$[E(\mathbf{X})]_i = E(X_i), \qquad i = 1, \ldots, n.$$

The covariance of $\mathbf{X}$ is the $n \times n$ matrix $\operatorname{cov}(\mathbf{X})$ with components

$$[\operatorname{cov}(\mathbf{X})]_{ij} = E((X_i - \mu_i)(X_j - \mu_j)), \qquad 1 \le i, j \le n, \tag{4.3}$$

where $\mu_i = E(X_i)$.

Example 4.13. A continuous random vector $\mathbf{X}$ has a nondegenerate Gaussian, or normal, distribution if its joint probability density function has the form

$$p_{\mathbf{X}}(\mathbf{x}; \boldsymbol{\mu}, C) = \frac{1}{\sqrt{(2\pi)^n \det(C)}} \exp\left(-\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu})^T C^{-1} (\mathbf{x} - \boldsymbol{\mu}) \right), \tag{4.4}$$

where $\mathbf{x}, \boldsymbol{\mu} \in \mathbb{R}^n$, C is an $n \times n$ symmetric positive definite matrix, and $\det(\cdot)$ denotes matrix determinant. The mean is given by $E(\mathbf{X}) = \boldsymbol{\mu}$ and the covariance matrix is $\operatorname{cov}(\mathbf{X}) = C$. These parameters characterize the distribution, and we indicate this situation by $\mathbf{X} \sim \text{Normal}(\boldsymbol{\mu}, C)$.

Example 4.14. A discrete random vector $\mathbf{X}$ has a Poisson distribution with independent components if its joint probability mass function has the form

$$p_{\mathbf{X}}(\mathbf{x}; \boldsymbol{\lambda}) = \begin{cases} \Pi_{i=1}^n \frac{e^{-\boldsymbol{\lambda}_i} \boldsymbol{\lambda}_i^{x_i}}{x_i!}, & \mathbf{x} \in \mathbf{Z}_+^n, \\ 0 & \text{elsewhere,} \end{cases} \tag{4.5}$$

where $\mathbf{Z}_+^n$ denotes the set of n-vectors $\mathbf{x}$ whose components x_i each lie in the set of nonnegative integers. $\boldsymbol{\lambda}$ is called the Poisson parameter vector, and it characterizes the distribution. This is denoted by $\mathbf{X} \sim \text{Poisson}(\boldsymbol{\lambda})$. The components λ_i of $\boldsymbol{\lambda}$ are all nonnegative real numbers. One can show that $E(\mathbf{X}) = \boldsymbol{\lambda}$ and $\operatorname{cov}(\mathbf{X}) = \operatorname{diag}(\lambda_i, \ldots, \lambda_n)$. See Exercise 4.4.

4.2 Maximum Likelihood Estimation

Suppose a random vector $\mathbf{X}$ has a joint probability density/mass function $p_{\mathbf{X}}(\mathbf{x}; \boldsymbol{\theta})$, where $\boldsymbol{\theta}$ is an unknown parameter vector, and a data vector $\mathbf{d} = (d_1, \ldots, d_n)$ is a given realization of $\mathbf{X}$. By this we mean that $\mathbf{d}$ is an outcome of a random experiment. In terms of the abstract framework of Definition 4.1, $\mathbf{d} = \mathbf{X}(s)$ for some s in the underlying sample space $\mathcal{S}$.

Definition 4.15. A maximum likelihood estimator (MLE) for $\boldsymbol{\theta}$ given $\mathbf{d}$ is a parameter $\hat{\theta}$ that maximizes the likelihood function

$$L(\boldsymbol{\theta}) \stackrel{\text{def}}{=} p_{\mathbf{X}}(\mathbf{d}; \boldsymbol{\theta}). \tag{4.6}$$

Monotone transformations like the logarithm preserve maximizers and can simplify MLE computations. Hence, an MLE is a maximizer of the log likelihood function,

$$\ell(\boldsymbol{\theta}) = \log p_{\mathbf{X}}(\mathbf{d}; \boldsymbol{\theta}). \tag{4.7}$$

Example 4.16. Suppose $\mathbf{d}$ is a realization of a Gaussian random vector $\mathbf{X} \sim \text{Normal}(\boldsymbol{\mu}, C)$, where the covariance matrix C is known but the mean vector $\boldsymbol{\mu}$ is unknown. From (4.4), the log likelihood function becomes

$$\ell(\boldsymbol{\mu}) = -\frac{1}{2}(\mathbf{d} - \boldsymbol{\mu})^T C^{-1}(\mathbf{d} - \boldsymbol{\mu}) + c,$$

where $c = \frac{1}{2}(n \log(2\pi) + \log(\det(C)))$ is independent of the unknown $\boldsymbol{\mu}$. Since C is positive definite, the MLE is $\hat{\mu}_{\text{MLE}} = \mathbf{d}$.

Example 4.17. Suppose that $\mathbf{d}$ is a realization of $\mathbf{X} \sim \text{Poisson}(\boldsymbol{\lambda})$, where the Poisson parameter $\boldsymbol{\lambda}$ is unknown. Then from (4.5),

$$-\ell(\boldsymbol{\lambda}) = \sum_{i=1}^{n} (\lambda_i - d_i \log \lambda_i) + c, \tag{4.8}$$

where $c = \sum_{i=1}^{n} \log(d_i!)$. One can show (see Exercise 4.5) that $-\ell(\boldsymbol{\lambda})$ is strictly convex on the interior of $\mathbb{R}^n_+$, and that $\hat{\lambda}_{\text{MLE}} = \mathbf{d}$.

Additional information can be incorporated by viewing the parameter vector as a realization of a random vector. This gives rise to the Bayesian approach to estimation.

4.3 Bayesian Estimation

We begin with a discussion of conditional probability and conditional expectation. Let $\mathbf{X} = (X_1, \ldots, X_n)$ and $\mathbf{Y} = (Y_1, \ldots, Y_m)$ be jointly distributed discrete random vectors, i.e., the components X_i and Y_j are all discrete random variables defined on the same probability space. Then $(\mathbf{X}, \mathbf{Y})$ is also a discrete random vector.

Definition 4.18. The joint probability mass function for $(\mathbf{X}, \mathbf{Y})$ is given by

$$p_{(\mathbf{X},\mathbf{Y})}(\mathbf{x}, \mathbf{y}) = \mathcal{P}\{\mathbf{X} = \mathbf{x},\ \mathbf{Y} = \mathbf{y}\}, \qquad (\mathbf{x}, \mathbf{y}) \in \mathbb{R}^n \times \mathbb{R}^m.$$

The marginal probability mass function of $\mathbf{X}$ is then defined to be

$$p_{\mathbf{X}}(\mathbf{x}) = \sum_{\mathcal{P}\{\mathbf{Y}=\mathbf{y}\}>0} p_{(\mathbf{X},\mathbf{Y})}(\mathbf{x}, \mathbf{y}), \qquad \mathbf{x} \in \mathbb{R}^n. \tag{4.9}$$

Table 4.1.

Outcome	X_1	X_2	$Y = X_1 + X_2$	Probability
(T,T)	0	0	0	1/4
(T,H)	0	1	1	1/4
(H,T)	1	0	1	1/4
(H,H)	1	1	2	1/4

The conditional probability mass function for $\mathbf{Y}$ given $\mathbf{X} = \mathbf{x}$ is given by

$$p_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}) = \frac{p_{(\mathbf{X},\mathbf{Y})}(\mathbf{x}, \mathbf{y})}{p_{\mathbf{X}}(\mathbf{x})}, \tag{4.10}$$

provided the denominator is nonzero.

Remark 4.19. If $\mathbf{X}$ and $\mathbf{Y}$ are independent random vectors, then (see Exercise 4.7) the conditional probability mass function of $\mathbf{Y}$ given $\mathbf{X} = \mathbf{x}$ does not depend on $\mathbf{x}$, and it satisfies

$$p_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}) = p_{\mathbf{Y}}(\mathbf{y}). \tag{4.11}$$

Definition 4.20. Let $\phi : \mathbb{R}^n \to \mathbb{R}^k$ be a measurable mapping. The conditional expectation of $\phi(\mathbf{Y})$ given $\mathbf{X} = \mathbf{x}$ is

$$E(\phi(\mathbf{Y})|\mathbf{X} = \mathbf{x}) = \sum_{\mathcal{P}\{\mathbf{Y}=\mathbf{y}\}>0} \phi(\mathbf{y})\, p_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}), \qquad \mathbf{x} \in \mathbb{R}^n. \tag{4.12}$$

This is a mapping from $\mathbb{R}^n$ into $\mathbb{R}^k$.

Note that $E(\phi(\mathbf{Y})|\mathbf{X} = \mathbf{x})$ need not be in Range$(\phi(\mathbf{Y}))$. This is illustrated by the following example.

Example 4.21. Suppose two independent fair coins are tossed. To model the outcome, let X_1 and X_2 be independent random variables with the same distribution as the random variable X in Example 4.5, and set $Y = X_1 + X_2$. Y represents the total number of heads that appear. Table 4.1 summarizes the possible outcomes and their probabilities.

We now apply Definition 4.20 with $\phi(\mathbf{Y}) = \mathbf{Y}$. The marginal probability mass function of Y is given by $p_Y(0) = \mathcal{P}\{Y = 0\} = 1/4$, $p_Y(1) = 1/2$, $p_Y(2) = 1/4$, and the expected value is $E(Y) = 1$. The conditional probability mass function of Y given $X_1 = x_1$, $p_{Y|X_1}(y|x_1)$, is given in Table 4.2.

The conditional expectation $E(Y|X_1 = x_1)$ equals $1/2$ when $x_1 = 0$ and it equals $3/2$ when $x_1 = 1$. This illustrates that specifying whether the first coin is tails or heads significantly changes the probabilities and the expectation associated with the total number of heads.

Remark 4.22. For continuous random vectors $\mathbf{X}$, $\mathbf{Y}$, one can define analogous concepts like joint and marginal probability density functions, conditional probability density function of $\mathbf{Y}$ given $\mathbf{X} = \mathbf{x}$, and conditional expectation of $\phi(\mathbf{Y})$ given $\mathbf{X} = \mathbf{x}$. To do this, one essentially replaces summations in (4.9)–(4.12) by appropriate integrals.

Table 4.2.

	$x_1 = 0$	$x_1 = 1$
$y = 0$	1/2	0
$y = 1$	1/2	1/2
$y = 2$	0	1/2

The following proposition can greatly simplify computations involving conditional distributions.

Proposition 4.23. *Let* $\mathbf{X} = (X_1, \ldots, X_n)$ *and* $\mathbf{Y} = (Y_1, \ldots, Y_m)$ *be jointly distributed random vectors, let* $\mathbf{g} : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^p$ *be a measurable function, and set* $\mathbf{Z} = \mathbf{g}(\mathbf{X}, \mathbf{Y})$. *Then the conditional probability mass/density function for* $\mathbf{Z}$ *given* $\mathbf{X} = \mathbf{x}$ *is given by* $p_{\mathbf{g}(\mathbf{X},\mathbf{Y})|\mathbf{X}}(\mathbf{z}|\mathbf{x})$.

Proof. For simplicity, assume that $n = m = p = 1$ and that X and Y are discrete random variables. The joint probability mass function for Z and X evaluated at any $(z_i, x_j) \in \text{Range}(Z) \times \text{Range}(X)$ is given by

$$\begin{aligned} p_{Z,X}(z_i, x_j) &= \mathcal{P}\{Z = z_i,\ X = x_j\} \\ &= \mathcal{P}\{g(x_j, Y) = z_i,\ X = x_j\}. \end{aligned} \tag{4.13}$$

The marginal probability mass function for X evaluated at x_j is

$$\begin{aligned} p_X(x_j) &= \sum_{z_i \in \text{Range}(Z)} \mathcal{P}\{g(x_j, Y) = z_i, X = x_j\} \\ &= \sum_{z_i \in \text{Range}(g_j)} \mathcal{P}\{Y \in g_j^{-1}(z_i), X = x_j\}, \end{aligned}$$

where $g_j = g(x_j, \cdot)$ maps Range(Y) into $\mathbb{R}^1$. But $z_i \in \text{Range}(g_j)$ if and only if $z_i = g_j(y_i)$ for some $y_i \in \text{Range}(Y)$. Hence,

$$p_X(x_j) = \sum_{y_i \in \text{Range}(Y)} \mathcal{P}\{Y = y_i, X = x_j\}. \tag{4.14}$$

The ratio of the right-hand sides of (4.13) and (4.14) gives the conditional probability mass function for the random variable $g(X, Y)$ given $X = x_j$. □

The following result, known as Bayes' law, relates the conditional random vector $\mathbf{X}|_{\mathbf{Y}=\mathbf{y}}$ to the inverse conditional random vector, $\mathbf{Y}|_{\mathbf{X}=\mathbf{x}}$.

Theorem 4.24. *Let* $\mathbf{X}$ *and* $\mathbf{Y}$ *be jointly distributed random vectors. Then*

$$p_{\mathbf{X}|\mathbf{Y}}(\mathbf{x}|\mathbf{y}) = \frac{p_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}) p_{\mathbf{X}}(\mathbf{x})}{p_{\mathbf{Y}}(\mathbf{y})}. \tag{4.15}$$

Definition 4.25. In the context of (4.15), suppose $\mathbf{X}$ represents the variable of interest and $\mathbf{Y}$ represents an observable quantity that depends on X. Then $p_{\mathbf{X}}(\mathbf{x})$ is called the

a priori probability mass/density function, or simply the prior, while $p_{\mathbf{X}|\mathbf{Y}}(\mathbf{x}|\mathbf{y})$ is called the a posteriori mass/density function. A maximizer of $p_{\mathbf{X}|\mathbf{Y}}(\mathbf{x}|\mathbf{y})$ with respect to $\mathbf{x}$ is called a maximum a posteriori estimator, or MAP estimator.

The following example establishes a connection between MAP estimation and Tikhonov regularization.

Example 4.26. Let $\mathbf{X}$ and $\mathbf{N}$ be independent, jointly distributed random vectors with $\mathbf{X} \sim \text{Normal}(\mathbf{0}_n, C_{\mathbf{X}})$ and $\mathbf{N} \sim \text{Normal}(\mathbf{0}_m, C_{\mathbf{N}})$, where $\mathbf{0}_p$ denotes the zero vector in $\mathbb{R}^p$. Let K be an $m \times n$ matrix, and define a new random vector

$$\mathbf{Z} = K\mathbf{X} + \mathbf{N}. \tag{4.16}$$

Suppose a realization $\mathbf{z} \in \mathbb{R}^m$ from the random vector $\mathbf{Z}$ is given. We wish to derive the MAP estimator of $\mathbf{x}$ based on the observation $\mathbf{z} = K\mathbf{x} + \mathbf{n}$. From Proposition 4.23 and the independence of $\mathbf{N}$ and $\mathbf{X}$,

$$\begin{aligned} p_{\mathbf{Z}|\mathbf{X}}(\mathbf{z}|\mathbf{x}) &= p_{K\mathbf{X}+\mathbf{N}|\mathbf{X}}(\mathbf{z}|\mathbf{x}) \\ &= p_{K\mathbf{x}+\mathbf{N}}(\mathbf{z}). \end{aligned} \tag{4.17}$$

But $K\mathbf{x} + \mathbf{N} \sim \text{Normal}(K\mathbf{x}, C_{\mathbf{N}})$. From Example 4.13 and (4.17),

$$p_{\mathbf{Z}|\mathbf{X}}(\mathbf{z}|\mathbf{x}) = \frac{\exp(-(\mathbf{z} - K\mathbf{x})^T C_{\mathbf{N}}^{-1}(\mathbf{z} - K\mathbf{x})/2)}{\sqrt{(2\pi)^m \det(C_{\mathbf{N}})}}. \tag{4.18}$$

From (4.4), the prior is given by

$$p_{\mathbf{X}}(\mathbf{x}) = \frac{\exp(-(\mathbf{x}^T C_{\mathbf{X}}^{-1}\mathbf{x})/2)}{\sqrt{(2\pi)^n \det(C_{\mathbf{X}})}}. \tag{4.19}$$

Combining (4.15) with (4.19) and (4.18), one obtains the a posteriori log likelihood function,

$$\begin{aligned} \ell(\mathbf{x}|\mathbf{z}) &= \log p_{\mathbf{X}|\mathbf{Z}}(\mathbf{x}|\mathbf{z}) \\ &= -\frac{1}{2}(\mathbf{z} - K\mathbf{x})^T C_{\mathbf{N}}^{-1}(\mathbf{z} - K\mathbf{x}) - \frac{1}{2}\mathbf{x}^T C_{\mathbf{X}}^{-1}\mathbf{x} + c, \end{aligned} \tag{4.20}$$

where c is constant with respect to $\mathbf{x}$. Then the MAP estimator, obtained by maximizing (4.20) with respect to $\mathbf{x}$, is given by

$$\mathbf{x} = (K^T C_{\mathbf{N}}^{-1} K + C_{\mathbf{X}}^{-1})^{-1} K^T C_{\mathbf{N}}^{-1}\mathbf{z}. \tag{4.21}$$

Note that if $C_{\mathbf{N}} = \sigma_{\mathbf{N}}^2 I_m$ and $C_{\mathbf{X}} = \sigma_{\mathbf{X}}^2 I_n$, then

$$\mathbf{x} = (K^T K + \alpha I_m)^{-1} K^T \mathbf{z}, \tag{4.22}$$

where $\alpha = (\sigma_{\mathbf{N}}/\sigma_{\mathbf{X}})^2$, and the MAP estimator has the same form as the solution obtained by Tikhonov regularization; see (1.15). In this case the square root of α can be interpreted as a noise-to-signal ratio.

4.4 Linear Least Squares Estimation

Let $\mathbf{X} = (X_1, \ldots, X_n)$ and $\mathbf{Z} = (Z_1, \ldots, Z_n)$ be jointly distributed, real-valued random vectors with finite expected squared components,

$$E(X_i^2) < \infty, \quad i = 1, \ldots, n, \qquad E(Z_j^2) < \infty, \quad j = 1, \ldots, m.$$

Definition 4.27. The cross correlation matrix for $\mathbf{Y}$ and $\mathbf{Z}$ is the $n \times m$ matrix $\Gamma_{\mathbf{XZ}} = E(\mathbf{X}\mathbf{Z}^T)$ with entries

$$[\Gamma_{\mathbf{XZ}}]_{ij} = E(X_i Z_j), \qquad i = 1, \ldots, n, \ j = 1, \ldots, m. \tag{4.23}$$

The autocorrelation matrix for $\mathbf{X}$ is $\Gamma_{\mathbf{XX}} = E(\mathbf{X}\mathbf{X}^T)$ with entries

$$[\Gamma_{\mathbf{XX}}]_{ij} = E(X_i X_j), \qquad 1 \le i, j \le n. \tag{4.24}$$

Note that $\Gamma_{\mathbf{ZX}} = \Gamma_{\mathbf{XZ}}^T$ and $\Gamma_{\mathbf{XX}}$ is symmetric and positive semidefinite. Also, if $E(\mathbf{X}) = \mathbf{0}$, then the auto correlation reduces to the covariance, $\Gamma_{\mathbf{XX}} = \text{cov}(\mathbf{X})$.

Definition 4.28. Let A be an $n \times n$ matrix with entries a_{ij}. The trace of A is the sum of the diagonal components:

$$\text{trace}(A) = \sum_{i=1}^{n} a_{ii}. \tag{4.25}$$

Note that the trace is a linear operator, i.e.,

$$\text{trace}(\alpha A + \beta B) = \alpha \text{trace}(A) + \beta \text{trace}(B), \tag{4.26}$$

and that $\text{trace}(A) = \text{trace}(A^T)$. The following proposition relates least squares and trace computations. See Exercise 4.12 for proof.

Proposition 4.29. *If a random vector* $\mathbf{X}$ *has finite expected squared components, then*

$$E(||\mathbf{X}||^2) = \text{trace}(\Gamma_{\mathbf{XX}}). \tag{4.27}$$

The next result relates the eigenvalues to the trace of a symmetric matrix. See Exercise 4.13 for a proof.

Proposition 4.30. *If A is symmetric, then the trace of A is the sum of the eigenvalues λ_i of A, i.e.,*

$$\text{trace}(A) = \sum_{i=1}^{n} \lambda_i. \tag{4.28}$$

4.4.1 Best Linear Unbiased Estimation

Consider the linear model

$$\mathbf{Z} = K\mathbf{x} + \mathbf{N}, \tag{4.29}$$

where K is an $m \times n$ matrix, $\mathbf{x} \in \mathbb{R}^n$ is deterministic, and $\mathbf{N}$ is a random n-vector with

$$E(\mathbf{N}) = \mathbf{0}, \tag{4.30}$$

and

(4.31) $$C_{\mathbf{N}} = \text{cov}(\mathbf{N})$$

is a known, nonsingular, $n \times n$ matrix.

Definition 4.31. The best linear unbiased estimator for $\mathbf{x}$ from $\mathbf{Z}$ is the vector $\hat{\mathbf{X}}_{\text{BLUE}}$, which minimizes

(4.32) $$J(\hat{\mathbf{X}}) = E\left(||\hat{\mathbf{X}} - \mathbf{x}||^2\right)$$

subject to the constraints

(4.33) $$\hat{\mathbf{X}} = B\mathbf{Z}, \qquad B \in \mathbb{R}^{n\times m},$$

(4.34) $$E(\hat{\mathbf{X}}) = \mathbf{x}.$$

$\hat{\mathbf{x}}_{\text{BLUE}}$ is called linear because of (4.33) and unbiased because of (4.34).

Theorem 4.32 (Gauss–Markov). *If K has full rank, then the best linear unbiased estimator is given by*

(4.35) $$\hat{\mathbf{X}}_{\text{BLUE}} = \hat{B}\mathbf{Z},$$

where

(4.36) $$\hat{B} = (K^T C_N^{-1} K)^{-1} K^T C_N^{-1}.$$

Proof. Note that since K has full rank, $K^T C_N^{-1} K$ is invertible. Consider a candidate solution

$$B = \hat{B} + M, \qquad M \in \mathbb{R}^{n\times m}.$$

Conditions (4.33)–(4.34) together with (4.29) and (4.30) give

$$\begin{aligned}\mathbf{x} &= (\hat{B} + M)E(K\mathbf{x} + \mathbf{N}) \\ &= \hat{B}K\mathbf{x} + MK\mathbf{x} \\ &= \mathbf{x} + MK\mathbf{x}.\end{aligned}$$

If this holds for all $\mathbf{x} \in \mathbb{R}^n$, then

(4.37) $$MK = 0.$$

Next, consider

$$\begin{aligned}J(B) &= E\left(||B\mathbf{Z} - \mathbf{x}||^2\right) \\ &= \text{trace}\left\{E\left[(B\mathbf{Z} - \mathbf{x})(B\mathbf{Z} - \mathbf{x})^T\right]\right\} \quad \text{by Proposition (4.29)} \\ &= \text{trace}(BK\mathbf{x}\mathbf{x}^T K^T B^T) + \text{trace}(BC_N B^T) - 2\text{trace}(BK\mathbf{x}\mathbf{x}^T) \\ &\qquad + \text{trace}(\mathbf{x}\mathbf{x}^T).\end{aligned}$$

A straightforward computation utilizing condition (4.37) shows that

(4.38) $$J(\hat{B} + M) = J(\hat{B}) + \text{trace}\left(MC_N M^T\right).$$

As a consequence of Proposition 4.30 and the fact that $MC_N M^T$ is positive semidefinite, $\text{trace}(MC_N M^T) \geq 0$. Let $\Theta_{n\times m}$ denote the $n \times m$ zero matrix. If $M \neq \Theta_{n\times m}$, then $MC_N M^T \neq \Theta_{n\times n}$, since C_N is assumed to be nonsingular. But then $MC_N M^T$ has at least one positive eigenvalue. This implies by Proposition 4.30 that $\text{trace}(MC_N M^T)$

> 0 whenever $M \neq \Theta_{n\times m}$. Thus $J(\hat{B} + M) \geq J(\hat{B})$ with equality if and only if $M = \Theta_{n\times m}$. □

Remark 4.33. If the noise covariance matrix $C_N = \sigma^2 I$ and K has full rank, then

$$\hat{\mathbf{X}}_{\text{BLUE}} = (K^T K)^{-1} K^T \mathbf{Z} = K^\dagger \mathbf{Z}. \tag{4.39}$$

This corresponds in the deterministic case to the solution to the least squares problem $\min_{\mathbf{x}} ||K\mathbf{x} - \mathbf{z}||$. More generally, the best linear unbiased estimator (4.35)–(4.36) corresponds to the solution to the deterministic weighted least squares problem,

$$\min_{\mathbf{x}\in\mathbb{R}^n} ||K\mathbf{x} - \mathbf{y}||_{C_N^{-1}}, \tag{4.40}$$

where

$$||\mathbf{v}||_W = \sqrt{\langle \mathbf{v}, \mathbf{v}\rangle_W} = \sqrt{\mathbf{v}^T W \mathbf{v}}$$

denotes the energy norm.

4.4.2 Minimum Variance Linear Estimation

In light of Remark 4.33, the best linear unbiased estimator is unsuitable for the solution of noisy ill-conditioned linear systems. To remedy this situation, we assume that the $\mathbf{x}$ is a realization of a random vector $\mathbf{X}$. What follows can be viewed as a linear least squares analogue of Bayesian estimation.

Definition 4.34. Suppose $\mathbf{X}$ and $\mathbf{Z}$ are jointly distributed random vectors whose components have finite expected squares. The minimum variance linear estimator of $\mathbf{X}$ from $\mathbf{Z}$ is given by

$$\hat{\mathbf{X}} = \hat{B}\mathbf{Z},$$

where

$$\hat{B} = \arg\min_{B\in\mathbb{R}^{n\times m}} E(||B\mathbf{Z} - \mathbf{X}||^2).$$

Proposition 4.35. *If $\Gamma_{\mathbf{ZZ}}$ is nonsingular, then the minimum variance linear estimator of $\mathbf{X}$ from $\mathbf{Z}$ is given by*

$$\hat{\mathbf{X}} = \left(\Gamma_{\mathbf{XZ}}\Gamma_{\mathbf{ZZ}}^{-1}\right)\mathbf{Z}. \tag{4.41}$$

See Exercise 4.15 for proof.

Now consider the linear model (4.16). Unlike as in Example 4.26, we will not assume that $\mathbf{X}$ and $\mathbf{N}$ have Gaussian distributions. We will, however, assume that $\mathbf{N}$ has mean zero (see (4.30)) and that $\mathbf{X}$ and $\mathbf{N}$ are independent. Consequently,

$$\Gamma_{\mathbf{XN}} = \Theta_{n\times m}, \tag{4.42}$$

$$\Gamma_{\mathbf{XZ}} = \Gamma_{\mathbf{XX}} K^T, \tag{4.43}$$

$$\Gamma_{\mathbf{ZZ}} = K\Gamma_{\mathbf{XX}} K^T + C_{\mathbf{N}}, \tag{4.44}$$

where $C_{\mathbf{N}} = \text{cov}(\mathbf{N})$. From this and (4.41) we obtain the following expression for the minimum variance linear estimator:

$$\hat{\mathbf{X}} = \Gamma_{\mathbf{XX}} K^T [K\Gamma_{\mathbf{XX}} K^T + C_{\mathbf{N}}]^{-1}\mathbf{Z} \tag{4.45}$$

$$= [K^T C_{\mathbf{N}}^{-1} K + \Gamma_{\mathbf{XX}}^{-1}]^{-1} K^T C_{\mathbf{N}}^{-1}\mathbf{Z}. \tag{4.46}$$

The second equality is valid if $\Gamma_{\mathbf{XX}}$ is nonsingular. See Exercise 4.17. Comparing (4.46) with (4.21), we make the following observation.

Remark 4.36. If we assume the linear data model (4.16), where $\mathbf{X}$ and $\mathbf{N}$ are independent random vectors with zero means, then the form of the minimum variance linear estimator (4.46) is the same as that of the MAP estimator (4.21). Note that the derivation of the MAP estimator required the additional assumption that $\mathbf{X}$ and $\mathbf{N}$ are Gaussian random vectors.

4.5 The EM Algorithm

Let $\mathbf{Y}$ be a random vector with a parameter-dependent probability distribution. The EM algorithm is an iterative procedure that, given a realization of $\mathbf{Y}$, yields a sequence of approximations to a maximum likelihood estimator for the parameter. The algorithm relies on an auxiliary random vector $\mathbf{X}$ that corresponds to hidden or missing data. $\mathbf{X}$ and $\mathbf{Y}$ together make up the complete data. A very general development can be found in [83]. For simplicity, we suppose that $\mathbf{X}$ and $\mathbf{Y}$ are discrete and let them have a joint probability mass function $p_{(\mathbf{X},\mathbf{Y})}(\mathbf{x}, \mathbf{y}; \theta)$, where θ denotes the parameter of interest. Then the conditional probability mass function for $\mathbf{X}$ given $\mathbf{Y}$ is

$$p_{\mathbf{X}|\mathbf{Y}}(\mathbf{x}|\mathbf{y}; \theta) = \frac{p_{(\mathbf{X},\mathbf{Y})}(\mathbf{x}, \mathbf{y}; \theta)}{p_{\mathbf{Y}}(\mathbf{y}; \theta)}, \tag{4.47}$$

where the denominator gives the marginal probability mass function of Y,

$$p_{\mathbf{Y}}(\mathbf{y}; \theta) = \sum_{\mathbf{x}} p_{(\mathbf{X},\mathbf{Y})}(\mathbf{x}, \mathbf{y}; \theta). \tag{4.48}$$

By $\sum_{\mathbf{x}}$ we mean the sum over components $\mathbf{x}$ for which $P\{\mathbf{X} = \mathbf{x}\} > 0$. The log likelihood function for $\mathbf{Y}$ given observed data $\mathbf{y}$ takes the form

$$\begin{aligned} l_{\mathbf{Y}}(\theta; \mathbf{y}) &= \log p_{\mathbf{Y}}(\mathbf{y}; \theta) \\ &= \log p_{(\mathbf{X},\mathbf{Y})}(\mathbf{x}, \mathbf{y}; \theta) - \log p_{\mathbf{X}|\mathbf{Y}}(\mathbf{x}|\mathbf{y}; \theta) \\ &= l_{(\mathbf{X},\mathbf{Y})}(\theta; \mathbf{x}, \mathbf{y}) - l_{\mathbf{X}|\mathbf{Y}}(\theta; \mathbf{x}|\mathbf{y}). \end{aligned}$$

Then for any fixed parameter θ_ν,

$$\begin{aligned} l_{\mathbf{Y}}(\theta; \mathbf{y}) &= l_{\mathbf{Y}}(\theta; \mathbf{y}) \sum_{\mathbf{x}} p_{\mathbf{X}|\mathbf{Y}}(\mathbf{x}|\mathbf{y}; \theta_\nu) \quad \text{by (4.47)–(4.48)} \\ &= \sum_{\mathbf{x}} l_{\mathbf{Y}}(\theta; \mathbf{y})\ p_{\mathbf{X}|\mathbf{Y}}(\mathbf{x}|\mathbf{y}; \theta_\nu) \\ &= \sum_{\mathbf{x}} l_{(\mathbf{X},\mathbf{Y})}(\theta; \mathbf{x}, \mathbf{y})\ p_{\mathbf{X}|\mathbf{Y}}(\mathbf{x}|\mathbf{y}; \theta_\nu) - \sum_{\mathbf{x}} l_{\mathbf{X}|\mathbf{Y}}(\theta; \mathbf{x}|\mathbf{y})\ p_{\mathbf{X}|\mathbf{Y}}(\mathbf{x}|\mathbf{y}; \theta_\nu) \\ &\stackrel{\text{def}}{=} Q(\theta|\mathbf{y}; \theta_\nu) - H(\theta|\mathbf{y}; \theta_\nu). \end{aligned}$$

Proposition 4.37. *If*

$$Q(\theta_{\nu+1}|\mathbf{y}; \theta_\nu) \geq Q(\theta_\nu|\mathbf{y}; \theta_\nu),$$

then

$$l_{\mathbf{Y}}(\theta_{\nu+1}; \mathbf{y}) \geq l_{\mathbf{Y}}(\theta_\nu; \mathbf{y}).$$

Proof. For any parameter $\theta_{\nu+1}$,

$$l_{\mathbf{Y}}(\theta_{\nu+1}; \mathbf{y}) - l_{\mathbf{Y}}(\theta_\nu; \mathbf{y}) = [Q(\theta_{\nu+1}|\mathbf{y}; \theta_\nu) - Q(\theta_\nu|\mathbf{y}; \theta_\nu)] + [H(\theta_\nu|\mathbf{y}; \theta_\nu) - H(\theta_{\nu+1}|\mathbf{y}; \theta_\nu)].$$

It suffices to show that the second bracketed term on the right-hand side is nonnegative. Note that

$$\begin{aligned} H(\theta_\nu|\mathbf{y}; \theta_\nu) - H(\theta_{\nu+1}|\mathbf{y}; \theta_\nu) &= -\sum_{\mathbf{x}} [l_{\mathbf{X}|\mathbf{Y}}(\theta_{\nu+1}; \mathbf{x}|\mathbf{y}) - l_{\mathbf{X}|\mathbf{Y}}(\theta_\nu; \mathbf{x}|\mathbf{y})]\ p_{\mathbf{X}|\mathbf{Y}}(\mathbf{x}|\mathbf{y}; \theta_\nu) \\ &= -\sum_{\mathbf{x}} \log\left(\frac{p_{\mathbf{X}|\mathbf{Y}}(\mathbf{x}|\mathbf{y}; \theta_{\nu+1})}{p_{\mathbf{X}|\mathbf{Y}}(\mathbf{x}|\mathbf{y}; \theta_\nu)}\right)\ p_{\mathbf{X}|\mathbf{Y}}(\mathbf{x}|\mathbf{y}; \theta_\nu) \\ &\geq -\log \sum_{\mathbf{x}} p_{\mathbf{X}|\mathbf{Y}}(\mathbf{x}|\mathbf{y}; \theta_\nu) \\ &= 0. \end{aligned}$$

The last inequality follows from the convexity of the negative log (see Exercise 4.18), while the equality that follows it comes from (4.47)–(4.48). □

This proposition motivates the following iterative procedure.

Algorithm 4.5.1. The EM Algorithm.
To maximize the log likelihood function $l_{\mathbf{Y}}(\theta; \mathbf{y})$, given a realization $\mathbf{y}$ of a random vector $\mathbf{Y}$ and an initial guess θ_0 for the parameter θ,

for $\nu = 0, 1, \ldots,$ repeat,

1. Compute $Q(\theta|\mathbf{y}; \theta_\nu)$, the conditional expectation of the log likelihood function for the complete data, given the observed $\mathbf{y}$ and the MLE approximation θ_ν. In the discrete case, this takes the form

$$Q(\theta|\mathbf{y}; \theta_\nu) = \sum_{\mathbf{x}} l_{(\mathbf{X},\mathbf{Y})}(\theta; \mathbf{x}, \mathbf{y})\ p_{\mathbf{X}|\mathbf{Y}}(\mathbf{x}|\mathbf{y}; \theta_\nu). \tag{4.49}$$

2. Compute a maximizer $\theta_{\nu+1}$ of $Q(\theta|\mathbf{y}; \theta_\nu)$.

Step 1 is called the E-step, and step 2 is called the M-step. That the sequence $\{\theta_\nu\}$ actually converges to a maximum likelihood estimator can be confirmed under fairly mild conditions. See [83] and the references therein.

4.5.1 An Illustrative Example

The following development is taken from Vardi and Lee [111]. In applications like nonnegative image reconstruction (see Chapter 9) it is important to find a nonnegative solution $\mathbf{f}$ to a linear system

$$K\mathbf{f} = \mathbf{g},$$

where the $m \times n$ coefficient matrix K and the vector $\mathbf{g} \in \mathbb{R}^n$ have nonnegative components. Of course such a solution need not be attained, so we seek to minimize some measure of the discrepancy between the data $\mathbf{g}$ and the model $K\mathbf{f}$. Here we minimize the Kullback–Leibler information divergence (see (2.52)):

$$\rho_{KL}(\mathbf{g}, K\mathbf{f}) = \sum_{i=1}^{m} g_i(\log g_i - \log[K\mathbf{f}]_i), \tag{4.50}$$

subject to

$$f_j \geq 0, \qquad j = 1, \ldots, n. \tag{4.51}$$

By a suitable rescaling (see Exercise 4.19) one may assume

$$\sum_{j=1}^{n} f_j = 1, \tag{4.52}$$

together with the following conditions on the entries of K:

$$k_{ij} \geq 0, \qquad i = 1, \ldots, m, \quad j = 1, \ldots, n, \tag{4.53}$$

$$\sum_{i=1}^{m} k_{ij} = 1, \qquad j = 1, \ldots, n. \tag{4.54}$$

Then we have

$$g_i \geq 0, \qquad i = 1, \ldots, m, \tag{4.55}$$

$$\sum_{i=1}^{m} g_i = 1, \tag{4.56}$$

$$[K\mathbf{f}]_i \geq 0, \tag{4.57}$$

$$\sum_{i=1}^{m} [K\mathbf{f}]_i = 1. \tag{4.58}$$

These conditions guarantee that $(\mathbf{g}, K\mathbf{f})$ lies in the domain of ρ_{KL}.

Minimizing (4.50) under the above conditions is equivalent to maximizing

$$J(\mathbf{f}) = \sum_{i=1}^{m} g_i \log[Kf]_i \tag{4.59}$$

subject to the constraints (4.51) and (4.52). To apply the EM algorithm, we first construct random variables X and Y, with support $\{1, \ldots, n\}$ and $\{1, \ldots, m\}$, respectively, and with joint probability mass function

$$P\{X = j, Y = i\} = p_{(X,Y)}(j, i; \mathbf{f}) = k_{ij} f_j. \tag{4.60}$$

Here $\mathbf{f} \in \mathbb{R}^n$ is the parameter to be estimated. The conditions (4.51)–(4.54) guarantee that $p_{(X,Y;)}(j, i|\mathbf{f})$ is indeed a probability mass function. See Exercise 4.20. The marginal probability mass function for Y is then

$$p_Y(i; \mathbf{f}) = \sum_{j=1}^{n} p_{(X,Y)}(j, i; \mathbf{f}) = \sum_{j=1}^{n} k_{ij} f_j = [K\mathbf{f}]_i. \tag{4.61}$$

Suppose we are given observed data $\mathbf{g} \in \mathbb{R}^m$ satisfying (4.55)–(4.56). If each g_i is a rational number, there exists a positive integer r such that

$$N_i = r g_i \tag{4.62}$$

is an integer for each i. The assumption that each g_i is rational can be relaxed [111]. Now take r independent, identically distributed copies of Y to obtain a random vector $\mathbf{Y}$, and let

$\mathbf{y} = (y_1, \ldots, y_r)$ be a realization for which N_i gives the number of indices k with $y_k = i$. Then the log likelihood function for Y, given data $\mathbf{y}$ and parameter $\mathbf{f}$, is

$$\begin{aligned} l_Y(\mathbf{f}; \mathbf{y}) &= \sum_{k=1}^{r} \log p_Y(y_k; \mathbf{f}) \\ &= \sum_{k=1}^{r} \left(\sum_{i=1}^{m} \delta_{y_k,i} \right) \log p_Y(y_k; \mathbf{f}) \\ &= \sum_{i=1}^{m} \sum_{k=1}^{r} \delta_{y_k,i} \log p_Y(y_k; \mathbf{f}) \\ &= \sum_{i=1}^{m} N_i \log p_Y(y_i; \mathbf{f}) \\ &= r \sum_{i=1}^{m} g_i \log([K\mathbf{f}]_i) \qquad \text{by (4.61) and (4.62).} \end{aligned}$$

This establishes the connection between maximum likelihood estimation and nonnegatively constrained linear equations; see (4.59).

In a similar manner, we can construct r copies of X to obtain a random vector $\mathbf{X}$ for which the pairs (X_k, Y_k) are independent and distributed according to (4.60). Here $\mathbf{X}$ is the hidden data vector, and $\mathbf{X}$ and $\mathbf{Y}$ together make up the complete data. Take a realization $(\mathbf{x}, \mathbf{y})$ with pairs (x_k, y_k), $k = 1, \ldots, r$, for which $N_{ij}(\mathbf{y}, \mathbf{x})$ denotes the number of indices k such that $y_k = i$ and $x_k = j$. Note that for each i,

$$\sum_{j=1}^{n} N_{ij}(\mathbf{x}, \mathbf{y}) = N_i = r g_i, \tag{4.63}$$

the number of occurrences of $y_k = i$. The log likelihood function for the complete data is given by

$$l_{\mathbf{X},\mathbf{Y}}(\mathbf{f}; \mathbf{x}, \mathbf{y}) = \sum_{i=1}^{m} \sum_{i=1}^{m} N_{ij}(\mathbf{x}, \mathbf{y}) \, [\log k_{ij} + \log f_j].$$

From (4.47)–(4.48) and (4.60),

$$p_{X|Y}(j|i; \mathbf{f}_\nu) = \frac{k_{ij} f_j^\nu}{\sum_{l=1}^{n} k_{il} f_l^\nu} \stackrel{\text{def}}{=} \hat{p}_{ij}^\nu, \tag{4.64}$$

where f_j^ν denotes the jth component of $\mathbf{f}_\nu$. Then by (4.49),

$$\begin{aligned} Q(\mathbf{f}|\mathbf{y}; \mathbf{f}_\nu) &= \sum_{\mathbf{x}} \left(\sum_{i=1}^{m} \sum_{j=1}^{n} N_{ij}(\mathbf{x}, \mathbf{y}) \, [\log k_{ij} + \log f_j^\nu] \right) \hat{p}_{ij}^\nu \\ &= \sum_{i=1}^{m} \sum_{j=1}^{n} r g_i \, [\log k_{ij} + \log f_j^\nu] \, \hat{p}_{ij}^\nu. \end{aligned} \tag{4.65}$$

The second equality follows from (4.63). This completes the E-step of the algorithm.

To implement the M-step, we maximize Q with respect to $\mathbf{f}$ subject to the constraints (4.51)–(4.52). This yields (see Exercise 4.21) the vector $\mathbf{f}_{v+1}$ with components

$$f_j^{v+1} = f_j^v \sum_{i=1}^{m} k_{ij} \left(\frac{g_i}{\sum_{l=1}^{n} k_{il} f_l^v} \right), \qquad j = 1, \ldots, n. \tag{4.66}$$

Exercises

4.1. Show that the covariance matrix C (see (4.3)) is symmetric and positive semidefinite. Provide an example showing that C need not be positive definite.

4.2. Show that if a random vector has independent components and the covariance matrix C exists, then C is diagonal.

4.3. Suppose that for each $i = 1, \ldots, n$, d_i is a realization of a Gaussian random variable X_i having mean μ and variance $\sigma^2 > 0$. Show that if the X_i's are independent, then the maximum likelihood estimator for μ is $\sum_{i=1}^{n} d_i/n$ and the maximum likelihood estimator for σ^2 is $\sum_{i=1}^{n}(d_i - \mu)^2/n$.

4.4. For the Poisson random vector in Example 4.14, show that $E(\mathbf{X}) = \boldsymbol{\lambda}$ and $\operatorname{cov}(\mathbf{X}) = \operatorname{diag}(\lambda_i, \ldots, \lambda_n)$.

4.5. Show that the negative Poisson log likelihood function in (4.8) is strictly convex on the interior of the nonnegative orthant $\mathbb{R}^n_+$ and that it has $\mathbf{d}$ as its unique minimizer.

4.6. Construct tables analogous to those in Example 4.21 for the toss of three independent, fair coins.

4.7. Verify equation (4.11) under the assumption that X and Y are independent, jointly distributed, discrete random variables.

4.8. Prove Theorem 4.24.

4.9. Show that the expression (4.21) gives the MAP estimator in Example 4.26.

4.10. Show that under the assumptions of Example 4.26, the right-hand side of equation (4.21) gives the conditional expectation, $E(\mathbf{X}|\mathbf{Z} = \mathbf{z})$. This can be most easily done using characteristic functions. These are essentially the expected values of the Fourier transforms of random variables.

4.11. From (4.21), derive expression (4.22) with $\alpha = (\sigma_{\mathbf{N}}/\sigma_{\mathbf{X}})^2$.

4.12. Prove Proposition 4.29.

4.13. If A is symmetric, it has an orthogonal eigendecomposition $A = U \operatorname{diag}(\lambda_i) U^T$ with $U^T U = U U^T = I$. Use this fact, along with $a_{ii} = \mathbf{e}_i^T A \mathbf{e}_i$, to prove the equality (4.28).

4.14. Verify equation (4.38).

4.15. Prove Proposition 4.35.

4.16. Verify equations (4.43)–(4.45).

4.17. Show that $\Gamma_{\mathbf{XX}} K^T [K \Gamma_{\mathbf{XX}} K^T + C_{\mathbf{N}}]^{-1} = [K^T C_{\mathbf{N}}^{-1} K + \Gamma_{\mathbf{XX}}^{-1}]^{-1} K^T C_{\mathbf{N}}^{-1}$.

4.18. Show that if J is convex, $\sum_i w_i = 1$, and $w_i \geq 0$, then

$$J\left(\sum_i z_i w_i \right) \leq \sum_i J(z_i)\, w_i.$$

Show that the inequality is strict if the z_i's are not all equal and each $w_i > 0$.

4.19. Show that by replacing $[K]_{ij} = k_{ij}$ by $k_{ij}/\sum_i k_{ij}$, one can obtain a matrix S for which $\sum_j f_j = \sum_i [K\mathbf{f}]_i$.

4.20. Show that conditions (4.51)–(4.54) guarantee that equation (4.60) gives a probability mass function.

4.21. Show that equation (4.66) gives a maximizer for Q in (4.65) subject to the equality constraint (4.52). *Hint*: Show that

$$\frac{\partial}{\partial f_l}\left[Q - \lambda\left(\sum_{j=1}^{n} f_j - 1\right)\right] = 0$$

and that this implies $f_j = r/\lambda \sum_i \hat{p}_{ij}^k g_i$. Then use (4.52) to obtain $\lambda = r$. Verify also that the inequality constraint (4.51) holds provided each $f_j^k \geq 0$.

Chapter 5
Image Deblurring

In this chapter we consider a two-dimensional analogue of (1.1):

$$g(x, y) = \int_0^1 \int_0^1 k(x - x', y - y')\, f(x', y')\, dx'\, dy'. \tag{5.1}$$

In image reconstruction, the estimation of f from observations of g is referred to as the two-dimensional image deblurring problem [64]. An imaging application is presented in section 5.1. Since the integral operator has convolution form, (5.1) is also known as the two-dimensional deconvolution problem.

The kernel function k in (5.1) is typically smooth, so from Example 2.13, the corresponding integral operator is compact. Hence by Theorem 2.14, the deblurring problem is ill-posed and some form of regularization must be applied to accurately reconstruct f from noisy data. Numerical implementation is complicated by the fact that the discrete systems arising from equation (5.1) may have a very large number of unknowns. This complication can be alleviated by taking advantage of convolution structure. In particular, certain discretizations of (5.1) give rise to linear systems with Toeplitz block structure. Such systems can be efficiently solved using conjugate gradient iteration with preconditioners based on the fast Fourier transform (FFT). Computational methods are described in detail in sections 5.2 and 5.3.

5.1 A Mathematical Model for Image Blurring

A very general model for the blurring of images [7] is

$$g(x, y) = \int\int_{\mathbb{R}^2} k(x, x', y, y')\, f(x', y')\, dx'\, dy'. \tag{5.2}$$

In optics, f is called the light source, or object. The kernel function k is known as the point spread function (PSF), and g is called the (blurred) continuous image. Equation (5.2) can be used to model the diffraction of light from the source as it propagates through a medium like the atmosphere [99]. It can also model distortion due to imperfections in optical devices like telescopes and microscopes. See [7] for other applications.

The continuous image g represents an energy density, with units of energy per unit area or, equivalently, number of photons per unit area. The image is often recorded with a device known as a CCD camera. This consists of an array of disjoint rectangles called pixels, Ω_{ij},

$0 \le i \le n_x - 1, 0 \le j \le n_y - 1$, onto which the photons fall and are counted. The energy falling on an individual array element is then given by

$$g_{ij} = \int\int_{\Omega_{ij}} g(x, y)\, dx\, dy. \tag{5.3}$$

A stochastic model for the data recorded by the ijth pixel of a CCD array is given in the notation of Chapter 4 by

$$D_{ij} \sim \text{Poisson}(g_{ij}) + \text{Normal}(0, \sigma^2). \tag{5.4}$$

The Poisson component models the photon count, while the additive Gaussian term accounts for background noise in the recording electronics [99]. We denote a realization of the random variable D_{ij} by d_{ij}. The $n_x \times n_y$ array d, whose components are the d_{ij}'s, is called the (noisy, blurred) discrete image. For each index pair (i, j), d_{ij} is a realization of a Gaussian random variable with zero mean and variance σ^2 added to a realization of a Poisson random variable with mean and variance g_{ij}; see Examples 4.13 and 4.14. These random variables are assumed to be independent of each other and independent of the random variables corresponding to the other pixels.

A fully discrete model may be obtained by truncating the region of integration in (5.2) to be the union of the Ω_{ij}'s and then applying midpoint quadrature to both (5.2) and (5.3). Assume that each Ω_{ij} has area $\Delta x \times \Delta y$, and let (x_i, y_j) denote the midpoint. Then

$$g_{ij} = \sum_{\mu=0}^{n_x-1} \sum_{\nu=0}^{n_y-1} k(x_i, x_\mu, y_j, y_\nu)\, f(x_\mu, y_\nu)\, \Delta x\, \Delta y + \epsilon_{ij}^{\text{quad}}, \tag{5.5}$$

where $\epsilon_{ij}^{\text{quad}}$ denotes quadrature error. Combining (5.4) and (5.5),

$$d_{ij} = \sum_{\mu=0}^{n_x-1} \sum_{\nu=0}^{n_y-1} t_{i,\mu,j,\nu} f_{\mu,\nu} + \eta_{ij}, \tag{5.6}$$

where now $f_{\mu,\nu} = f(x_\mu, y_\nu)$, the term η_{ij} incorporates the various stochastic error realizations and quadrature errors, and $t_{i,\mu,j,\nu} = k(x_i, x_\mu, y_j, y_\nu)\, \Delta x\, \Delta y$. We refer to the array t as the discrete PSF.

The blurring process is sometimes assumed to be invariant under spatial translation. This means that the PSF can be represented as a function of two variables, rather than four variables,

$$k(x, x', y, y') = k(x - x', y - y'), \tag{5.7}$$

and equation (5.2) reduces to (5.1). This representation greatly simplifies computations. Since the integral in (5.1) then has convolution form, given the source $f = f(x, y)$ and the PSF $k = k(x, y)$, one can in principle compute the continuous image g using the convolution theorem,

$$g = \mathcal{F}^{-1}\{\mathcal{F}\{k\}\, \mathcal{F}\{f\}\}. \tag{5.8}$$

Here the continuous Fourier transform of a (possibly complex-valued) function f defined on $\mathbb{R}^d$ ($d = 2$ for two-dimensional imaging) is given by

$$\mathcal{F}\{f\}(\boldsymbol{\omega}) = \int_{\mathbb{R}^d} f(\mathbf{x})\, e^{-\hat{\imath} 2\pi \mathbf{x}^T \boldsymbol{\omega}}\, d\mathbf{x}, \qquad \boldsymbol{\omega} \in \mathbb{R}^d, \tag{5.9}$$

with $\hat{\imath} \stackrel{\text{def}}{=} \sqrt{-1}$. The inverse continuous Fourier transform is given by

$$\mathcal{F}^{-1}\{g\}(\mathbf{x}) = \int_{\mathbb{R}^d} g(\boldsymbol{\omega})\, e^{\hat{\imath} 2\pi \mathbf{x}^T \boldsymbol{\omega}}\, d\omega, \qquad \mathbf{x} \in \mathbb{R}^d. \tag{5.10}$$

One can formally derive from (5.8) the Fourier inversion formula,

$$f = \mathcal{F}^{-1}\left\{\frac{\mathcal{F}\{g\}}{\mathcal{F}\{k\}}\right\}. \tag{5.11}$$

If $\mathcal{F}\{k\}$ takes on zero values, this formula is not valid. If it takes on small nonzero values, this reconstructed f is unstable with respect to perturbations in the data g. These situations correspond to violations of conditions (ii) and (iii) in Definition 2.7 of well-posedness.

Discrete computations are also greatly simplified by the representation (5.7). In equation (5.6), the discrete PSF can be represented as a two-dimensional array, rather than as a four-dimensional array,

$$t_{i,\mu,j,\nu} = t_{i-\mu,j-\nu}, \qquad 0 \le i, \mu \le n_x - 1,\ 0 \le j, \nu \le n_y - 1. \tag{5.12}$$

Equation (5.6) then reduces to

$$d_{ij} = \sum_{\mu=0}^{n_x-1} \sum_{\nu=0}^{n_y-1} t_{i-\mu,j-\nu} f_{\mu,\nu} + \eta_{ij} \tag{5.13}$$

with

$$t_{ij} = k(i\Delta x, j\Delta y)\ \Delta x \Delta y. \tag{5.14}$$

The discrete convolution product in (5.13) defines a linear operator. A discrete analogue of the continuous Fourier transform can be used to efficiently compute regularized solutions. Details are given in section 5.2.

5.1.1 A Two-Dimensional Test Problem

The two-dimensional test problem arises in atmospheric optics, an application described in detail in [99]. The data, a simulated image of a satellite in earth orbit viewed with a ground-based telescope, was generated according to the model (5.13). In this model, the continuous PSF takes the form

$$k = |\mathcal{F}^{-1}\{Ae^{\hat{\imath}\phi}\}|^2. \tag{5.15}$$

Here A is called the aperture function and ϕ is the phase. The phase represents distortions in a planar wavefront, emanating from a point source at infinity, due to propagation through an optically thin layer of material (in this case, the atmosphere) with a variable index of refraction. The aperture function represents the region in the plane over which light is collected. For a large reflecting telescope, the aperture function is typically the indicator function for an annulus. Both the phase and the support of the aperture function can be seen in the top plot in Figure 5.1. The middle plot shows the corresponding PSF. Figure 5.2 shows a simulated light source, along with the corresponding blurred, noisy image. This was obtained by convolving the PSF with the source and then adding noise to the resulting blurred image.

The bottom plot in Figure 5.1 shows the power spectrum, $|\mathcal{F}\{k\}|^2$, of the PSF. Several phenomena can be seen from the power spectrum. First, its support (i.e., the region in

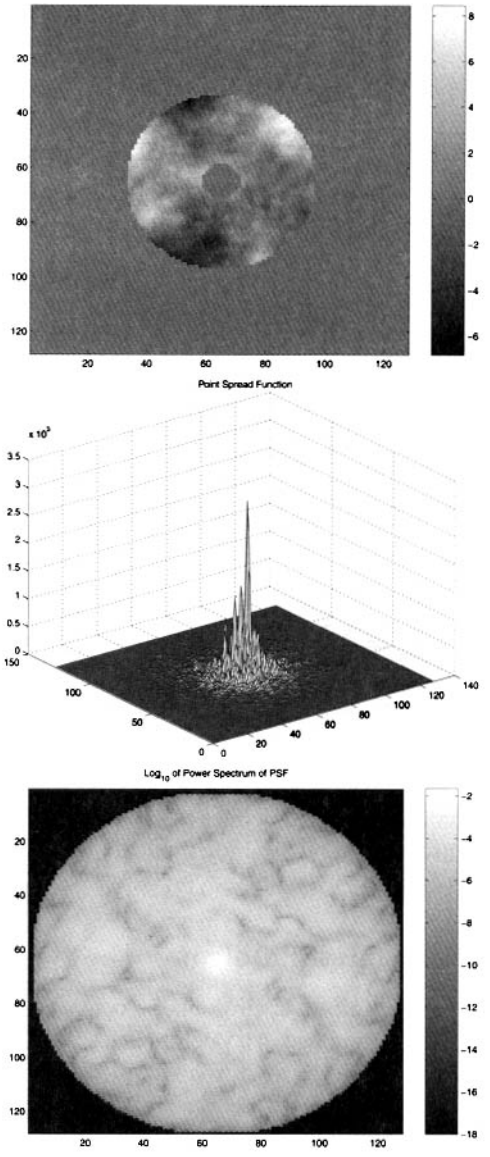


Figure 5.1. *Atmospheric image blurring operator. At the top is a gray-scale plot of the component-wise product $A\phi$ of the aperture function A and the phase ϕ. The middle plot shows the corresponding PSF, $k = |\mathcal{F}^{-1}\{Ae^{i\phi}\}|^2$. The bottom plot shows a logarithmically scaled gray-scale plot of the power spectrum of the PSF.*

which it is nonzero) is a disk. For this reason, the PSF is called band limited, or diffraction limited. See Exercise 5.3 for insight. The exterior of this disk corresponds to the null space of the convolution integral operator in (5.1), and the image deblurring problem violates the uniqueness condition (ii) of well-posedness in Definition 2.7. Functions in the null space of this operator are highly oscillatory. Thus high frequency information about the source

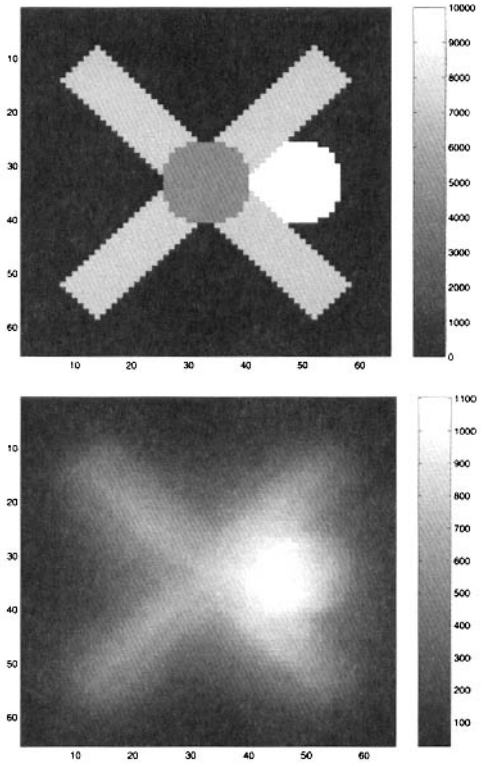


Figure 5.2. *Atmospheric image data. At the top is a gray-scale plot of the source or object. The bottom plot shows a gray-scale plot of the blurred, noisy image.*

(true image) is not present in the image data. Second, near the edge of the disk the power spectrum takes on very small, but nonzero, values. This implies that the stability condition (iii) of Definition 2.7 does not hold.

5.2 Computational Methods for Toeplitz Systems

Linear systems with block Toeplitz structure arise from the discrete convolution product in equation (5.13). We next discuss computational techniques to efficiently solve such systems. The symbol $\mathbb{C}^n$ denotes the set of vectors with n complex components. Here the indices of the components of a generic vector $\mathbf{f} \in \mathbb{C}^n$ will vary from 0 through $n-1$, i.e., $\mathbf{f} = (f_0, \ldots, f_{n-1})$. $\mathbb{C}^n$ is a Hilbert space under the Euclidean inner product and induced norm:

$$\langle \mathbf{f}, \mathbf{g} \rangle = \sum_{j=0}^{n-1} f_j \, \overline{g}_j, \qquad \|\mathbf{f}\| = \sqrt{\sum_{j=0}^{n-1} |f_j|^2}. \tag{5.16}$$

Given $z = \alpha + \hat{\imath}\beta \in \mathbb{C}$, $\overline{z} = \alpha - \hat{\imath}\beta$ denotes the complex conjugate, and $|z| = \sqrt{\alpha^2 + \beta^2}$ denotes magnitude. We denote the set of complex-valued $n_x \times n_y$ arrays by $\mathbb{C}^{n_x \times n_y}$. A generic array $f \in \mathbb{C}^{n_x \times n_y}$ will be indexed by f_{ij}, $i = 0, \ldots, n_x - 1$, $j = 0, \ldots, n_y - 1$.

5.2.1 Discrete Fourier Transform and Convolution

What follows is a discrete analogue of the continuous Fourier transform (5.9).

Definition 5.1. The discrete Fourier transform (DFT) is a mapping on $\mathbb{C}^n$ given by

$$[\mathcal{F}\{\mathbf{f}\}]_i = \frac{1}{\sqrt{n}} \sum_{j=0}^{n-1} f_j \, e^{-\hat{\imath} 2\pi ij/n}, \qquad i = 0, 1, \ldots, n-1. \tag{5.17}$$

As in (5.9), $\hat{\imath} = \sqrt{-1}$. We can express the DFT as a matrix-vector product, $\mathcal{F}\{\mathbf{f}\} = F\mathbf{f}$, where $F \in \mathbb{C}^{n \times n}$ is the Fourier matrix. This has components

$$[F]_{ij} = \frac{e^{-\hat{\imath} 2\pi ij/n}}{\sqrt{n}}, \qquad 0 \le i, j \le n-1. \tag{5.18}$$

The inverse DFT is given by

$$\begin{aligned}[\mathcal{F}^{-1}\{\mathbf{g}\}]_i &= \frac{1}{\sqrt{n}} \sum_{j=0}^{n-1} g_j \, e^{\hat{\imath} 2\pi ij/n} \\ &= [F^* \mathbf{g}]_i, \qquad i = 0, \ldots, n-1.\end{aligned} \tag{5.19}$$

This follows from the fact that the Fourier matrix F is a unitary matrix (see Exercise 5.5), i.e., $F^*F = I$. Here the superscript $*$ denotes matrix conjugate transpose.

Remark 5.2. Our definitions of the DFT and its inverse are somewhat nonstandard. Typically the DFT is defined without the factor of $1/\sqrt{n}$ in (5.17), and the inverse DFT is defined with a factor of $1/n$ rather than $1/\sqrt{n}$ in (5.19). This coincides with the functions **fft**$(\cdot)$ and **ifft**$(\cdot)$, which are given in section 5.2.2. We selected our definitions so that both the DFT and its inverse preserve Euclidean inner products and norms (see Exercise 5.6).

Definition 5.3. The discrete convolution product of vectors $\mathbf{t} = (t_{1-n}, \ldots, t_0, t_1, \ldots, t_{n-1})$ and $\mathbf{f} \in \mathbb{C}^n$ is given by

$$[\mathbf{t} \star \mathbf{f}]_i = \sum_{j=0}^{n-1} t_{i-j} f_j, \qquad i = 0, \ldots, n-1. \tag{5.20}$$

Definition 5.4. A discrete vector $\mathbf{t}$ is called n-periodic if

$$t_i = t_j \qquad \text{whenever } i = j \bmod n. \tag{5.21}$$

Given $\mathbf{t} = (t_0, t_1, \ldots, t_{n-1}) \in \mathbb{C}^n$, by the periodic extension of $\mathbf{t}$ of size $2n-1$, we mean the n-periodic vector $\mathbf{t}^{ext} = (t^{ext}_{1-n}, \ldots, t^{ext}_0, t^{ext}_1, \ldots, t^{ext}_{n-1})$ for which $t^{ext}_i = t_i$ for $i = 0, \ldots, n-1$.

Definition 5.5. We adopt the following notation for component-wise multiplication and component-wise division of vectors. For $\mathbf{f}, \mathbf{g} \in \mathbb{C}^n$,

$$[\mathbf{f}. * \mathbf{g}]_i = f_i g_i, \qquad [\mathbf{f}./\mathbf{g}]_i = f_i / g_i, \ g_i \neq 0. \tag{5.22}$$

This notation extends to two-dimensional arrays in an obvious manner.

The next proposition relates the discrete Fourier transform to the discrete convolution product.

Proposition 5.6. *If* $\mathbf{t}, \mathbf{f} \in \mathbb{C}^n$ *and* $\mathbf{t}^{ext}$ *is the periodic extension of* $\mathbf{t}$ *of size* $2n-1$, *then*

$$\frac{1}{\sqrt{n}}\,\mathbf{t}^{ext} \star \mathbf{f} = \mathcal{F}^{-1}\{\mathcal{F}\{\mathbf{t}\}. * \mathcal{F}\{\mathbf{f}\}\}. \tag{5.23}$$

Proof. Set $w = \exp(-\hat{\imath}2\pi/n)$. Then by (5.18), $\sqrt{n}[F]_{ij} = w^{ij}$. Consequently,

$$\begin{aligned}
\sqrt{n}\,[\mathcal{F}\{\mathbf{t}_{ext} \star \mathbf{f}\}]_k &= \sum_{i=0}^{n-1}\left(\sum_{j=0}^{n-1} t_{i-j}^{ext} f_j\right) w^{ik} \\
&= \sum_{j=0}^{n-1} f_j \left(\sum_{i=0}^{n-1} t_{i-j}^{ext} w^{ik}\right) \\
&= \sum_{j=0}^{n-1} f_j \left(\sum_{\ell=-j}^{n-1-j} t_{\ell}^{ext} w^{(\ell+j)k}\right) \\
&= \sum_{j=0}^{n-1} f_j w^{jk} \sum_{\ell=-j}^{n-1-j} t_{\ell}^{ext} w^{\ell k} \\
&= \sqrt{n}\mathcal{F}\{\mathbf{f}\}\ \sqrt{n}\mathcal{F}\{\mathbf{t}\}.
\end{aligned}$$

The last equality follows from the n-periodicity of both $\mathbf{t}^{ext}$ and $w^{\cdot\cdot k}$. □

Proposition 5.6 can be extended to compute two-dimensional discrete convolution products.

Definition 5.7. The two-dimensional DFT is the mapping on $\mathbb{C}^{n_x \times n_y}$ given by

$$[\mathcal{F}\{f\}]_{ij} = \frac{1}{\sqrt{n_x n_y}} \sum_{i'=0}^{n_x-1}\sum_{j'=0}^{n_y-1} f_{i',j'} e^{-\hat{\imath}2\pi\,(ii'/n_x + jj'/n_y)}, \tag{5.24}$$

$0 \le i \le n_x - 1$, $0 \le j \le n_y - 1$. The inverse two-dimensional DFT is obtained by replacing $-\hat{\imath}$ by $\hat{\imath}$ in equation (5.24).

Definition 5.8. The (two-dimensional) discrete convolution product of an array t, having components t_{ij}, $1 - n_x \le i \le n_x - 1$, $1 - n_y \le j \le n_y - 1$, with an array $f \in \mathbb{C}^{n_x \times n_y}$, is given by

$$[t \star f]_{ij} = \sum_{i'=0}^{n_x-1}\sum_{j'=0}^{n_y-1} t_{i-i',j-j'} f_{i',j'}, \qquad 0 \le i \le n_x - 1,\ 0 \le j \le n_y - 1. \tag{5.25}$$

Definition 5.9. A two-dimensional array t is called (n_x, n_y)-periodic if

$$\begin{aligned}
t_{i,j} &= t_{i',j} \qquad \text{whenever } i = i' \bmod n_x, \\
t_{i,j} &= t_{i,j'} \qquad \text{whenever } j = j' \bmod n_y.
\end{aligned}$$

Let $t \in \mathbb{C}^{n_x \times n_y}$. By the periodic extension of t of size $(2n_x - 1) \times (2n_y - 1)$, we mean the (n_x, n_y)-periodic array t^{ext}, with components t^{ext}_{ij}, $1-n_x \le i \le n_x - 1$, $1-n_y \le j \le n_y - 1$, for which $t^{ext}_{ij} = t_{ij}$ whenever $0 \le i \le n_x - 1$, $0 \le j \le n_y - 1$.

Proposition 5.10. *If t, $f \in \mathbb{C}^{n_x \times n_y}$ and t^{ext} is the periodic extension of t of size $(2n_x - 1) \times (2n_y - 1)$, then*

$$\frac{1}{\sqrt{n_x n_y}}\, t^{ext} \star f = \mathcal{F}^{-1}\{\mathcal{F}\{t\} . * \mathcal{F}\{f\}\}. \tag{5.26}$$

5.2.2 The FFT Algorithm

If the one-dimensional DFT (5.17) were implemented using conventional matrix-vector multiplication, then its computational cost would be $\mathcal{O}(n^2)$, where n is the length of the vector being transformed. The FFT algorithm reduces this computational cost to $\mathcal{O}(n \log n)$. First discovered by Cooley and Tukey [26], this algorithm is used in a broad range of applications in addition to image processing, ranging from time series analysis to the numerical solution of differential equations.

To derive the FFT algorithm, first define $\tilde{F}_n = \sqrt{n}F$, where F is the $n \times n$ Fourier matrix (see (5.18)). The components of $\tilde{F}_n$ are w_n^{ij} with

$$w_n = e^{-i2\pi/n}. \tag{5.27}$$

In the computations to follow, we assume that n is an even integer, and we set $m = n/2$ and $w_m = w_n^2 = \exp(-i2\pi/m)$.

Given any $\mathbf{f} = (f_0, f_1, \ldots, f_{n-1}) \in \mathbb{C}^n$, for $i = 0, 1, \ldots, n-1$,

$$\begin{aligned}
[\tilde{F}_n \mathbf{f}]_i &= \sum_{\ell=0}^{m-1} w_n^{i(2\ell)} f_{2\ell} + \sum_{\ell=0}^{m-1} w_n^{i(2\ell+1)} f_{2\ell+1} \\
&= \sum_{\ell=0}^{m-1} w_m^{i\ell} f'_\ell + w_n^i \sum_{\ell=0}^{m-1} w_m^{i\ell} f''_\ell \\
&= [\tilde{F}_m \mathbf{f}']_i + w_n^i [\tilde{F}_m \mathbf{f}'']_i,
\end{aligned} \tag{5.28}$$

where $\mathbf{f}' = (f_0, f_2, \ldots, f_{n-2})$ and $\mathbf{f}'' = (f_1, f_3, \ldots, f_{n-1})$. Note that for $i = 0, 1, \ldots, m-1$,

$$[\tilde{F}_m \mathbf{f}']_{m+i} = \sum_{\ell=0}^{m-1} w_m^{m\ell} w_m^{i\ell} f'_\ell = [\tilde{F}_m \mathbf{f}']_i, \tag{5.29}$$

since $w_m^m = 1$. Similarly,

$$[\tilde{F}_m \mathbf{f}'']_{m+i} = [\tilde{F}_m \mathbf{f}'']_i. \tag{5.30}$$

In addition, since $w_n^m = \exp(-i\pi) = -1$,

$$w_n^{m+i} = -w_n^i. \tag{5.31}$$

Combining (5.28)–(5.31) and replacing m by $n/2$, we obtain

$$[\tilde{F}_n \mathbf{f}]_i = [\tilde{F}_{n/2} \mathbf{f}']_i + w_n^i [\tilde{F}_{n/2} \mathbf{f}'']_i, \qquad i = 0, 1, \ldots, n/2 - 1, \tag{5.32}$$

$$[\tilde{F}_n \mathbf{f}]_{n/2+i} = [\tilde{F}_{n/2} \mathbf{f}']_i - w_n^i [\tilde{F}_{n/2} \mathbf{f}'']_i, \qquad i = 0, 1, \ldots, n/2 - 1. \tag{5.33}$$

The recursion (5.32)–(5.33) forms the basis for the FFT algorithm (see [110] for implementation details). It reduces the computation of the transform of an n-vector to a pair of transforms of vectors of size $n/2$.

To analyze the computational cost, let FFT(n) represent the number of floating point multiplications required to evaluate $\tilde{F}_n\mathbf{f}$. In equations (5.32)–(5.33), we see that given the vectors $\tilde{F}_{n/2}\mathbf{f}'$ and $\tilde{F}_{n/2}\mathbf{f}''$, only $n/2$ multiplications are needed to evaluate $\tilde{F}_n\mathbf{f}$. Hence,

$$\text{FFT}(n) = n/2 + 2\,\text{FFT}(n/2). \tag{5.34}$$

Since FFT(1) = 0, if we assume that $n = 2^k$, then

$$\text{FFT}(n) = n/2 \times k = n/2 \times \log_2(n). \tag{5.35}$$

See Exercise 5.8.

Note that the inverse DFT (5.19) differs from the forward transform (5.17) only in the replacement of $w_n = \exp(-\hat{\imath}2\pi/n)$ by its complex conjugate, $\overline{w}_n = \exp(\hat{\imath}2\pi/n)$. Thus the algorithm and the cost for computing the inverse discrete Fourier transform are both essentially the same as for the forward transform.

In the material to follow, we indicate multiplication by the scaled Fourier matrix $\tilde{F}_n$ by $\mathbf{fft}(\cdot)$. Consequently, given $\mathbf{f} = (f_0, f_1, \ldots, f_{n-1}) \in \mathbb{C}^n$,

$$[\mathbf{fft}(\mathbf{f})]_i = \sqrt{n}\,[\mathcal{F}\{\mathbf{f}\}]_i = \sum_{j=0}^{n-1} f_j\, e^{-\hat{\imath}2\pi ij/n}, \qquad i = 0, 1, \ldots, n-1. \tag{5.36}$$

The inverse of **fft** is given by

$$[\mathbf{ifft}(\mathbf{f})]_i = \frac{1}{\sqrt{n}}\,[\mathcal{F}^{-1}\{\mathbf{f}\}]_i = \frac{1}{n}\sum_{j=0}^{n-1} f_j\, e^{\hat{\imath}2\pi ij/n}, \qquad i = 0, 1, \ldots, n-1. \tag{5.37}$$

Thus the discrete convolution result (5.23) can be expressed as

$$\mathbf{t}^{ext} \star \mathbf{f} = \mathbf{ifft}(\mathbf{fft}(\mathbf{t}). * \mathbf{fft}(\mathbf{f})). \tag{5.38}$$

Two-Dimensional FFTs

We now address the computation of two-dimensional DFTs (see Definition 5.7). Setting $e^{-\hat{\imath}2\pi\,(ii'/n_x + jj'/n_y)} = e^{-\hat{\imath}2\pi\, ii'/n_x} e^{-\hat{\imath}2\pi\, jj'/n_y} \stackrel{\text{def}}{=} w_{n_x}^{ii'} w_{n_y}^{jj'}$, we obtain from (5.24)

$$\sqrt{n_x n_y}\,[\mathcal{F}\{f\}]_{ij} = \sum_{j'=0}^{n_y-1} \left(\sum_{i'=0}^{n_x-1} f_{i'j'} w_{n_x}^{ii'} \right) w_{n_y}^{jj'}. \tag{5.39}$$

The quantity inside the parentheses can be expressed as $\mathbf{fft}(f_{\cdot,j'})$. By this, we mean that the one-dimensional scaled DFT (5.36) has been applied to each of the n_y columns of the array f. Denote the result by $\hat{f}_x$. Applying **fft** to each of the n_x rows of $\hat{f}_x$ then gives the result in (5.39). From this, we see that the number of multiplications required to evaluate (5.39) is given by

$$\text{FFT2}(n_x, n_y) = n_y \times \text{FFT}(n_x) + n_x \times \text{FFT}(n_y) = \frac{N}{2}\log_2(N), \tag{5.40}$$

where $N = n_x n_y$ is the number of components in the array. Evaluation of two-dimensional inverse DFTs using one-dimensional inverse DFTs can be carried out in the same manner, and the computational cost is the same.

In a manner analogous to (5.36)–(5.37), we define the two-dimensional scaled DFT and its inverse,

$$[\mathbf{fft2}(f)]_{ij} = \sqrt{n_x n_y}[\mathcal{F}\{f\}]_{ij} = \sum_{i'=0}^{n_x-1}\sum_{j'=0}^{n_y-1} f_{i'j'} \exp(-\hat{\imath}2\pi(ii'/n_x + jj'/n_y)),$$

$$[\mathbf{ifft2}(f)]_{ij} = \frac{[\mathcal{F}^{-1}\{f\}]_{ij}}{\sqrt{n_x n_y}} = \frac{1}{n_x n_y}\sum_{j'=0}^{n_y-1} f_{i'j'} \exp(\hat{\imath}2\pi(ii'/n_x + jj'/n_y)).$$

The two-dimensional discrete convolution result (5.26) can then be rewritten as

$$t^{ext} \star f = \mathbf{ifft2}(\mathbf{fft2}(t). * \mathbf{fft2}(f)). \tag{5.41}$$

We next examine matrix representations of discrete convolution operators.

5.2.3 Toeplitz and Circulant Matrices

Definition 5.11. A matrix is called Toeplitz if it is constant along diagonals. An $n \times n$ Toeplitz matrix T has the form

$$T = \begin{bmatrix} t_0 & t_{-1} & \cdots & t_{2-n} & t_{1-n} \\ t_1 & t_0 & t_{-1} & \ddots & t_{2-n} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ t_{n-2} & \ddots & t_1 & t_0 & t_{-1} \\ t_{n-1} & t_{n-2} & \cdots & t_1 & t_0 \end{bmatrix}. \tag{5.42}$$

For any vector $\mathbf{f} \in \mathbb{C}^n$, the matrix-vector product $T\mathbf{f}$ has discrete convolution form

$$[T\mathbf{f}]_i = \sum_{j=0}^{n-1} t_{i-j} f_j = [\mathbf{t} \star \mathbf{f}]_i, \qquad i = 0, \ldots, n-1,$$

where $\mathbf{t} = (t_{1-n}, \ldots, t_{-1}, t_0, t_1, \ldots, t_{n-1}) \in \mathbb{C}^{2n-1}$. We indicate this situation by $T = \mathbf{toeplitz}(\mathbf{t})$.

Definition 5.12. An $n \times n$ matrix C is called circulant if it is Toeplitz and its rows are circular right shifts of the elements of the preceding row. In this case we can write

$$C = \begin{bmatrix} c_0 & c_{n-1} & \cdots & c_2 & c_1 \\ c_1 & c_0 & c_{n-1} & \ldots & c_2 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ c_{n-2} & \ddots & c_1 & c_0 & c_{n-1} \\ c_{n-1} & c_{n-2} & \cdots & c_1 & c_0 \end{bmatrix}. \tag{5.43}$$

Then $C = \mathbf{toeplitz}(\mathbf{c}^{ext})$, where $\mathbf{c}^{ext} = (c_1, \ldots, c_{n-1}, c_0, c_1 \ldots, c_{n-1})$ is the n-periodic extension of size $2n-1$ of $\mathbf{c} = (c_0, c_1, \ldots, c_{n-1})$. We indicate this situation by $C = \mathbf{circulant}(\mathbf{c})$. Note that $\mathbf{c} = C_{\cdot,1}$, the first column of C.

For a detailed discussion of circulant matrices and their properties, see [29]. We next establish the relationship between circulant matrices and the discrete Fourier transform.

Definition 5.13. The *circulant right shift matrix* is given by

$$R = \begin{bmatrix} 0 & 0 & \cdots & 0 & 1 \\ 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & 0 & \cdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 & 0 \end{bmatrix}. \tag{5.44}$$

It has the property that $R(x_0, x_1, \ldots, x_{n-2}, x_{n-1}) = (x_{n-1}, x_0, \ldots, x_{n-3}, x_{n-2})$.

Proposition 5.14. *If* $C = \mathbf{circulant}(c_0, c_1, \ldots, c_{n-1})$, *then*

$$C = \sum_{j=0}^{n-1} c_j R^j. \tag{5.45}$$

Moreover, $\{\frac{1}{\sqrt{n}} R^j\}_{j=0}^{n-1}$ *forms an orthonormal set under the Frobenius inner product.*

Lemma 5.15. *Let* $w = \exp(-\hat{\imath} 2\pi/n)$. *Then*

$$R = F^* \operatorname{diag}(1, w, w^2, \ldots, w^{n-1})\, F. \tag{5.46}$$

Proof. From (5.18), $[F]_{ij} = w^{ij}/\sqrt{n}$. Consequently,

$$[F^* \operatorname{diag}(1, w, w^2, \ldots, w^{n-1})\, F]_{ij} = \frac{1}{n} \sum_{k=0}^{n-1} \overline{w}^{ik} w^k w^{jk} = \frac{1}{n} \sum_{k=0}^{n-1} w^{(-i+1+j)k}.$$

Equation (5.46) follows from Exercise 5.4 and from (5.44). □

Corollary 5.16. *If* $C = \mathbf{circulant}(\mathbf{c})$, *then*

$$C = F^* \operatorname{diag}(\hat{\mathbf{c}})\, F, \tag{5.47}$$

where F *is the Fourier matrix* (5.18) *and*

$$\hat{\mathbf{c}} = \sqrt{n} F \mathbf{c} = \mathbf{fft}(\mathbf{c}). \tag{5.48}$$

The components of $\hat{\mathbf{c}}$ *are the eigenvalues of* C, *and the columns of* F^* *are the corresponding eigenvectors.*

Proof. From (5.45)–(5.46),

$$C = \sum_{j=0}^{n-1} c_j F^* \left[\operatorname{diag}(w^i)\right]^j F = F^* \operatorname{diag}\left(\sum_{j=0}^{n-1} c_j w^{ij}\right) F.$$

Equations (5.47)–(5.48) now follow from (5.36). □

Remark 5.17. From (5.47)–(5.48) and section 5.2.2, circulant matrix-vector products $\mathbf{v} = C\mathbf{f}$ can be computed at $\mathcal{O}(n \log n)$ cost by (i) computing $\hat{\mathbf{f}} = \mathbf{fft}(\mathbf{f})$; (ii) computing the component-wise vector product $\hat{\mathbf{v}} = \hat{\mathbf{c}}. * \hat{\mathbf{f}}$; and (iii) computing $\mathbf{v} = \mathbf{ifft}(\hat{\mathbf{v}})$. Similarly, nonsingular circulant systems can be solved at $\mathcal{O}(n \log n)$ cost using the fact that

$$C^{-1} = F^* \mathrm{diag}(1./\hat{\mathbf{c}}) F. \tag{5.49}$$

Remark 5.18. Toeplitz matrix-vector products can be efficiently computed by combining circulant embedding with FFTs. Let T be the $n \times n$ Toeplitz matrix in (5.42), let $\mathbf{v} \in R^n$, and define $S = \mathbf{toeplitz}(\mathbf{s})$, with

$$\mathbf{s} = (t_1, t_2, \ldots, t_{n-1}, 0, t_{1-n}, \ldots, t_{-2}, t_{-1}).$$

Then

$$C^{ext} \begin{bmatrix} \mathbf{v} \\ \mathbf{0}_{n\times 1} \end{bmatrix} = \begin{bmatrix} T\mathbf{v} \\ S\mathbf{v} \end{bmatrix}, \qquad \text{where} \quad C^{ext} = \begin{bmatrix} T & S \\ S & T \end{bmatrix}.$$

The $2n \times 2n$ block matrix C^{ext}, which we call the circulant extension of T, can be expressed as $C^{ext} = \mathbf{circulant}(\mathbf{c}^{ext})$, where

$$\mathbf{c}^{ext} = (t_0, t_1, \ldots, t_{n-1}, 0, t_{1-n}, \ldots, t_{-1}) \in \mathbb{C}^{2n}.$$

Consequently, to compute $\mathbf{w} = T\mathbf{v}$, (i) compute $\hat{\mathbf{c}}^{ext} = \mathbf{fft}(\mathbf{c}^{ext})$ and $\hat{\mathbf{v}}^{ext} = \mathbf{fft}((\mathbf{v}, \ \mathbf{0}_{\mathbf{n}\times\mathbf{1}}))$; (ii) compute $\hat{\mathbf{w}}^{ext} = \hat{\mathbf{c}}^{ext}. * \hat{\mathbf{v}}^{ext}$; (iii) compute $\mathbf{w}^{ext} = \mathbf{ifft}(\hat{\mathbf{w}}^{ext})$; and (iv) extract the first n components of $\mathbf{w}^{ext}$ to obtain $\mathbf{w}$.

5.2.4 Best Circulant Approximation

In this section we consider the computation of circulant approximations to square matrices. These approximations can be used to construct preconditioners for Toeplitz systems and will form the basic components of the level 1 and level 2 block circulant preconditioners discussed in section 5.3.3.

Let $\mathcal{C}_n$ denote the set of $n \times n$ circulant matrices. That this is a linear subspace of $\mathbb{C}^{n\times n}$ is an immediate consequence of the Proposition 5.14. We now apply approximation theoretic tools from section 2.

Definition 5.19. Given $A \in \mathbb{C}^{n\times n}$, the best circulant approximation to A in the Frobenius norm is given by

$$C(A) = \arg\min_{C\in\mathcal{C}_n} ||C - A||_{Fro}. \tag{5.50}$$

Proposition 5.20. *Let $A \in \mathbb{R}^{n\times n}$. Then $C(A) = \mathbf{circulant}(c_0, c_1, \ldots, c_{n-1})$, where*

$$c_j = \frac{1}{n}\langle A, R^j \rangle_{Fro}, \qquad j = 0, 1, \ldots, n-1.$$

The following lemma can be used to verify that best circulant approximation preserves symmetry and positive definiteness. For a proof, see Exercise 5.16.

Lemma 5.21.

$$C(A) = F^* \Lambda F, \tag{5.51}$$

where F is the Fourier matrix and Λ is the diagonal matrix whose diagonal entries are the same as those of FAF^*.

Theorem 5.22. *The mapping $A \mapsto C(A)$ is a (linear) projection operator. This mapping preserves symmetry and positive definiteness.*

$C(A)$ has some nice theoretical approximation properties which make it suitable for preconditioning Toeplitz systems. For details, see [65, Chapter 5].

The following result indicates that the cost of setting up the best circulant approximation to a Toeplitz matrix is $\mathcal{O}(n)$. See Exercise 5.18 for proof.

Corollary 5.23. *Let $T =$ **toeplitz(t)**, where $\mathbf{t} = (t_{1-n}, \ldots, t_{-1}, t_0, t_1, \ldots, t_{n-1})$. Then $C(T) =$ **circulant(c)**, where $\mathbf{c}$ has entries*

$$c_j = \frac{(n-j)t_j + jt_{j-n}}{n}, \qquad j = 0, 1, \ldots, n-1.$$

One can also replace the best circulant approximation by the best cosine and sine approximations with respect to the Frobenius norm. See [15, 16] for details. See also Exercise 5.19.

5.2.5 Block Toeplitz and Block Circulant Matrices

We next examine analogues of Toeplitz and circulant matrices that arise in two-dimensional convolution.

Definition 5.24. An $n_x n_y \times n_x n_y$ matrix T is called block Toeplitz with Toeplitz blocks (BTTB) if it has the block form

$$T = \begin{bmatrix} T_0 & T_{-1} & \cdots & T_{1-n_y} \\ T_1 & T_0 & T_{-1} & \vdots \\ \vdots & \ddots & \ddots & T_{-1} \\ T_{n_y-1} & \cdots & T_1 & T_0 \end{bmatrix}, \tag{5.52}$$

where each block T_j is an $n_x \times n_x$ Toeplitz matrix.

Definition 5.25. Given an array $v \in \mathbb{C}^{n_x \times n_y}$, one can obtain a vector $\mathbf{v} \in \mathbb{C}^{n_x n_y}$ by stacking the columns of v. This defines a linear operator **vec** : $\mathbb{C}^{n_x \times n_y} \to \mathbb{C}^{n_x n_y}$,

$$\mathbf{vec}(v) = [v_{1,1} \ldots v_{n_x,1} v_{1,2} \ldots v_{n_x,2} \ldots v_{1,n_y} \ldots v_{n_x,n_y}]^T. \tag{5.53}$$

This corresponds to lexicographical column ordering of the components in the array v. The symbol **array** denotes the inverse of the **vec** operator,

$$\mathbf{array}(\mathbf{vec}(v)) = v, \qquad \mathbf{vec}(\mathbf{array}(\mathbf{v})) = \mathbf{v}, \tag{5.54}$$

whenever $v \in \mathbb{C}^{n_x \times n_y}$ and $\mathbf{v} \in \mathbb{C}^{n_x n_y}$.

We can now relate block Toeplitz matrix-vector multiplication to discrete two-dimensiona convolution.

Proposition 5.26. *The two-dimensional convolution product* (5.25) *can be expressed as*

$$t \star f = \mathbf{array}(T\ \mathbf{vec}(f)), \tag{5.55}$$

where T is the $n_x n_y \times n_x n_y$ BTTB matrix of the form (5.52) *with $T_j = \mathbf{toeplitz}(t_{\cdot,j})$. Here $t_{\cdot,j}$ denotes the jth column of the $(2n_x - 1) \times (2n_y - 1)$ array t. We indicate this situation by $T = \mathbf{bttb}(t)$.*

Definition 5.27. An $n_x n_y \times n_x n_y$ matrix C is block circulant with circulant blocks (BCCB) if (i) C is BTTB; (ii) the $n_x \times n_x$ block rows of C are all circulant right shifts of each other; and (iii) each block is a circulant matrix. In other words,

$$C = \begin{bmatrix} C_0 & C_{n_y-1} & \cdots & C_2 & C_1 \\ C_1 & C_0 & C_{n_y-1} & \ddots & C_2 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ C_{n_y-2} & \ddots & C_1 & C_0 & C_{n_y-1} \\ C_{n_y-1} & C_{n_y-2} & \cdots & C_1 & C_0 \end{bmatrix}, \tag{5.56}$$

where each C_j is an $n_x \times n_x$ circulant matrix.

Proposition 5.28. *Suppose we have a BTTB matrix $C = \mathbf{bttb}(c^{ext})$, where c^{ext} is the periodic extension of $c \in \mathbb{C}^{n_x \times n_y}$ of size $(2n_x - 1) \times (2n_y - 1)$. Then C is BCCB, and we can obtain c from the first column $C_{\cdot,1}$ of C in the representation* (5.56) *by taking*

$$c = \mathbf{array}(C_{\cdot,1}). \tag{5.57}$$

Moreover, we can generate the jth block in (5.56) *from the jth column of c via*

$$C_j = \mathbf{circulant}(c_{\cdot,j}).$$

We indicate this situation by $C = \mathbf{bccb}(c)$.

The computation of the two-dimensional DFT in (5.24) can be carried out by first applying the one-dimensional DFT to the columns of the array f and then applying the one-dimensional DFT to the rows of the resulting array. This can be expressed in terms of matrices.

Definition 5.29. The tensor product of an $m \times n$ matrix A and a $p \times q$ matrix B is the $(mp) \times (nq)$ matrix

$$A \otimes B = \begin{bmatrix} a_{11}B & a_{12}B & \cdots & a_{1n}B \\ a_{21}B & a_{22}B & \cdots & a_{2n}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}B & a_{m2}B & \cdots & a_{mn}B \end{bmatrix}. \tag{5.58}$$

Proposition 5.30. *Given $f \in \mathbb{C}^{n_y \times n_y}$,*

$$\mathcal{F}\{f\} = \mathbf{array}((F_y \otimes F_x)\ \mathbf{vec}(f)),$$

where F_y and F_x are, respectively, Fourier matrices (5.18) *of size $n_y \times n_y$ and $n_x \times n_x$.*

The following result indicates how to compute BCCB matrix-vector products using two-dimensional DFTs, and it provides the eigenvalues of the BCCB matrix.

Proposition 5.31. *Let $C = \mathbf{bccb}(c)$, where $c \in \mathbb{C}^{n_x \times n_y}$. Then*

$$C = F^* \operatorname{diag}(\mathbf{vec}(\hat{c}))\, F, \tag{5.59}$$

where $F = F_y \otimes F_x$ and

$$\hat{c} = \sqrt{n_x n_y}\,\mathcal{F}\{c\} = \mathbf{fft2}(c).$$

The components of $\hat{c}$ are the eigenvalues of C.

This proposition leads to the following scheme for computing BCCB matrix-vector products. If $\mathbf{f} = \mathbf{vec}(f)$ and C has a representation (5.59), then

$$C\mathbf{f} = \mathbf{vec}(\mathcal{F}^{-1}\{\hat{c}. * \mathcal{F}\{f\}) = \mathbf{vec}(\mathbf{ifft2}(\hat{c}. * \mathbf{fft2}(f))). \tag{5.60}$$

In a manner analogous to the one-dimensional case (see Remark 5.18), BTTB matrix-vector products $T\mathbf{f}$ can be computed using block circulant extension combined with the two-dimensional DFT. Let $T = \mathbf{bttb}(t)$, where $t \in \mathbb{C}^{(2n_x-1)\times(2n_y-1)}$. We first extend t by zeros along the top and left margins, obtaining a $(2n_x) \times (2n_y)$ array

$$\tilde{t} = \begin{bmatrix} 0 & 0 & \cdots & 0 & \cdots & 0 \\ 0 & t_{1-n_x,1-n_y} & \cdots & t_{1-n_x,0} & \cdots & t_{1-n_x,n_y-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & t_{0,1-n_y} & \cdots & t_{0,0} & \cdots & t_{0,n_y-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & t_{n_x-1,1-n_y} & \cdots & t_{n_x-1,0} & \cdots & t_{n_x-1,n_y-1} \end{bmatrix}. \tag{5.61}$$

Next, partition $\tilde{t}$ into four $n_x \times n_y$ subblocks,

$$\tilde{t} = \begin{bmatrix} \tilde{t}_{11} & \tilde{t}_{12} \\ \tilde{t}_{21} & \tilde{t}_{22} \end{bmatrix}, \tag{5.62}$$

and then reorder the blocks to generate

$$c^{ext} = \begin{bmatrix} \tilde{t}_{22} & \tilde{t}_{21} \\ \tilde{t}_{12} & \tilde{t}_{11} \end{bmatrix}. \tag{5.63}$$

Note that the entry $t_{0,0}$ lies in the upper left corner of c^{ext}. The matrix $C^{ext} = \mathbf{bccb}(c^{ext})$ is the block analogue of the circulant extension in Remark 5.18.

Now to compute $T\mathbf{f}$, where $\mathbf{f} \in \mathbb{C}^{n_x n_y}$, first construct $f = \mathbf{array}(\mathbf{f}) \in \mathbb{C}^{n_x \times n_y}$. Then extend f by zeros to obtain $f^{ext} \in \mathbb{C}^{2n_x \times 2n_y}$,

$$f^{ext} = \begin{bmatrix} f & 0_{n_x \times n_y} \\ 0_{n_x \times n_y} & 0_{n_x \times n_y} \end{bmatrix}. \tag{5.64}$$

Next, compute $\mathbf{g} = C^{ext}\mathbf{vec}(f_{ext})$. Finally, $T\mathbf{f}$ can be obtained by extracting the leading $n_x \times n_y$ subblock of $\mathbf{array}(\mathbf{g})$ and then applying to this subblock the **vec** operator. The next algorithm follows from (5.60).

Algorithm 5.2.1. BTTB Matrix-Vector Product Computation by Block Circulant Extension.
Let $f = \mathbf{array}(\mathbf{f}) \in \mathbb{C}^{n_x \times n_y}$, and let $T = \mathbf{bttb}(t)$, where $t \in \mathbb{C}^{(2n_x-1)\times(2n_y-1)}$. To compute $\mathbf{g} = T\mathbf{f}$ or, equivalently, to compute $g = t \star f$,

Construct $c^{ext} \in \mathbb{C}^{2n_x \times 2n_y}$ from t via (5.61)–(5.63).
$\hat{c}^{ext} := \mathbf{fft2}(c^{ext})$.
Extend f to a $(2n_x) \times (2n_y)$ array, f^{ext} via (5.64).
$\hat{f}^{ext} := \mathbf{fft2}(f^{ext})$.
$\hat{g}^{ext} := \hat{c}^{ext} .* \hat{f}^{ext}$.
$g^{ext} := \mathbf{ifft2}(\hat{g}^{ext})$.
Extract the leading $n_x \times n_y$ subblock of g^{ext} to obtain g.
Then $\mathbf{g} = \mathbf{vec}(g)$.

5.3 Fourier-Based Deblurring Methods

The discrete, noisy data model (5.13) has a matrix-vector representation

$$\mathbf{d} = T\mathbf{f} + \eta, \tag{5.65}$$

where $T = \mathbf{bttb}(t)$ (see Proposition 5.26), $\mathbf{d} = \mathbf{vec}(d)$, $\mathbf{f} = \mathbf{vec}(f)$, and $\eta = \mathbf{vec}(\eta)$. We refer to T as the blurring matrix. Given T and $\mathbf{d}$, we wish to estimate $\mathbf{f}$. For now we assume that the (unknown) error term η is predominantly Gaussian. This suggests a least squares fit-to-data functional; see Example 4.26. We consider the discrete Tikhonov functional with a quadratic penalty term:

$$\mathcal{T}_\alpha(\mathbf{f}) = \frac{1}{2}||T\mathbf{f} - \mathbf{d}||^2 + \frac{\alpha}{2}\mathbf{f}^* L\mathbf{f}. \tag{5.66}$$

Here L is symmetric positive definite and is called the penalty matrix, and $\alpha > 0$ is the regularization parameter. A minimizer of $\mathcal{T}_\alpha$ solves the linear system

$$(T^*T + \alpha L)\mathbf{f} = T^*\mathbf{d}. \tag{5.67}$$

Consider the squared L^2-norm penalty functional $J(f)$; see (2.46). Applying midpoint quadrature on an equispaced grid as in section 5.1, we obtain

$$J(f) = \frac{1}{2}\int\int f(x,y)^2\, dx\, dy \approx \frac{\Delta x\, \Delta y}{2} \sum_{i=0}^{n_x-1}\sum_{j=0}^{n_y-1} f_{ij}^2. \tag{5.68}$$

By incorporating the factor of $\Delta x\, \Delta y$ into the regularization parameter α and reordering the f_{ij}'s into a column vector $\mathbf{f}$, the right-hand side of (5.68) corresponds to the penalty matrix $L = I$, the identity matrix, in (5.66)–(5.67).

To incorporate an a priori assumption of smoothness, one can apply the Sobolev H^1 penalty functional (2.47). The corresponding penalty operator is the negative Laplacian; cf. (2.49) with diffusion coefficient $\kappa = 1$. Standard finite difference approximation of derivatives yields (up to a constant multiple) the negative discrete Laplacian,

$$[\mathcal{L}f]_{ij} = 4f_{ij} - f_{i+1,j} - f_{i-1,j} - f_{i,j+1} - f_{i,j-1}. \tag{5.69}$$

If we assume periodic boundary conditions

$$f_{0,j} = f_{n_x+1,j} \quad \text{and} \quad f_{i,0} = f_{i,n_y+1},$$

and we apply lexicographical column ordering of the unknowns, we obtain the $n_x n_y \times n_x n_y$ penalty matrix with $n_y \times n_y$ block representation

$$L = \begin{bmatrix} L_0 & -I & \Theta & \dots & \Theta & -I \\ -I & L_0 & -I & \Theta & \ddots & \Theta \\ \Theta & -I & L_0 & -I & \Theta & \vdots \\ \vdots & & \ddots & \ddots & \ddots & \\ \Theta & \ddots & \Theta & -I & L_0 & -I \\ -I & \Theta & \dots & \Theta & -I & L_0 \end{bmatrix}. \tag{5.70}$$

Here I represents the $n_x \times n_x$ identity matrix, Θ represents the $n_x \times n_x$ zero matrix, and L_0 is the $n_x \times n_x$ matrix of the form

$$L_0 = \begin{bmatrix} 4 & -1 & 0 & \dots & 0 & -1 \\ -1 & 4 & -1 & 0 & \ddots & 0 \\ 0 & -1 & 4 & -1 & 0 & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \ddots & 0 & -1 & 4 & -1 \\ -1 & 0 & \dots & 0 & -1 & 4 \end{bmatrix}. \tag{5.71}$$

Remark 5.32. To replace periodic boundary conditions with homogeneous Dirichlet boundary conditions, replace the upper right and lower left $-I$'s in (5.70) with Θ's, and drop the corner -1's from L_0 in (5.71). See [102]. Additional modifications to the main diagonal of L are needed to incorporate homogeneous Neumann, or no-flux, boundary conditions. In this case, each of the rows of L must sum to zero, so the 4's on the main diagonal may be replaced by 3's or 2's, depending on the number of off-diagonal -1's in a given row. Again, see [102]. In both the Dirichlet and Neumann cases, BCCB structure is lost. In the Dirichlet case, the matrix L is BTTB.

Remark 5.33. Regardless of whether the boundary conditions are periodic, Dirichlet, or Neumann, the matrix L corresponding to (5.69) is sparse, and matrix-vector products can be computed in $5n_x n_y + \mathcal{O}(1)$ operations. L is also symmetric and positive semidefinite. In the Dirichlet case, L is positive definite.

5.3.1 Direct Fourier Inversion

If both the blurring matrix T and the penalty matrix L are BCCB, then the Tikhonov system (5.67) can be solved directly using two-dimensional FFTs. In this case, L has an array representation

$$\mathcal{L}f = \mathbf{ifft2}(\hat{\ell}. * \mathbf{fft2}(f)), \qquad f \in \mathbb{R}^{n_x \times n_y}. \tag{5.72}$$

For example, the identity matrix has such a representation with $\hat{\ell} = 1_{n_x \times n_y}$, the $n_x \times n_y$ array of ones. For the discrete Laplacian with periodic boundary conditions (see (5.70)–(5.71)),

one takes $\hat{\ell} = \mathbf{fft2}(\mathbf{array}(L_{\cdot,1}))$, where $L_{\cdot,1}$ denotes the first column of the matrix L in (5.70). Let $T = \mathbf{bccb}(t)$ with $t \in \mathbb{R}^{n_x \times n_y}$ (Proposition 5.28). The following algorithm yields the solution to (5.67).

Algorithm 5.3.1. Tikhonov Regularization for BCCB Systems.
Given $d = \mathbf{array}(\mathbf{d}) \in \mathbb{R}^{n_x \times n_y}$, given $t \in \mathbb{R}^{n_x \times n_y}$ for which $T = \mathbf{bccb}(t)$, and given $\ell \in \mathbb{R}^{n_x \times n_y}$ for which $L = \mathbf{bccb}(\ell)$, to solve the system
$(T^*T + \alpha L)\mathbf{f} = \mathbf{d}$,

$\hat{\ell} := \mathbf{fft2}(\ell)$; *% Fourier representer for L*
$\hat{t} := \mathbf{fft2}(t)$; *% Fourier representer for T*
$\hat{d} := \mathbf{fft2}(d)$;
$\hat{f} := \mathrm{conj}(\hat{t}) .* \hat{d} ./ (|\hat{t}|^2 + \alpha\hat{\ell})$;
$f := \mathbf{ifft2}(\hat{f})$;

Then $\mathbf{f} = \mathbf{vec}(f)$.

Figure 5.3 shows some two-dimensional image reconstructions computed using this algorithm. Two penalty matrices L are employed: the identity $L = I$ and the negative discrete Laplacian with periodic boundary conditions; see (5.70)–(5.71). As should be expected, the best reconstruction obtained with the identity penalty matrix is less smooth than the corresponding best reconstruction obtained using the discrete Laplacian.

If either L or T is not BCCB, then Fourier transform techniques cannot be used directly to solve system (5.67). The use of direct matrix decomposition methods like the LU factorization or the Cholesky factorization [46] to solve this system is often precluded by the size $n_x n_y$ and the fact that T is not sparse. With certain boundary conditions, other fast transform techniques, e.g., fast cosine transform for Neumann boundary conditions, may be directly applied to (5.67). Otherwise, iterative techniques like the conjugate gradient Algorithm 3.2 must be used.

5.3.2 CG for Block Toeplitz Systems

Assume now that the blurring matrix T in (5.66) is BTTB and the penalty matrix L is symmetric positive semidefinite with $\mathcal{O}(n)$ nonzero entries, where $n = n_x n_y$ denotes the size of T and L. Let

$$A = T^*T + \alpha L \tag{5.73}$$

denote the coefficient matrix in (5.67). We assume that A is nonsingular, so it is SPD. Corollary 3.8 then guarantees that the CG Algorithm 3.2 will generate iterates that converge to the solution of system (5.67). From Algorithm 5.2.5, section 5.2.2, and Remark 5.33 we see that the cost of applying the matrix A to a vector is $\mathcal{O}(n \log n)$. This dominates the $\mathcal{O}(n)$ cost of the inner product computations in each CG iteration; see Remark 3.10. Given this relatively low cost per iteration, CG is often a viable method for solving (5.67), even when the convergence rate is somewhat slow.

Remark 5.34. CG iterations can be applied to minimize the unregularized least squares cost functional $J(\mathbf{f}) = ||T\mathbf{f} - \mathbf{d}||^2$ or, equivalently, to solve the "normal equations":

$$T^*T\mathbf{f} = T^*\mathbf{d}.$$

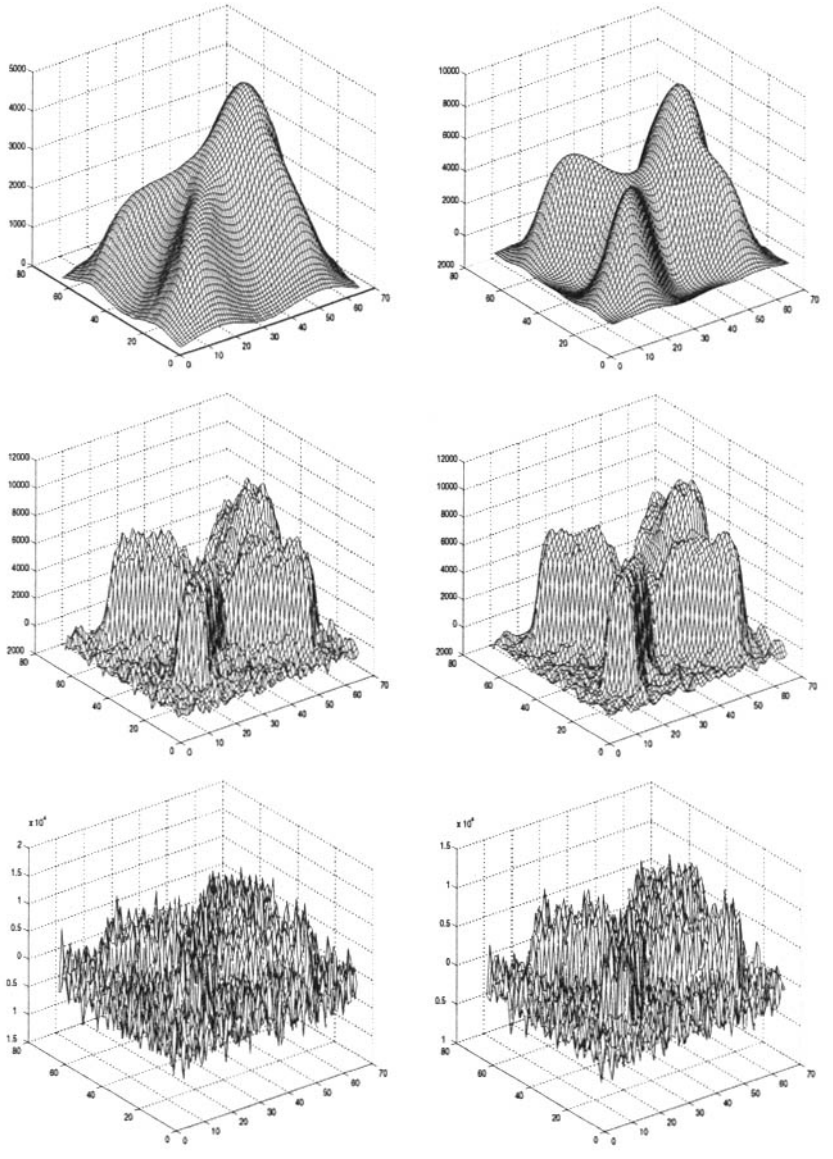


Figure 5.3. *Reconstructions from data in Figures* 5.1–5.2, *generated using Algorithm* 5.3.1. *Reconstructions on the left were obtained using the identity penalty matrix. Those on the right were obtained using the negative Laplacian with periodic boundary conditions. Images in the top row were generated with regularization parameter* $\alpha = 10^{-2}$, *in the second row,* $\alpha = 10^{-4}$, *and in the third row,* $\alpha = 10^{-6}$.

See [9] for implementation details. As with Landweber iteration, (section 1.4), the CG iteration count plays the role of the regularization parameter. See [50] and the references therein, or see [35, Chapter 7].

Preconditioning can sometimes significantly speed up CG convergence for system (5.67).

5.3.3 Block Circulant Preconditioners

The Block Circulant Extension Preconditioner

The inversion of Toeplitz systems by circulant preconditioning was first suggested by Strang [105]. See [14] for an extensive review of this subject. In this section we discuss various block generalizations of circulant preconditioners.

The following preconditioner is based on the extension idea underlying Algorithm 5.2.5. It is similar to the Toeplitz approximate inverse preconditioners of Hanke and Nagy [53]. To apply this preconditioner to the Tikhonov system (5.67), we require both the blurring matrix T and the penalty matrix L to be BTTB.

Algorithm 5.3.2. Preconditioning the Tikhonov System by Block Circulant Extension. Let $T = \mathbf{bttb}(t)$ and $L = \mathbf{bttb}(\ell)$, where $t, \ell \in \mathbb{C}^{(2n_x-1)\times(2n_y-1)}$. Let $r \in \mathbb{C}^{n_x \times n_y}$ and $\mathbf{r} = \mathbf{vec}(r)$. To compute $\mathbf{s} = M^{-1}\mathbf{r}$, where M is the block circulant extension preconditioner for $A = T^*T + \alpha L$,

Assemble $c_t \in \mathbb{C}^{2n_x \times 2n_y}$ and $c_\ell \in \mathbb{C}^{2n_x \times 2n_y}$ from t and ℓ
via (5.61)–(5.63).
$\hat{c}_t := \mathbf{fft2}(c_t)$.
$\hat{c}_\ell := \mathbf{fft2}(c_\ell)$.
Extend r to a $(2n_x) \times (2n_y)$ array, $r_{ext} = \begin{bmatrix} r & 0_{n_x \times n_y} \\ 0_{n_x \times n_y} & 0_{n_x \times n_y} \end{bmatrix}$.
$\hat{r}_{ext} := \mathbf{fft2}(r_{ext})$.
$\hat{s}_{ext} := \hat{r}_{ext} ./ (|\hat{c}_t|^2 + \alpha\, \hat{c}_\ell)$.
$s_{ext} := \mathbf{ifft2}(\hat{s}_{ext})$.
Extract the leading $n_x \times n_y$ subblock of s_{ext} to obtain s.

Then $\mathbf{s} = \mathbf{vec}(s)$.

Level 1 and level 2 block circulant preconditioners are derived from the best circulant approximation, presented in section 5.2.4. See [20, 17, 18] for further details. Unlike the block circulant extension preconditioner, these can be applied even when the matrices T and L are not BTTB.

Level 1 Block Circulant Preconditioning

We begin with the construction of the level 1 block circulant approximation. Assume for simplicity that T is BTTB with representation (5.52). Its level 1 approximation, which we denote by $C_1(T)$ is obtained by replacing each of the blocks T_j by its best circulant

approximation $C(T_j)$, i.e.,

$$C_1(T) = \begin{bmatrix} C(T_0) & C(T_{-1}) & \cdots & C(T_{1-n_y}) \\ C(T_1) & C(T_0) & C(T_{-1}) & \vdots \\ \vdots & \ddots & \ddots & C(T_{-1}) \\ C(T_{n_y-1}) & \cdots & C(T_1) & C(T_0) \end{bmatrix}. \tag{5.74}$$

Note that $C_1(T)$ is block Toeplitz with circulant blocks. More generally, if T is a block matrix with block components T_{ij}, then $C_1(T)$ is the block matrix with circulant blocks $C(T_{ij})$. Similarly, one can compute the level 1 approximation to the penalty matrix L. For instance, for the discrete Laplacian (5.70) with Dirichlet boundary conditions,

$$C_1(L) = \begin{bmatrix} C(L_0) & -I & \Theta & \dots & \Theta & \Theta \\ -I & C(L_0) & -I & \Theta & \ddots & \Theta \\ \Theta & -I & C(L_0) & -I & \Theta & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \Theta \\ \Theta & \ddots & \Theta & -I & C(L_0) & -I \\ \Theta & \Theta & \dots & \Theta & -I & C(L_0) \end{bmatrix}; \tag{5.75}$$

see Remark 5.32. This computation is greatly simplified by the fact that $C(I) = I$ and $C(\Theta) = \Theta$, consequences of Theorem 5.22. If Neumann boundary conditions are used instead of Dirichlet boundary conditions, the upper left and lower right $C(L_0)$ blocks must be modified.

As a consequence of Theorem 5.22, the level 1 circulant approximation to general block matrices is linear, i.e.,

$$C_1(\alpha A + \beta B) = \alpha C_1(A) + \beta C_1(B), \tag{5.76}$$

for any block matrices A and B and scalars α and β. One can also show [20] that $C_1(A)$ is SPD whenever A is.

Now consider the construction of a preconditioner for $A = T^*T + \alpha L$. From (5.76), $C_1(A) = C_1(T^*T) + \alpha C_1(L)$. Unfortunately, T^*T need not be BTTB (see Exercise 5.26), and $C_1(T^*T)$ may be difficult to compute. However, $C_1(T^*T)$ may be well approximated by $C_1(T)^* C_1(T)$. Hence, we take as a preconditioner for A

$$M_1(A) = C_1(T)^* C_1(T) + \alpha C_1(L). \tag{5.77}$$

To implement this level 1 preconditioner, we rely on the fact that the discrete Fourier transform can be used to diagonalize circulant matrices; see Corollary 5.16. Applying this to the blocks of (5.74) gives

$$C(T_j) = F_x^* \Lambda_j F_x, \qquad j = 1 - n_y, \dots, 0, \dots, n_y - 1,$$

where F_x denotes the $n_x \times n_x$ Fourier matrix, and Λ_j denotes the $n_x \times n_x$ diagonal matrix whose diagonal entries are the eigenvalues of $C(T_j)$. These entries are the components of $\mathbf{fft2}(\mathbf{c}_j)$, where $C(T_j) = \mathbf{circulant}(\mathbf{c}_j)$. From this we obtain

$$C_1(T) = (I_y \otimes F_x)^* \, T(\Lambda) \, (I_y \otimes F_x), \tag{5.78}$$

where I_y denotes the $n_y \times n_y$ identity matrix and $T(\Lambda)$ is the block Toeplitz matrix with the Λ_j's as its diagonal blocks,

$$T(\Lambda) = \begin{bmatrix} \Lambda_0 & \Lambda_{-1} & \cdots & \Lambda_{1-n_y} \\ \Lambda_1 & \Lambda_0 & \Lambda_{-1} & \vdots \\ \vdots & \ddots & \ddots & \Lambda_{-1} \\ \Lambda_{n_y-1} & \cdots & \Lambda_1 & \Lambda_0 \end{bmatrix}.$$

There exists a permutation matrix P, corresponding to a reindexing of unknowns from column lexicographical order to row lexicographical order, for which $P^T T(\Lambda) P$ is block diagonal. Hence,

$$C_1(T) = (I_y \otimes F_x)^* P^T \operatorname{diag}(D_1, \ldots, D_{n_x}) P (I_y \otimes F_x), \tag{5.79}$$

where the diagonal blocks D_k are each dense $n_y \times n_y$ matrices with components

$$[D_k]_{ij} = [\Lambda_{i-j}]_{k,k}, \qquad 1 \le i, j \le n_y, \quad k = 1, \ldots, n_x.$$

Assume that L has an analogous representation:

$$C_1(L) = (I_y \otimes F_x)^* P^T \operatorname{diag}(E_1, \ldots, E_{n_x}) P (I_y \otimes F_x). \tag{5.80}$$

From (5.79)–(5.80) and the fact that $I_y \otimes F_x$ and P are both unitary matrices, we obtain the following representation for the preconditioner (5.77):

$$M_1(A) = (I_y \otimes F_x)^* P^T \operatorname{diag}(D_k^2 + \alpha E_k) P (I_y \otimes F_x). \tag{5.81}$$

Now consider the computation of

$$\mathbf{w} = M_1(A)^{-1}\mathbf{v} = (I_y \otimes F_x)^* P^T \operatorname{diag}(D_k^2 + \alpha E_k)^{-1} P (I_y \otimes F_x),$$

where $\mathbf{v} = \mathbf{vec}(v)$ and v is an $n_x \times n_y$ array. The matrix-vector product $\hat{\mathbf{v}} = (I_y \otimes F_x)\mathbf{v}$ corresponds to applying one-dimensional DFTs to the columns of v. Let $\hat{v} = \mathbf{array}(\hat{\mathbf{v}})$. The computation $\hat{\mathbf{w}} = P^T \operatorname{diag}(D_k^2 + \alpha E_k)^{-1} P \hat{\mathbf{v}}$ can be carried out by solving linear systems

$$(D_k^2 + \alpha E_k)\, \hat{w}_{k,\cdot} = \hat{v}_{k,\cdot}, \qquad k = 1, \ldots, n_x, \tag{5.82}$$

each of size $n_y \times n_y$, where $\hat{v}_{k,\cdot}$ denotes the kth row of $\hat{v}$, and storing $\hat{w}_{k,\cdot}$ as the kth row of $\hat{w} = \mathbf{array}(\hat{\mathbf{w}})$. Finally, the computation $\mathbf{w} = (I_y \otimes F_x)^* \hat{\mathbf{w}}$ corresponds to applying inverse DFTs to the columns of $\hat{w}$. This yields $w = \mathbf{array}(\mathbf{w})$.

Level 2 Block Circulant Preconditioning

Suppose we have a block matrix T with a level 1 block circulant approximation $C_1(T)$ of the form (5.79). The level 2 block circulant approximation to T, which we denote by $C_2(T)$, can be obtained simply by replacing each of the diagonal block matrices D_k by its best circulant approximation $C(D_k)$. Let

$$C(D_k) = F_y^* \operatorname{diag}(\hat{\mathbf{d}}_k) F_y,$$

where $\operatorname{diag}(\hat{\mathbf{d}}_k)$ denotes the diagonal matrix whose diagonal entries comprise the components of the vector $\hat{\mathbf{d}}_k \in \mathbb{C}^{n_y}$. This yields the representation

$$\begin{aligned} C_2(T) &= (I_y \otimes F_x)^* P^T (I_y \otimes F_y)^* \\ &\quad \times \operatorname{diag}(\hat{\mathbf{d}}_1, \ldots, \hat{\mathbf{d}}_{n_x}) (I_y \otimes F_y) P (I_y \otimes F_x) \\ &= (F_y \otimes F_x)^* \operatorname{diag}(\hat{\mathbf{d}}_1, \ldots, \hat{\mathbf{d}}_{n_x}) (F_y \otimes F_x), \end{aligned} \tag{5.83}$$

where now $\text{diag}(\mathbf{d}_1, \ldots, \mathbf{d}_{n_x})$ denotes the $n_x n_y \times n_x n_y$ diagonal matrix with diagonal components equal to those of $\mathbf{vec}([\mathbf{d}_1 \ldots \mathbf{d}_{n_x}])$.

As was the case with the level 1 approximation, the level 2 approximation is linear (see (5.76)) and preserves symmetry and positive definiteness [20].

Based on (5.77) and (5.83), as a preconditioner for $A = T^*T + \alpha L$ we take

$$\begin{aligned} M_2(A) &= C_2(T)^* C_2(T) + \alpha C_2(L) \\ &= (F_y \otimes F_x)^* \, \text{diag}(|\hat{\mathbf{d}}_k|.^2 + \hat{\mathbf{e}}_k) \, (F_y \otimes F_x). \end{aligned} \tag{5.84}$$

The computation $\mathbf{w} = M_2(A)^{-1}\mathbf{v}$ is straightforward. The matrix-vector product $\hat{\mathbf{v}} = (F_y \otimes F_x)\,\mathbf{v}$ corresponds to applying the two-dimensional DFT to $v = \mathbf{array}(\mathbf{v})$. Next, to compute $\hat{\mathbf{w}} = \text{diag}(|\hat{\mathbf{d}}_k|.^2 + \hat{\mathbf{e}}_k)^{-1}\hat{\mathbf{v}}$, take $\hat{w} = \mathbf{array}(\hat{\mathbf{w}})$ to consist of columns

$$\hat{w}_{:,k} = \hat{v}_{:,k}./(|\hat{\mathbf{d}}_k|.^2 + \hat{\mathbf{e}}_k), \qquad k = 1, \ldots, n_y.$$

Finally, apply the inverse two-dimensional DFT to $\hat{w}$ to obtain $w = \mathbf{array}(\mathbf{w})$.

5.3.4 A Comparison of Block Circulant Preconditioners

We now compare the numerical performance of the various block circulant preconditioners. The test problem used here is an atmospheric optics deblurring problem very similar to that presented in Figures 5.1–5.2. The reconstruction method is Tikhonov regularization with the identity regularization operator, and the value used for the regularization parameter is $\alpha = 10^{-4}$. The image lies on a 128×128 pixel grid. Hence the number of unknowns is $n_x n_y = 128^2 = 16384$. Figure 5.4 shows the iterative solution error norm, $||f_\alpha^\nu - f_\alpha||$. Here f_α denotes the exact solution to the regularized system (5.67) and f_α^ν denotes the approximate solution obtained after ν PCG iterations. Note that unpreconditioned CG converged the most slowly, followed by PCG with level 2 preconditioning and PCG with level 1 preconditioning. PCG with block circulant extension preconditioning converged the fastest. Keep in mind that convergence rates may change as one varies parameters like n_x, n_y, and α.

The total computational cost of an iterative method equals the cost per iteration multiplied by the number of iterations required to meet a particular stopping tolerance. Except for the level 1 preconditioner, the cost per iteration is dominated by two-dimensional forward and inverse FFTs. As in section 5.2.2, let $\text{FFT2}(m, n)$ denote the cost of applying a forward or inverse two-dimensional FFT to an $m \times n$ array. Without preconditioning, each CG iteration costs $4 \times \text{FFT2}(2n_x, 2n_y) + \mathcal{O}(n_x n_y)$. The FFT costs are $2 \times \text{FFT2}(2n_x, 2n_y)$ to apply the BTTB matrix T on a $2n_x \times 2n_y$ grid and $2 \times \text{FFT2}(2n_x, 2n_y)$ to apply T^*; see Algorithm 5.2.5. Each application of the block circulant extension preconditioner costs an additional $2 \times \text{FFT2}(2n_x, 2n_y) + \mathcal{O}(n_x n_y)$; see Algorithm 5.3.3. This brings the total cost of each iteration of PCG with the block circulant extension preconditioner to essentially $6 \times \text{FFT2}(2n_x, 2n_y)$. Given the dramatic improvement in the convergence rate shown in Figure 5.4, it is clear that block circulant preconditioning greatly reduces the total cost when compared to unpreconditioned CG.

The cost of applying the level 2 block preconditioner is dominated by $2 \times \text{FFT2}(n_x, n_y)$ (note that no block circulant extension is required). Hence, the total cost per iteration of level 2 PCG is slightly more than two-thirds that of block circulant extension PCG. However, because of the differences in convergence rates, the overall cost of level 2 PCG is less than that of unpreconditioned CG but more than that for PCG with block circulant extension preconditioning.

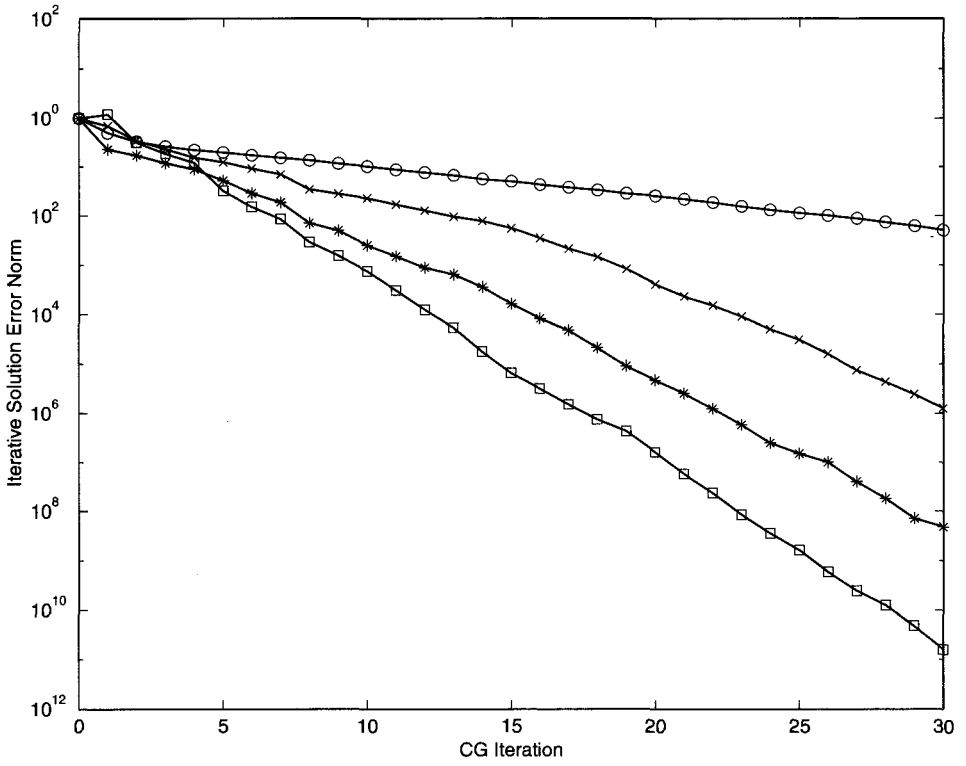


Figure 5.4. *Performance of block circulant preconditioners for a two-dimensional image deblurring problem. The iterative solution error norm, $||f_\alpha^v - f_\alpha||$, is plotted against iteration count v. The circles denote unpreconditioned CG; x's denote PCG with level 2 preconditioning; asterisks denote PCG with level 1 preconditioning; and squares denote PCG with preconditioning by block circulant extension.*

Finally, we examine the cost of the level 1 preconditioner. Let FFT(n) denote the cost of applying a forward or inverse FFT to a vector of length n. The cost of the FFTs in each application of this preconditioner is then $2n_y \times \text{FFT}(n_x)$. However, one must also solve the block systems (5.82). Assuming that Cholesky factorizations of the (dense) blocks have been precomputed, this cost is $n_y \times (n_x^2 + \mathcal{O}(n_x))$. This dominates the FFT costs. To compare this with the cost of the other preconditioners, assume $n_x = n_y$. Then for large n_x the level 1 cost is essentially n_x^3. One the other hand, the cost of the other preconditioners and of unpreconditioned CG is $\mathcal{O}(n_x^2 \log n_x)$—a quantity that becomes substantially *smaller* than n_x^3 for large n_x. Given this, and given the convergence behavior seen in Figure 5.4, PCG with level 1 preconditioning is more expensive than either unpreconditioned CG or PCG with any of the other preconditioners.

5.4 Multilevel Techniques

Multilevel techniques, which include wavelet and multigrid methods, are interesting alternatives to Fourier-based methods for the solution to the large linear systems that arise in image reconstruction. Rieder [97] combined wavelet decomposition techniques with Jacobi and

Gauss–Seidel type iterative methods to solve Fredholm first kind integral equations. Hanke and Vogel [55, 118] used Rieder's ideas to develop a class of two-level preconditioners to solve linear systems arising from regularized inverse problems. See also Jacobsen's MSc. thesis [63]. Riley and Vogel [98] showed that two-level preconditioners can be competitive with block circulant preconditioners in image reconstruction applications.

Unlike circulant preconditioners, multilevel techniques do not require Toeplitz structure. Work by Vogel [117] indicates that no advantage is gained by using more than two levels.

Exercises

5.1. Apply midpoint quadrature to obtain the discrete convolution in (5.13) from the continuous version (5.1). Show that $t_{ij} = k((i-1)\Delta x, (j-1)\Delta y)\ \Delta x \Delta y$, where $\Delta x = 1/n_x$, $\Delta y = 1/n_y$.

5.2. Interpret the DFT (5.17) as a truncation of domain combined with a quadrature applied to the continuous Fourier transform (5.9). Obtain an error bound for this approximation in terms of grid spacing and domain truncation.

5.3. Let A be in indicator function for the interval $[-1, 1]$,

$$A(x) = \begin{cases} 1, & -1 \le x \le 1, \\ 0 & \text{otherwise.} \end{cases}$$

Prove that $\hat{k} = \mathcal{F}\{|\mathcal{F}^{-1}\{A\}|^2\}$ is the ramp function,

$$\hat{k}(x) = \begin{cases} 2 - |x|, & -2 \le x \le 2, \\ 0 & \text{otherwise.} \end{cases}$$

5.4. Let $\omega = \exp(-\hat{\imath} 2\pi/n)$, where $\hat{\imath} = \sqrt{-1}$. Show that

$$\sum_{j=0}^{n-1} \omega^{jk} = \begin{cases} n & \text{if } k = 0, \\ 0 & \text{if } k = 1, 2, \ldots, n-1. \end{cases}$$

5.5. Prove that the Fourier matrix (5.18) is unitary. *Hint:* Show that $n[F^* F]_{jk} = \sum_{\ell=0}^{n-1} w^{\ell(-j+k)}$, where $w = \exp(-\hat{2}\pi/n)$. Then apply Exercise 5.4.

5.6. Prove that the DFT preserves Euclidean inner products and hence that it preserves Euclidean norms.

5.7. Prove Proposition 5.10.

5.8. Derive equation (5.35), given (5.34), FFT(1) = 0, and $n = 2^k$.

5.9. Verify that (5.23) and (5.38) are equivalent.

5.10. Verify the equalities in (5.40).

5.11. Prove Proposition 5.14.

5.12. Use Corollary 5.16 to prove Proposition 5.6.

5.13. Given the Fourier eigendecomposition (5.47) of a circulant matrix C, explain how to directly obtain the SVD of C.

5.14. Prove Proposition 5.20.

5.15. Show that if U is a unitary matrix, then $||UA||_{Fro} = ||A||_{Fro}$.

5.16. Prove Lemma 5.21. *Hint:* If C is circulant, then by Corollary 5.16 $C = F^*\Lambda F$ for some diagonal matrix Λ. Now use the fact that unitary transformations like F preserve Frobenius norm.

5.17. Prove Theorem 5.22.

5.18. Prove Corollary 5.23.

5.19. The $n \times n$ Cosine matrix Cos has entries

$$[\text{Cos}]_{ij} = \begin{cases} \frac{1}{\sqrt{n}}, & j = 1,\ i = 1, \ldots, n, \\ \sqrt{\frac{2}{n}} \cos\left(\frac{(2i-1)(j-1)\pi}{2n}\right), & j = 2, \ldots, n,\ i = 1, \ldots, n. \end{cases}$$

Show that this is a unitary matrix.

5.20. Prove Proposition 5.26.

5.21. Prove Proposition 5.28.

5.22. Prove Proposition 5.30.

5.23. Prove that the tensor product of Fourier matrices is unitary.

5.24. Prove Proposition 5.31.

5.25. Explain how to modify Algorithm 5.2.5 to handle the case when the array t has size $n_x \times n_y$ rather than $(2n_x - 1) \times (2n_y - 1)$. This happens when the PSF is taken to be the recorded image of an approximate point source.

5.26. Provide a simple counterexample to show that the product of Toeplitz matrices need not be Toeplitz.

5.27. Verify equation (5.78).

5.28. Give an explicit representation of the column-to-row permutation matrix P in equation (5.79).

5.29. Show that $(I_y \otimes F_y)\ P\ (I_y \otimes F_x) = F_y \otimes F_x$, thereby verifying equation (5.83).

5.30. Let T denote the Toeplitz matrix arising from the one-dimensional test problem of Chapter 1. Solve the linear system $(T^*T + \alpha I)\mathbf{f} = T^*\mathbf{d}$ using the preconditioned CG method with a preconditioner constructed from $C(T)$. Compare the results with unpreconditioned CG in terms of convergence rates and computational cost. How does the choice of the regularization parameter α affect convergence rates?

5.31. In the computations of section 5.3.4, the penalty matrix L was taken to be the identity. Replace this L with the negative discrete Laplacian with Dirichlet boundary conditions, and repeat these computations.

5.32. Conduct a careful numerical study, similar to that presented in section 5.3.4, in which level 1 and level 2 block circulant preconditioners are replaced by block cosine preconditioners. See [15, 16] for implementation details.

Chapter 6

Parameter Identification

By parameter identification, we usually mean the estimation of coefficients in a differential equation from observations of the solution to that equation. These coefficients are called system parameters, and the solution and its derivatives constitute the state variables. The forward problem is to compute the state variables given the system parameters and appropriate boundary conditions. The forward problem is typically well-posed. Parameter identification, the inverse problem of interest, is typically ill-posed. Moreover, even when the forward problem is linear in the state variable, the parameter identification problem is generally nonlinear. Specialized computational techniques with which to deal with this nonlinearity are presented later in this chapter. First we provide some illustrative examples.

Example 6.1. The damped harmonic oscillator equation [10] takes the form

$$m\frac{d^2x}{dt^2} + c\frac{dx}{dt} + kx = f(t), \qquad t > 0, \tag{6.1}$$

with initial conditions $x(0) = x_0$, $\frac{dx}{dt}(0) = v_0$. This is used to model the oscillations of a mass-spring system. See [10] for details. In this context, $x(t)$ represents the displacement of the mass at time t, and $\frac{dx}{dt}$ represents the velocity. x and $\frac{dx}{dt}$ comprise the state variables. The system parameters are the mass m, the damping coefficient c, the spring constant k, and the external forcing function $f(t)$. The forward problem is to determine the state variables, given the system parameters and the initial state (x_0, v_0). A number of inverse problems can be formulated, depending on what information is available. For example, damping and forcing may be absent. One may displace the system and, from observations of its resulting motion, try to determine m and k. This problem is ill-posed in the sense that one cannot uniquely determine both k and m from the observed data. The solution to equation (6.1), with $c = f = v_0 = 0$, is given by

$$x(t) = x_0 \cos(\omega t), \qquad \omega = \sqrt{\frac{k}{m}}, \tag{6.2}$$

where ω represents the frequency of oscillation of the system. From certain observations of the state, e.g., the displacement $x(t)$ over an interval of length $2\pi/\omega$, one can uniquely determine ω. However, from ω one can determine only the ratio k/m. From (6.2) one can formally compute

$$\omega = \frac{1}{t}\cos^{-1}\left(\frac{x(t)}{x_0}\right).$$

This establishes that the dependence of ω on $x(t)$ is nonlinear. Hence the dependence of either k or m (given that the other is known) on ω is nonlinear.

Example 6.2. The one-dimensional steady-state diffusion equation is

$$-\frac{d}{dx}\left(\kappa(x)\frac{du}{dx}\right) = f(x), \qquad 0 < x < 1, \tag{6.3}$$

with appropriate boundary conditions, e.g., Dirichlet conditions,

$$u(0) = u_L, \qquad u(1) = u_R. \tag{6.4}$$

This is used to model the steady-state temperature distribution within a thin metal rod [75]. In this setting, the state variable is the temperature distribution $u(x)$, $0 \le x \le 1$. The system parameters are the diffusion coefficient $\kappa(x)$ and the heat source term $f(x)$. Since these are functions rather than scalars, equation (6.3) is called a distributed parameter system. The forward problem, (6.3) together with Dirichlet boundary conditions, is well-posed when κ is smooth and bounded away from zero.

Consider the following inverse problem: Given $f(x)$ and $u(x)$ for $0 \le x \le 1$, estimate $\kappa(x)$. Formally,

$$\kappa(x) = \int_{y=0}^{x} f(y)\,dy \Big/ \frac{du}{dx}. \tag{6.5}$$

One can see from this expression that the dependence of κ on $\frac{du}{dx}$ (and hence on u itself) is nonlinear. This expression also illustrates several manifestations of ill-posedness. If $\frac{du}{dx} = 0$ within a subinterval of $0 < x < 1$, then one cannot uniquely determine $\kappa(x)$ within that subinterval. More subtle is the lack of continuous dependence on the state variable. Consider the parameterized perturbation $\delta u = \epsilon \sin(x/\epsilon^2)$. This perturbation vanishes as the parameter $\epsilon \to 0$. The corresponding perturbation to the derivative, $\frac{d}{dx}\delta u = \cos(x/\epsilon^2)/\epsilon$, becomes arbitrarily large as $\epsilon \to 0$. From (6.5) we see that the corresponding perturbation in $\kappa(x)$ can become arbitrarily large as $\epsilon \to 0$.

6.1 An Abstract Framework

What follows is a brief, nonrigorous presentation of a framework for distributed parameter identification using what is known as the output least squares formulation. For a rigorous development, see [6]. Consider the abstract distributed parameter system

$$A(q)u = f, \tag{6.6}$$

where q represents the distributed parameter to be estimated, $A(q)$ represents a parameter-dependent operator, and u represents the corresponding state variable. Equation (6.6) is called the state equation. In Example 6.2 above, q represents the diffusion coefficient κ, and $A(\kappa) = -\frac{d}{dx}(\kappa(x)\frac{d}{dx}(\cdot))$ is the diffusion operator in (6.3).

Assume that the observed data can be expressed as

$$d = Cu + \eta, \tag{6.7}$$

where η represents noise in the data. Here C is called the state-to-observation map. For example, if the state variable is measured at n discrete points x_i, then

$$[Cu]_i = u(x_i), \qquad i = 1, \ldots, n.$$

For computations to be carried out below, we will need several abstract function spaces. Let $\mathcal{Q}$ denote the parameter space, which contains the parameter q, let $\mathcal{U}$ denote the state space, which contains the state variable u, and let $\mathcal{Y}$ denote the observation space, which contains the observed data d. For simplicity, all three spaces are assumed to be Hilbert spaces.

The inverse problem is to estimate q in (6.6) given data d in (6.7). Several approaches can be taken to solve this inverse problem. One is to solve the constrained regularized least squares minimization problem

$$\min_{u\in\mathcal{U},q\in\mathcal{Q}} \frac{1}{2}||Cu-d||_{\mathcal{Y}}^2+\alpha J(q) \quad \text{subject to} \quad A(q)u=f. \tag{6.8}$$

Here $J(q)$ is a regularization functional, incorporated to impose stability or a priori information or both, and α is a positive regularization parameter. This approach is called regularized output least squares.

Assume the forward problem, solving for u in (6.6), is well-posed, and denote the solution by

$$u=A(q)^{-1}f.$$

One can then obtain from (6.8) the unconstrained regularized least squares minimization problem,

$$\min_{q\in\mathcal{Q}} T(q), \qquad T(q)=\frac{1}{2}||F(q)-d||_{\mathcal{Y}}^2+\alpha J(q), \tag{6.9}$$

where now

$$F(q)=CA(q)^{-1}f. \tag{6.10}$$

Here $F:\mathcal{Q}\to\mathcal{Y}$ is often referred to as the parameter-to-observation map.

6.1.1 Gradient Computations

In practical implementations, a distributed parameter q must be discretized before the minimization problems (6.8) or (6.9)–(6.10) can be solved. In this section, we assume that q can be represented in terms of a vector $\mathbf{q}\in\mathbb{R}^n$. Likewise, the variables u and f and the operator $A(\mathbf{q})$ are assumed to be discretized. Assume a discretization $T_n:\mathbb{R}^n\to\mathbb{R}$ of the regularized least squares functional T in (6.9).

One of the key components in any implementation of the optimization methods of Chapter 3 is the evaluation of gradient vectors. These gradients can be approximated by finite differences,

$$[\operatorname{grad} T_n(\mathbf{q})]_i \approx \frac{T_n(\mathbf{q}+\tau_i\mathbf{e}_i)-T_n(\mathbf{q})}{\tau_i}, \qquad i=1,\ldots,n, \tag{6.11}$$

where τ_i is relatively small compared to the ith component of the discretized parameter $\mathbf{q}$. The practical choice of τ_i requires a compromise between mathematical accuracy of the approximation (6.11) and computer roundoff error considerations. See [32, p. 103]. For distributed parameter identification, finite difference gradient computations tend to be quite expensive. In the context of (6.9)–(6.10), each gradient evaluation requires n evaluations of $F(\mathbf{q})=CA(\mathbf{q})^{-1}f$, and each computation of $A(\mathbf{q})^{-1}f$ entails the approximate solution of a differential equation. For instance, in Example 6.2, $A(\mathbf{q})^{-1}f$ represents the numerical solution of the diffusion equation (6.3).

6.1.2 Adjoint, or Costate, Methods

Adjoint methods for parameter identification, introduced by Chavent and Lemonier [22], can drastically reduce the cost of gradient evaluations. To illustrate, assume the operator $A(\mathbf{q}) : \mathcal{U} \to \mathcal{U}$ is linear, invertible, and Fréchet differentiable with derivative denoted by $\frac{dA}{d\mathbf{q}}$. Differentiation of the identity

$$A(\mathbf{q})A(\mathbf{q})^{-1} = I$$

yields

$$\frac{d}{d\mathbf{q}}A(\mathbf{q})^{-1} = -A(\mathbf{q})^{-1}\frac{dA}{d\mathbf{q}}A(\mathbf{q})^{-1}. \tag{6.12}$$

Consider the least squares fit-to-data term in (6.9):

$$J_{LS}(\mathbf{q}) = \frac{1}{2}||F(\mathbf{q}) - d||_{\mathcal{Y}}^2. \tag{6.13}$$

Let $\mathbf{e}_i$ denote the ith standard unit vector. Setting $r(\mathbf{q}) = F(\mathbf{q}) - d$ and using the fact that $\frac{d}{d\tau}A(\mathbf{q} + \tau\mathbf{e}_i)|_{\tau=0} = \frac{dA}{d\mathbf{q}}\mathbf{e}_i$, we obtain a representation for the components of the gradient $\mathbf{g}_{LS}$ of J_{LS}. For $i = 1, \ldots, n$,

$$\begin{aligned}
[\mathbf{g}_{LS}]_i &\stackrel{\text{def}}{=} \frac{d}{d\tau}J_{LS}(\mathbf{q} + \tau\mathbf{e}_i)\,|_{\tau=0} \\
&= \left\langle \frac{d}{d\tau}F(\mathbf{q} + \tau\mathbf{e}_i)|_{\tau=0}, r(\mathbf{q}) \right\rangle_{\mathcal{Y}} \\
&= -\left\langle CA(\mathbf{q})^{-1}\left(\frac{dA}{d\mathbf{q}}\mathbf{e}_i\right)A(\mathbf{q})^{-1}f, r(\mathbf{q}) \right\rangle_{\mathcal{Y}} \\
&= \left\langle \left(\frac{dA}{d\mathbf{q}}\mathbf{e}_i\right)A(\mathbf{q})^{-1}f, A^*(\mathbf{q})^{-1}C^*r(\mathbf{q}) \right\rangle_{\mathcal{U}}. \qquad (6.14)
\end{aligned}$$

The last equality follows by taking Hilbert space adjoints, while the preceding equality follows from (6.10) and (6.12). If we denote the solution to the state equation (6.6) by u and we let z denote the solution to the costate, or adjoint, equation

$$A^*(\mathbf{q})z = -C^*r(\mathbf{q}), \tag{6.15}$$

then we obtain

$$[\mathbf{g}_{LS}]_i = \left\langle \left(\frac{dA}{d\mathbf{q}}\mathbf{e}_i\right)u, z \right\rangle_{\mathcal{U}}, \qquad i = 1, \ldots, n. \tag{6.16}$$

In contrast to the finite difference computation (6.11), the costate gradient computation (6.16) requires only one inversion of the operator $A(\mathbf{q})$ to obtain u and one inversion of its adjoint to obtain z.

Costate methods can also be applied to time-dependent distributed parameter systems

$$u_t = -A(q)u + f.$$

However, numerical implementation can be much more problematic than in the steady-state case; see [121].

6.1.3 Hessian Computations

Newton's method for optimization requires Hessian as well as gradient computations. As was the case with the gradient, the Hessian of T_n can be computed by finite difference approximations requiring only the evaluation of T_n. See [32, p. 103] for details. For distributed parameter identification, this approach is considerably more expensive than finite difference gradient computations, since it requires roughly $n^2/2$ inversions of $A(\mathbf{q})$. To evaluate Hessian matrix-vector products, one can apply finite differences to the gradient, i.e.,

$$H(\mathbf{q})\mathbf{v} \approx \frac{\mathbf{g}(\mathbf{q}+\tau\mathbf{v}) - \mathbf{g}(\mathbf{q})}{\tau}, \tag{6.17}$$

where $\mathbf{g}(\mathbf{q})$ denotes the gradient, which can be evaluated using the adjoint approach (6.16). One can also use an adjoint approach directly to compute $H(\mathbf{q})\mathbf{v}$.

Given that one can efficiently compute Hessian matrix-vector products, one can in principle assemble the Hessian using the fact that

$$[H(\mathbf{q})]_{ij} = \langle H(\mathbf{q})\mathbf{e}_i, \mathbf{e}_j \rangle_{\mathbf{q}}, \qquad 1 \le i, j \le n. \tag{6.18}$$

In practice, both assembly and direct inversion of the Hessian can be avoided by using iterative linear solution techniques.

6.1.4 Gauss–Newton Hessian Approximation

The Hessian of the least squares functional (6.13) can be expressed as

$$H(\mathbf{q}) = H^{GN}(\mathbf{q}) + \frac{d^2F}{d\mathbf{q}^2} r(\mathbf{q}),$$

where

$$H^{GN}(\mathbf{q}) = \left(\frac{dF}{d\mathbf{q}}\right)^* \frac{dF}{d\mathbf{q}}. \tag{6.19}$$

H^{GN} is called the Gauss–Newton approximation to the Hessian. This has several computational advantages. First, it can sometimes be much easier to compute than the full Hessian, since it does not involve the second derivative term $\frac{d^2F}{d\mathbf{q}^2}$, which has a tensor representation. Moreover, the Gauss–Newton approximation is positive semidefinite, and it is positive definite if the first derivative $\frac{dF}{d\mathbf{q}}$ has full rank. This guarantees that the Gauss–Newton step is a descent direction; see Definition 3.2. This also allows the use of the conjugate gradient method in the computation of the Gauss–Newton step.

A disadvantage of the Gauss–Newton Hessian approximation is the possible loss of local quadratic convergence. If the residual $r(\mathbf{q})$ tends to zero, then one obtains local quadratic convergence. Otherwise, the rate may be linear, or the iteration may not even converge if $r(\mathbf{q})$ remains too large. See [32, p. 221] for details; see also Figure 6.3. Local convergence can be restored by replacing H^{GN} with $H^{GN} + \gamma I$, where γ is a positive parameter that may vary with iteration count. The resulting quasi-Newton method is called the Levenberg–Marquardt method. See [32, p. 227].

6.2 A One-Dimensional Example

To determine the thermal conductivity of a thin metal rod of unit length, we conduct a pair of experiments. Holding the ends of the rod at a fixed temperature, we first apply a

concentrated heat source at location $x = 1/3$, and we measure the resulting steady-state temperature distribution. We then move the heat source to location $x = 2/3$ and again measure the distribution. We modeled the temperature distributions using the steady-state diffusion equation in Example 6.2. The diffusion coefficient $\kappa(x)$ represents the thermal conductivity. The boundary conditions were taken to be homogeneous Dirichlet (see (6.4)) with $u_L = u_R = 0$. The forcing functions are

$$f^1(x) = \delta(x - 1/3) \quad \text{and} \quad f^2(x) = \delta(x - 2/3),$$

where the superscript denotes the experiment and $\delta(\cdot)$ denotes the Dirac delta. The data are modeled as

$$d_i^e = u^e(x_i) + \eta_i^e, \qquad i = 1, \ldots, n-1, \quad e = 1, 2. \tag{6.20}$$

Here d_i^e represents the temperature observation for experiment e taken at the node point $x_i = ih$, $h = 1/n$, and η_i^e represents measurement error.

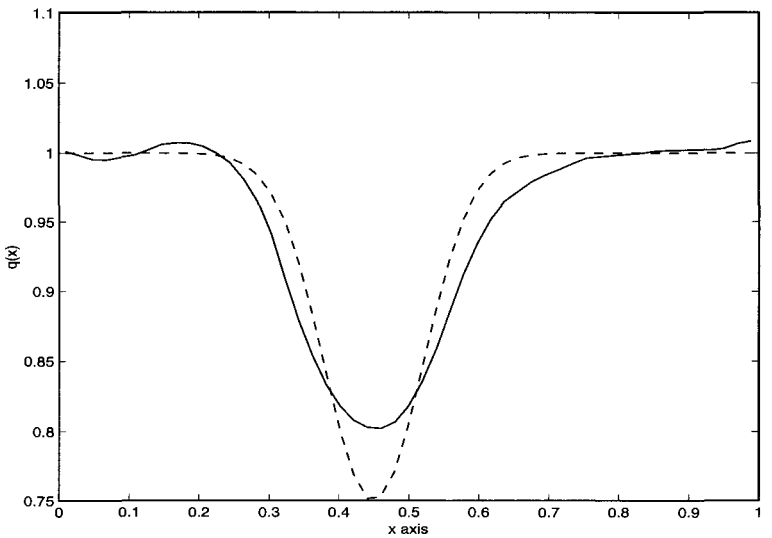


Figure 6.1. *True and estimated distributed parameters in a one-dimensional steady-state diffusion equation. The dashed line denotes the true diffusion coefficient; the solid line denotes the estimated diffusion coefficient obtained using a regularized least squares approach.*

We performed a simulation in which the true diffusion coefficient $\kappa(x)$ is represented by the dashed line in Figure 6.1. The data (6.20) are shown in Figure 6.2. To obtain the $u^e(x_i)$, we solved a pair of equations of the form (6.3) using a standard Galerkin finite element discretization [5]. Galerkin's method for (6.3) yields an approximation of the form

$$u(x) = \sum_{i=1}^{n-1} u_i \, \phi_i(x), \tag{6.21}$$

where the coefficient vector $\mathbf{u} = (u_1, \ldots, u_{n-1})$ solves the linear system $A\mathbf{u} = \mathbf{f}$, with

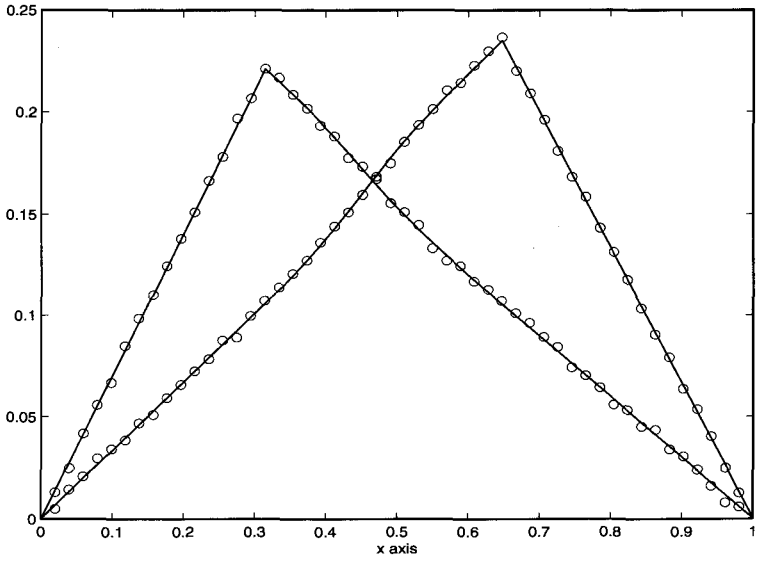


Figure 6.2. *Observed data used for distributed parameter identification in a one-dimensional steady-state diffusion equation. The left curve represents the solution corresponding to a point source at $x = 1/3$. The right curve represents the solution corresponding to a point source at $x = 2/3$. Circles represent observed data.*

stiffness matrix A and load vector $\mathbf{f}$ having components

$$[A]_{ij} = \int_0^1 \kappa(x) \frac{d\phi_j}{dx} \frac{d\phi_i}{dx}\, dx, \qquad [\mathbf{f}]_i = \int_0^1 f(x)\, \phi_i(x)\, dx. \tag{6.22}$$

Corresponding to the point source at $x = 1/3$ in the first experiment, we obtain a load vector $\mathbf{f}^1$ with components

$$[\mathbf{f}^1]_i = \phi_i(1/3), \qquad i = 1, \ldots, n-1.$$

To obtain the components of $\mathbf{f}^2$, we evaluate the basis functions ϕ_i at $x = 2/3$ rather than $x = 1/3$.

The basis functions ϕ_i were taken to be piecewise linear "hat" functions, characterized by $\phi_i(x_j) = \delta_{ij}$, where the x_j's are node points. Midpoint quadrature was used to evaluate the stiffness matrix, yielding

$$A(\kappa) = \frac{1}{h} \begin{bmatrix} \kappa_1 + \kappa_2 & -\kappa_2 & 0 & \cdots & 0 \\ -\kappa_2 & \kappa_2 + \kappa_3 & -\kappa_3 & 0 & \ddots \\ 0 & -\kappa_3 & \ddots & \ddots & \ddots \\ \vdots & \ddots & \ddots & \ddots & -\kappa_{n-1} \\ 0 & \cdots & 0 & -\kappa_{n-1} & \kappa_{n-1} + \kappa_n \end{bmatrix}, \tag{6.23}$$

where $\kappa_i = \kappa(x_i^{mid})$ with $x_i^{mid} = (x_{i-1} + x_i)/2$ for $i = 1, \ldots, n$. This matrix can also be derived (up to a factor of h) using finite difference techniques. See Exercise 6.5. $A(\kappa)$

is symmetric and can be shown to be positive definite, provided each $\kappa_i > 0$. Since it is tridiagonal, the coefficients of the simulated temperature profiles,

$$\mathbf{u}^e = A(\kappa)^{-1}\mathbf{f}^e, \qquad e = 1, 2, \tag{6.24}$$

can be computed at $\mathcal{O}(n)$ cost.

Our goal is to estimate κ given the discretized data model (6.20), (6.24). To do this, we minimize the regularized least squares functional

$$T(\kappa) = \frac{1}{2}\sum_{e=1}^{2} ||A(\kappa)^{-1}\mathbf{f}^e - \mathbf{d}^e||^2 + \alpha J_{reg}(\kappa). \tag{6.25}$$

The regularization term is taken to be a discretization of the H^1 regularization functional (see (2.47)),

$$\begin{aligned} J_{reg}(\kappa) &= \frac{1}{2}\sum_{i=1}^{n-1}(\kappa_{i+1} - \kappa_i)^2\, h \\ &= \frac{\alpha}{2}\kappa^T L \kappa, \end{aligned} \tag{6.26}$$

where L is an $n \times n$ one-dimensional discrete Laplacian matrix with Neumann boundary conditions:

$$L = \frac{1}{h}\begin{bmatrix} 1 & -1 & 0 & \dots & 0 & 0 \\ -1 & 2 & -1 & 0 & \ddots & 0 \\ 0 & -1 & 2 & -1 & 0 & \vdots \\ \vdots & & \ddots & \ddots & \ddots & \\ 0 & \ddots & 0 & -1 & 2 & -1 \\ 0 & 0 & \dots & 0 & -1 & 1 \end{bmatrix}. \tag{6.27}$$

Note that L is symmetric and positive semidefinite, and its null space is spanned by the vector $\mathbf{1} = (1, \dots, 1)$.

The gradient of the regularization term $\alpha J_{reg}(\kappa)$ in (6.25) is $\mathbf{g}_{reg} = \alpha L\kappa$, and its Hessian is αL. To obtain gradients of the least squares fit-to-data terms in (6.25), we use the costate method. From equation (6.16), we need to compute

$$\frac{dA}{d\kappa}\mathbf{e}_i \stackrel{\text{def}}{=} \frac{d}{d\tau}A(\kappa + \tau\mathbf{e}_i)|_{\tau=0}. \tag{6.28}$$

One can decompose (6.23) into

$$A(\kappa) = \frac{1}{h}B_x^T\ \text{diag}(\kappa)\ B_x, \tag{6.29}$$

where B_x is the $n \times (n-1)$ bidiagonal matrix

$$B_x = \begin{bmatrix} 1 & 0 & \cdots & 0 & 0 \\ -1 & 1 & 0 & \cdots & 0 \\ 0 & -1 & 1 & 0 & \ddots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \ddots & 0 & -1 & 1 \\ 0 & 0 & \cdots & 0 & -1 \end{bmatrix}, \tag{6.30}$$

obtained from the discretization of the first derivative operator $\frac{d}{dx}$ with homogeneous Dirichlet boundary conditions. From (6.16) and (6.28)–(6.29) we obtain the least squares gradients corresponding to each experiment $e = 1, 2$,

$$\begin{aligned}
[\mathbf{g}_{LS}^e]_i &= \left\langle \left(\frac{dA}{d\kappa} \mathbf{e}_i \right) \mathbf{u}^e, \mathbf{z}^e \right\rangle_{n-1} \\
&= \frac{1}{h} \langle B_x^T \operatorname{diag}(\mathbf{e}_i) B_x \mathbf{u}^e, \mathbf{z}^e \rangle_{n-1} \\
&= \frac{1}{h} \langle \operatorname{diag}(\mathbf{e}_i) B_x \mathbf{u}^e, B_x \mathbf{z}^e \rangle_n \\
&= \frac{1}{h} [B_x \mathbf{u}^e]_i [B_x \mathbf{z}^e]_i, \qquad i = 1, \ldots, n. \qquad (6.31)
\end{aligned}$$

Here $\mathbf{u}^e$ and $\mathbf{z}^e$ denote the solutions to the discrete state and costate equations,

$$A(\kappa)\mathbf{u} = \mathbf{f}^e, \quad A^T(\kappa)\mathbf{z} = \mathbf{r}^e, \qquad e = 1, 2,$$

and $\langle \cdot, \cdot \rangle_n$ denotes the usual Euclidean inner product on $\mathbb{R}^n$. The gradient of the $T(\kappa)$ in (6.25) is then given by $\mathbf{g} = \mathbf{g}_{LS}^1 + \mathbf{g}_{LS}^2 + \mathbf{g}_{reg}$.

We used both a Newton iteration and a Gauss–Newton iteration to minimize the functional (6.25). For Newton's method, Hessian matrix-vector products for the least squares terms were computed using (6.17). For the Gauss–Newton method, the adjoint formula in Exercise 6.4 was used to compute the corresponding (approximate) Hessian matrix-vector products. Equation (6.18) was then used to assemble the least squares Hessians at each iteration. The Hessian of the regularization term is given by αL and was added on. The estimated parameter is shown in Figure 6.1, and numerical performance results are shown in Figure 6.3. By the iterative solution error at iteration ν we mean the difference between the minimizer of (6.25) and the approximate minimizer at iteration ν, obtained using a particular numerical method. Note that the Gauss–Newton method converges at a linear rate in this example. While the quadratic convergence rate for Newton's method is asymptotically much faster, Newton's method requires five iterations before it produces a smaller iterative solution error than Gauss–Newton.

For problems in more than one space dimension, the costate gradient formulas are analogous to (6.31). However, because of large system sizes, the computation of solutions to the state and costate equations can be much more involved; see [115].

6.3 A Convergence Result

In this section we apply some of the analytical tools of Chapter 2 to obtain a convergence result that pertains to parameter identification. Our goal here is to sketch some key ideas. For a more detailed analysis, see [35, Chapter 10]. As in section 6.1, we consider minimization of the Tikhonov functional

$$T_\alpha(q) = ||F(q) - d||_{\mathcal{Y}}^2 + \alpha J(q), \qquad \alpha > 0.$$

We assume $\mathcal{Q}$ and $\mathcal{Y}$ are Hilbert spaces, $d \in \mathcal{Y}$ is fixed, and the mapping $F : \mathcal{Q} \to \mathcal{Y}$ is weakly continuous. This means that $||F(q_n) - F(q)||_{\mathcal{Y}} \to 0$ whenever q_n converges weakly to q in $\mathcal{Q}$. Then the mapping $q \mapsto ||F(q) - d||_{\mathcal{Y}}^2$ is weakly lower semicontinuous. See Exercise 6.9. We also assume that the regularization functional $J : \mathcal{Q} \to \mathbb{R}$ is weakly lower semicontinuous and coercive. As a consequence of Theorem 2.30, T_α has a minimizer over any closed convex subset of $\mathcal{Q}$.

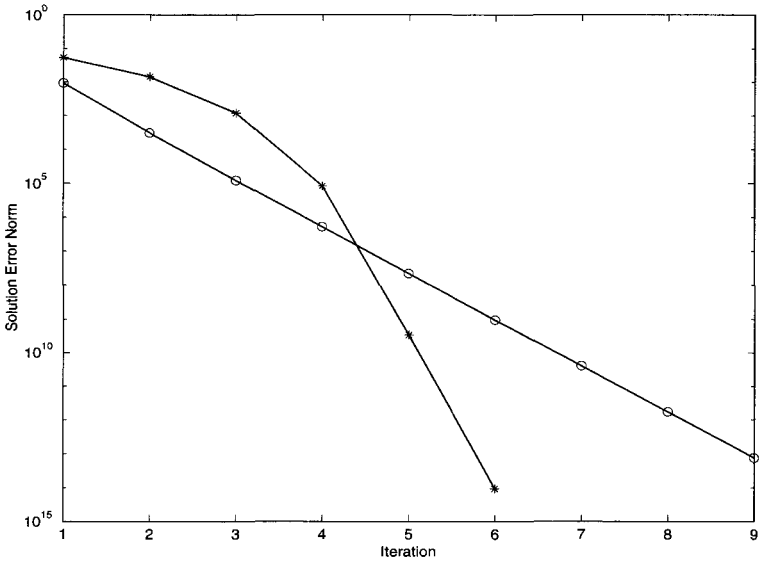


Figure 6.3. *Numerical performance of unconstrained optimization methods for parameter identification using penalized least squares. Asterisks denote the norm of the iterative solution error for Newton's method, and circles represent the iterative solution error for the Gauss–Newton method.*

Now consider a sequence of idealized experiments in which we collect noisy data

$$d_n = F(q_{\text{true}}) + \eta_n, \qquad n = 1, 2, \ldots,$$

and we can control the noise level so that it vanishes in the limit,

$$\delta_n \stackrel{\text{def}}{=} ||\eta_n|| \to 0 \quad \text{as} \quad n \to \infty. \tag{6.32}$$

For each experiment we pick a regularization parameter $\alpha_n > 0$ and compute a minimizer of the corresponding Tikhonov functional,

$$q_n \stackrel{\text{def}}{=} \arg\min_q T_n(q), \qquad T_n(q) \stackrel{\text{def}}{=} ||F(q) - d_n||_{\mathcal{Y}}^2 + \alpha_n J(q).$$

Our goal is to establish conditions under which $q_n \to q_{\text{true}}$, where q_{true} represents the underlying true parameter. We make the following assumptions on the regularization parameters α_n and the data error:

$$\alpha_n \to 0 \quad \text{as} \quad n \to \infty, \tag{6.33}$$

$$\frac{\delta_n^2}{\alpha_n} \to 0 \quad \text{as} \quad n \to \infty. \tag{6.34}$$

This means that the regularization parameter goes to zero, but at a slower rate than the square of the data error norm.

Lemma 6.3. *Let assumptions* (6.33)–(6.34) *hold. Then*

$$||F(q_n) - d_n|| \to 0 \quad \text{as} \quad n \to \infty, \tag{6.35}$$

and there exists a constant B for which

$$J(q_n) \leq B, \quad n = 1, 2, \ldots. \tag{6.36}$$

Proof. Note that

$$\begin{aligned} ||F_n(q_n) - d_n||^2 &\leq T_n(q_n) \\ &\leq T_n(q_{\text{true}}) \\ &= ||F_n(q_{\text{true}}) - d_n||^2 + \alpha_n J(q_{\text{true}}). \end{aligned}$$

The second equality follows from the fact that q_n minimizes T_n. Dividing by α_n and noting that $\delta_n = ||F(q_{\text{true}}) - d_n||$, we obtain

$$\frac{||F(q_n) - d_n||^2}{\alpha_n} \leq \frac{\delta_n^2}{\alpha_n} + J(q_{\text{true}}).$$

But the first term on the right-hand side vanishes as $n \to 0$ by assumption (6.34). Thus the left-hand side is uniformly bounded. But then by assumption (6.33), the numerator of the left-hand side must vanish as $n \to 0$, yielding (6.35). Similarly,

$$J(q_n) \leq \frac{T_n(q_n)}{\alpha_n} \leq \frac{T_n(q_{\text{true}})}{\alpha_n} = \frac{\delta_n^2}{\alpha_n} + J(q_{\text{true}}).$$

Equation (6.36) follows from (6.34). □

We make an additional local uniqueness assumption,

$$F(q) \neq F(q_{\text{true}}) \quad \text{whenever} \quad q \neq q_{\text{true}}. \tag{6.37}$$

Theorem 6.4. *Suppose that J is weakly lower semicontinuous and coercive, F is weakly continuous, and assumptions* (6.33)–(6.34) *and* (6.37) *hold. Then q_n has a subsequence that converges weakly to q_{true}.*

Proof. By (6.36) and the coercivity of J, the q_n's are bounded. Consequently, there is a subsequence q_{n_j} that converges weakly to some $q^* \in \mathcal{Q}$. By the triangle inequality,

$$||F(q_{n_j}) - F(q_{\text{true}})|| \leq ||F(q_{n_j}) - d_{n_j}|| + \delta_{n_j}.$$

Taking the limit as $j \to \infty$ and using (6.35) and (6.32), we obtain $F(q^*) = F(q_{\text{true}})$. Then by (6.37), $q_{\text{true}} = q^*$. □

Exercises

6.1. Show that if $c = 0$ and $f(t) = f_0$ is known, then one can uniquely determine both k and m in equation (6.1) from observations of the displacement $x(t)$.

6.2. For the data presented in Figure 6.2, numerically implement the naive parameter estimation scheme (6.5). Use standard forward differences to approximate $\frac{du}{dx}$ and use trapezoidal quadrature to approximate the integral term.

6.3. Suppose $f = f(q)$ in equation (6.6). Use adjoint computations to obtain a formula for the components of the gradient of (6.13) in this case.

6.4. Use adjoint computations to obtain the following formula for the components of the Gauss–Newton Hessian approximation:

$$[H^{GN}]_{ij} = \left\langle \frac{dA}{d\mathbf{q}} \mathbf{e}_j \, u, \, A(\mathbf{q})^{-2} \frac{dA}{d\mathbf{q}} \mathbf{e}_i \, u \right\rangle_{\mathcal{U}}.$$

6.5. Use finite difference discretization of the differential operator $-\frac{d}{dx}(\kappa(x)\frac{d}{dx})$ to obtain the stiffness matrix in (6.23) multiplied by a factor of $1/h$.

6.6. Verify the matrix decomposition (6.29).

6.7. Confirm the string of equalities preceding (6.31).

6.8. Apply the BFGS method to solve the one-dimensional parameter identification problem in section 6.2.

6.9. Prove that if $F : \mathcal{Q} \to \mathcal{Y}$ is weakly continuous, then the mapping $q \mapsto ||F(q) - d||^2$ is weakly lower semicontinuous.

Chapter 7

Regularization Parameter Selection Methods

Figures 1.4–1.6 in Chapter 1 illustrate the importance of selecting the regularization parameter properly. With too little regularization, reconstructions have highly oscillatory artifacts due to noise amplification. With too much regularization, the reconstructions are too smooth. Ideally, one should select a regularization parameter α so that the corresponding regularized solution f_α minimizes some indicator of solution fidelity, e.g., some measure of the size of the solution error,

$$e_\alpha \stackrel{\text{def}}{=} f_\alpha - f_{\text{true}}. \tag{7.1}$$

In statistics this quantity is known as the estimation error. Obviously, computations involving e_α are not practical, since the true solution f_{true} is unknown. Instead we need error indicators that depend on available information.

Given the data model

$$\mathbf{d} = K\mathbf{f}_{\text{true}} + \boldsymbol{\eta}, \tag{7.2}$$

an alternative error indicator is the predictive error,

$$\mathbf{p}_\alpha \stackrel{\text{def}}{=} Ke_\alpha = Kf_\alpha - Kf_{\text{true}}. \tag{7.3}$$

The predictive error is also not directly computable, but it can sometimes be accurately estimated. Many of the techniques presented in this chapter are computable quantities related to predictive error.

We emphasize methods for selecting regularization parameters that make use of statistical information about the noise in the data. To do this we assume that (7.2) is a semistochastic discrete linear data model with additive noise. This means that the noise $\boldsymbol{\eta}$ is assumed to be a realization of a random n-vector, and K is an $n \times m$ matrix. $f_{\text{true}} \in \mathbb{R}^m$, which represents the true solution, and the matrix K are assumed to be deterministic. Hence the expression *semistochastic* to describe this model. (The model is fully stochastic if both $\boldsymbol{\eta}$ and f_{true} are random vectors [38].) In our analysis we allow f_{true} to lie in an infinite-dimensional Hilbert space, and we allow K to map this Hilbert space into $\mathbb{R}^n$. The model (7.2) then becomes semidiscrete.

For additional information on practical aspects of regularization parameter selection, see [109, 52], [123, Chapter 4], and the book by Hansen [58] and the references therein.

We now introduce some tools with which to derive and analyze regularization parameter selection methods. Unless otherwise indicated, $\|\cdot\|$ denotes the standard Euclidean norm.

Definition 7.1. A random vector $\boldsymbol{\eta} = (\eta_1, \ldots, \eta_n)$ is a discrete white noise vector provided that $E(\boldsymbol{\eta}) = \mathbf{0}$ and $\text{cov}(\boldsymbol{\eta}) = \sigma^2 I$, i.e., for $1 \leq i, j \leq n$,

$$E(\eta_i) = 0, \qquad E(\eta_i \eta_j) = \sigma^2 \delta_{ij}. \tag{7.4}$$

The scalar quantity σ^2 is called the variance of the white noise.

An example of a discrete white noise vector is a Gaussian random vector with zero mean and covariance equal to $\sigma^2 I$; see Example 4.13.

Recall that the trace of a square matrix is the sum of the diagonal components; see Definition 4.28.

Lemma 7.2 (Trace Lemma). *Let $f \in \mathcal{H}$, where $\mathcal{H}$ is a deterministic, real Hilbert space, let $\boldsymbol{\eta}$ be a discrete white noise vector, and let $B : \mathbb{R}^n \to \mathcal{H}$ be a (bounded) linear operator. Then*

$$E(\|f + B\boldsymbol{\eta}\|_{\mathcal{H}}^2) = \|f\|_{\mathcal{H}}^2 + \sigma^2 \text{trace}(B^* B). \tag{7.5}$$

Proof.

$$\begin{aligned} E(\|f + B\boldsymbol{\eta}\|_{\mathcal{H}}^2) &= E(\|f\|_{\mathcal{H}}^2) + 2E(\langle f, B\boldsymbol{\eta}\rangle_{\mathcal{H}}) + E(\langle B\boldsymbol{\eta}, B\boldsymbol{\eta}\rangle_{\mathcal{H}}) \\ &= \|f\|_{\mathcal{H}}^2 + 2E[(B^* f)^T \boldsymbol{\eta}] + E[\boldsymbol{\eta}^T B^* B \boldsymbol{\eta}] \\ &= \|f\|_{\mathcal{H}}^2 + 2\sum_{i=1}^{n} [B^* f]_i E(\eta_i) + \sum_{i=1}^{n}\sum_{j=1}^{n} [B^* B]_{ij} E(\eta_i \eta_j). \end{aligned}$$

Now use white noise properties (7.4). □

7.1 The Unbiased Predictive Risk Estimator Method

The unbiased predictive risk estimator (UPRE) method is also known as the C_L method. It was originally developed by Mallow [84] for model selection in linear regression, and it has since been adapted for the solution of inverse problems. The UPRE method is based on a statistical estimator of the mean squared norm of predictive error (7.3):

$$\frac{1}{n}\|\mathbf{p}_\alpha\|^2 = \frac{1}{n}\|K\mathbf{f}_\alpha - K\mathbf{f}_{\text{true}}\|^2. \tag{7.6}$$

This quantity is called the predictive risk.

To derive the method, we assume that the noise $\boldsymbol{\eta}$ in (7.2) is a random vector (rather than a realization of a random vector, as assumed previously). Hence, related quantities like $\mathbf{d}$, $\mathbf{f}_\alpha$, and $\mathbf{e}_\alpha$ are also random vectors, and the predictive risk (7.6) is a random variable. In addition, we assume that the regularized solution depends linearly on the data,

$$f_\alpha = R_\alpha \mathbf{d}, \tag{7.7}$$

where R_α is the $n \times m$ regularization matrix. An example is standard Tikhonov regularization, in which $R_\alpha = (K^T K + \alpha I)^{-1} K^T$; see section 1.2.1.

We define the $n \times n$ influence matrix to be

$$A_\alpha = K R_\alpha, \tag{7.8}$$

and we assume that A_α is symmetric; see Exercise 7.1. The predictive error can then be expressed as

$$\mathbf{p}_\alpha = (A_\alpha - I)K\mathbf{f}_{\text{true}} + A_\alpha \boldsymbol{\eta}. \tag{7.9}$$

Note that the first term on the right-hand side is deterministic, while the second is stochastic. We make the additional assumption that $\boldsymbol{\eta}$ is a discrete white noise vector; see Definition 7.1. From the Trace Lemma 7.2 and the assumption that A_α is symmetric, we obtain the following expression for the expected value of the predictive risk:

$$E\left(\frac{1}{n}||\mathbf{p}_\alpha||^2\right) = \frac{1}{n}||(A_\alpha - I)K\mathbf{f}_{\text{true}}||^2 + \frac{\sigma^2}{n}\text{trace}(A_\alpha^2). \tag{7.10}$$

The regularized residual is defined to be

$$\begin{aligned} \mathbf{r}_\alpha &\stackrel{\text{def}}{=} K\mathbf{f}_\alpha - \mathbf{d} = (A_\alpha - I)\mathbf{d} \\ &= (A_\alpha - I)K\mathbf{f}_{\text{true}} + (A_\alpha - I)\boldsymbol{\eta}. \end{aligned} \tag{7.11}$$

The Trace Lemma 7.2 gives

$$\begin{aligned} E\left(\frac{1}{n}||\mathbf{r}_\alpha||^2\right) &= \frac{1}{n}||(A_\alpha - I)K\mathbf{f}_{\text{true}}||^2 + \frac{\sigma^2}{n}\text{trace}(A_\alpha^2) \\ &\quad - \frac{2\sigma^2}{n}\text{trace}(A_\alpha) + \sigma^2. \end{aligned} \tag{7.12}$$

Here we used the symmetry of A_α together with the linearity of the trace operator (4.26). Comparing (7.10) and (7.12), we obtain

$$E\left(\frac{1}{n}||\mathbf{p}_\alpha||^2\right) = E\left(\frac{1}{n}||\mathbf{r}_\alpha||^2\right) + \frac{2\sigma^2}{n}\text{trace}(A_\alpha) - \sigma^2. \tag{7.13}$$

We define the UPRE to be

$$U(\alpha) = \frac{1}{n}||\mathbf{r}_\alpha||^2 + \frac{2\sigma^2}{n}\text{trace}(A_\alpha) - \sigma^2. \tag{7.14}$$

From (7.13), the expected value of $U(\alpha)$ is equal to the expected value of the predictive risk. In other words, $U(\alpha)$ is an unbiased estimator for the expected value of the predictive risk. The UPRE regularization parameter selection method is to pick

$$\alpha_U = \arg\min_\alpha U(\alpha). \tag{7.15}$$

Remark 7.3. Although they have the same expected values, given a particular noise realization, it does not necessarily follow that $U(\alpha)$ and the predictive risk $||\mathbf{p}_\alpha||^2/n$ take on the same values as α varies. Moreover, they need not have the same minimizers. However, the minimizers for the respective functions $U(\alpha)$ and $||\mathbf{p}_\alpha||^2/n$ should be close provided that the variability about the respective expected values is small and the respective functions are not too flat at their minimizers. For an analysis that indicates that this is the case, see Lukas [82]. Numerical results, including those presented in Figures 7.2 and 7.3, also indicate that these minimizers are close.

It is also not clear that a value of α which minimizes the predictive risk will yield a small value for a quantity like $||\mathbf{e}_\alpha||$, derived from the estimation error. Again, numerical results suggest that this is the case. We revisit this issue later in section 7.6.

7.1.1 Implementation of the UPRE Method

In principle, our goal is to find the global minimizer for $U(\alpha)$ in (7.14). The following observations are relevant:

(i) $U(\alpha)$ may have several local minimizers. Hence a local optimization method like Newton's method need not produce the desired global minimizer.

(ii) Certain regularization methods—e.g., truncated SVD—may yield functions $U(\alpha)$ that have jump discontinuities. In this case, a minimization method that requires derivatives of $U(\alpha)$ is not appropriate.

(iii) The regularized solution typically changes little with small changes in α. Hence, one needs to compute only a relatively rough approximation to the minimizer of $U(\alpha)$. In addition, one often can determine a priori that the regularization parameter must lie within a certain given range.

As a consequence of these observations, the following strategy is often effective: First, find a range $[\alpha_{\min}, \alpha_{\max}]$ for values of the regularization parameter, e.g., by analyzing the spectrum of K and the noise statistics or by numerical experimentation. Second, select a grid $\alpha_{\min} \le \alpha_1 < \alpha_2 < \cdots \le \alpha_{\max}$. Space the α_i's so that $\Delta \log \alpha_i \stackrel{\text{def}}{=} \log \alpha_{i+1} - \log \alpha_i$ is constant. The number of grid points α_i can often be quite small, e.g., no more than 20. Third, evaluate U at each α_i and select the smallest value. Of course, such a strategy is not practical for a regularization method like Landweber iteration. With iterative methods, compute $U(\nu)$ at every iteration ν and terminate the iteration when $U(\nu)$ appears to grow significantly.

If an SVD of K is available, the evaluation of $U(\alpha)$ is straightforward. Assume that $K = U\text{diag}(s_i)V^T$ and R_α has a linear filtering representation (see section 7.6):

$$R_\alpha \mathbf{d} = \sum_{s_i>0} w_\alpha(s_i^2) \frac{\hat{d}_i}{s_i} \mathbf{v}_i. \tag{7.16}$$

Here the left and right singular vectors $\mathbf{u}_i$ and $\mathbf{v}_i$ make up the columns of U and V, respectively, and $\hat{d}_i = \mathbf{u}_i^T \mathbf{d}$. Then

$$\text{trace}(A_\alpha) = \sum_{s_i>0} w_\alpha(s_i^2) \tag{7.17}$$

and

$$\frac{1}{n}||\mathbf{r}_\alpha||^2 = \frac{1}{n}\sum_{s_i>0}[w_\alpha(s_i^2) - 1]^2 \hat{d}_i^2 + \frac{1}{n}\sum_{s_i=0}\hat{d}_i^2. \tag{7.18}$$

For each fixed value of α, one can then evaluate

$$U(\alpha) = \frac{1}{n}\sum_{s_i>0}[w_\alpha(s_i^2) - 1]^2 \hat{d}_i^2 + 2\frac{\sigma}{n}\sum_{s_i>0} w_\alpha(s_i^2) + \frac{1}{n}\sum_{s_i=0}\hat{d}_i^2 - \sigma^2. \tag{7.19}$$

If the $\hat{d}_i$'s are precomputed, the cost of evaluating $U(\alpha)$ is only $\mathcal{O}(n)$.

Very similar computations can be carried out if K has (block) circulant structure; see sections 5.2.3 and 5.2.5. Suppose $K = \textbf{toeplitz}(\mathbf{t})$, and let $\hat{\mathbf{t}}$ denote the discrete Fourier transform of $\mathbf{t}$. The singular values of K are the magnitudes of the components of $\hat{\mathbf{t}}$, i.e., $s_i = |\hat{t}|_i$. Consequently, one can replace the s_i^2's in (7.19) by $|\hat{\mathbf{t}}|_i^2$'s. In place of the squared

singular components $\hat{d}_i^2$, one must then take the square of the magnitudes of the components of the DFT of $\mathbf{d}$. If K is block circulant with circulant blocks, replace one-dimensional DFTs with two-dimensional DFTs.

When the matrix K is large and an SVD is not available, the exact evaluation of the term trace(A_α) in (7.14) can be quite expensive. Several approaches may be taken to approximate this trace. Perhaps the most obvious is to approximate the eigenvalues of the influence matrix A_α using an iterative technique like the Lanczos method [46]. See [69].

7.1.2 Randomized Trace Estimation

Another approach to approximating the trace of symmetric matrices like A_α, known as randomization [44, 61], is based on the proof of the Trace Lemma 7.2. In particular, if $\mathbf{W}$ is a white noise vector with zero mean and unit variance ($\sigma^2 = 1$ in (7.4)), then an unbiased estimator for trace(A_α) is the random variable

$$t(\alpha) = \mathbf{W}^T A_\alpha \mathbf{W}. \tag{7.20}$$

This yields the following practical algorithm for estimating trace(A_α): (i) generate M independent realizations $\mathbf{w}_i$ of $\mathbf{W}$, (ii) compute $t_i(\alpha) = \mathbf{w}_i^T A_\alpha \mathbf{w}_i$ for each i, and then (iii) take the sample mean $\bar{t}(\alpha) = \sum_{i=1}^{M} t_i(\alpha)/M$ to be the trace estimate. The accuracy of this estimate depends on the variability of the $t_i(\alpha)$'s, and this variability can be quantified in terms of the variance of $t(\alpha)$. Hutchinson [61] showed that this variance is minimized by taking $\mathbf{W}$ to be a random vector whose components are independent and take on values $+1$ and -1, each with probability $1/2$. To simulate such a random vector, one can generate a realization $\mathbf{v}$ of a random vector $\mathbf{V}$ whose components are independent and uniformly distributed on the interval $[0, 1]$. Then take $\mathbf{w}$ to have components

$$w_i = \begin{cases} +1 & \text{if } v_i \geq 1/2, \\ -1 & \text{if } v_i < 1/2. \end{cases} \tag{7.21}$$

Utilizing M independent realizations of $\mathbf{W}$ reduces the variance by a factor of M^{-1}.

7.1.3 A Numerical Illustration of Trace Estimation

We applied Hutchinson's randomization scheme to estimate the trace of the influence matrix A_α that arises when standard Tikhonov regularization is applied to the deblurring problem of section 5.1.1. In this case,

$$A_\alpha = K^T(K^T K + \alpha I)^{-1} K,$$

and we took $K = \mathbf{bccb}(t)$, the block circulant–circulant block matrix generated by the discrete point spread function t; see section 5.2.5. The trace can be expressed in terms of the components $\hat{t}_i$ of the discrete Fourier transform $\hat{t}$ of t,

$$\text{trace}(A_\alpha) = \sum_{\hat{t}_i \neq 0} \frac{|\hat{t}_i|^2}{|\hat{t}_i|^2 + \alpha}.$$

We used a single realization $\mathbf{w}$ of $\mathbf{W}$, obtained via formula (7.21), to compute the trace estimator

$$\bar{t}(\alpha) = \mathbf{w}^T A_\alpha \mathbf{w}.$$

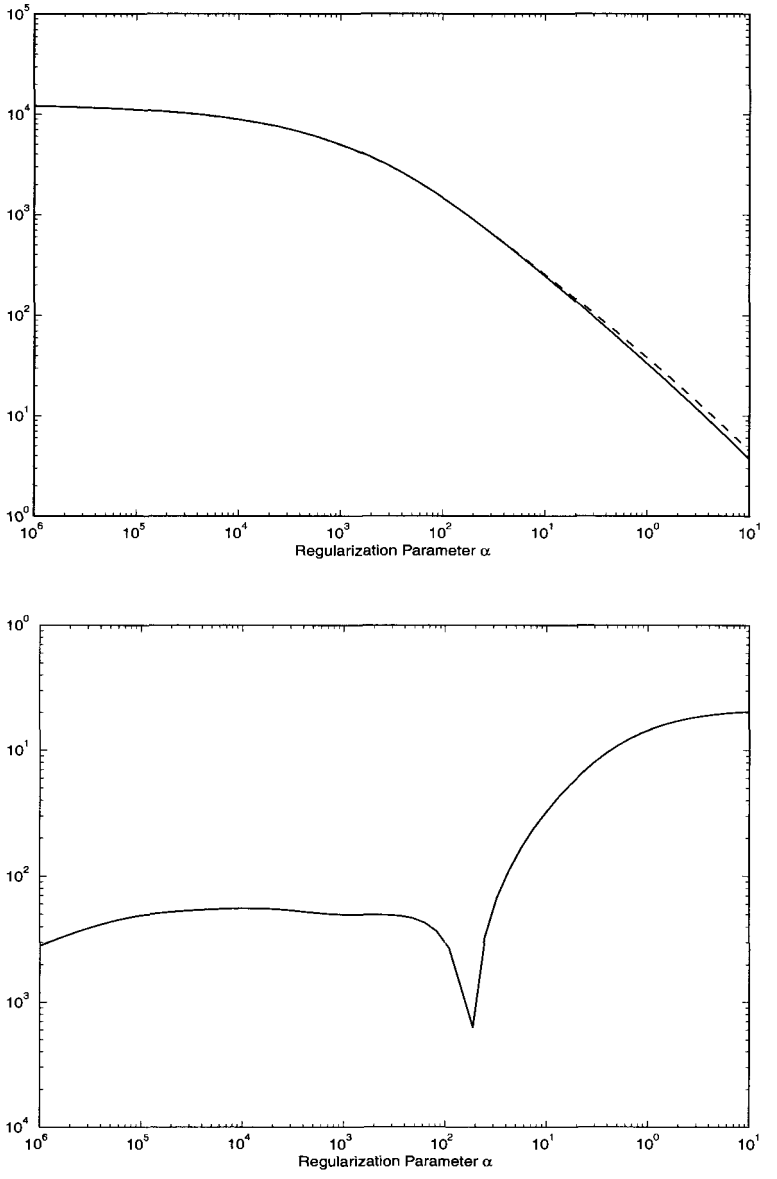


Figure 7.1. *Randomized estimate of the trace of the influence matrix for Tikhonov regularization applied to an image deblurring problem. In the top graph, the solid line denotes* $\text{trace}(A_\alpha)$, *and the dashed line represents a randomized estimate* $\bar{t}(\alpha)$. *The bottom graph shows the relative error,* $|\text{trace}(A_\alpha) - \bar{t}(\alpha)|/\text{trace}(A_\alpha)$.

Evaluation of the matrix-vector product $A_\alpha \mathbf{w}$ was carried out using Fourier transforms; see Algorithm 5.3.1. Results are presented in Figure 7.1. Note that for values of the regularization parameter less than about 10^{-2}, the relative error for the trace estimate is less than 1%.

7.1.4 Nonlinear Variants of UPRE

The UPRE method can be extended to handle certain nonlinear operators K and more general regularization techniques. Note that the residual $\mathbf{r}_\alpha = K(\mathbf{f}_\alpha) - \mathbf{d}$ in (7.14) is well defined even if K is nonlinear. For linear Tikhonov regularization with a penalty operator L, the influence matrix takes the form

$$A_\alpha = K^*(K^*K + \alpha L)^{-1}K. \tag{7.22}$$

If K is nonlinear, one can replace the K in (7.22) by the Fréchet derivative $K'(f_\alpha)$. Similarly, if a nonquadratic penalty functional is used, the L in (7.22) can be replaced by the Hessian of the penalty functional or some other linearization.

7.2 Generalized Cross Validation

The method of generalized cross validation (GCV) [122, 123], is an alternative to UPRE that does not require prior knowledge of the variance σ^2 of the white noise η in the model (7.2). To apply this method, one selects the regularization parameter α that minimizes the GCV functional:

$$GCV(\alpha) = \frac{\frac{1}{n}||\mathbf{r}_\alpha||^2}{[\frac{1}{n}\text{trace}(I - A_\alpha)]^2}, \tag{7.23}$$

where $\mathbf{r}_\alpha = K\mathbf{f}_\alpha - \mathbf{d} = (A_\alpha - I)\mathbf{d}$ is the regularized residual. Like the UPRE, the GCV function is an estimator for the predictive risk (7.6). For an analysis of this method, see section 7.6.5.

The implementation of the GCV method is very similar to UPRE implementation, described in section 7.1.1. For example, if one has an SVD for K and one precomputes the expansion coefficients $\hat{d}_i = \mathbf{u}_i^T\mathbf{d}$ for the data, the evaluation of the GCV functional can be carried out with $\mathcal{O}(n)$ cost via

$$GCV(\alpha) = \frac{\frac{1}{n}\sum_{s_i>0}[w_\alpha(s_i^2) - 1]^2\,\hat{d}_i^2 + \frac{1}{n}\sum_{s_i=0}\hat{d}_i^2}{\left[1 - \frac{1}{n}\sum_{s_i>0} w_\alpha(s_i^2)\right]^2}. \tag{7.24}$$

Remark 7.4. The expression $\text{trace}(I - A_\alpha)$ in the denominator of (7.23) can be viewed as a measure of the degrees of freedom in the regularized approximation. In the case of TSVD regularization, this expression is precisely the number of terms in the TSVD expansion (1.12).

Randomization can also be used to approximate the trace quantity in the denominator of (7.23). See [47] for an interesting hybrid Lanczos–randomization approach that iteratively computes estimates and error bounds for the GCV functional. Nonlinearity can be handled as with UPRE [92, 123].

7.2.1 A Numerical Comparison of UPRE and GCV

A numerical comparison of the GCV method and the UPRE method is presented in Figure 7.2 for Tikhonov regularization and in Figure 7.3 for TSVD regularization. The data came

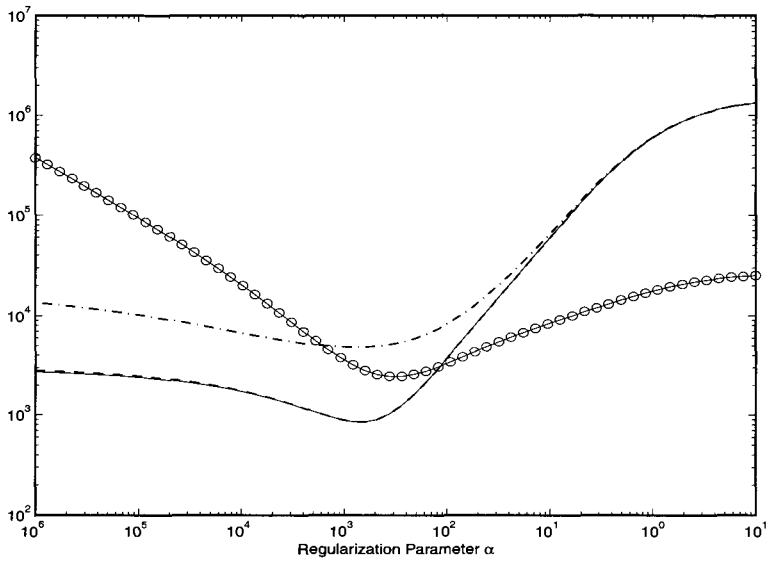


Figure 7.2. *Error indicators for the Tikhonov regularization method. The solid line represents the UPRE $U(\alpha)$; the dashed line, which is almost indistinguishable from the solid line, represents the predictive risk, $||\mathbf{p}_\alpha||^2/n$; the dot-dashed line indicates $GCV(\alpha)$; and the squared norm of the estimation error, $||e_\alpha||^2$, is indicated by the circles.*

from the two-dimensional image deblurring problem described in section 5.1.1, and discrete white noise was added to the simulated data. Formulas (7.19) and (7.24) were used for function evaluation, and the s_i's and $\hat{d}_i$'s were computed using the two-dimensional FFT.

From these plots we can make several observations for this particular test problem:

(i) The graphs of the UPRE $U(\alpha)$ and the predictive risk (7.6) are nearly indistinguishable. Both have exactly the same minimizer for Tikhonov regularization (Figure 7.2). For TSVD regularization (Figure 7.3), the difference between minimizer of $U(\alpha)$ and the predictive risk minimizer is very slight (about 10%), and the resulting reconstructed images are almost identical.

(ii) For TSVD regularization (Figure 7.3), the graph of the predictive risk (dashed line) and the graph of the norm of the estimation error $||e_\alpha||$ (indicated by circles) are qualitatively much different. However, their minimizers are identical. On the other hand, for Tikhonov regularization (Figure 7.2), the minimizer of the norm of the estimation error is slightly larger than the minimizer of the predictive risk.

(iii) The minimizers of the predictive risk and of the GCV function (dot-dashed line) are in close agreement for both Tikhonov and TSVD regularization.

These observations are consistent with the analytic results presented in section 7.6.

7.3 The Discrepancy Principle

The discrepancy principle was introduced in a discrete, deterministic setting in section 1.2.3. It can also be formulated for the discrete data model (7.2), where η is a realization of discrete

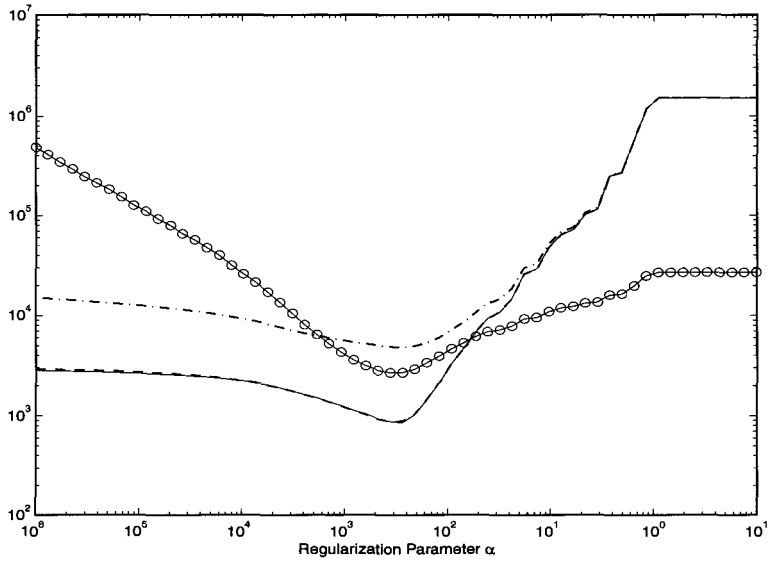


Figure 7.3. *Error indicators for the TSVD regularization method. The solid line represents the UPRE $U(\alpha)$; the dashed line, which is almost indistinguishable from the solid line, represents the predictive risk, $||\mathbf{p}_\alpha||^2/n$; the dot-dashed line indicates $GCV(\alpha)$; and the squared norm of the estimation error, $||e_\alpha||^2$, is indicated by the circles.*

white noise. In this case, one seeks the regularization parameter α for which

$$\frac{1}{n}||\mathbf{r}_\alpha||^2 = \sigma^2, \tag{7.25}$$

where $\mathbf{r}_\alpha = Kf_\alpha - \mathbf{d}$ is the regularized residual. This is motivated by the observation that if $f_\alpha \approx f_{\text{true}}$, then $E(\frac{1}{n}||Kf_\alpha - \mathbf{d}||^2) \approx E(\frac{1}{n}||\eta||^2) = \sigma^2$. For an analysis of this method in a stochastic setting, see section 7.6.4.

7.3.1 Implementation of the Discrepancy Principle

The practical implementation of the discrepancy principle requires the solution of the nonlinear equation

$$0 = F(\alpha) \stackrel{\text{def}}{=} \frac{1}{n}||\mathbf{r}_\alpha||^2 - \sigma^2, \tag{7.26}$$

where white noise variance σ^2 is assumed to be known. Several observations are in order:

(i) If the regularization parameter is discrete, equation (7.26) need not have a solution. This is the case with Landweber iteration, where the regularization parameter is the iteration count, and with TSVD when the regularization parameter is taken to be the number of components m that are not filtered out. Then to implement TSVD regularization, one chooses the largest integer m for which $F(m) \geq 0$. The implementation for iterative methods is similar.

(ii) The function $F(\alpha)$ in (7.26) is often monotonic in α. For example, given that the regularized solution $\mathbf{f}_\alpha = R_\alpha \mathbf{d}$ has a filter representation (7.16), then

$$F(\alpha) = \sum_{s_i>0} [w_\alpha(s_i^2) - 1]^2 \, \hat{d}_i^2 + \sum_{s_i=0} \hat{d}_i^2 - \sigma^2, \qquad \hat{d}_i = \mathbf{u}_i^T \mathbf{d},$$

and $F(\alpha)$ is monotonic whenever w_α is monotonic. This is the case with TSVD and Tikhonov regularization, for example. In this case, bracketing procedures like the bisection method [3, p. 56] may be applied.

(iii) In certain cases, e.g., for Tikhonov regularization, $F(\alpha)$ is continuously differentiable. Then more rapidly convergent schemes like the one-dimensional Newton's method or secant method [3, p. 65] can be applied. It should be noted that while Newton's method has extremely fast local convergence, the cost of each iteration is typically higher, since it requires the derivative $F'(\alpha)$. Implementation of the secant method requires only values of the function $F(\alpha)$.

(iv) For several reasons, one need compute only a rough approximation to the solution to (7.26). First, the regularized solution typically changes little with small changes in α. Second, our analysis in section 7.6.4 shows that the discrepancy method is not optimal. One need not work very hard to find a highly accurate solution to an approximate problem.

7.4 The L-Curve Method

To implement the L-curve method, one plots the log of the squared norm of the regularized solution against the squared norm of the regularized residual for a range of values of regularization parameter. This curve typically has an L shape. The L-curve criterion for regularization parameter selection is to pick the parameter value corresponding to the "corner" of this curve. For examples and implementation details for this method, see [56, 52, 60].

One crucial detail is the precise characterization of the corner of the L-curve. Hansen and O'Leary [60] advocated the point of maximum curvature. Let $\mathbf{f}_\alpha$ denote the regularized solution and let $\mathbf{r}_\alpha = K\mathbf{f}_\alpha - \mathbf{d}$ denote the regularized residual. Define

$$X(\alpha) \stackrel{\text{def}}{=} \log R(\alpha), \qquad R(\alpha) \stackrel{\text{def}}{=} ||\mathbf{r}_\alpha||^2, \tag{7.27}$$

$$Y(\alpha) \stackrel{\text{def}}{=} \log S(\alpha), \qquad S(\alpha) \stackrel{\text{def}}{=} ||\mathbf{f}_\alpha||^2. \tag{7.28}$$

Assume $X(\alpha)$ and $Y(\alpha)$ vary smoothly with α. This is the case, e.g., with Tikhonov regularization. One selects the value of α that maximizes the curvature function

$$\kappa(\alpha) \stackrel{\text{def}}{=} \frac{X''(\alpha)Y'(\alpha) - X'(\alpha)Y''(\alpha)}{\left(X'(\alpha)^2 + Y'(\alpha)^2\right)^{3/2}}, \tag{7.29}$$

where the prime ($'$) denotes differentiation with respect to α. Given an SVD $K = U\text{diag}(s_i)V^T$ and a filter representation (7.16), one can express

$$R(\alpha) = \sum_{s_i>0} [w_\alpha(s_i^2) - 1]^2 \, \hat{d}_i^2 + \sum_{s_i=0} \hat{d}_i^2, \qquad S(\alpha) = \sum_{s_i>0} w_\alpha(s_i^2) \frac{\hat{d}_i}{s_i} \mathbf{v}_i, \tag{7.30}$$

and one can show (see Exercise 7.6) that

$$R'(\alpha) = -\alpha S'(\alpha). \tag{7.31}$$

From this one can derive a formula for the curvature that depends only on $R(\alpha)$, $S(\alpha)$, and $R'(\alpha)$:

$$\kappa(\alpha) = -\frac{R(\alpha)S(\alpha)[\alpha R(\alpha) + \alpha^2 S(\alpha)] + [R(\alpha)S(\alpha)]^2/S'(\alpha)}{[R^2(\alpha) + \alpha^2 S^2(\alpha)]^{3/2}}. \tag{7.32}$$

7.4.1 A Numerical Illustration of the L-Curve Method

We computed the L-curve that arises when standard Tikhonov regularization is applied to the deblurring problem of section 5.1.1. We also computed the curvature function $\kappa(\alpha)$ using the representation (7.32). Plots of both the L-curve and the curvature are given in Figure 7.4. The corner of the L-curve, defined as the point of maximum curvature, occurs at $\alpha = 5.4 \times 10^{-4}$. The optimal regularization parameter, defined as the minimizer of estimation error norm $||f_\alpha - f_{\text{true}}||$, is $\alpha = 2.8 \times 10^{-3}$. In this particular example, the L-curve method underestimates the optimal parameter by nearly an order of magnitude.

The analysis in section 7.6.6, which follows [114], shows that the L-curve regularization parameter selection procedure is nonconvergent in a certain statistical sense. This example is consistent with that analysis. Hanke [51] has also demonstrated the nonconvergence of the L-curve method in a continuous, deterministic setting.

7.5 Other Regularization Parameter Selection Methods

A number of variants of the discrepancy principle have been proposed and analyzed. See [35, section 4.4] for examples. A particular example is the minimum bound method. In the context of the data model (7.2), this method requires the minimization of the functional

$$B(\alpha) \stackrel{\text{def}}{=} \frac{1}{n}\mathbf{r}_\alpha^T (nK^T K)^\dagger \mathbf{r}_\alpha + \gamma \frac{\sigma^2}{\alpha}. \tag{7.33}$$

Here $\mathbf{r}_\alpha = Kf_\alpha - \mathbf{d}$ is the regularized residual, $\gamma \geq 1$ is a parameter, and σ^2 is the variance in the (white) noise. This method was originally formulated in a deterministic setting by Raus [95] and Gfrerer [42]. It was analyzed in a discrete, stochastic setting by Lukas [81].

In many applications, the linear data model with additive white noise (7.2) does not apply. It is often the case in medical and astronomical imaging that the data error is dominated by Poisson noise. For example, see section 5.1 and, in particular, equation (5.4). In this case, the noise level is not known a priori, so techniques like the discrepancy principle and the UPRE method cannot be used. Moreover, while the noise components are independent, they may have drastically differing variances, so the discrete white noise assumption may not hold. This may cause the GCV method to fail.

Veklerov and Llacer [112, 77] developed a novel stopping rule for iterative methods like the EM algorithm, which minimizes the negative Poisson log likelihood functional; see Example 4.17 and section 4.5. Their approach can be adapted to select regularization parameters for methods like Tikhonov regularization. To illustrate the key ideas, assume the data components d_i are independent and Poisson distributed with mean $[K\mathbf{f}_{\text{true}}]_i$. We denote this by $\mathbf{d} \sim \text{Poisson}(K\mathbf{f}_{\text{true}})$. Given a particular value α of the regularization parameter, apply a statistical hypothesis test for the conjecture that $\mathbf{d} \sim \text{Poisson}(K\mathbf{f}_\alpha)$. If the test succeeds, α is an acceptable regularization parameter; otherwise, α is rejected. This approach may yield a range of acceptable values of α, or it may yield no acceptable values. If the latter situation were to occur, it would suggest that the data model was inadequate or that the prior information (incorporated through the regularization functional) was incorrect.

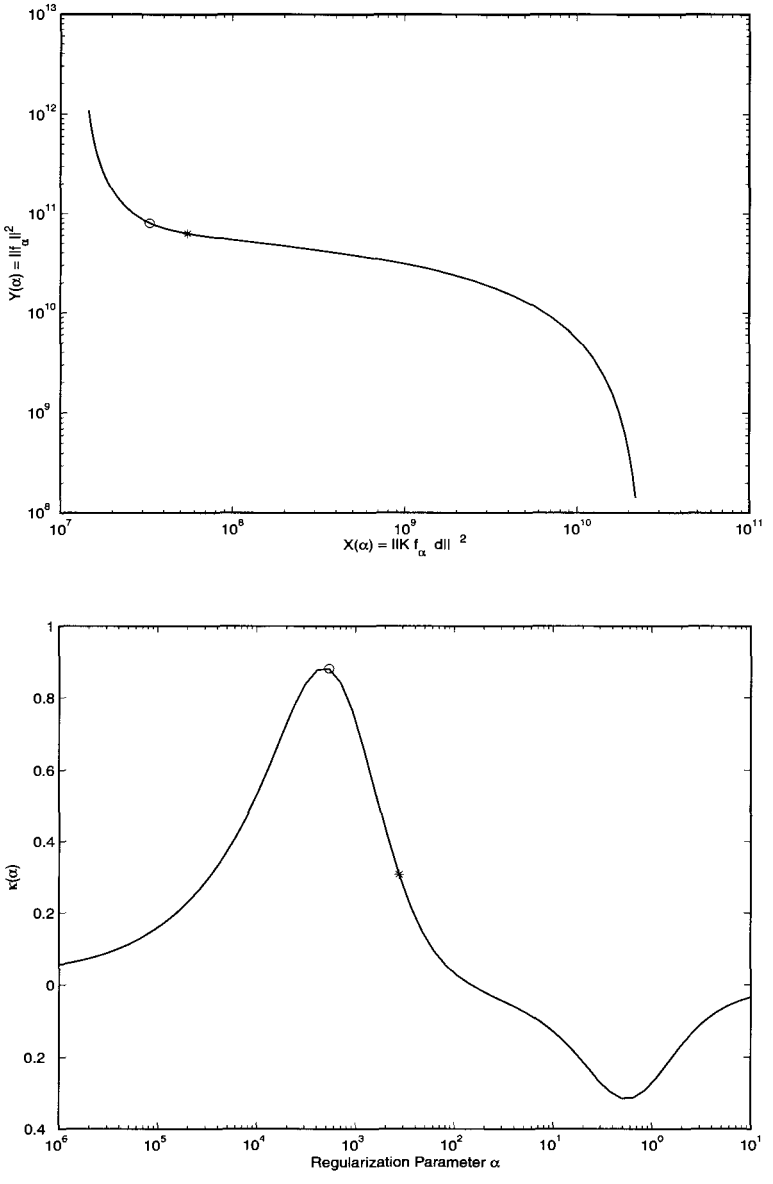


Figure 7.4. *L-curve for Tikhonov regularization applied to an image deblurring problem. The top graph shows the L-curve. As the regularization parameter α increases, the curve moves from the upper left to the lower right. The bottom graph shows the curvature of the L-curve. The circle on both graphs indicates the point of maximum curvature. The asterisk indicates the point where the estimation error is minimized.*

7.6 Analysis of Regularization Parameter Selection Methods

There are several ways to compare and contrast computational methods for regularization parameter selection. Perhaps the most straightforward approach is to apply the methods to a collection of test problems. Such a collection was compiled by Hansen [57, 58, 59]. One can conduct Monte Carlo studies based on these test problems.

A more systematic approach is to lay out some assumptions and then conduct a rigorous analysis based on these assumptions to determine attributes like asymptotic rates of convergence. These attributes can then be compared to determine which method is best for a particular class of problems. For example, in section 1.2.1 we assumed a discrete, deterministic data model. We then analyzed the behavior of the solution error $\mathbf{e}_\alpha \stackrel{\text{def}}{=} \mathbf{f}_\alpha - \mathbf{f}_{\text{true}}$ in terms of the error level $\delta \stackrel{\text{def}}{=} ||\boldsymbol{\eta}||$. In particular, we showed for certain regularization methods that the regularization parameter α could be selected so that $\mathbf{e}_\alpha \to 0$ as $\delta \to 0$. Under an additional assumption, the source condition (1.25), we obtained a rate of convergence (1.29). This type of analysis can be extended to a continuous, deterministic setting with data $d = Kf + \eta$ for which d, f_{true}, and η all lie in infinite-dimensional Hilbert spaces like $L^2(0, 1)$. For details, see [35, 70].

Perhaps the greatest shortcoming of the continuous, deterministic analysis lies in the treatment of error. A Hilbert space like $L^2(0, 1)$ cannot contain a realization of continuous white noise, which is a generalization of discrete white noise [38]. Even in a finite-dimensional setting, there may be significant differences between deterministic and stochastic analyses. The deterministic analysis in section 1.2.1 provides worst-case error bounds. It is possible, but very unlikely, that these bounds will be attained. In contrast, a statistical analysis will yield what may be viewed as average-case error bounds.

In the analysis that follows, we assume a somewhat more complicated data model than (7.2). As before, we assume that the data are discrete and the noise is stochastic. However, we assume the true solution lies in an infinite-dimensional space. We also make very specific assumptions regarding the decay rates of singular values of the (linear) operator and generalized Fourier coefficients of the true solution. Our analysis is similar in many respects to that of Wahba [122, 123], Kay [67], Davies and Anderssen [28], and Lukas [81, 82].

7.6.1 Model Assumptions and Preliminary Results

Consider the data model

$$\mathbf{d} = K_n f_{\text{true}} + \boldsymbol{\eta}, \tag{7.34}$$

where the true solution f_{true} lies in a deterministic, infinite-dimensional Hilbert space $\mathcal{H}$ and the linear operator K_n maps $\mathcal{H}$ into $\mathbb{R}^n$. K_n can be viewed as a range discretization of an underlying continuous operator. See, for example, equations (5.2) and (5.3), used to model image formation. We take $\boldsymbol{\eta}$ to be a discrete white noise vector; see Definition 7.1. We refer to (7.34) as a semidiscrete, semistochastic linear data model.

As in our previous fully discrete development, we denote the regularized solution by f_α and the estimation error by $e_\alpha = f_\alpha - f_{\text{true}}$. Both are now random functions. We again use lowercase rather than uppercase letters to represent random quantities. The parameter n in (7.34) denotes the number of components in the data vector $\mathbf{d}$. Our goal is to establish

mean square convergence:

$$E(\|e_\alpha\|_{\mathcal{H}}^2) \to 0 \quad \text{as} \quad n \to \infty.$$

We stress that the variance σ^2 of the discrete white noise vector $\boldsymbol{\eta}$ is fixed, independent of n. In this case (see Exercise 7.9),

$$E(\|\boldsymbol{\eta}\|^2/n) = \sigma^2 \quad \text{for each } n. \tag{7.35}$$

If we view $\boldsymbol{\eta}$ as a discretization of some underlying continuous error that is parameterized by n, then (7.35) implies that the error level stays constant. Nevertheless, we can still demonstrate mean square convergence for certain regularization methods, provided certain assumptions are met. This counterintuitive result highlights the difference between the deterministic framework and the stochastic framework.

For each n, let the partially discrete linear operator $K_n : \mathcal{H} \to \mathbb{R}^n$ have a singular system $\{\mathbf{u}_{in}, s_{in}, v_{in}\}_{i=1}^n$ with singular values

$$s_{1n} \geq s_{2n} \geq \cdots \geq s_{nn} > 0.$$

Note that $v_{in} \in \mathcal{H}$ with $\langle v_{in}, v_{jn}\rangle_{\mathcal{H}} = \delta_{ij}$, while $\mathbf{u}_{in} \in \mathbb{R}^n$ with $\langle \mathbf{u}_{in}, \mathbf{u}_{jn}\rangle_{\mathbb{R}^n} = \mathbf{u}_{in}^T \mathbf{u}_{jn} = \delta_{ij}$. As in section 7.1, we assume that the regularized solution has a linear filter representation,

$$f_\alpha = \sum_{i=1}^n w_\alpha(s_{in}^2) \frac{\mathbf{u}_{in}^T \mathbf{d}}{s_{in}} v_{in} \stackrel{\text{def}}{=} R_\alpha \mathbf{d}. \tag{7.36}$$

Here $R_\alpha : \mathbb{R}^n \to \mathcal{H}$ is the regularization operator. Then by (7.34) and (7.36) the estimation error

$$\begin{aligned} e_\alpha &\stackrel{\text{def}}{=} f_\alpha - f_{\text{true}} = (R_\alpha K_n - I) f_{\text{true}} + R_\alpha \boldsymbol{\eta} \\ &= \sum_{i=1}^n [w_\alpha(s_{in}^2) - 1] \hat{f}_{in} v_{in} + \left(\sum_{i=1}^n \hat{f}_{in} v_{in} - f_{\text{true}} \right) \\ &\quad + \sum_{i=1}^n w_\alpha(s_{in}^2) \frac{\hat{\eta}_{in}}{s_{in}} v_{in}, \end{aligned} \tag{7.37}$$

where

$$\hat{f}_{in} = \langle f_{\text{true}}, v_{in}\rangle_{\mathcal{H}}, \qquad \hat{\eta}_{in} = \mathbf{u}_{in}^T \boldsymbol{\eta}, \qquad i = 1, \ldots, n. \tag{7.38}$$

Also as in section 7.1, the influence matrix is

$$\begin{aligned} A_\alpha &\stackrel{\text{def}}{=} K_n R_\alpha \\ &= U \operatorname{diag}(w_\alpha(s_{in}^2))\, U^T. \end{aligned} \tag{7.39}$$

Here U is the orthogonal matrix whose ith column is the left singular vector $\mathbf{u}_{in}$ of K_n. We also define the predictive error,

$$\begin{aligned} \mathbf{p}_\alpha &\stackrel{\text{def}}{=} K_n e_\alpha \\ &= (A_\alpha - I) K_n f_{\text{true}} + A_\alpha \boldsymbol{\eta} \\ &= \sum_{i=1}^n (w_\alpha(s_{in}^2) - 1)\, s_{in} \hat{f}_{in} \mathbf{u}_{in} + \sum_{i=1}^n w_\alpha(s_{in}^2)\, \hat{\eta}_{in} \mathbf{u}_{in}, \end{aligned} \tag{7.40}$$

and the regularized residual,

$$\begin{aligned}\mathbf{r}_\alpha &\stackrel{\text{def}}{=} K_n f_\alpha - \mathbf{d} \\ &= (A_\alpha - I)K_n f_{\text{true}} + (A_\alpha - I)\boldsymbol{\eta} \\ &= \sum_{i=1}^{n}(w_\alpha(s_{in}^2) - 1)\, s_{in}\hat{f}_{in}\mathbf{u}_{in} + \sum_{i=1}^{n}(w_\alpha(s_{in}^2) - 1)\,\hat{\eta}_{in}\mathbf{u}_{in}. \end{aligned} \tag{7.41}$$

The predictive risk is the scalar quantity $||\mathbf{p}_\alpha||^2/n$.

The following lemma provides representations for trace terms appearing in error indicators like $E(||e_\alpha||_{\mathcal{H}}^2)$ and $E(||\mathbf{p}_\alpha||^2/n)$.

Lemma 7.5. *The influence matrix is symmetric, with representation*

$$\text{trace}(A_\alpha) = \sum_{i=1}^{n} w_\alpha(s_{in}^2) \tag{7.42}$$

and

$$\text{trace}(A_\alpha^* A_\alpha) = \text{trace}(A_\alpha^2) = \sum_{i=1}^{n} w_\alpha(s_{in}^2)^2. \tag{7.43}$$

In addition,

$$\text{trace}(R_\alpha^* R_\alpha) = \sum_{i=1}^{n} \frac{w_\alpha(s_{in}^2)^2}{s_{in}^2}. \tag{7.44}$$

Proof. Equations (7.42) and (7.43) follow immediately from (7.39) and (4.28). To verify (7.44), let U be as in (7.39). By adjoint and orthogonality properties and representation (7.36),

$$\begin{aligned}[U^T(R_\alpha^* R_\alpha)U]_{ij} &= (R_\alpha^* R_\alpha \mathbf{u}_{in})^T \mathbf{u}_{jn} \\ &= \langle R_\alpha \mathbf{u}_{in}, R_\alpha \mathbf{u}_{jn}\rangle_{\mathcal{H}} \\ &= \left\langle \sum_{k=1}^{n} w_\alpha(s_{kn}^2)\frac{\langle \mathbf{u}_{kn}, \mathbf{u}_{in}\rangle_n}{s_{kn}} v_{kn}, \sum_{k=1}^{n} w_\alpha(s_{kn}^2)\frac{\langle \mathbf{u}_{kn}, \mathbf{u}_{jn}\rangle_n}{s_{kn}} v_{kn} \right\rangle_{\mathcal{H}} \\ &= \left\langle \frac{w_\alpha(s_{in}^2)}{s_{in}} v_{in}, \frac{w_\alpha(s_{jn}^2)}{s_{jn}} v_{jn} \right\rangle_{\mathcal{H}} \\ &= \frac{w_\alpha(s_{jn}^2)^2}{s_{in}^2}\delta_{ij}. \end{aligned}$$

Equation (7.44) follows from the fact that the matrix trace is invariant under orthogonal transformations (see Exercise 7.10). □

From the Trace Lemma 7.2 and (7.37), we obtain the following proposition.

Proposition 7.6. *Let P_n denote the projection of $\mathcal{H}$ onto* $\text{Null}(K_n)^\perp$, *and let* $f_{\text{Null}(K_n)} = (I - P_n)f_{\text{true}}$ *denote the orthogonal projection of* f_{true} *onto* $\text{Null}(K_n)$. *Then*

$$E(||e_\alpha||_{\mathcal{H}}^2) = ||(R_\alpha K_n - P_n)f_{\text{true}}||_{\mathcal{H}}^2 + ||(I - P_n)f_{\text{true}}||_{\mathcal{H}}^2$$

$$+ \sigma^2 \operatorname{trace}(R_\alpha^* R_\alpha)$$

$$= \sum_{i=1}^{n} [w_\alpha(s_{in}^2) - 1]^2 f_{in}^2 + \|f_{\text{Null}(K_n)}\|_{\mathcal{H}}^2 + \sigma^2 \sum_{i=1}^{n} \frac{w_\alpha(s_{in}^2)^2}{s_{in}^2}. \tag{7.45}$$

Proof. Note that the third term in (7.45) comes from (7.44). □

Remark 7.7. Equation (7.45) is somewhat analogous to error expressions that arose in section 1.2.1. The third term on the right-hand side of (7.45) quantifies the effects of noise amplification. In the statistics literature, this is often referred to as the variance in the estimation error. The second term in (7.45) represents truncation error due to finite-dimensional projection, while the first term represents truncation error due to the regularization. Together, the first and second terms represent what statisticians call the bias in the estimation error.

We obtain similar bias-variance decompositions for the predictive risk and the mean squared regularized residual. See Exercise 7.11 for proof.

Proposition 7.8.

$$E\left(\frac{1}{n}\|\mathbf{p}_\alpha\|^2\right) = \frac{1}{n}\|(A_\alpha - I)K_n f_{\text{true}}\|^2 + \frac{\sigma^2}{n}\operatorname{trace}(A_\alpha^2)$$

$$= \frac{1}{n}\sum_{i=1}^{n}(w_\alpha(s_{in}^2) - 1)^2 s_{in}^2 f_{in}^2 + \frac{\sigma^2}{n}\sum_{i=1}^{n} w_\alpha(s_{in}^2)^2, \tag{7.46}$$

and

$$E\left(\frac{1}{n}\|\mathbf{r}_\alpha\|^2\right) = \frac{1}{n}\|(A_\alpha - I)K_n f_{\text{true}}\|^2 + \frac{\sigma^2}{n}\operatorname{trace}[(A_\alpha - I)^2]$$

$$= \frac{1}{n}\sum_{i=1}^{n}(w_\alpha(s_{in}^2) - 1)^2 s_{in}^2 f_{in}^2 + \frac{\sigma^2}{n}\sum_{i=1}^{n}(w_\alpha(s_{in}^2) - 1)^2. \tag{7.47}$$

At this point we could apply an analysis similar to that in sections 1.2.1 and 1.2.2 to obtain bounds for $E(\|e_\alpha\|_{\mathcal{H}}^2)$. For example, with TSVD regularization (1.11), one can bound the third term in (7.45) by σ^2/α (see (1.21)). Similarly, assuming a source condition $f_{\text{true}} = K_n^* \mathbf{z}$, one can bound the first term by $\alpha\|\mathbf{z}\|^2$. Proceeding as in section 1.2.2, one can select $\alpha = \sigma/\|\mathbf{z}\|$ to obtain

$$E(\|e_\alpha\|_{\mathcal{H}}^2) \le \|f_{\text{Null}(K_n)}\|_{\mathcal{H}}^2 + 2\sigma\,\|\mathbf{z}\|. \tag{7.48}$$

However, the right-hand side will not go to zero, since σ^2 is fixed.

To obtain sharper bounds, we assume a specific algebraic decay rate for the singular values of K_n,

$$s_{in}^2 = n\,c\,i^{-p}, \qquad c > 0, \quad p > 1. \tag{7.49}$$

Note that the s_{in}^2 are the eigenvalues of $K_n^* K_n$.

Remark 7.9. Implicit in the rate (7.49) is the assumption that the underlying continuous operator K has singular values s_i for which

$$s_i^2 = \frac{s_{in}^2}{n} = c\,i^{-p}. \tag{7.50}$$

The factor of $1/n$ arises from the fact that the Euclidean inner product on $\mathbb{R}^n$ must be scaled to behave like a quadrature approximation to a continuous inner product like the inner product on $L^2(0, 1)$. Hence (7.49) is a realistic assumption [103]. The decay rate parameter p in (7.49) reflects the smoothing properties of the operator K and the ill-posedness of operator equation $Kf = d$. As p increases, the problem becomes more ill-posed, and the intrinsic information content [43] decreases.

Remark 7.10. Because of equation (7.50), regularization parameters for fully discrete and partially discrete problems will differ by a factor of n. To simplify the analysis, we redefine the regularization filter functions (1.11) and (1.13) to incorporate this factor. We define the discrete filter function for TSVD,

$$w_\alpha^{\text{TSVD}}(s^2) = \begin{cases} 1 & \text{if } s^2 \geq n\alpha, \\ 0 & \text{if } s^2 < n\alpha, \end{cases} \tag{7.51}$$

and for Tikhonov regularization,

$$w_\alpha^{\text{Tikh}}(s^2) = \frac{s^2}{s^2 + n\alpha}. \tag{7.52}$$

We also assume an algebraic decay rate for the generalized Fourier coefficients of the true solution f_{true},

$$\hat{f}_{in}^2 \stackrel{\text{def}}{=} \langle f_{\text{true}}, v_{in} \rangle_{\mathcal{H}}^2 = bi^{-q}, \qquad b > 0, \quad q > 1. \tag{7.53}$$

As the rate parameter q increases, the true solution becomes smoother.

Remark 7.11. An alternate assumption that is equivalent to the source condition $f_{\text{true}} \in \text{Range}(K^*)$ in the setting of section 1.2.2 is

$$|||f_{\text{true}}|||_{K^*}^2 \stackrel{\text{def}}{=} \lim_{n\to\infty} \sum_{i=1}^{n} \frac{\hat{f}_{in}^2}{s_i^2} = \frac{b}{c} \sum_{i=1}^{\infty} i^p \hat{f}_{in}^2 < \infty. \tag{7.54}$$

This assumption is equivalent to that made in the analysis of Wahba [122, 123] and Kay [67]. At first glance, it appears that (7.54) is less restrictive than (7.53) because no specific decay rate for the f_{in} is specified. However, substitution of (7.53) into (7.54) yields

$$|||f_{\text{true}}|||_{K^*}^2 = \frac{b}{c} \sum_{i=1}^{\infty} i^{2-q} < \infty, \tag{7.55}$$

a condition that holds if and only if $q > p + 1$ with $p > 1$. On the other hand, (7.53) is assumed to hold for any $q > 1$. If $1 < q \leq p+1$, then f_{true} is too rough to lie in $\text{Range}(K^*)$. Thus, neither of the two assumptions (7.53) and (7.54) implies the other.

We make one additional assumption, regarding convergence of the projections $f_{\text{Null}(K_n)}$ of the true solution onto the null space of the K_n (see Proposition 7.6):

$$\|f_{\text{Null}(K_n)}\| \to 0 \quad \text{as} \quad n \to \infty. \tag{7.56}$$

This assumption holds if the K_n converge strongly to some underlying operator K and $\text{Null}(K) = \{0\}$. If K has a nontrivial null space, then instead of establishing mean square

convergence of f_α to f_{true}, we can show that f_α converges to the projection of f_{true} onto $\text{Null}(K)^\perp$.

Remark 7.12. Here is a summary of our assumptions:

(i) We take the semidiscrete, semistochastic data model (7.34), where η denotes discrete white noise and n denotes the number of data points.

(ii) The operator $K_n : \mathcal{H} \to \mathbb{R}^n$ has a singular system $\{\mathbf{u}_{in}, s_{in}, v_{in} \mid i = 1, \ldots, n\}$. The singular values s_{in} are all positive and decay at an algebraic rate given in (7.49).

(iii) The true solution f_{true} in model (7.34) has generalized Fourier coefficients $\hat{f}_{in} = \langle f_{\text{true}}, v_{in} \rangle_{\mathcal{H}}$ which decay at an algebraic rate given in (7.53).

(iv) The orthogonal projection of the true solution onto the null space of K_n, which we denote by $f_{\text{Null}(K_n)}$, vanishes as $n \to \infty$.

(v) The regularized solution f_α has a linear filtering representation (7.36).

In the next sections, we make use of big "O" and little "o" notation and asymptotic equality. Little "o" notation was defined in (2.1).

Definition 7.13. The symbol $\simeq$ denotes asymptotic equality, i.e.,

$$f(\alpha) \simeq g(\alpha) \text{ as } \alpha \to \alpha_* \quad \text{if and only if} \quad \lim_{\alpha \to \alpha_*} \frac{f(\alpha)}{g(\alpha)} = 1. \tag{7.57}$$

Big "O" notation is defined as usual: $f(\alpha) = \mathcal{O}(g(\alpha))$ as $\alpha \to \alpha_*$ means there exists a constant $C > 0$ for which

$$\limsup_{\alpha \to \alpha_*} \left| \frac{f(\alpha)}{g(\alpha)} \right| \leq C.$$

7.6.2 Estimation and Predictive Errors for TSVD

From the truncated SVD error indicator plots in Figure 7.3 we saw that estimation error norm and predictive risk achieved their minima at exactly the same value of the regularization parameter. We now explain why one should expect this to happen. We also derive asymptotic rates of convergence for the expected error norms and for the regularization parameter that minimizes these norms.

As in [113] we apply TSVD regularization with the regularization parameter taken to be the index $m = \max\{i \mid s_{in}^2 \geq n\alpha\}$. Since the singular values are assumed to be nonincreasing, m is the number of singular values at or above the cut-off level $n\alpha$. Then the filter function (7.51) can be expressed as $w_m(s_{in}^2) = 1$, if $i \leq m$, and $w_m(s_{in}^2) = 0$ otherwise. From (7.45) and the rate assumptions (7.49) and (7.53), the expected value of the squared norm of the estimation error for TSVD can be expressed as

$$E(\|e_m^{\text{TSVD}}\|_{\mathcal{H}}^2) = \|f_{\text{Null}(K_n)}\|_{\mathcal{H}}^2 + b \sum_{i=m+1}^{n} i^{-q} + \frac{\sigma^2}{n} \frac{1}{c} \sum_{i=1}^{m} i^p. \tag{7.58}$$

What follows is a discrete analogue of Theorem 2.37, which characterizes minimizers for smooth functions.

Lemma 7.14. *Let* $F : \mathbf{Z}(n) \to \mathbb{R}$, *where* $\mathbf{Z}(n) \stackrel{\text{def}}{=} \{0, 1, \ldots, n\}$. *Set* $\Delta_m^+ F(m) \stackrel{\text{def}}{=} F(m+1) - F(m)$ *and* $\Delta_m^- F(m) \stackrel{\text{def}}{=} F(m) - F(m-1)$. *If* $m_* = \arg\min_{m \in \mathbf{Z}(n)} F(m)$ *with* $0 < m_* < n$, *then* $\Delta_m^- F(m_*) \le 0$ *and* $\Delta_m^+ F(m_*) \ge 0$.

Applying this result to equation (7.58), we obtain the following.

Proposition 7.15. *Let* $m_* = \arg\min_{m \in \mathbf{Z}(n)} E(\|e_m^{\mathrm{TSVD}}\|_{\mathcal{H}}^2)$, *and assume* $0 < m_* < n$. *Then*

$$\Delta_m^- E(\|e_{m_*}^{\mathrm{TSVD}}\|_{\mathcal{H}}^2) = -b m_*^{-q} + \frac{\sigma^2}{n} \frac{1}{c} m_*^p \le 0, \tag{7.59}$$

$$\Delta_m^+ E(\|e_{m_*}^{\mathrm{TSVD}}\|_{\mathcal{H}}^2) = -b(m_*+1)^{-q} + \frac{\sigma^2}{n} \frac{1}{c} (m_*+1)^p \ge 0. \tag{7.60}$$

Consequently,

$$-1 + \left(\frac{1}{bc} \frac{\sigma^2}{n}\right)^{-\frac{1}{p+q}} \le m_* \le \left(\frac{1}{bc} \frac{\sigma^2}{n}\right)^{-\frac{1}{p+q}}. \tag{7.61}$$

To show that the bounds (7.61) characterize a minimizer, we need to show that neither $m = 0$ nor $m = n$ are minimizers. The case $m = n$ requires knowledge of the asymptotic behavior of $E(\|e_m^{\mathrm{TSVD}}\|_{\mathcal{H}}^2)$ for large n. To determine this behavior, apply the idea underlying the integral comparison test for series convergence to the sums in (7.58):

$$\frac{m^{p+1}}{p+1} = \int_0^m x^p \, dx \le \sum_{i=1}^m i^p \le \int_0^{m+1} x^p \, dx = (1 + \mathcal{O}(m^{-1})) \frac{m^{p+1}}{p+1}$$

and

$$\left[1 + \mathcal{O}\left(m^{-1}\right) + \mathcal{O}\left(\left(\frac{m}{n}\right)^{1-q}\right)\right] \frac{m^{1-q}}{q-1} \le \sum_{i=m+1}^n i^{-q} \le \frac{m^{1-q}}{q-1}$$

as $n \to \infty$, $m \to \infty$, and $m/n \to 0$. Then from (7.58),

$$E(\|e_m^{\mathrm{TSVD}}\|_{\mathcal{H}}^2) = \|f_{\mathrm{Null}(K_n)}\|_{\mathcal{H}}^2 + \left[\frac{b}{q-1} + \mathcal{O}\left(m^{-1}\right) + \mathcal{O}\left(\left(\frac{m}{n}\right)^{1-q}\right)\right] m^{1-q}$$
$$+ \frac{\sigma^2}{n} \left(\frac{1}{c(p+1)} + \mathcal{O}(m^{-1})\right) m^{p+1}. \tag{7.62}$$

Theorem 7.16 (Minimizer of Estimation Error for TSVD).
Let $m_e = \arg\min_{m \in \mathbf{Z}(m)} E(\|e_m^{\mathrm{TSVD}}\|_{\mathcal{H}}^2)$, *the minimizer of the expected value of the squared norm of the estimation error for TSVD regularization. If n is sufficiently large, then*

$$m_e = \mathrm{int}\left[(bc)^{\frac{1}{p+q}} \left(\frac{\sigma^2}{n}\right)^{-\frac{1}{p+q}}\right], \tag{7.63}$$

where int[·] *denotes the greatest integer function. Moreover, the value at the minimizer satisfies*

$$E(\|e_{m_e}^{\mathrm{TSVD}}\|_{\mathcal{H}}^2) \simeq \|f_{\mathrm{Null}(K_n)}\|_{\mathcal{H}}^2 + C_1^{\mathrm{TSVD}} \left(\frac{\sigma^2}{n}\right)^{\frac{q-1}{p+q}}, \tag{7.64}$$

where $C_1^{\mathrm{TSVD}} = (p+q)/((p+1)(q-1)) \left(b^{p+1}/c^{q-1}\right)^{1/(p+q)}$.

Proof. Expression (7.63) follows immediately from the bounds (7.61). However, this m_e need not be a minimizer. Substitution of (7.63) in (7.62) gives (7.64), since m_e^{-1} and m_e/n both tend to zero as $n \to \infty$. Note that by assumption (7.56), $||f_{\text{Null}(K_n)}||_{\mathcal{H}} \to 0$, and so $E(||e_{m_e}^{\text{TSVD}}||_{\mathcal{H}}^2)$ converges to zero as $n \to \infty$. On the other hand, from (7.58), $E(||e_0^{\text{TSVD}}||_{\mathcal{H}}^2) \geq b$ and $E(||e_n^{\text{TSVD}}||_{\mathcal{H}}^2) \to \infty$ as $n \to \infty$. This establishes that m_e really is a minimizer, provided n is sufficiently large. □

Remark 7.17. Because of the rate assumption (7.49), the relationship between the index parameter m and the filter parameter α in (7.51) is given by $\alpha \approx cm^{-p}$. From (7.63), this gives the behavior of the optimal filter parameter for TSVD regularization,

$$\alpha_{\text{TSVD}} \approx C_2^{\text{TSVD}} \left(\frac{\sigma^2}{n}\right)^{\frac{p}{p+q}}, \tag{7.65}$$

where $C_2^{\text{TSVD}} = (c^q/b^p)^{1/(p+q)}$.

We now analyze the predictive error for TSVD regularization. From (7.46), the form of the TSVD filter (7.51), and the rate assumptions (7.49) and (7.53), the expected predictive risk is given by

$$E\left(\frac{1}{n}||\mathbf{p}_m^{\text{TSVD}}||^2\right) = bc \sum_{i=m+1}^{n} i^{-p-q} + \frac{m\sigma^2}{n}. \tag{7.66}$$

Hence, by Lemma 7.14, if $E(||\mathbf{p}_m^{\text{TSVD}}||_n^2/n)$ has a minimizer m that lies between 1 and $n-1$, then

$$\Delta_m^- E\left(\frac{1}{n}||\mathbf{p}_m^{\text{TSVD}}||^2\right) = bc\, m^{-p-q} + \frac{\sigma^2}{n} \leq 0, \tag{7.67}$$

$$\Delta_m^+ E\left(\frac{1}{n}||\mathbf{p}_m^{\text{TSVD}}||^2\right) = bc\,(m+1)^{-p-q} + \frac{\sigma^2}{n} \geq 0. \tag{7.68}$$

Note that if one multiplies (7.67) by m^p/c, then one obtains the corresponding result (7.59) for estimation error. Similarly, one can multiply (7.68) by $(m+1)^p/c$ to obtain (7.60). Consequently, if one can preclude the boundary cases $m = 0$ and $m = n$, then $E(||\mathbf{p}_m^{\text{TSVD}}||^2/n)$ and $E(||e_m^{\text{TSVD}}||_{\mathcal{H}}^2)$ have the same minimizer.

Theorem 7.18 (Minimizer of Predictive Risk for TSVD). *For TSVD regularization, for sufficiently large n the minimizer of the expected predictive risk, $E(||\mathbf{p}_m^{\text{TSVD}}||^2/n)$, is identical to the minimizer m_e of the expected value of the squared norm of the estimation error, $E(||e_m^{\text{TSVD}}||_{\mathcal{H}}^2)$; see Theorem* 7.16.

7.6.3 Estimation and Predictive Errors for Tikhonov Regularization

The Tikhonov error indicator plots in Figure 7.2 show that the minimizer of the estimation error norm is slightly larger than the minimizer of the predictive risk. This is in contrast to the situation for TSVD regularization. The following analysis explains this. We also analyze a phenomenon called saturation, in which convergence rates are qualitatively different, depending on whether the true solution is sufficiently smooth.

The following quantities arise in the analysis of Tikhonov regularization. For $s \geq -1$, $p > 1$, $h > 0$, and j, n positive integers, define

$$S^s_{p,j}(n,h) = \sum_{i=1}^{n} \frac{(ih)^s}{[1+(ih)^p]^j} h. \tag{7.69}$$

For $s > -1$, also define

$$I^s_{p,j} = \int_0^\infty \frac{u^s}{(1+u^p)^j} du. \tag{7.70}$$

Proposition 7.19. *Suppose $jp - s - 1 > 0$. As $h \to 0$, $n \to \infty$, and $nh \to \infty$,*

$$S^s_{p,j}(n,h) = \begin{cases} I^s_{p,j} + \mathcal{O}((nh)^{s-jp+1}) + \mathcal{O}(h) & \text{if } s > 0, \\ I^s_{p,j} + \mathcal{O}((nh)^{s-jp+1}) + \mathcal{O}(h^{s+1}) & \text{if } -1 < s \leq 0, \\ -\log(h) + \mathcal{O}(1) & \text{if } s = -1. \end{cases} \tag{7.71}$$

Proof. Let $f(u) = u^s/(1+u^p)^j$. For $s > -1$, by the triangle inequality,

$$|S^s_{p,j}(n,h) - I^s_{p,j}| \leq |S^s_{p,j}(\infty,h) - S^s_{p,j}(n,h)| + |I^s_{p,j} - S^s_{p,j}(\infty,h)|.$$

To deal with the first term on the right-hand side, $0 \leq S^s_{p,j}(\infty,h) - S^s_{p,j}(n,h) \leq \int_{nh}^\infty u^{s-jp}\, ds = \mathcal{O}((nh)^{s-jp+1})$. If $s > 0$, $f'(u)$ is bounded for $u \geq 0$, and the second term on the right-hand side can be viewed as error for the rectangular rule quadrature approximation, which is $\mathcal{O}(h)$ accurate. On the other hand, if $-1 < s \leq 0$, $f(u)$ is decreasing, so $0 \leq \int_0^\infty f(u)\, du - S^s_{p,j}(\infty,h) \leq \int_0^h f(u)\, du = \mathcal{O}(h^{s+1})$. Now consider $s = -1$. As above, $0 \leq S^s_{p,j}(\infty,h) - S^s_{p,j}(n,h) = \mathcal{O}((nh)^{-jp})$. On the other hand, $0 \leq S^s_{p,j}(\infty,h) - \int_h^\infty f(u)\, du \leq f(h)\, h \leq 1$, and

$$\int_h^\infty f(u)\, du = -\log h + \int_h^1 (f(u) - u^{-1})\, du + \int_1^\infty f(u)\, du. \tag{7.72}$$

The second and third terms on the right-hand side are both $\mathcal{O}(1)$. □

One can relate $I^s_{p,j}$ to certain special functions.

Proposition 7.20. *If $s + 1 > 0$ and $jp - s - 1 > 0$, then*

$$I^s_{p,j} = \frac{1}{p} B\left(\frac{jp-s-1}{p}, \frac{s+1}{p}\right) = \frac{\Gamma\left(\frac{jp-s-1}{p}\right)\Gamma\left(\frac{s+1}{p}\right)}{p(j-1)!}, \tag{7.73}$$

where $B(\cdot,\cdot)$ and $\Gamma(\cdot)$ denote the beta and gamma functions, respectively:

$$B(z,w) = \int_0^1 t^{z-1}(1-t)^{w-1}\, dt, \quad \Gamma(z) = \int_0^\infty t^{z-1}e^{-t}\, dt, \qquad z > 1, w > 1.$$

Proof. To obtain the first equality in (7.73), substitute $t = 1/(1+u^p)$. The second equality follows from the fact that $B(z,w) = \Gamma(z)\Gamma(w)/\Gamma(z+w)$ and from the recursion $\Gamma(z+1) = z\Gamma(z)$. See [1]. □

The analysis of Tikhonov regularization is somewhat similar to that for TSVD regularization. From (7.45), the form of the filter function (7.52), and the decay assumptions (7.49) and (7.53), the expected value of the squared norm of the estimation error for Tikhonov regularization can be expressed as

$$\begin{aligned} E(\|e_\alpha^{\text{Tikh}}\|_{\mathcal{H}}^2) &= \|f_{\text{Null}(K_n)}\|_{\mathcal{H}}^2 + \sum_{i=1}^{n} \left(\frac{n\alpha}{s_{in}^2 + n\alpha} \right)^2 f_{in}^2 + \sigma^2 \sum_{i=1}^{n} \frac{s_{in}^2}{(s_{in}^2 + n\alpha)^2} \\ &= \|f_{\text{Null}(K_n)}\|_{\mathcal{H}}^2 + \sum_{i=1}^{n} \left(\frac{\frac{\alpha}{c} i^p}{1 + \frac{\alpha}{c} i^p} \right)^2 bi^{-q} + \frac{\sigma^2}{n} \sum_{i=1}^{n} \frac{\frac{1}{c} i^p}{(1 + \frac{\alpha}{c} i^p)^2}. \end{aligned} \tag{7.74}$$

Let T_2 and T_3 denote the second and third terms on the right-hand side of (7.74). To approximate these sums by integrals, we take $h = (\alpha/c)^{1/p}$. Then if $h \to 0, n \to \infty$, and $nh \to \infty$, one can express the third term in (7.74) as

$$T_3 = \frac{\sigma^2}{n} \alpha^{\frac{-p-1}{p}} c^{\frac{1}{p}} \sum_{i=1}^{n} \frac{(ih)^p}{[1 + (ih)^p]^2} h = \frac{\sigma^2}{n} \alpha^{\frac{-p-1}{p}} c^{\frac{1}{p}} \left[I_{p,2}^p + o(1) \right]. \tag{7.75}$$

The second equality in (7.75) follows from Proposition 7.19 with $j = 2$ and $s = p$. To the second term on the right-hand side of (7.74) we can apply this same approach when $s = 2p - q \geq -1$:

$$\begin{aligned} T_2 &= \alpha^{\frac{q-1}{p}} b c^{\frac{1-q}{p}} \sum_{i=1}^{n} \frac{(ih)^{2p-q}}{[1 + (ih)^p]^2} h \\ &= \begin{cases} \alpha^{\frac{q-1}{p}} b c^{\frac{1-q}{p}} [I_{p,2}^{2p-q} + o(1)] & \text{if } 2p - q > -1, \\ \alpha^2 b c^{-2} [(-\log \alpha)/p + \mathcal{O}(1)] & \text{if } 2p - q = -1 \end{cases} \end{aligned} \tag{7.76}$$

as $h \to 0, n \to \infty$, and $nh \to \infty$. On the other hand, when $2p - q < -1$ we can write

$$T_2 = \alpha^2 \frac{b}{c^2} \sum_{i=1}^{n} \frac{i^{2p-q}}{(1 + \frac{\alpha}{c} i^p)^2} = \alpha^2 (|||f_{\text{true}}|||_{K^*K}^2 + o(1)), \tag{7.77}$$

where $|||f_{\text{true}}|||_{K^*K}^2$ is given in (7.55). Combining (7.74)–(7.77), we obtain

$$\begin{aligned} E(\|e_\alpha^{\text{Tikh}}\|_{\mathcal{H}}^2) &= \|f_{\text{Null}(K_n)}\|_{\mathcal{H}}^2 + \frac{\sigma^2}{n} \alpha^{\frac{-p-1}{p}} c^{\frac{1}{p}} \left[I_{p,2}^p + o(1) \right] \\ &+ \begin{cases} \alpha^{\frac{q-1}{p}} b c^{\frac{1-q}{p}} [I_{p,2}^{2p-q} + o(1)] & \text{if } 2p - q > -1, \\ \alpha^2 b c^{-2} [(-\log \alpha)/p + \mathcal{O}(1)] & \text{if } 2p - q = -1, \\ \alpha^2 (|||f_{\text{true}}|||_{K^*K}^2 + o(1)) & \text{if } 2p - q < -1 \end{cases} \end{aligned} \tag{7.78}$$

as $\alpha \to 0, n \to \infty$, and $n\alpha \to \infty$. Similarly,

$$\begin{aligned} \frac{1}{2n} \frac{d}{d\alpha} E(\|e_\alpha^{\text{Tikh}}\|_{\mathcal{H}}^2) &= \alpha \sum_{i=1}^{n} \frac{s_{in}^2 f_{in}^2}{(s_{in}^2 + n\alpha)^3} - \sigma^2 \sum_{i=1}^{n} \frac{s_{in}^2}{(s_{in}^2 + n\alpha)^3} \\ &= -\sigma^2 \alpha^{-2} \left(\frac{\alpha}{n} \right)^{-\frac{1}{p}} c^{-\frac{1}{p}} [I_{p,3}^{2p} + o(1)] \\ &+ \begin{cases} \alpha^{\frac{q-p-1}{p}} b c^{\frac{1-q}{p}} [I_{p,3}^{2p-q} + o(1)] & \text{if } 2p - q > -1, \\ \alpha b c^{-2} [(-\log \alpha)/p + \mathcal{O}(1)] & \text{if } 2p - q = -1, \\ \alpha [|||f|||_{K^*K}^2 + o(1)] & \text{if } 2p - q < -1 \end{cases} \end{aligned} \tag{7.79}$$

as $\alpha \to 0$, $n \to \infty$, and $n\alpha \to 0$.

Theorem 7.21 (Minimizer of Estimation Error for Tikhonov Regularization). *Let* $\alpha_e = \arg\min_{\alpha \geq 0} E(\|e_\alpha^{\text{Tikh}}\|_{\mathcal{H}}^2)$, *the minimizer of the expected value of the squared norm of the estimation error for Tikhonov regularization. Then as* $n \to \infty$,

$$\alpha_e \simeq \begin{cases} C_1 \left(\frac{\sigma^2}{n}\right)^{\frac{p}{p+q}} & \text{if } 2p - q > -1, \\ \beta_e & \text{if } 2p - q = -1, \\ C_2 \left(\frac{\sigma^2}{n}\right)^{\frac{p}{3p+1}} & \text{if } 2p - q < -1, \end{cases} \tag{7.80}$$

where β_e *is the solution to*

$$-\beta^{\frac{3p+1}{p}} \log \beta = \frac{\sigma^2}{n} b^{-1} c^2 p \, I_{p,3}^{2p},$$

and

$$C_1 = \left(\frac{c^{q/p} I_{p,3}^{2p}}{b I_{p,3}^{2p-q}}\right)^{\frac{p}{p+q}}, \qquad C_2 = \left(\frac{c^{1/p} I_{p,3}^{2p}}{\|\|f_{\text{true}}\|\|_{K^*K}^2}\right)^{\frac{p}{3p+1}}.$$

Moreover, the value of the estimation error at the minimizer satisfies

$$E(\|e_{\alpha_e}^{\text{Tikh}}\|_{\mathcal{H}}^2) \simeq \|f_{\text{Null}(K_n)}\|_{\mathcal{H}}^2 + \begin{cases} C_3 \left(\frac{\sigma^2}{n}\right)^{\frac{q-1}{p+q}} & \text{if } 2p - q > -1, \\ C_4 \beta_e^{-p-1} & \text{if } 2p - q = -1, \\ C_5 \left(\frac{\sigma^2}{n}\right)^{\frac{2p}{3p+1}} & \text{if } 2p - q < -1. \end{cases} \tag{7.81}$$

Proof. Expression (7.80) comes from solving the equation $\frac{d}{d\alpha} E(\|e_\alpha^{\text{Tikh}}\|_{\mathcal{H}}^2) = 0$ for α. From (7.80), in case $2p - q > -1$, $h = (\alpha_e/c)^{1/p}$ is proportional to $n^{-1/(p+q)}$, and nh is proportional to $n^{(p+q-1)/(p+q)}$. Since we assume $p, q > 1$, then $h \to 0$, and $nh \to \infty$ as $n \to \infty$. Hence the $o(1)$ terms in (7.78) tend to zero as $n \to \infty$ in this case. The same holds in case $2p - q \leq -1$. This validates the asymptotic rate in (7.80), provided that $0 < \alpha_e < \infty$. Finally, evaluating at $\alpha = 0$, $E(\|e_0^{\text{Tikh}}\|_{\mathcal{H}}^2) \to \infty$ as $n \to \infty$ and $\lim_{\alpha \to \infty} E(\|e_\alpha^{\text{Tikh}}\|_{\mathcal{H}}^2) = \|f_{\text{true}}\|_{\mathcal{H}}^2$, while from (7.81), $E(\|e_{\alpha_e}^{\text{Tikh}}\|_{\mathcal{H}}^2) \to 0$ as $n \to \infty$, confirming that $0 < \alpha_e < \infty$. □

Remark 7.22. Comparing (7.80) with the corresponding expression (7.65) for TSVD, we see that when $2p - q > -1$, the respective regularization parameters that minimize the expected squared estimation error for TSVD and Tikhonov are both proportional to $(\sigma^2/n)^{p/(p+q)}$. However, when $2p - q < -1$, the Tikhonov regularization parameter α_e is proportional to $(\sigma^2/n)^{p/(3p+1)}$, an expression that is independent of the parameter q specifying the decay rate of the Fourier coefficients of the true solution f_{true}; see (7.53). The condition that $2p - q < -1$ implies that f_{true} lies in the range of $K_n^* K_n$. This quantitative change in behavior also occurs in the deterministic analysis of Tikhonov regularization (see [35, section 4.2] or [70, pp. 40–41]) and is called saturation. Note that saturation does not occur for TSVD regularization, either in the deterministic setting or in the stochastic setting.

We next examine the predictive error for Tikhonov regularization. From (7.46) and

(7.52),

$$E\left(\frac{1}{n}\|\mathbf{p}_\alpha^{\text{Tikh}}\|^2\right) = \sum_{i=1}^{n}\left(\frac{\alpha}{s_{in}^2+n\alpha}\right)^2 \frac{s_{in}^2}{n} f_{in}^2 + \frac{\sigma^2}{n}\sum_{i=1}^{n}\left(\frac{s_{in}^2}{s_{in}^2+n\alpha}\right)^2. \tag{7.82}$$

If a minimizer occurs for $0 < \alpha < \infty$, then, proceeding as in equation (7.79),

$$\begin{aligned} 0 = \frac{1}{2n}\frac{d}{d\alpha}E(\|\mathbf{p}_\alpha^{\text{Tikh}}\|^2) &= n\alpha\sum_{i=1}^{n}\frac{s_{in}^4 f_{in}^2}{(s_{in}^2+n\alpha)^3} - \sigma^2\sum_{i=1}^{n}\frac{s_{in}^4}{(s_{in}^2+n\alpha)^3} \\ &= \alpha b c^{-1}\sum_{i=1}^{n}\frac{i^{p-q}}{(1+\frac{\alpha}{c}i^p)^3} - \frac{\sigma^2}{n}c^{-1}\sum_{i=1}^{n}\frac{i^p}{(1+\frac{\alpha}{c}i^p)^3} \qquad (7.83) \\ &= -\frac{\sigma^2}{n}\alpha^{\frac{-p-1}{p}}c^{\frac{1}{p}}[I_{p,3}^p + o(1)] \\ &\quad + \begin{cases} \alpha^{\frac{q-1}{p}}bc^{\frac{1-q}{p}}[I_{p,3}^{p-q}+o(1)] & \text{if } p-q > -1, \\ \alpha bc^{-1}[(-\log\alpha)/p + \mathcal{O}(1)] & \text{if } p-q = -1, \\ \alpha[|||f_{\text{true}}|||_{K^*}^2 + o(1)] & \text{if } p-q < -1, \end{cases} \qquad (7.84) \end{aligned}$$

where we define

$$|||f_{\text{true}}|||_{K^*}^2 = \frac{b}{c}\sum_{i=1}^{\infty} i^{p-q} = \lim_{n\to\infty}\sum_{i=1}^{n}\frac{f_{in}^2}{s_i^2}. \tag{7.85}$$

Note that $|||f_{\text{true}}|||_{K^*} < \infty$ is equivalent to the source condition $f_{\text{true}} \in \text{Range}(K^*)$.

Theorem 7.23 (Minimizer of Predictive Risk for Tikhonov Regularization). *Let* $\alpha_{\text{pred}} = \arg\min_{\alpha\geq 0} E(\frac{1}{n}\|\mathbf{p}_\alpha^{\text{Tikh}}\|^2)$, *the minimizer of the expected predictive risk for Tikhonov regularization. As* $n \to \infty$,

$$\frac{\alpha_{\text{pred}}}{n} \simeq \begin{cases} C_1^{\text{pred}}\left(\frac{\sigma^2}{n}\right)^{\frac{p}{p+q}} & \text{if } p-q > -1, \\ \beta_{\text{pred}} & \text{if } p-q = -1, \\ C_2^{\text{pred}}\left(\frac{\sigma^2}{n}\right)^{\frac{p}{2p+1}} & \text{if } p-q < -1, \end{cases} \tag{7.86}$$

where β_{pred} *is the solution to*

$$-\beta^{\frac{2p+1}{p}}\log\beta = \frac{\sigma^2}{n}b^{-1}c^{\frac{p+1}{p}}p\,I_{p,3}^p$$

and

$$C_1^{\text{pred}} = b^{-1}c^{q/p}\frac{I_{p,3}^p}{I_{p,3}^{p-q}}, \quad C_2^{\text{pred}} = \frac{c^{1/p}I_{p,3}^p}{|||f_{\text{true}}|||_{K^*}^2}.$$

Remark 7.24. From equation (7.86) we see that saturation, the qualitative change in behavior of the regularization parameter, occurs for the Tikhonov predictive error when $p - q = -1$. In contrast, for the Tikhonov estimation error, saturation occurs when $p - 2q = -1$; see (7.80) and Remark 7.22. The condition $p - q < -1$ corresponds to $f_{\text{true}} \in \text{Range}(K^*)$ rather than $f_{\text{true}} \in \text{Range}(K^*K)$, the case with the Tikhonov estimation error.

7.6.4 Analysis of the Discrepancy Principle

To analyze the discrepancy principle (see section 7.3), we will study the behavior of solutions to the nonlinear equation

$$
\begin{aligned}
0 = F(\alpha) &\stackrel{\text{def}}{=} E\left(\frac{1}{n}||\mathbf{r}_\alpha||^2\right) - \sigma^2 \\
&= \sum_{i=1}^{n} (w_\alpha(s_{in}^2) - 1)^2 \frac{s_{in}^2}{n} f_{in}^2 - \frac{\sigma^2}{n} \sum_{i=1}^{n} \left[2w_\alpha(s_{in}^2) - w_\alpha^2(s_{in}^2)\right].
\end{aligned} \tag{7.87}
$$

The second equality follows from (7.47).

Discrepancy Principle Analysis for TSVD

Applying TSVD regularization with index parameter m and the rate assumptions (7.49) and (7.53), we obtain from (7.87) (see Exercise 7.20),

$$
0 = E\left(\frac{1}{n}||\mathbf{r}_m^{\text{TSVD}}||^2\right) - \sigma^2 = bc \sum_{i=m+1}^{n} i^{-p-q} - \frac{\sigma^2}{n} m. \tag{7.88}
$$

The following result is obtained using techniques from section 7.6.2.

Theorem 7.25 (Regularization Parameter Choice for the Discrepancy Principle Applied to TSVD). *Let m_{discrep} denote the solution to* (7.88)*, obtained when the discrepancy principle is used with TSVD regularization. Then*

$$
m_{\text{discrep}} \simeq \left[\frac{bc}{p+q-1}\right]^{\frac{1}{p+q}} \left(\frac{\sigma^2}{n}\right)^{-\frac{1}{p+q}}. \tag{7.89}
$$

Remark 7.26. The discrepancy principle parameter choice (7.89) behaves in a manner similar to the estimation error minimizer (7.63). In particular, both share the rate $(\sigma^2/n)^{-1/(p+q)}$, but the rate constants differ by a factor of $(p+q-1)^{1/(p+q)}$, where $p, q > 1$. Hence, the discrepancy principle should be expected to give a truncation m which is smaller than the minimizer of the estimation error. This tends to yield a regularized solution that is overly smooth.

Similar results are obtained when the discrepancy principle is used to select the Tikhonov regularization parameter.

Discrepancy Principle Analysis for Tikhonov Regularization

From (7.87), (7.52), and the rate assumptions (7.49) and (7.53),

$$
\begin{aligned}
0 = E\left(\frac{1}{n}||\mathbf{r}_\alpha^{\text{Tikh}}||^2\right) - \sigma^2 &= \sum_{i=1}^{n} \left(\frac{n\alpha}{s_{in}^2 + n\alpha}\right)^2 \frac{s_{in}^2}{n} f_{in}^2 - \frac{\sigma^2}{n} \sum_{i=1}^{n} \frac{2n\alpha s_{in}^2 + s_{in}^4}{(s_{in}^2 + n\alpha)^2} \\
&= \alpha^2 bc^{-1} \sum_{i=1}^{n} \frac{i^{p-q}}{(1 + \frac{\alpha}{c} i^p)^2} - \frac{\sigma^2}{n} \sum_{i=1}^{n} \frac{2\frac{\alpha}{c} i^p + 1}{(1 + \frac{\alpha}{c} i^p)^2} \\
&= -\frac{\sigma^2}{n} \alpha^{-\frac{1}{p}} c^{\frac{1}{p}} \left[2I_{p,2}^p + I_{p,2}^0 + o(1)\right]
\end{aligned}
$$

$$+\begin{cases} \alpha^{\frac{p+q-1}{p}} bc^{\frac{1-q}{p}} \left[I_{p,2}^{p-q} + o(1) \right] & \text{if } p-q > -1, \\ \alpha^2 bc^{-1}[(-\log\alpha)/p + \mathcal{O}(1)] & \text{if } p-q = -1, \\ \alpha^2[|||f_{\text{true}}|||_{K^*}^2 + o(1)] & \text{if } p-q < -1. \end{cases} \tag{7.90}$$

Theorem 7.27 (Regularization Parameter Choice for the Discrepancy Principle Applied to Tikhonov Regularization). *Let α_{discrep} denote the solution to* (7.90)*, obtained when the discrepancy principle is used with Tikhonov regularization. Then*

$$\alpha_{\text{discrep}} \simeq \begin{cases} C_1^{\text{discrep}} \left(\frac{\sigma^2}{n}\right)^{\frac{p}{p+q}} & \text{if } p-q > -1, \\ \beta_{\text{discrep}} & \text{if } p-q = -1, \\ C_2^{\text{discrep}} \left(\frac{\sigma^2}{n}\right)^{\frac{p}{2p+1}} & \text{if } p-q < -1, \end{cases} \tag{7.91}$$

where β_{discrep} is the solution to

$$-\beta^{\frac{2p+1}{p}} \log\beta = \frac{\sigma^2}{n} b^{-1} c^{\frac{p+1}{p}} p(2I_{p,2}^p + I_{p,2}^0)$$

and

$$C_1^{\text{discrep}} = \left(\frac{(2I_{p,2}^p + I_{p,2}^0)}{I_{p,2}^{p-q}} \frac{c^{q/p}}{b} \right)^{\frac{p}{p+q}}, \qquad C_2^{\text{discrep}} = \left(\frac{c^{1/p}(2I_{p,2}^p + I_{p,2}^0)}{|||f_{\text{true}}|||_{K^*}^2} \right)^{\frac{p}{2p+1}}.$$

Remark 7.28. Comparing (7.91) and (7.86), we see that the regularization parameters α_P and α_{discrep} both decay at the same rate as $n \to \infty$.

7.6.5 Analysis of GCV

To simplify notation, define the expected value of the predictive risk (see (7.40)):

$$P(\alpha) \stackrel{\text{def}}{=} E\left(\frac{1}{n}||\mathbf{p}_\alpha||^2\right). \tag{7.92}$$

Also define the expected value of the mean squared residual (see (7.41)):

$$R(\alpha) \stackrel{\text{def}}{=} E\left(\frac{1}{n}||\mathbf{r}_\alpha||^2\right). \tag{7.93}$$

From equations (7.46)–(7.47) we have

$$R(\alpha) = P(\alpha) - 2\frac{\sigma}{n}\,\text{trace}A_\alpha + \sigma^2. \tag{7.94}$$

The expected value of the GCV function can be expressed as

$$EV(\alpha) \stackrel{\text{def}}{=} E\left(\frac{\frac{1}{n}||\mathbf{r}_\alpha||^2}{[\frac{1}{n}\text{trace}(I - A_\alpha)]^2}\right) = \frac{R(\alpha)}{[1 - \frac{1}{n}\text{trace}A_\alpha]^2}. \tag{7.95}$$

Let $G(\alpha)$ denote the denominator and assume differentiability with respect to α. A necessary condition for an unconstrained minimizer is that $\frac{d}{d\alpha}EV(\alpha) = 0$. Applying the quotient rule for differentiation and then multiplying by $G(\alpha) > 0$ gives

$$\begin{aligned} 0 &= R'(\alpha) - R(\alpha)\frac{G'(\alpha)}{G(\alpha)} \\ &= P'(\alpha) + \epsilon(\alpha), \end{aligned} \tag{7.96}$$

where

$$\epsilon(\alpha) \stackrel{\text{def}}{=} -\frac{2}{n}\left(\frac{d}{d\alpha}\text{trace}A_\alpha\right)\left[\sigma^2 - \frac{R(\alpha)}{1-\frac{1}{n}\text{trace}A_\alpha}\right]. \tag{7.97}$$

If we can show that $\epsilon(\alpha)$ is small relative to $P'(\alpha)$, then the minimizer of $EV(\alpha)$ will be asymptotically equal to the minimizer of $P(\alpha)$. The keys are to establish that $\frac{1}{n}\text{trace}A_\alpha = o(1)$ and $R(\alpha) = \sigma^2 + o(1)$.

GCV Analysis for Tikhonov Regularization

Consider the case of Tikhonov regularization. Using the techniques of section 7.6.3, as $h = (\alpha/c)^{1/p} \to 0$, $n \to \infty$, and $nh \to \infty$,

$$\frac{1}{n}\text{trace}A_\alpha = \frac{1}{n}\sum_{i=1}^{n}\frac{s_{in}^2}{s_{in}^2 + n\alpha} = \frac{1}{n\alpha^{1/p}}\left[c^{1/p}I_{p,1}^0 + o(1)\right] = o(1),$$

and hence

$$\frac{1}{1-\frac{1}{n}\text{trace}A_\alpha} = \frac{1}{1-o(1)} = 1 + o(1). \tag{7.98}$$

Similarly,

$$\frac{d}{d\alpha}\text{trace}A_\alpha = -n\sum_{i=1}^{n}\frac{s_{in}^2}{s_{in}^2 + n\alpha} = -\alpha^{\frac{-p-1}{p}}c^{1/p}[I_{p,2}^p + o(1)], \tag{7.99}$$

and from (7.90)

$$R(\alpha) = \sigma^2 + o(1). \tag{7.100}$$

Combining (7.97)–(7.100), we obtain

$$\epsilon(\alpha) = n^{-1}\alpha^{\frac{-p-1}{p}} \times o(1) \tag{7.101}$$

as $h \to 0$, $n \to \infty$, and $nh \to \infty$. From expression (7.84), we see that $P'(\alpha)$ is asymptotically proportional to $n^{-1}\alpha^{\frac{-p-1}{p}}$. Thus $\epsilon(\alpha) = P'(\alpha) \times o(1)$, and $\epsilon(\alpha)$ is small relative to $P'(\alpha)$ as $h \to 0$, $n \to \infty$, and $nh \to \infty$. Then from (7.96) we obtain the following result.

Theorem 7.29 (GCV for Tikhonov Regularization). *Let*

$$\alpha_V = \arg\min_{\alpha>0} EV^{\text{Tikh}}(\alpha),$$

the minimizer of the expected value of the GCV function for Tikhonov regularization. Then $\alpha_V \simeq \alpha_{\text{pred}}$ *as* $n \to \infty$, *where* α_{pred} *denotes the minimizer of the expected value of the predictive risk for Tikhonov regularization (see* (7.86)*).*

GCV Analysis for TSVD Regularization

The analysis for TSVD regularization is somewhat similar to that for Tikhonov regularization, although $EV(\alpha)$ is no longer differentiable. As before, take the index parameter m to be the regularization parameter. The key idea is to apply Lemma 7.14 and replace (7.96) by

$$\Delta^- R(m) - R(m-1)\frac{\Delta^- G(m)}{G(m-1)} \le 0, \tag{7.102}$$

$$\Delta^+ R(m) - R(m)\frac{\Delta^+ G(m)}{G(m)} \ge 0, \tag{7.103}$$

where now $R(m) = bc\sum_{i=1}^{m} i^{-p-q} + \sigma^2[1 - m/n]$ and $G(m) = [1 - m/n]^2$. See [113] for details.

Theorem 7.30 (GCV for TSVD Regularization). *Let*

$$m_V = \arg\min_{m\in\mathbf{Z}(n)} EV^{\mathrm{TSVD}}(m),$$

the minimizer of the expected value of the GCV function for TSVD regularization. Then $m_V \simeq m_e$ as $n \to \infty$, where m_e denotes the minimizer of the expected value of the squared estimation error for TSVD; see (7.63).

7.6.6 Analysis of the L-Curve Method

To carry out an analysis of the L-curve method, we need to modify expressions (7.30)–(7.32). In particular, define the expected L-curve components:

$$R(\alpha) \stackrel{\text{def}}{=} E\left(\frac{1}{n}||\mathbf{r}_\alpha||^2\right), \qquad S(\alpha) \stackrel{\text{def}}{=} E\left(||f_\alpha||_{\mathcal{H}}^2\right). \tag{7.104}$$

Now consider the case of Tikhonov regularization. Taking the representation (7.36) for the regularized solution and the filter function (7.52), we obtain

$$R(\alpha) = n\alpha^2 \sum_{i=1}^{n} \frac{1}{(s_{in}^2 + n\alpha)^2} s_{in}^2 f_{in}^2 + \sigma^2 n\alpha^2 \sum_{i=1}^{n} \frac{1}{(s_{in}^2 + n\alpha)^2}, \tag{7.105}$$

$$S(\alpha) = \sum_{i=1}^{n} \frac{s_{in}^4}{(s_{in}^2 + n\alpha)^2} f_{in}^2 + \sigma^2 \sum_{i=1}^{n} \frac{s_{in}^2}{(s_{in}^2 + n\alpha)^2}. \tag{7.106}$$

As before,

$$R'(\alpha) = -\alpha S'(\alpha), \tag{7.107}$$

but now

$$S'(\alpha) = -2n \sum_{i=1}^{n} \frac{s_{in}^4 f_{in}^2}{(s_{in}^2 + n\alpha)^3} - 2\sigma^2 n \sum_{i=1}^{n} \frac{s_{in}^4}{(s_{in}^2 + n\alpha)^3}. \tag{7.108}$$

The expression (7.32) for $\kappa(\alpha)$ in terms of $R(\alpha)$, $S(\alpha)$, and $S'(\alpha)$ is unchanged.

Assume that $\alpha = \alpha(n)$ is selected so that as $n \to \infty$,

$$\alpha(n) \to 0, \tag{7.109}$$

$$n\,\alpha(n) \to \infty, \tag{7.110}$$

$$R(\alpha(n)) \simeq \sigma^2, \tag{7.111}$$

$$S(\alpha(n)) \simeq ||f_{\text{true}}||_{\mathcal{H}}^2. \tag{7.112}$$

These conditions must be satisfied if the assumptions in Remark 7.12 hold and $f_{\alpha(n)}$ is to converge to f_{true} in mean square. See Exercise 7.25. Substituting these conditions into (7.32) gives

$$\kappa(\alpha(n)) \simeq \frac{||f_{\text{true}}||_{\mathcal{H}}^4}{\sigma^2 S'(\alpha(n))} \quad \text{as} \quad n \to \infty. \tag{7.113}$$

But then by the analysis in section 7.6.3, and equation (7.79) in particular,

$$|S'(\alpha(n))| \le C\alpha(n)^{-2-1/p} n^{1+1/p}. \tag{7.114}$$

Consequently, $S'(\alpha(n)) \to \infty$ as $n \to \infty$. By (7.113), this implies that $\kappa(\alpha(n)) \to 0$ as $n \to \infty$. The following result follows immediately.

Theorem 7.31. *Let $\alpha_L(n)$ maximize the curvature $\kappa(\alpha)$ of the expected Tikhonov L-curve, given in* (7.32) *with R and S as in* (7.105)–(7.106). *If $\kappa(\alpha_L(n))$ does not tend to zero as $n \to \infty$, then $E(\|f_{\alpha_L(n)} - f_{\text{true}}\|_{\mathcal{H}}^2)$ does not converge to zero.*

This means that either the expected L-curve becomes very flat as n becomes large, or the corner of the L-curve does not give a value of α that yields mean square convergence.

Remark 7.32. Note that the L-curve method is not directly applicable to select a regularization parameter for TSVD, since the TSVD filter (7.51) has jump discontinuities. However, if one replaces derivatives by discrete differences, as in section 7.6.2, then one can carry out an analysis very similar to that for Tikhonov regularization. In particular, let the regularization parameter be the number of singular components m that are not filtered out, and let $F'(m) = F(m+1) - F(m)$. One can derive expressions for $R(m)$, $S(m)$, and $S'(m)$ analogous to (7.105)–(7.108). Under assumptions analogous to (7.109)–(7.112), one can establish either that the TSVD L-curve becomes very flat or that the regularization parameter value m_L corresponding to the corner does not yield mean square convergence.

7.7 A Comparison of Methods

We now introduce some notation to describe the asymptotic convergence results obtained in section 7.6, derived under the assumptions in Remark 7.12. For a particular regularization method, define the expected optimal regularization parameters for estimation and prediction to be

$$\alpha_e(n) = \arg\min_{\alpha} E(\|e_\alpha\|_{\mathcal{H}}^2), \qquad \alpha_p(n) = \arg\min_{\alpha} E(\|\mathbf{p}_\alpha\|^2/n). \tag{7.115}$$

Note that both of these optimal parameters depend on the data size n in model (7.34). Let $\alpha(n)$ denote the expected parameter obtained from a particular regularization parameter selection method; e.g., $\alpha_V(n)$ denotes the minimizer of the expected GCV functional (7.95).

Definition 7.33. A regularization parameter selection method is e-optimal if there exists n_0 for which

$$\alpha(n) = \alpha_e(n) \quad \text{whenever } n \geq n_0.$$

The method is asymptotically e-optimal if

$$\alpha(n) \simeq \alpha_e(n) \quad \text{as} \quad n \to \infty.$$

The method is order e-optimal if there exists a positive constant r, called the order constant, for which

$$\alpha(n) \simeq r\, \alpha_e(n) \quad \text{as} \quad n \to \infty.$$

The method is convergent if

$$E(\|e_{\alpha(n)}\|_{\mathcal{H}}^2) \to 0 \quad \text{as} \quad n \to \infty.$$

Otherwise, the method is nonconvergent. Analogous definitions are obtained when e for estimation is replaced by p for prediction.

Table 7.1.

	TSVD	Tikhonov
UPRE	[Thm. 7.18] p-optimal [Thm. 7.16] e-optimal	[Thm. 7.23] p-optimal [Thm. 7.21] order e-optimal for $s > -1$ [Thm. 7.21] e-convergent for $s \leq -1$
GCV	[Thm. 7.30] asymptotically p-optimal [Thm. 7.30] asymptotically e-optimal	[Thm. 7.29] asymptotically p-optimal [Thm. 7.29] order e-optimal for $s > -1$ [Thm. 7.29] e-convergent for $s \leq -1$
Discrepancy principle	[Thm. 7.25] order p-optimal [Thm. 7.25] order e-optimal	[Thm. 7.27] order p-optimal [Thm. 7.27] order e-optimal for $s > -1$ [Thm. 7.27] e-convergent for $s \leq -1$
L-curve	[Remark 7.32] nonconvergent	[Thm. 7.31] nonconvergent

Table 7.1 summarizes the optimality properties of the methods previously analyzed. The two columns correspond to the two regularization methods, truncated SVD and Tikhonov regularization. The four rows correspond to the four regularization parameter selection methods that were considered.

The parameter $s = p - q$, where p quantifies the decay rate of the eigenvalues of $K_n^* K_n$ (see (7.49)), and q gives the decay rate of the generalized coefficients of f_{true} (see (7.53)). Recall from Remark 7.24 that the condition $s \leq -1$ corresponds to the source condition $f_{\text{true}} \in \text{Range}(K_n^*)$. Since K_n is typically a discretization of a smoothing operator, this is a smoothness condition; see Remark 7.11. If $s \leq -1$, then f_{true} is smooth; otherwise, f_{true} is rough.

The square brackets $[\cdot]$ indicate the theorems or remarks corresponding to the results in the table. Since the UPRE is an unbiased estimator for predictive risk, theorems giving convergence rates for predictive error provide rates for the UPRE method.

Table 7.1 shows that the UPRE method has the most desirable convergence properties. However, this method assumes discrete white noise in the data model (7.34), and it requires knowledge of the variance of the noise. The same holds for the discrepancy principle in the stochastic setting. The GCV method also assumes white noise, but it does not make use of prior knowledge of the variance. Hence, GCV is the method of choice in many practical applications. The L-curve method has been advocated for applications where no prior information about the noise is available. However, if such prior information is available, the analysis suggests that other methods should be used instead.

The proper use of prior statistical information about the noise in the data should be a requirement for the solution of practical ill-posed problems.

Exercises

7.1. Given the singular value $K = U\text{diag}(s_i)V^T$ decomposition and the linear filtering representation (7.16) for the regularization matrix R_α, compute a representation for the influence matrix $A_\alpha = KR_\alpha$, and use it to show that A_α is symmetric.

7.2. Use the results of Exercise 7.1 to verify equations (7.17) and (7.18).

7.3. Show that (7.23) implies (7.24).

7.4. Use the randomized trace estimate from section 7.1.3 to evaluate the UPRE and the

GCV function for the image deblurring application in section 7.2.1. How do these results compare with the exact computations?

7.5. Implement the discrepancy principle for the image deblurring application of section 7.2.1.

7.6. Given the representations in (7.30), derive (7.31).

7.7. Verify equation (7.32).

7.8. Implement the minimum bound method (see (7.33)) for the image deblurring application.

7.9. Verify equation (7.35).

7.10. Prove that if A is symmetric and U is orthogonal, then $\text{trace}(A) = \text{trace}(U^T AU)$.

7.11. Prove Proposition 7.8.

7.12. Verify equation (7.48).

7.13. Prove Proposition 7.15.

7.14. Prove Theorem 7.18. To do this, show that for sufficiently large n, neither $m = 0$ nor $m = n$ minimize $E(||\mathbf{p}_m^{\text{TSVD}}||^2/n)$.

7.15. Under the assumptions of Proposition 7.19, verify that the second and third terms on the right-hand side of (7.72) are both $\mathcal{O}(1)$.

7.16. Verify equation (7.73) using the substitution $t = 1/(1 + u^p)$ in (7.70).

7.17. Confirm the second equality in equation (7.77).

7.18. Confirm equation (7.79).

7.19. Compute the constants C_3, C_4, and C_5 in equation (7.81).

7.20. Verify equation (7.88).

7.21. Prove Theorem 7.25.

7.22. Prove Theorem 7.27.

7.23. Confirm equations (7.96)–(7.97).

7.24. Prove Theorem 7.30.

7.25. Assume that the conditions in Remark 7.12 hold and $E(||f_{\alpha(n)} - f_{\text{true}}||_{\mathcal{H}}^2) \to 0$ as $n \to \infty$. Verify that conditions (7.109)–(7.112) must be satisfied.

7.26. Carry out an analysis of the L-curve method for TSVD regularization, using information given in Remark 7.32.

7.27. Analyze the minimum bound method using the techniques of section 7.6. How does this method compare with the other methods of this section?

This page intentionally left blank

Chapter 8
Total Variation Regularization

In section 1.3 of Chapter 1 we provided a very brief introduction to total variation regularization. In this chapter we take a closer look at both computational and theoretical issues.

8.1 Motivation

In a real analysis course [100], one sometimes sees the following definition of the total variation (TV) of a function f defined on the interval $[0, 1]$:

$$\mathrm{TV}(f) \stackrel{\text{def}}{=} \sup \sum_i |f(x_i) - f(x_{i-1})|, \tag{8.1}$$

where the supremum is taken over all partitions $0 = x_0 < x_1 < \cdots < x_n = 1$ of the interval. If f is piecewise constant with a finite number of jump discontinuities, then $\mathrm{TV}(f)$ gives the sum of magnitudes of the jumps. If f is smooth, one can multiply and divide the right-hand side of (8.1) by $\Delta x_i = x_i - x_{i-1}$ and take the limit as the $\Delta x_i \to 0$ to obtain the representation

$$\mathrm{TV}(f) = \int_0^1 \left| \frac{df}{dx} \right| dx. \tag{8.2}$$

An obvious generalization of (8.2) to two space dimensions is

$$\mathrm{TV}(f) = \int_0^1 \int_0^1 |\nabla f| \, dx \, dy, \tag{8.3}$$

where $\nabla f = (\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y})$ denotes the gradient and $|(x, y)| = \sqrt{x^2 + y^2}$ denotes the Euclidean norm. An extension of this representation, valid even when f is not smooth, is

$$\mathrm{TV}(f) = \sup_{\vec{v} \in \mathcal{V}} \int_0^1 \int_0^1 f(x, y) \, \mathrm{div}\, \vec{v} \, dx \, dy, \tag{8.4}$$

where $\mathcal{V}$ consists of vector-valued functions $\vec{v} = (v_1(x, y), v_2(x, y))$ whose Euclidean norm is bounded by 1 and whose components v_i are continuously differentiable and vanish on the boundary of the unit square. $\mathrm{div}\, \vec{v} = \frac{\partial v_1}{\partial x} + \frac{\partial v_2}{\partial y}$ gives the divergence of $\vec{v}$. We will take (8.4) to be the definition of total variation; see [45] and section 8.4.

From the expression (8.4), one can develop theoretical properties of TV. For instance, one can establish that minimization of the Tikhonov–TV functional

$$T_\alpha(f) = \frac{1}{2}||Kf - g||^2 + \alpha\ \mathrm{TV}(f) \tag{8.5}$$

yields a regularization scheme in the sense of section 2.2 for the operator equation $Kf = g$; see [2] and section 8.4.

$\mathrm{TV}(f)$ can be interpreted geometrically as the lateral surface area of the graph of f. In particular, let S be a region with a smooth boundary ∂S contained within the unit square. Take $f(x, y) = H > 0$ for (x, y) in the interior of S and $f(x, y) = 0$ in the exterior. $\mathrm{TV}(f)$ is then the length of the boundary ∂S multiplied by the height H of the jump in f. For example, if S is the disk of radius $1/4$ centered at $(1/2, 1/2)$, then $\mathrm{TV}(f) = 2\pi \times 1/4 \times H$. With this geometric insight, one can begin to understand why total variation is an effective regularization functional. If f has many large amplitude oscillations, then it has large lateral surface area, and hence $\mathrm{TV}(f)$ is large. This is a property that TV shares with the more standard Sobolev H^1 "squared norm of the gradient" regularization functionals; see (2.47). Unlike the H^1 functional, with total variation one can effectively reconstruct functions with jump discontinuities. This is illustrated in one dimension in Figure 1.5. In two-dimensional image deblurring, total variation regularization tends to produce qualitatively correct reconstructions of blocky images [34]. By blocky, we mean the image is nearly piecewise constant with jump discontinuities, and the length of the curves on which the discontinuities occur is relatively small. The image in Figure 5.2 is blocky. The reconstruction in Figure 8.1, obtained with total variation regularization, does a much better job of preserving this blocky structure than do the reconstructions in Figure 5.3, which were generated using conventional regularization techniques.

We now turn our attention to numerical implementation.

8.2 Numerical Methods for Total Variation

We wish to obtain regularized solutions to operator equations $Kf = g$. In principle, this can be done by minimizing the Tikhonov–TV functional (8.5). However, the representations (8.2) and (8.3) are not suitable for the implementation of the numerical methods of Chapter 3, due to the nondifferentiability of the Euclidean norm at the origin. To overcome this difficulty, one can take an approximation to the Euclidean norm $|\mathbf{x}|$ like $\sqrt{|\mathbf{x}|^2 + \beta^2}$, where β is a small positive parameter. This yields the following approximation to $\mathrm{TV}(f)$, valid for a smooth function f defined on the unit interval in one dimension:

$$J_\beta(f) = \int_0^1 \sqrt{\left(\frac{df}{dx}\right)^2 + \beta^2}\, dx. \tag{8.6}$$

In two space dimensions, this becomes

$$J_\beta(f) = \int_0^1 \int_0^1 \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2 + \beta^2}\, dx\, dy. \tag{8.7}$$

In the following section we consider minimization of the functional

$$T(\mathbf{f}) = \frac{1}{2}||K\mathbf{f} - \mathbf{d}||^2 + \alpha J(\mathbf{f}), \tag{8.8}$$

where J is a discretization of an approximation to the one-dimensional TV functional like (8.6), $\mathbf{d}$ represents discrete data, and K is a matrix; see the example in section 1.1.

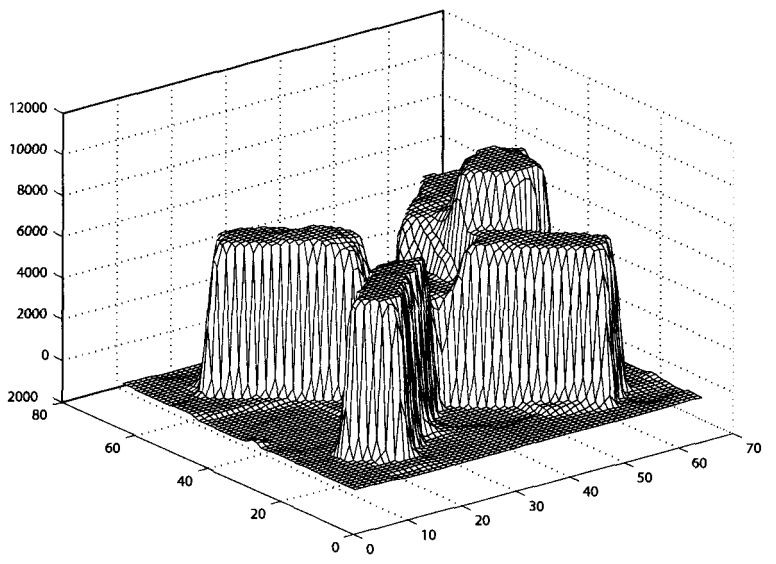


Figure 8.1. *Reconstructed image obtained with total variation regularization. The data for this reconstruction are described in section* 5.1.1. *A comparison with reconstructions in Figure* 5.3 *obtained with standard regularization techniques clearly shows that edge discontinuities and blocky structures are much better preserved with total variation.*

8.2.1 A One-Dimensional Discretization

To make the presentation less abstract, suppose $f(x)$ is a smooth function defined on the unit interval in $\mathbb{R}^1$ and $\mathbf{f} = (f_0, \ldots, f_n)$ with $f_i \approx f(x_i)$, $x_i = i\Delta x$, $\Delta x = 1/n$. Take the derivative approximation

$$D_i\mathbf{f} = (f_i - f_{i-1})/\Delta x, \qquad i = 1, \ldots, n. \tag{8.9}$$

Note the $(n+1) \times 1$ matrix representation, $D_i = [0, \ldots, 0, -1/\Delta x, 1/\Delta x, 0, \ldots, 0]$.

We assume a discretized penalty functional of the form

$$J(\mathbf{f}) = \frac{1}{2}\sum_{i=1}^{n} \psi\left((D_i\mathbf{f})^2\right)\Delta x, \tag{8.10}$$

where ψ is a smooth approximation to twice the square root function with the property

$$\psi'(t) > 0 \quad \text{whenever} \quad t > 0. \tag{8.11}$$

For example, the choice

$$\psi(t) = 2\sqrt{t + \beta^2} \tag{8.12}$$

leads to an approximation to (8.6). Another example [33] is

$$\psi(t) = \begin{cases} \frac{t}{\epsilon}, & t \le \epsilon^2, \\ 2\sqrt{t} - \epsilon, & t > \epsilon^2. \end{cases} \tag{8.13}$$

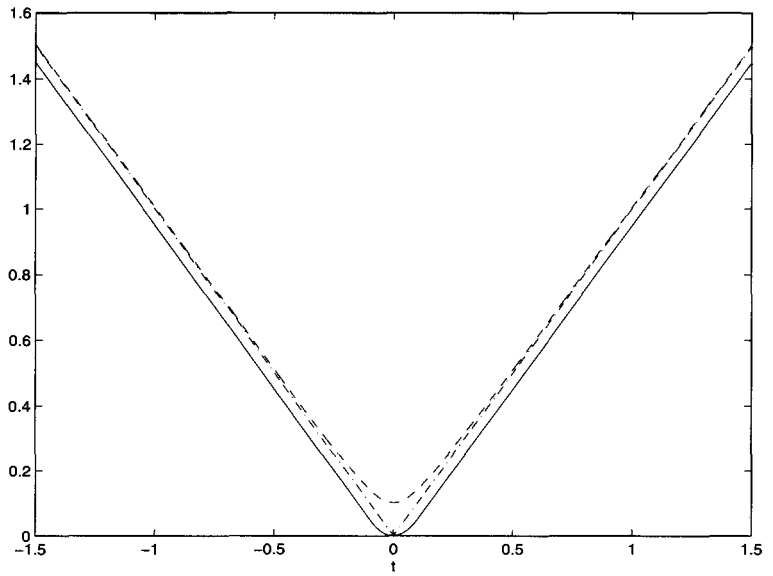


Figure 8.2. *Smooth approximations to the absolute value function. The dot-dashed curve represents the absolute value function; the solid curve represents the Huber function* $\varphi_\epsilon(t) = \psi_\epsilon(t^2)/2$ *(see* (8.13)*) with parameter* $\epsilon = 0.1$*; and the dashed curve represents the approximation* $\varphi_\beta(t) = \sqrt{t^2 + \beta^2}$ *(see* (8.12)*) with parameter* $\beta = 0.1$.

The composite function $\frac{1}{2}\psi(t^2)$ is the well-known Huber function from robust statistics. See Figure 8.2 for plots of (8.12) and (8.13).

To minimize (8.8) using the optimization techniques of Chapter 3, we need the gradient of J. For any $\mathbf{v} \in \mathbb{R}^{n+1}$,

$$\frac{d}{d\tau} J(\mathbf{f} + \tau\mathbf{v}) = \sum_{i=1}^{n} \psi'\left([D_i\mathbf{f}]^2\right)\ (D_i\mathbf{f})(D_i\mathbf{v})\Delta x \tag{8.14}$$

$$= \Delta x\ (D\mathbf{v})^T\ \mathrm{diag}(\psi'(\mathbf{f}))\ (D\mathbf{f})$$

$$= \langle \Delta x\ D^T\ \mathrm{diag}(\psi'(\mathbf{f}))\ D\mathbf{f}, \mathbf{v}\rangle, \tag{8.15}$$

where $\mathrm{diag}(\psi'(\mathbf{f}))$ denotes the $n \times n$ diagonal matrix whose ith diagonal entry is $\psi'\left((D_i\mathbf{f})^2\right)$, D is the $n \times (n+1)$ matrix whose ith row is D_i (see (8.9)), and $\langle \cdot, \cdot \rangle$ denotes the Euclidean inner product on $\mathbb{R}^{n+1}$. From this we obtain the gradient

$$\mathrm{grad}\, J(\mathbf{f}) = L(\mathbf{f})\mathbf{f}, \tag{8.16}$$

where

$$L(\mathbf{f}) = \Delta x\ D^T\ \mathrm{diag}(\psi'(\mathbf{f}))\ D \tag{8.17}$$

is a symmetric $(n+1) \times (n+1)$ matrix. $L(\mathbf{f})$ is positive semidefinite provided condition (8.11) holds.

To obtain the Hessian of J, from (8.14),

$$\frac{\partial^2 J}{\partial\tau\partial\xi}(\mathbf{f}+\tau\mathbf{v}+\xi\mathbf{w})|_{\tau,\xi=0} = \sum_{i=1}^{n} \psi'([D_i\mathbf{f}]^2)(D_i\mathbf{w})(D_i\mathbf{v})\Delta x$$

$$+\sum_{i=1}^{n} \psi''\left([D_i\mathbf{f}]^2\right)(D_i\mathbf{f})(D_i\mathbf{v})\,2(D_i\mathbf{f})(D_i\mathbf{w})\Delta x$$

$$= \langle \Delta x\,[\mathrm{diag}(\psi'(\mathbf{f})) + \mathrm{diag}(2(D\mathbf{f})^2\psi_i''(\mathbf{f}))]\,D\mathbf{v}, D\mathbf{w}\rangle, \tag{8.18}$$

where $\mathrm{diag}(2(D\mathbf{f})^2\psi''(\mathbf{f}))$ denotes the $n \times n$ diagonal matrix whose ith diagonal entry is $2(D_i\mathbf{f})^2\,\psi''\left([D_i\mathbf{f}]^2\right)$. Consequently,

$$\mathrm{Hess}\,J(\mathbf{f}) = L(\mathbf{f}) + L'(\mathbf{f})\mathbf{f}, \tag{8.19}$$

where $L(\mathbf{f})$ is given in (8.17) and

$$L'(\mathbf{f})\mathbf{f} = \Delta x\; D^T\,\mathrm{diag}(2(D\mathbf{f})^2\psi''(\mathbf{f}))\;D. \tag{8.20}$$

From (8.8) and (8.16)–(8.17), we obtain the gradient of the penalized least squares cost functional,

$$\mathrm{grad}\,T(\mathbf{f}) = K^T(K\mathbf{f}-\mathbf{d}) + \alpha L(\mathbf{f})\mathbf{f}. \tag{8.21}$$

From this and (8.19)–(8.20), we obtain the Hessian,

$$\mathrm{Hess}\;T(\mathbf{f}) = K^TK + \alpha L(\mathbf{f}) + \alpha L'(\mathbf{f})\mathbf{f}. \tag{8.22}$$

8.2.2 A Two-Dimensional Discretization

We now consider minimization of the penalized least squares functional (8.8), where J is a discretization of a two-dimensional total variation approximation like (8.7). The matrix K is a discretization of a linear operator which acts on functions of two variables, and the vector $\mathbf{d}$ denotes discrete data. See, for example, section 5.1.

Suppose $f = f_{ij}$ is defined on an equispaced grid in two space dimensions, $\{(x_i, y_j) \mid x_i = i\Delta x, y_j = j\Delta y,\ i = 0,\ldots,n_x, j = 0,\ldots,n_y\}$. In a manner analogous to the one-dimensional case, we define the discrete penalty functional $J : \mathbb{R}^{(n_x+1)\times(n_y+1)} \to \mathbb{R}$ by

$$J(f) = \frac{1}{2}\sum_{i=1}^{n_x}\sum_{j=1}^{n_y} \psi\left((D_{ij}^x f)^2 + (D_{ij}^y f)^2\right), \tag{8.23}$$

where

$$D_{ij}^x f = \frac{f_{i,j} - f_{i-1,j}}{\Delta x}, \qquad D_{ij}^y f = \frac{f_{i,j} - f_{i,j-1}}{\Delta y}. \tag{8.24}$$

To simplify notation, we dropped a factor of $\Delta x\,\Delta y$ from the right-hand side of (8.23). This factor can be absorbed in the regularization parameter α in (8.8). Gradient computations are similar to those in one dimension:

$$\frac{d}{d\tau}J(f+\tau v)|_{\tau=0} = \sum_{i=1}^{n_x}\sum_{j=1}^{n_y} \psi_{ij}'\left[(D_{ij}^x f)(D_{ij}^x v) + (D_{ij}^y f)(D_{ij}^y v)\right], \tag{8.25}$$

where $\psi_{ij}' = \psi'((D_{ij}^x f)^2 + (D_{ij}^y f)^2)$.

Now let $\mathbf{f} = \mathbf{vec}(f)$ and $\mathbf{v} = \mathbf{vec}(v)$, corresponding to lexicographical column ordering of the two-dimensional array components (see Definition 5.25); let D_x and D_y denote the resulting $n_x n_y \times (n_x + 1)(n_y + 1)$ matrices corresponding to the grid operators in (8.24); let $\mathrm{diag}(\psi'(\mathbf{f}))$ denote the $n_x n_y \times n_x n_y$ diagonal matrix whose diagonal entries are the ψ'_{ij}s; and let $\langle \cdot, \cdot \rangle$ denote the Euclidean inner product on $\mathbb{R}^{(n_x+1)(n_y+1)}$. Then

$$\frac{d}{d\tau} J(f + \tau v)|_{\tau=0} = \langle \mathrm{diag}(\psi'(\mathbf{f}))\ D_x\mathbf{f}, D_x\mathbf{v} \rangle + \langle \mathrm{diag}(\psi'(\mathbf{f}))D_y\mathbf{f}, D_y\mathbf{v} \rangle.$$

From this we obtain a gradient representation (8.16), but now

$$\begin{aligned} L(\mathbf{f}) &= D_x^T\, \mathrm{diag}(\psi'(\mathbf{f}))\, D_x + D_y^T\, \mathrm{diag}(\psi'(\mathbf{f}))\, D_y \\ &= [D_x^T\ D_y^T] \begin{bmatrix} \mathrm{diag}(\psi'(\mathbf{f})) & 0 \\ 0 & \mathrm{diag}(\psi'(\mathbf{f})) \end{bmatrix} \begin{bmatrix} D_x \\ D_y \end{bmatrix}. \end{aligned} \tag{8.26}$$

Remark 8.1. The matrix $L(\mathbf{f})$ can be viewed as a discretization of a steady-state diffusion operator

$$\begin{aligned} \mathcal{L}(f)u &= -\nabla \cdot (\psi'\, \nabla u) \\ &= -\frac{\partial}{\partial x}\left(\psi' \frac{\partial u}{\partial x}\right) - \frac{\partial}{\partial y}\left(\psi' \frac{\partial u}{\partial y}\right) \end{aligned} \tag{8.27}$$

with the diffusion coefficient

$$\psi' = \psi'(|\nabla f|^2) = \psi'\left(\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2\right)$$

and with "natural" (homogeneous Neumann) boundary conditions. Expression (8.27) gives the directional derivative in the direction u of the functional

$$J(f) = \frac{1}{2}\int_0^1 \int_0^1 \psi(|\nabla f|^2)\, dx\, dy. \tag{8.28}$$

An alternative approach to obtain the discrete operator $L(\mathbf{f})$ is to apply a discretization scheme directly to the continuous operator $\mathcal{L}(f)$ in (8.27). An example is the cell-centered finite difference scheme utilized in [120].

As in the one-dimensional case, one can compute a representation (8.19) for the Hessian of the penalty functional, with $L(\mathbf{f})$ given in (8.26) and

$$L'(\mathbf{f})\mathbf{f} = [D_x^T\ D_y^T] \begin{bmatrix} \mathrm{diag}(2(D_x\mathbf{f})^2\psi''(\mathbf{f})) & \mathrm{diag}(2(D_x\mathbf{f})(D_y\mathbf{f})\psi''(\mathbf{f})) \\ \mathrm{diag}(2(D_y\mathbf{f})(D_x\mathbf{f})\psi''(\mathbf{f})) & \mathrm{diag}(2(D_y\mathbf{f})^2\psi''(\mathbf{f})) \end{bmatrix} \begin{bmatrix} D_x \\ D_y \end{bmatrix}. \tag{8.29}$$

8.2.3 Steepest Descent and Newton's Method for Total Variation

In either the one- or the two-dimensional case, the gradient of the regularized cost functional has the form (8.21). To minimize (8.8), Algorithm 3.1 then gives us the following.

Algorithm 8.2.1. Steepest Descent for Total Variation–Penalized Least Squares.

$\nu := 0;$
$\mathbf{f}_0 :=$ initial guess;
begin steepest descent iterations
$\mathbf{g}_\nu := K^T(K\mathbf{f}_\nu - \mathbf{d}) + \alpha\, L(\mathbf{f}_\nu)\mathbf{f}_\nu;$ *% gradient*
$\tau_\nu := \arg\min_{\tau>0} T(\mathbf{f}_\nu - \tau\, \mathbf{g}_\nu);$ *% line search*
$\mathbf{f}_{\nu+1} := \mathbf{f}_\nu - \tau_\nu \mathbf{g}_\nu;$ *% update approximate solution*
$\nu := \nu + 1;$
end steepest descent iterations

Remark 8.2. Algorithm 8.2.3 is very similar to the discretized artificial time evolution approach of Rudin, Osher, and Fatemi [101]. In principle, to obtain a regularized solution to the operator equation $Kf = g$, they computed a steady-state solution of the time-dependent diffusion equation

$$\frac{\partial f}{\partial t} = -\alpha \mathcal{L}(f) f - K^*(Kf - g),$$

where $\mathcal{L}(f)$ is given in (8.27). After spatial discretization, they used explicit time marching with a fixed time step $\tau = \Delta t$ in place of the line search parameter τ_ν. See Exercise 8.4.

In both the one- and two-dimensional cases, the Hessian of the total variation–penalized least squares functional (8.8) has form (8.22). What follows is an implementation of Newton's method (section 3.3) to minimize (8.8). Some sort of globalization is essential to guarantee convergence of the Newton iterates [119]. Here we incorporate a line search.

Algorithm 8.2.2. Newton's Method for Total Variation–Penalized Least Squares.

$\nu := 0;$
$f_0 :=$ initial guess;
begin primal Newton iterations
$\mathbf{g}_\nu := K^T(K\mathbf{f}_\nu - \mathbf{d}) + \alpha\, L(\mathbf{f}_\nu)\mathbf{f}_\nu;$ *% gradient*
$H_J := L(\mathbf{f}_\nu) + L'(\mathbf{f}_\nu)\mathbf{f}_\nu;$ *% Hessian of penalty functional*
$H := K^T K + \alpha\, H_J;$ *% Hessian of cost functional*
$\mathbf{s}_\nu := -H^{-1}\mathbf{g}_\nu;$ *% Newton step*
$\tau_\nu := \arg\min_{\tau>0} T(\mathbf{f}_\nu + \tau\, \mathbf{s}_\nu);$ *% line search*
$\mathbf{f}_{\nu+1} := \mathbf{f}_\nu + \tau_\nu \mathbf{s}_\nu;$ *% update approximate solution*
$\nu := \nu + 1;$
end primal Newton iterations

8.2.4 Lagged Diffusivity Fixed Point Iteration

An alternative to the steepest descent method and Newton's method for the minimization of (8.8) is the lagged diffusivity fixed point iteration [119]:

$$\mathbf{f}_{\nu+1} = [K^T K + \alpha\, L(\mathbf{f}_\nu)]^{-1} K^T \mathbf{d} \tag{8.30}$$

$$= \mathbf{f}_\nu - [K^T K + \alpha\, L(\mathbf{f}_\nu)]^{-1} \operatorname{grad} T(\mathbf{f}_\nu). \tag{8.31}$$

The fixed point form (8.30) can be derived by first setting $\operatorname{grad} T(\mathbf{f}) = \mathbf{0}$ to obtain $(K^T K + \alpha\, L(\mathbf{f}))\mathbf{f} = K^T \mathbf{d}$; see (8.21). The discretized diffusion coefficient $\psi'(\mathbf{f})$ is then evaluated

at $\mathbf{f}_\nu$ to obtain $L(\mathbf{f}_\nu)$; see expressions (8.17) and (8.26) and Remark 8.1. Hence the expression "lagged diffusivity." The equivalent quasi-Newton form (8.31) can also be derived by dropping the term $\alpha\, L'(\mathbf{f})\mathbf{f}$ from the Hessian; see (8.22).

The following algorithm is based on the quasi-Newton form (8.31). The quasi-Newton form tends to be less sensitive to roundoff error than the fixed point form (8.30).

Algorithm 8.2.3. Lagged Diffusivity Fixed Point Method for Total Variation–Penalized Least Squares.

$\nu := 0;$
$\mathbf{f}_0 :=$ initial guess;
begin fixed point iterations
$L_\nu := L(\mathbf{f}_\nu);$ *% discretized diffusion operator*
$\mathbf{g}_\nu := K^T(K\mathbf{f}_\nu - \mathbf{d}) + \alpha\, L_\nu \mathbf{f}_\nu;$ *% gradient*
$H = K^T K + \alpha\, L_\nu;$ *% approximate Hessian*
$\mathbf{s}_{\nu+1} := -H^{-1}\mathbf{g}_\nu;$ *% quasi–Newton step*
$\mathbf{f}_{\nu+1} := \mathbf{f}_\nu + \mathbf{s}_\nu;$ *% update approximate solution*
$\nu := \nu + 1;$
end fixed point iterations

Remark 8.3. If $K^T K$ is positive definite, one can rigorously prove that this fixed point iteration converges globally [4, 41, 13, 120, 19], so no line search is needed. The approximate Hessian differs from the true Hessian (8.22) by the term $\alpha L'(\mathbf{f}_\nu)\mathbf{f}_\nu$. This term does not typically vanish as the iteration proceeds, so the rate of convergence of the lagged diffusivity iteration should be expected to be linear.

8.2.5 A Primal-Dual Newton Method

We first recall some basic results from convex analysis. In this discussion, φ is a convex functional defined on a convex set $C \subset \mathbb{R}^d$. For our purposes, $d = 1$ or 2, $C = \mathbb{R}^d$, and

$$\varphi(\mathbf{x}) = \frac{1}{2}\psi(|\mathbf{x}|^2) \tag{8.32}$$

with $|\mathbf{x}|^2 = \mathbf{x}^T\mathbf{x} = \sum_{i=1}^d x_i^2$. Relevant examples of the function ψ include $\psi(t) = 2\sqrt{t}$, which yields $\varphi(\mathbf{x}) = |\mathbf{x}|$, and the approximations (8.12) and (8.13).

Definition 8.4. The conjugate set C^* is defined by

$$C^* = \left\{ \mathbf{y} \in \mathbb{R}^d \mid \sup_{\mathbf{x}\in C}[\mathbf{x}^T\mathbf{y} - \varphi(\mathbf{x})] < \infty \right\}, \tag{8.33}$$

and the corresponding conjugate functional to φ is

$$\varphi^*(\mathbf{y}) = \sup_{\mathbf{x}\in C}\{\mathbf{x}^T\mathbf{y} - \varphi(\mathbf{x})\}. \tag{8.34}$$

This functional, which is also known as the Fenchel transform of φ, has the conjugate set C^* as its domain.

One can show [79, Proposition 1, p. 196] that the conjugate set C^* and the conjugate functional φ^* are, respectively, a convex set and a convex functional. The corresponding

second conjugates are defined in the obvious manner:

$$\varphi^{**} = (\varphi^*)^* \quad \text{and} \quad \mathcal{C}^{**} = (\mathcal{C}^*)^*.$$

In our finite dimensional Hilbert space setting, one can show [79, Proposition 2, p. 198] that $\varphi^{**} = \varphi$ and $\mathcal{C}^{**} = \mathcal{C}$. Consequently, from (8.34) we obtain the dual representation

$$\varphi(\mathbf{x}) = \sup_{y \in \mathcal{C}^*} \{\mathbf{x}^T \mathbf{y} - \varphi^*(\mathbf{y})\}. \tag{8.35}$$

We now derive the dual representation of the Euclidean norm, $\varphi(\mathbf{x}) = |\mathbf{x}|$, on $\mathbb{R}^d$. This is used in section 8.4 to define the TV functional. By the Cauchy–Schwarz inequality,

$$\mathbf{x}^T \mathbf{y} - |\mathbf{x}| \leq (|\mathbf{y}| - 1)|\mathbf{x}|, \tag{8.36}$$

with equality if and only if $\mathbf{y} = c\mathbf{x}$ for some $c \in \mathbb{R}$. If $|\mathbf{y}| > 1$, one can make (8.36) arbitrarily large by taking $\mathbf{y} = c\mathbf{x}$ and letting c increase. If $|\mathbf{y}| \leq 1$, then (8.36) is zero or negative, and its maximum value of zero is attained for $\mathbf{x} = 0$. Hence

$$\sup_{\mathbf{x} \in \mathbb{R}^d} \{\mathbf{x}^T \mathbf{y} - |\mathbf{x}|\} = \begin{cases} 0, & |\mathbf{y}| \leq 1, \\ +\infty, & |\mathbf{y}| \geq 1. \end{cases}$$

Thus the conjugate set is the unit ball,

$$\mathcal{C}^* = \{y \in \mathbb{R}^d \mid |y| \leq 1\},$$

and the conjugate functional $\varphi^*(\mathbf{y}) = 0$ for each $\mathbf{y} \in \mathcal{C}^*$. The dual representation (8.35) then yields

$$|\mathbf{x}| = \sup_{|\mathbf{y}| \leq 1} \mathbf{x}^T \mathbf{y}. \tag{8.37}$$

The following two examples give dual representations of convex approximations to the Euclidean norm derived from (8.12) and (8.13) via (8.32).

Example 8.5. Consider the convex functional

$$\varphi_\beta(\mathbf{x}) = \sqrt{|\mathbf{x}|^2 + \beta^2}, \qquad \beta > 0, \tag{8.38}$$

defined on $\mathcal{C} = \mathbb{R}^d$. One can show (see Exercise 8.7) that

$$\sup_{\mathbf{x} \in \mathbb{R}^d} \{\mathbf{x}^T \mathbf{y} - \varphi_\beta(\mathbf{x})\} = \begin{cases} -\beta\sqrt{1 - |\mathbf{y}|^2} & \text{if} \quad |\mathbf{y}| \leq 1, \\ +\infty & \text{if} \quad |\mathbf{y}| > 1. \end{cases} \tag{8.39}$$

Hence, the conjugate set $\mathcal{C}^*$ is the unit ball in $\mathbb{R}^d$, $\varphi_\beta^*(\mathbf{y}) = -\beta\sqrt{1 - |\mathbf{y}|^2}$, and by (8.35),

$$\varphi_\beta(\mathbf{x}) = \sup_{|\mathbf{y}| \leq 1} \left\{ \mathbf{x}^T \mathbf{y} + \beta\sqrt{1 - |\mathbf{y}|^2} \right\}. \tag{8.40}$$

Example 8.6. Consider the functional

$$\varphi_\epsilon(\mathbf{x}) = \begin{cases} \frac{|\mathbf{x}|^2}{2\epsilon} & \text{if} \quad |\mathbf{x}| \leq \epsilon, \\ |\mathbf{x}| - \frac{\epsilon}{2} & \text{if} \quad |\mathbf{x}| > \epsilon, \end{cases} \tag{8.41}$$

defined on $\mathcal{C} = \mathbb{R}^d$. The conjugate set $\mathcal{C}^*$ is again the unit ball, and the conjugate functional is given by

$$\varphi_\epsilon^*(\mathbf{y}) = \frac{\epsilon}{2}|\mathbf{y}|^2, \qquad \mathbf{y} \in \mathcal{C}^*.$$

See Exercise 8.8. Consequently,

$$\varphi_\epsilon(\mathbf{x}) = \sup_{|\mathbf{y}|\le 1} \left\{ \mathbf{x}^T\mathbf{y} - \frac{\epsilon}{2}|\mathbf{y}|^2 \right\}. \tag{8.42}$$

The following theorem relates the gradient of a convex functional φ to the gradient of its conjugate φ^*. See [30, p. 290] for a proof.

Theorem 8.7. *Suppose that φ is differentiable in a neighborhood of $\mathbf{x}_0 \in \mathcal{C} \subset \mathbb{R}^d$, and the mapping $F = \operatorname{grad}\varphi : \mathbb{R}^d \to \mathbb{R}^d$ is invertible in that neighborhood. Then φ^* is Frechet differentiable in a neighborhood of $\mathbf{y}_0 = \varphi(\mathbf{x}_0)$ with*

$$\operatorname{grad}\varphi^*(y) = F^{-1}(\mathbf{y}). \tag{8.43}$$

We now apply convex analysis to obtain a dual formulation for the two-dimensional penalty functional (8.23). Setting

$$\varphi(x_1, x_2) = \frac{1}{2}\psi(x_1^2 + x_2^2) \tag{8.44}$$

and employing the dual representation (8.35) with $\mathbf{y} = (u, v)$, we obtain

$$J(f) = \sum_{i,j} \sup_{(u_{ij}, v_{ij}) \in \mathcal{C}^*} \{(D_{ij}^x f)u_{ij} + (D_{ij}^y f)v_{ij} - \varphi^*(u_{ij}, v_{ij})\}.$$

As in section 8.2.2, we stack the array components f_{ij}, u_{ij}, and v_{ij} into column vectors $\mathbf{f}$, $\mathbf{u}$, and $\mathbf{v}$; we let D_x and D_y be matrix representers for the grid operators D_{ij}^x and D_{ij}^y; and we let $\langle \cdot, \cdot \rangle$ denote Euclidean inner product. Then the penalty functional can be rewritten as

$$\begin{aligned} J(\mathbf{f}) &= \sup_{(\mathbf{u},\mathbf{v}) \in \mathcal{C}^*} \{\langle D_x\mathbf{f}, \mathbf{u}\rangle + \langle D_y\mathbf{f}, \mathbf{v}\rangle - \langle \varphi^*(\mathbf{u}, \mathbf{v}), \mathbf{1}\rangle\} \\ &= \sup_{(\mathbf{u},\mathbf{v}) \in \mathcal{C}^*} \tilde{J}(\mathbf{u}, \mathbf{v}, \mathbf{f}), \end{aligned} \tag{8.45}$$

where

$$\tilde{J}(\mathbf{u}, \mathbf{v}, \mathbf{f}) = \langle \mathbf{f}, D_x^T\mathbf{u} + D_y^T\mathbf{v}\rangle - \langle \varphi^*(\mathbf{u}, \mathbf{v}), \mathbf{1}\rangle;$$

$\mathbf{1}$ denotes the vector of 1's; and by $(\mathbf{u}, \mathbf{v}) \in \mathcal{C}^*$ we mean each of the component pairs (u_{ij}, v_{ij}) lies in $\mathcal{C}^*$.

Minimization of the penalized least squares functional (8.8) is equivalent to computing the saddle point

$$(\mathbf{u}^*, \mathbf{v}^*, \mathbf{f}^*) = \arg\min_{\mathbf{f}} \max_{(\mathbf{u},\mathbf{v}) \in \mathcal{C}^*} \tilde{T}(\mathbf{u}, \mathbf{v}, \mathbf{f}), \tag{8.46}$$

where

$$\tilde{T}(\mathbf{u}, \mathbf{v}, \mathbf{f}) = \frac{1}{2}||K\mathbf{f} - \mathbf{d}||^2 + \alpha\tilde{J}(\mathbf{u}, \mathbf{v}, \mathbf{f}).$$

We refer to $\mathbf{f}$ as the primal variable and to $\mathbf{u}$ and $\mathbf{v}$ as the dual variables.

Since (8.46) is unconstrained with respect to $\mathbf{f}$, a first order necessary condition for a saddle point is

$$\mathbf{0} = \operatorname{grad}_{\mathbf{f}} \tilde{T} = K^T(K\mathbf{f} - \mathbf{d}) + \alpha(D_x^T\mathbf{u} + D_y^T\mathbf{v}). \tag{8.47}$$

An additional necessary condition is that the duality gap in (8.35) must vanish, i.e., for each grid index i, j,

$$\varphi(D_{ij}^x\mathbf{f}, D_{ij}^y\mathbf{f}) = (D_{ij}^x\mathbf{f})u_{ij} + (D_{ij}^y\mathbf{f})v_{ij} - \varphi^*(u_{ij}, v_{ij}). \tag{8.48}$$

Finally, the dual variables must lie in the conjugate set; i.e.,

$$(u_{ij}, v_{ij}) \in C^*. \tag{8.49}$$

We next examine the implications of (8.48). Suppose (8.48) holds for a point (u_{ij}, v_{ij}) in the interior of C^*. This is the case in each of Examples 8.5 and 8.6; see Exercises 8.7 and 8.8. Then

$$\begin{aligned}(0,0) &= \operatorname{grad}_{u_{ij},v_{ij}}[(D_{ij}^x\mathbf{f})u_{ij} + (D_{ij}^y\mathbf{f})v_{ij} - \varphi^*(u_{ij}, v_{ij})] \\ &= (D_{ij}^x\mathbf{f}, D_{ij}^y\mathbf{f}) - \operatorname{grad}\varphi^*(u_{ij}, v_{ij}) \\ &= (D_{ij}^x\mathbf{f}, D_{ij}^y\mathbf{f}) - \frac{1}{\psi'((D_{ij}^x\mathbf{f})^2 + (D_{ij}^y\mathbf{f})^2)}(u_{ij}, v_{ij}). \end{aligned} \tag{8.50}$$

The last equality follows from the representation (8.44) and Theorem 8.7. Equation (8.50) is equivalent to

$$D_{ij}^x f = \frac{u_{ij}}{\psi'_{ij}}, \qquad D_{ij}^y f = \frac{v_{ij}}{\psi'_{ij}}, \tag{8.51}$$

where $\psi'_{ij} = \psi'_{ij}((D_{ij}^x\mathbf{f})^2 + (D_{ij}^y\mathbf{f})^2)$. Returning to matrix notation, we have

$$D_x\mathbf{f} = B(\mathbf{f})\,\mathbf{u}, \qquad D_y\mathbf{f} = B(\mathbf{f})\,\mathbf{v}, \tag{8.52}$$

where

$$B(\mathbf{f}) = \operatorname{diag}(1/\psi'(\mathbf{f})). \tag{8.53}$$

We can reformulate the first order necessary conditions (8.47), (8.52) as a nonlinear system

$$\mathbf{G}(\mathbf{u}, \mathbf{v}, \mathbf{f}) \stackrel{\text{def}}{=} \begin{bmatrix} B(\mathbf{f})\mathbf{u} - D_x\mathbf{f} \\ B(\mathbf{f})\mathbf{v} - D_y\mathbf{f} \\ \alpha D_x^T\mathbf{u} + \alpha D_y^T\mathbf{v} + K^T(K\mathbf{f} - \mathbf{d}) \end{bmatrix} = \mathbf{0}. \tag{8.54}$$

The derivative of G can be expressed as

$$\mathbf{G}'(\mathbf{u}, \mathbf{v}, \mathbf{f}) = \begin{bmatrix} B(\mathbf{f}) & \mathbf{0} & B'(\mathbf{f})\mathbf{u} - D_x \\ \mathbf{0} & B(\mathbf{f}) & B'(\mathbf{f})\mathbf{v} - D_y \\ \alpha D_x^T & \alpha D_y^T & K^TK \end{bmatrix}. \tag{8.55}$$

Here $B'(\mathbf{f})\mathbf{u}$ has component representation

$$[B'(\mathbf{f})\mathbf{u}]_{ij} = \frac{-\psi''_{ij}}{(\psi'_{ij})^2}\, 2\left((D_{ij}^x\mathbf{f})D_{ij}^x + (D_{ij}^y\mathbf{f})D_{ij}^y\right) u_{ij}.$$

In matrix form,

$$B'(\mathbf{f})\mathbf{u} - D_x = -E_{11}D_x - E_{12}D_y, \tag{8.56}$$

with

$$E_{11} = \operatorname{diag}\left(1 + \frac{2\psi''(\mathbf{f})\ (D_x\mathbf{f})\ \mathbf{u}}{\psi'(\mathbf{f})^2}\right), \qquad E_{12} = \operatorname{diag}\left(\frac{2\psi''(\mathbf{f})\ (D_y\mathbf{f})\ \mathbf{u}}{\psi'(\mathbf{f})^2}\right), \tag{8.57}$$

where the products and quotients are computed pointwise. Similarly,

$$B'(\mathbf{f})\mathbf{v} - D_y = -E_{21}D_x - E_{22}D_y, \tag{8.58}$$

with

$$E_{21} = \operatorname{diag}\left(\frac{2\psi''(\mathbf{f})\ (D_x\mathbf{f})\ \mathbf{v}}{\psi'(\mathbf{f})^2}\right), \qquad E_{22} = \operatorname{diag}\left(1 + \frac{2\psi''(\mathbf{f})\ (D_y\mathbf{f})\ \mathbf{v}}{\psi'(\mathbf{f})^2}\right). \tag{8.59}$$

Newton's method for the system (8.54) requires solutions of systems of the form $\mathbf{G}'(\mathbf{u}, \mathbf{v}, \mathbf{f})\ (\Delta\mathbf{u}, \Delta\mathbf{v}, \Delta\mathbf{f}) = -\mathbf{G}(\mathbf{u}, \mathbf{v}, \mathbf{f})$. Substituting (8.56)–(8.59) and applying block row reduction to convert $\mathbf{G}'$ to block upper triangular form, we obtain

$$\begin{bmatrix} B(\mathbf{f}) & \mathbf{0} & -E_{11}D_x - E_{12}D_y \\ \mathbf{0} & B(\mathbf{f}) & -E_{21}D_x - E_{22}D_y \\ \mathbf{0} & \mathbf{0} & K^TK + \alpha\overline{L} \end{bmatrix} \begin{bmatrix} \Delta\mathbf{u} \\ \Delta\mathbf{v} \\ \Delta\mathbf{f} \end{bmatrix} = \begin{bmatrix} -\mathbf{g}_1 \\ -\mathbf{g}_2 \\ \mathbf{r} \end{bmatrix}. \tag{8.60}$$

Here $\mathbf{g}_i$ denotes the ith component of $\mathbf{G}$ (see (8.54))

$$\begin{aligned} \overline{L} &= \begin{bmatrix} D_x^T & D_y^T \end{bmatrix} \begin{bmatrix} B(\mathbf{f})^{-1} & \mathbf{0} \\ \mathbf{0} & B(\mathbf{f})^{-1} \end{bmatrix} \begin{bmatrix} E_{11} & E_{12} \\ E_{21} & E_{22} \end{bmatrix} \begin{bmatrix} D_x \\ D_y \end{bmatrix} \\ &= D_x^T B(\mathbf{f})^{-1} E_{11} D_x + D_x^T B(\mathbf{f})^{-1} E_{12} D_y + D_y^T B(\mathbf{f})^{-1} E_{21} D_x \\ &\quad + D_y^T B(\mathbf{f})^{-1} E_{22} D_y \end{aligned} \tag{8.61}$$

and

$$\begin{aligned} \mathbf{r} &= -\mathbf{g}_3 + \alpha D_x^T B(\mathbf{f})^{-1} D_x \mathbf{g}_1 + \alpha D_y^T B(\mathbf{f})^{-1} D_y \mathbf{g}_2 \\ &= K^T(\mathbf{d} - K\mathbf{f}) - \alpha D_x^T B(\mathbf{f})^{-1} D_x \mathbf{f} - \alpha D_y^T B(\mathbf{f})^{-1} D_y \mathbf{f}. \end{aligned} \tag{8.62}$$

Consequently,

$$\Delta\mathbf{f} = [K^TK + \alpha\overline{L}]^{-1}\mathbf{r}, \tag{8.63}$$

$$\Delta\mathbf{u} = -\mathbf{u} + B(\mathbf{f})^{-1}[D_x\mathbf{f} + E_{11}D_x\Delta\mathbf{f} + E_{12}D_y\Delta\mathbf{f}], \tag{8.64}$$

$$\Delta\mathbf{v} = -\mathbf{v} + B(\mathbf{f})^{-1}[D_y\mathbf{f} + E_{21}D_x\Delta\mathbf{f} + E_{22}D_y\Delta\mathbf{f}]. \tag{8.65}$$

We employ backtracking to the boundary to maintain the constraint (8.49). In other words, we compute

$$\overline{\tau} = \max\{0 \le \tau \le 1 \mid (u_{ij} + \tau\Delta u_{ij},\ v_{ij} + \tau\Delta v_{ij}) \in C^* \text{ for all } i, j\}.$$

We then update

$$\mathbf{u} := \mathbf{u} + \overline{\tau}\Delta\mathbf{u}, \qquad \mathbf{v} := \mathbf{v} + \overline{\tau}\Delta\mathbf{v}.$$

Practical experience [21] suggests that no globalization is needed in the update $\mathbf{f} := \mathbf{f} + \Delta\mathbf{f}$.

Algorithm 8.2.4. Primal-Dual Newton's Method for Total Variation–Penalized Least Squares Minimization in Two Space Dimensions.

$\nu := 0$;
$\mathbf{f}_0 :=$ initial guess for primal variable;
$\mathbf{u}_0, \mathbf{v}_0 :=$ initial guesses for dual variables;
begin primal-dual Newton iterations
$B_\nu^{-1} := \text{diag}(\psi'(\mathbf{f}_\nu))$;
$\mathbf{w} := 2\psi'(\mathbf{f}_\nu)./\psi''(\mathbf{f}_\nu)$;
$E_{11} := \text{diag}(\mathbf{w}. * (D_x\mathbf{f}_\nu). * \mathbf{u}_\nu)$;
$E_{12} := \text{diag}(\mathbf{w}. * (D_y\mathbf{f}_\nu). * \mathbf{u}_\nu)$;
$E_{21} := \text{diag}(\mathbf{w}. * (D_x\mathbf{f}_\nu). * \mathbf{v}_\nu)$;
$E_{22} := \text{diag}(\mathbf{w}. * (D_y\mathbf{f}_\nu). * \mathbf{v}_\nu)$;
$\overline{L}_\nu := D_x^T B_\nu^{-1} E_{11} D_x + D_x^T B_\nu^{-1} E_{12} D_y + D_y^T B_\nu^{-1} E_{21} D_x$
$+ D_y^T B_\nu^{-1} E_{22} D_y$; % *discretized diffusion operator*
$\mathbf{r}_\nu := K^T(\mathbf{d} - K\mathbf{f}_\nu) - \alpha(D_x^T B_\nu^{-1} D_x + D_y^T B_\nu^{-1} D_y)\mathbf{f}_\nu$;
$\Delta\mathbf{f} := (K^T K + \alpha\, \overline{L}_\nu)^{-1}\mathbf{r}_\nu$; % *Newton step*
$\Delta\mathbf{u} := -\mathbf{u}_\nu + B_\nu^{-1}[D_x\mathbf{f}_\nu + (E_{11}D_x + E_{12}D_y)\Delta\mathbf{f}]$;
$\Delta\mathbf{v} := -\mathbf{u}_\nu + B_\nu^{-1}[D_y\mathbf{f}_\nu + (E_{21}D_x + E_{22}D_y)\Delta\mathbf{f}]$;
$\mathbf{f}_{\nu+1} := \mathbf{f}_\nu + \Delta\mathbf{f}$; % *update primal variable*
$\tau_\nu := \max\{0 \le \tau \le 1 \mid (\mathbf{u}_\nu + \tau\Delta\mathbf{u}, \mathbf{v}_\nu + \tau\Delta\mathbf{v}) \in C^*\}$;
$\mathbf{u}_{\nu+1} := \mathbf{u}_\nu + \tau_\nu\Delta\mathbf{u}$; % *update dual variables*
$\mathbf{v}_{\nu+1} := \mathbf{v}_\nu + \tau_\nu\Delta\mathbf{v}$;
$\nu := \nu + 1$;
end primal-dual Newton iterations

Remark 8.8. For large-scale systems where the matrix $K^T K$ does not have a sparse matrix representation, the most expensive part of the algorithm is the inversion of the matrix $A \stackrel{\text{def}}{=} K^T K + \alpha\overline{L}$ in the computation of the component $\Delta\mathbf{f}$ of the Newton step. If K has block Toeplitz structure, then the techniques of section 5.2.5 can be used to compute matrix-vector products $A\mathbf{v}$ at a cost of $n \log n$ (note that the matrix $\overline{L}$ is sparse; see (8.61)). This suggests the use of iterative linear solvers like the CG Algorithm 3.2. Use of CG is precluded by the fact that $\overline{L}$ need not be symmetric, but one can replace $\overline{L}$ by its symmetric part, $(\overline{L} + \overline{L}^T)/2$, and still retain quadratic convergence of the primal-dual Newton iteration [21]. CG iteration can then be applied as a linear solver. Chan, Chan, and Wong [15] provided preconditioners for CG in this setting.

8.2.6 Other Methods

The computational methods presented in the previous sections are based on smooth approximations to the Euclidean norm of the gradient. Ito and Kunisch [62] presented an alternative approach that is based on the representation (8.4).

One can also replace the Euclidean norm of the gradient by other norms. Li and Santosa [74] made use of the ℓ^1 norm. After discretization in two dimensions, their penalty functional took the form

$$J(f) = \sum_i \sum_j |D_{ij}^x f| + |D_{ij}^y f|.$$

They then applied an interior point method to solve their minimization problem. It should be noted that unlike the Euclidean norm, the ℓ^1 norm is not rotationally invariant. This may have the unfortunate consequence of making the reconstruction dependent on the orientation of the computational grid.

The time evolution approach outlined in Remark 8.2 provides one example of a broad class of nonlinear PDE-based techniques called nonlinear diffusion methods. These methods have found important applications in computer vision and image processing. See [124] for details and references.

Finally, we note that there exist several other mathematical expressions for variation that are closely related but not equivalent to (8.4). See [73] for details and references.

8.3 Numerical Comparisons

In this section, we compare the performance of some of the solution methods presented in the previous sections.

8.3.1 Results for a One-Dimensional Test Problem

The one-dimensional test problem is described in section 1.1, and the data used are presented in Figure 1.1. To solve this problem, we minimized the discrete regularized least squares functional (8.8)–(8.10) with

$$\frac{1}{2}\psi((D_i\mathbf{f})^2) = \sqrt{\left(\frac{f_i - f_{i-1}}{\Delta x}\right)^2 + \beta^2} \tag{8.66}$$

using the four iterative solution methods presented in sections 8.2.3–8.2.5.

The matrix K in (8.8) is Toeplitz and not sparse. In the primal-dual Newton implementation, the conjugate set C^* is the interval $-1 \leq u \leq 1$; see Example 8.5.

The reconstruction obtained using a one-dimensional version of the primal-dual Newton Algorithm 8.2.5 is shown in Figure 1.5. The functional (8.8) is strictly convex for this test example, so the other methods yield essentially the same reconstruction provided the minimizer is computed to sufficient accuracy.

One of our primary measures of numerical performance is the relative iterative solution error norm,

$$e_\alpha^\nu = \frac{||\mathbf{f}_\alpha^\nu - \mathbf{f}_\alpha||}{||\mathbf{f}_\alpha||}, \tag{8.67}$$

where $\mathbf{f}_\alpha$ represents the minimizer of (8.8) and $\mathbf{f}_\alpha^\nu$ represents the numerical approximation to $\mathbf{f}_\alpha$ at iteration ν. In place of the exact $\mathbf{f}_\alpha$ we used an extremely accurate approximation obtained with the primal-dual Newton method.

The performances of the steepest descent method (Algorithm 8.2.3) and Newton's method with a line search (Algorithm 8.2.3) are compared in Figure 8.3. Initially the steepest descent method exhibits a rapid decrease in the iterative solution error, but almost no change in the reconstructions is observed after the first five steepest descent iterations. This is consistent with Theorem 3.5, since the Hessian is quite ill-conditioned.

With Newton's method very little progress occurs until about iteration 50. During the earlier iterations, the line search restricts the step size. In the last six iterations, there is a dramatic decrease in solution error, as the local quadratic convergence rate characteristic of Newton's method is finally attained. This behavior is consistent with the theory presented in

section 3.3. Note that the convergence constant c_* in (3.18) becomes large as the minimum eigenvalue of the Hessian becomes small and the Lipschitz constant γ becomes large. The former event occurs when the regularization parameter α is small; the latter occurs when the parameter β in (8.66) becomes small. When c_* is large, $\mathbf{f}_\alpha^\nu$ must be quite close to $\mathbf{f}_\alpha$ before the iteration will converge without a line search.

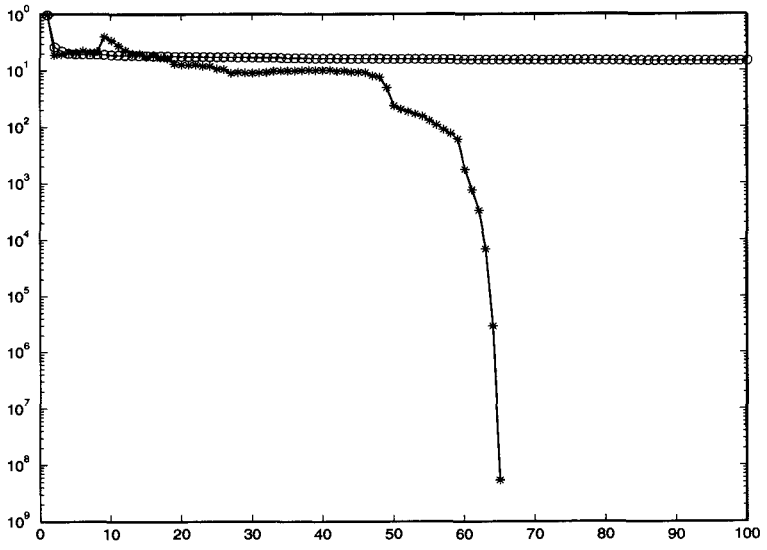


Figure 8.3. *Numerical performance of the steepest descent method and Newton's method on a one-dimensional test problem. The relative iterative solution error norm* (8.67) *is plotted against the iteration count. Circles represent the results for the steepest descent method (Algorithm* 8.2.3*), and asterisks represent the results for the primal Newton method (Algorithm* 8.2.3*).*

In Figure 8.4 the performance of the lagged diffusivity fixed point method (Algorithm 8.2.4) and the primal-dual Newton method (Algorithm 8.2.5) are compared. The lagged diffusivity fixed point method displays rapid decrease in the solution error during the first few iterations. Convergence then slows to a steady linear rate. The primal-dual Newton method also displays fairly fast initial convergence. After about eight iterations, a more rapid quadratic convergence rate can be seen.

Note that the steepest descent method requires a nonsparse matrix-vector multiplication at each iteration, since K is a full matrix. The other three methods require the inversion of nonsparse linear systems at each iteration. (We used Gaussian elimination to solve these systems.) Hence, the cost per iteration of the steepest descent method is significantly less than that of the other three methods. However, for this particular test problem, the extremely slow convergence rate of steepest descent negates the advantage of low computational cost per iteration. The other three methods all have roughly the same cost per iteration. Due to its more rapid convergence rate, the primal-dual Newton method is the most efficient method for this problem.

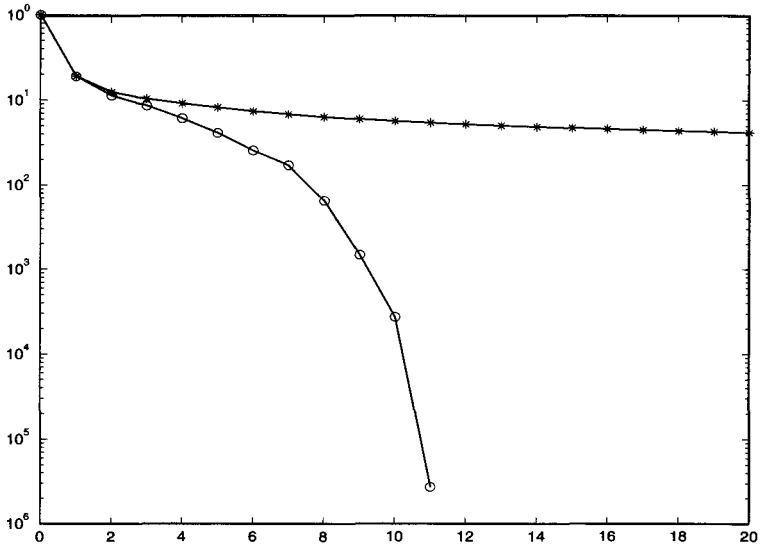


Figure 8.4. *Numerical performance of the lagged diffusivity fixed point iteration and the primal-dual Newton method on a one-dimensional test problem. Asterisks represent the relative iterative solution error norm for the fixed point method (Algorithm* 8.2.4*), and circles represent the results for the primal-dual Newton method (Algorithm* 8.2.5*).*

8.3.2 Two-Dimensional Test Results

The image deblurring test problem in this section is described in section 5.1.1, the test data are similar to that shown in Figure 5.2, and the reconstructions are similar to that shown in Figure 8.1. To obtain the reconstructions, we minimized a two-dimensional version of the penalized least squares functional (8.8)–(8.10) with

$$\frac{1}{2}\psi((D_{ij}^x \mathbf{f})^2 + (D_{ij}^y \mathbf{f})^2) = \sqrt{\left(\frac{f_{i,j} - f_{i-1,j}}{\Delta x}\right)^2 + \left(\frac{f_{i,j} - f_{i,j-1}}{\Delta x}\right)^2 + \beta^2}.$$

We present numerical performance results (see Figure 8.5) only for the lagged diffusivity fixed point Algorithm 8.2.4 and for the primal-dual Newton Algorithm 8.2.5. Comparison of the other methods is left to Exercise 8.14.

In the primal-dual Newton implementation, the conjugate set C^* is the unit ball in $\mathbb{R}^2$; see Example 8.5. The matrix K in (8.8) is block Toeplitz with Toeplitz blocks; see section 5.2.5. As in our one-dimensional test problem, K is not sparse.

As in one dimension, the lagged diffusivity fixed point convergence is rapid at first but slows to a steady linear rate after a few iterations. Primal-dual Newton displays fairly rapid initial convergence, with an increase in the convergence rate at later iterations. For this test problem, primal-dual Newton clearly converges at a much more rapid rate.

Both lagged diffusivity and primal-dual Newton require the solution of nonsparse linear systems at each iteration. In this two-dimensional application, these systems are large enough to discourage the use of direct matrix decomposition methods. Instead we applied the CG Algorithm 3.2 with no preconditioning. (See Remark 8.8 for primal-dual Newton

implementation details.) We found that we needed a very small (residual) CG stopping tolerance to maintain rapid convergence of the primal-dual Newton iterations. A much more relaxed CG stopping tolerance could be used without degrading the convergence of the lagged diffusivity iteration. Consequently, the cost per iteration of primal-dual Newton was significantly larger than the cost per iteration of lagged diffusivity fixed point. This may no longer be the case if preconditioning is applied; see [15] and Exercise 8.15. An overall cost comparison is difficult to carry out, since it depends on factors like stopping tolerances; values of parameters like α, β, and the system size; and the effectiveness of the preconditioner.

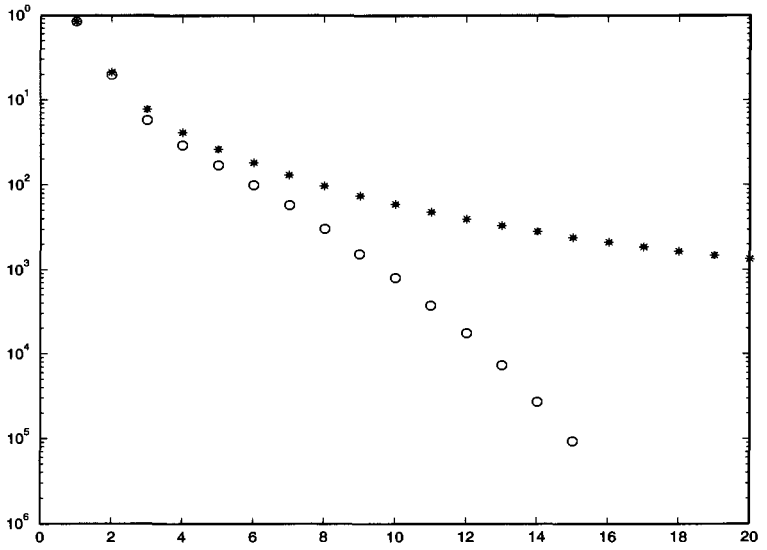


Figure 8.5. *Comparison of lagged diffusivity fixed point iteration and primal-dual Newton iteration for a two-dimensional image reconstruction problem. Asterisks denote relative iterative solution error for the fixed point iteration, and circles denote error for primal-dual Newton.*

Total variation methods have been applied to more general inverse problems. See [33] for an application to distributed parameter identification.

8.4 Mathematical Analysis of Total Variation

In this section, Ω denotes a simply connected, nonempty, open subset of $\mathbb{R}^d$, $d = 1, 2, \ldots$, with Lipschitz continuous boundary. In imaging applications, Ω is typically the unit square in $\mathbb{R}^2$. We use the symbol ∇ to denote the gradient of a smooth function $f : \mathbb{R}^d \to \mathbb{R}^1$, i.e., $\nabla f = (\frac{\partial f}{\partial x_1}, \ldots, \frac{\partial f}{\partial x_d})$. $C_0^1(\Omega; \mathbb{R}^d)$ denotes the space of vector-valued functions $\vec{v} = (v_1, \ldots, v_d)$ whose component functions v_i are each continuously differentiable and compactly supported on Ω, i.e., each v_i vanishes outside some compact subset of Ω. The divergence of $\vec{v}$ is given by

$$\operatorname{div} \vec{v} = \sum_{i=1}^{d} \frac{\partial v_i}{\partial x_i}.$$

The Euclidean norm is denoted by $|\cdot|$. In particular, $|\vec{v}(x)| = [\sum_{i=1}^{d} v_i(x)^2]^{1/2}$. The Sobolev space $W^{1,1}(\Omega)$ denotes the closure of $C_0^1(\Omega)$ with respect to the norm

$$\|f\|_{1,1} = \int_\Omega \left[|f| + \sum_{i=1}^{d} \left| \frac{\partial f}{\partial x_i} \right| \right].$$

The following definition is taken from Giusti [45].

Definition 8.9. The total variation of a function $f \in L^1(\Omega)$ is defined by

$$\mathrm{TV}(f) = \sup_{\vec{v} \in \mathcal{V}} \int_\Omega f \operatorname{div} \vec{v}\, dx, \tag{8.68}$$

where the space of test functions

$$\mathcal{V} = \{\vec{v} \in C_0^1(\Omega; \mathbb{R}^d) \mid |\vec{v}(x)| \le 1 \text{ for all } x \in \Omega\}. \tag{8.69}$$

Remark 8.10. Equation (8.68) can be viewed as a weak form of

$$\mathrm{TV}(f) = \int_\Omega |\nabla f|\, dx.$$

Using the dual representation (8.37) for the Euclidean norm and formally applying integration by parts,

$$\begin{aligned} \int_\Omega |\nabla f|\, dx &= \int_\Omega \sup_{|\vec{v}| \le 1} \nabla f^T \vec{v}\, dx \\ &= \sup_{|\vec{v}| \le 1} \left[\int_{\partial\Omega} f\, \vec{v}^T \hat{n}\, dS - \int_\Omega f \operatorname{div} \vec{v}\, dx \right], \end{aligned}$$

where $\partial\Omega$ denotes the boundary of Ω and $\hat{n}$ denotes the outward unit normal to $\partial\Omega$. If $\vec{v}$ is compactly supported in Ω, then the boundary integral term vanishes. Note that $|\vec{v}| \le 1$ if and only if $|-\vec{v}| \le 1$, so we can drop the minus sign to obtain (8.68)–(8.69).

Example 8.11. Let $\Omega = [0, 1] \subset \mathbb{R}^1$, and define

$$f(x) = \begin{cases} f_0, & x < 1/2, \\ f_1, & x > 1/2, \end{cases}$$

where f_0, f_1 are constants. For any $v \in C_0^1[0, 1]$,

$$\int_0^1 f(x)v'(x)\, dx = \int_0^{1/2} f(x)v'(x)\, dx + \int_{1/2}^1 f(x)v'(x)\, dx = (f_0 - f_1)\, v(1/2).$$

This quantity is maximized over all $v \in \mathcal{V}$ when $v(1/2) = \operatorname{sign}(f_0 - f_1)$. This yields $\mathrm{TV}(f) = |f_1 - f_0|$, which agrees with (8.1).

Example 8.12. Let E be a set contained in $\Omega \subset \mathbb{R}^d$, with $d \ge 2$, and assume its boundary ∂E is C^2. Let $f(x) = f_0$ if $x \in E$ and $f(x) = 0$ otherwise. In this case,

$$\mathrm{TV}(f) = f_0 \operatorname{Area}(\partial E),$$

where Area$(\cdot)$ denotes surface area. (This reduces to arc length when the dimension $d = 2$.) To verify, for any $\vec{v} \in C_0^1(\Omega; \mathbb{R}^d)$ the divergence theorem yields

$$\int_\Omega f \operatorname{div} \vec{v}\, dx = f_0 \int_E \operatorname{div} \vec{v}\, dx = f_0 \int_{\partial E} \vec{v}^T \hat{n}\, dS, \tag{8.70}$$

where $\hat{n}(x)$ denotes the outward unit normal to ∂E at x and dS denotes surface integration. Imposing $|\vec{v}(x)| \leq 1$, we obtain from (8.68) that $\mathrm{TV}(f) \leq f_0 \mathrm{Area}(\partial S)$. Since E has C^2 boundary, its outward unit normal $\hat{n}(x)$ will be a C^1 vector-valued function, which can be extended to a function $\vec{v} \in C_0^1(\Omega; \mathbb{R}^d)$ for which $|\vec{v}(x)| \leq 1$. Then by (8.68) and (8.70), $\mathrm{TV}(f) \geq f_0 \int_{\partial E} |\hat{n}(x)|^2\, dS = f_0 \mathrm{Area}(\partial S)$.

Proposition 8.13. *If $f \in W^{1,1}(\Omega)$, then*

$$\mathrm{TV}(f) = \int_\Omega |\nabla f|. \tag{8.71}$$

Proof. If $f \in C^1(\Omega)$ and $\vec{v} \in C_0^1(\Omega; \mathbb{R}^d)$, then integration by parts yields

$$\int_\Omega f \operatorname{div} \vec{v}\, dx = -\int_\Omega \nabla f^T \vec{v}\, dx.$$

Take

$$\vec{w}(x) = \begin{cases} -\frac{\nabla f}{|\nabla f|} & \text{if } \nabla f(x) \neq 0, \\ 0 & \text{if } \nabla f(x) = 0. \end{cases}$$

One can pick $\vec{v} \in C_0^1(\Omega; \mathbb{R}^d)$ with components arbitrarily close to those of $\vec{w}$ with respect to the L^2 norm, and, hence, (8.71) holds for any $f \in C^1(\Omega)$. By a standard denseness argument, this also holds for $f \in W^{1,1}(\Omega)$. □

Definition 8.14. The space of functions of bounded variation, denoted by $BV(\Omega)$, consists of functions $f \in L^1(\Omega)$ for which

$$\|f\|_{BV} \stackrel{\text{def}}{=} \|f\|_{L^1(\Omega)} + \mathrm{TV}(f) < \infty. \tag{8.72}$$

Theorem 8.15. *$\|\cdot\|_{BV}$ is a norm, and $BV(\Omega)$ is a Banach space under this norm. The TV functional is a seminorm on this space.*

See Giusti [45] for a proof of this theorem. Proposition 8.13 and Examples 8.11 and 8.12 show that $W^{1,1}(\Omega)$ is a proper subspace of $BV(\Omega)$.

The following three theorems pertain to the important properties of compactness, convexity, and semicontinuity. Proofs can be found in [2].

Theorem 8.16. *Let S be a BV-bounded set of functions. For $\Omega \subset \mathbb{R}^d$, S is a relatively compact subset of $L^p(\Omega)$ for $1 \leq p < d/(d-1)$ and is weakly relatively compact in $L^{d/(d-1)}(\Omega)$. In case the dimension $d = 1$, we set $d/(d-1) = +\infty$.*

Theorem 8.17. *The* TV *functional* (8.68)*, defined on the space $BV(\Omega)$, is convex but not strictly convex. The restriction of this functional to $W^{1,1}(\Omega)$ is strictly convex.*

Theorem 8.18. *The* TV *functional is weakly lower semicontinuous with respect to the L^p norm topology for $1 \leq p < \infty$.*

We next examine the existence, uniqueness, and stability of minimizers of the BV-penalized least squares functional

$$T(f) = ||Kf - d||^2_{L^2(\Omega)} + \alpha ||f||_{BV}, \qquad \alpha > 0. \tag{8.73}$$

Theorem 8.19. *Let $1 \leq p < d/(d-1)$, and let C be a closed, convex subset of $L^p(\Omega)$. Assume $K : L^p(\Omega) \to L^2(\Omega)$ is linear, bounded, and* $\text{Null}(K) = \{0\}$. *Then, for any fixed $d \in L^2(\Omega)$, the functional in* (8.73) *has a unique constrained minimizer,*

$$f_* = \arg\min_{f \in C} T(f).$$

Proof. Existence follows arguments similar to those of Theorem 2.30. See [2] for details. Note that since K is linear with a trivial null space and the squared Hilbert space norm is strictly convex, the mapping $f \mapsto ||Kf - d||^2_{L^2(\Omega)}$ is strictly convex. Uniqueness follows from strict convexity. □

The following stability result is proved in [2].

Theorem 8.20. *Suppose the hypotheses of Theorem* 8.19 *hold. Then the minimizer f_* is stable with respect to*

(i) *perturbations d_n of the data d for which $||d_n - d||_{L^2(\Omega)} \to 0$;*

(ii) *perturbations K_n of the operator K for which $||K_n(f) - K(f)||_{L^2(\Omega)} \to 0$ uniformly on compact subsets in $L^p(\Omega)$;*

(iii) *perturbations α_n of the regularization parameter $\alpha > 0$.*

Similar existence-uniqueness-stability results can be obtained when the BV norm (8.72) is replaced by the TV functional (8.68), yielding

$$T(f) = ||Kf - d||^2_{L^2(\Omega)} + \alpha \text{TV}(f). \tag{8.74}$$

The condition that K has a trivial null space can also be weakened somewhat. The following result is an example. See [2] for a proof.

Theorem 8.21. *Let C be a closed, convex subset of $L^p(\Omega)$ with $1 \leq p < d/(d-1)$. Let $K : L^p(\Omega) \to L^2(\Omega)$ be linear and bounded. Assume that $K1 \neq 0$, where 1 denotes the function $1(x) = 1$ for all $x \in \Omega$. Then the functional in* (8.74) *has a unique constrained minimizer over C.*

8.4.1 Approximations to the TV Functional

As in section 8.2.5, we replace the Euclidean norm $|\cdot|$ by a smooth, convex approximation φ. For smooth f one can define a corresponding approximation to the TV functional,

$$J(f) = \int_\Omega \varphi(\nabla f)\, dx, \tag{8.75}$$

which is analogous to the representation (8.71).

To obtain an extension of the functional in (8.75) that is valid for nonsmooth f in a manner analogous to (8.68), we make use of the dual representation

$$J(f) = \sup_{\vec{v}\in\mathcal{V}} \int_\Omega [-f \text{ div } \vec{v} - \varphi^*(\vec{v}(x))]\, dx, \tag{8.76}$$

where

$$\mathcal{V} = \{\vec{v} \in C_0^1(\Omega; \mathbb{R}^d) \mid \vec{v}(x) \in C^* \text{ for all } x \in \Omega\}.$$

Equation (8.76) is obtained from (8.35) by replacing x with ∇f, replacing y with $\vec{v}(x)$, and integrating by parts; see Remark 8.10. Motivated by Examples 8.5 and 8.6, we define

$$J_\beta(f) = \sup_{\vec{v}\in\mathcal{V}} \int_\Omega \left[-f \text{ div } \vec{v} + \beta\sqrt{1 - |\vec{v}(x)|^2}\right] dx \tag{8.77}$$

and

$$J_\epsilon(f) = \sup_{\vec{v}\in\mathcal{V}} \int_\Omega \left[-f \text{ div } \vec{v} - \frac{\epsilon}{2}|\vec{v}(x)|^2\right] dx, \tag{8.78}$$

where $\mathcal{V}$ is given in (8.69).

The following results establish stability of total variation regularized solutions with respect to perturbations (8.77) and (8.78) of the TV functional as the parameters β and ϵ tend to zero. See [2] for proofs.

Proposition 8.22. *Both J_β and J_ϵ are convex and weakly lower semicontinuous. Moreover, $J_\beta(f) \to \text{TV}(f)$ as $\beta \to 0$ and $J_\epsilon(f) \to \text{TV}(f)$ as $\epsilon \to 0$, uniformly on BV-bounded sets.*

Theorem 8.23. *Total variation regularized solutions are stable with respect to certain perturbations in the penalty functional. In particular, if* TV(f) *in* (8.74) *is replaced by either $J_\beta(f)$ or $J_\epsilon(f)$ and $\alpha > 0$ is fixed, then the corresponding regularized solutions $f_{\alpha,\beta}$ and $f_{\alpha,\epsilon}$ converge to the total variation regularized solution in L^p norm, $1 \le p < d/(d-1)$, as $\beta \to 0$ and $\epsilon \to 0$.*

Exercises

8.1. Prove that $L(\mathbf{f})$ in (8.17) is a positive semidefinite matrix. What is the null space of $L(\mathbf{f})$?

8.2. Let the functional J be as in (8.28). Recall from Remark 2.35 that its Gateaux, or directional, derivative at f in the direction h is given by

$$\delta J(f; h) = \frac{d}{d\tau} J(f + \tau h)|_{\tau=0}.$$

Show that for smooth f and h,

$$\delta J(f; h) = \int_0^1 \int_0^1 \psi'(|\nabla f|^2)\, \nabla f^T \nabla h \, dx\, dy.$$

Then show that $\delta J(f; h) = \langle \mathcal{L}(f) f, h\rangle$, provided that the normal derivative of f vanishes on the boundary of the unit square. Here $\mathcal{L}(f)$ is given in (8.27) and $\langle \cdot, \cdot \rangle$ denotes the L^2 inner product on the unit square.

8.3. Derive the two-dimensional representation for $L'(\mathbf{f})\mathbf{f}$ in (8.29).

8.4. Suppose that explicit time marching, or the forward Euler method, is applied to the system of ODEs:

$$\frac{d\mathbf{f}}{dt} = -\text{grad}\, T(f),$$

where the functional T is given in (8.8). Show that the resulting iteration is equivalent to that of the steepest descent Algorithm 8.2.3, except that the line search parameter $\tau_\nu = \Delta t$ is fixed.

8.5. Show that the right-hand side of (8.30) is equivalent to (8.31).

8.6. Verify (8.35) directly. *Hint:* Use the Cauchy–Schwarz inequality to show that the left-hand side is bounded by the right-hand side. Then show that the bound is attained.

8.7. Verify equation (8.39).

8.8. With φ_ϵ given in Example 8.6, verify that

$$\sup_{\mathbf{x}\in\mathbb{R}^d} \{\mathbf{x}^T\mathbf{y} - \varphi_\epsilon(\mathbf{x})\} = \begin{cases} \frac{\epsilon}{2}|\mathbf{y}|^2 & \text{if} \quad |\mathbf{y}| \le 1, \\ +\infty & \text{if} \quad |\mathbf{y}| > 1. \end{cases}$$

8.9. By applying block Gaussian elimination to the right-hand side of (8.55), derive the expression for $\overline{L}$ in (8.61). Also, derive (8.62).

8.10. For the matrix $\overline{L}$ in (8.61), prove that the symmetric part $(\overline{L} + \overline{L}^T)/2$ is positive semidefinite.

8.11. For the one-dimensional test problem of section 8.3.1, conduct a numerical study of the effects of varying the parameters α and β on the performance of each of the four algorithms applied in that section. In particular, what are the effects on numerical performance of making β very small?

8.12. What is the qualitative effect on the reconstructions in the one-dimensional test problem of *increasing* the parameter β?

8.13. For the one-dimensional test problem, replace the approximation (8.38) to the absolute value by (8.41). Explain why one cannot then implement either the primal Newton method or the primal-dual Newton method. Implement and compare results for the remaining two methods, the steepest descent method and the lagged diffusivity fixed point method.

8.14. For the two-dimensional test problem of section 8.3.2, implement both the steepest descent Algorithm 8.2.3 and the Newton Algorithm 8.2.3. How do these methods compare in terms of convergence rates and computational cost?

8.15. In the implementation of the lagged diffusivity fixed point method and the primal-dual Newton method for two-dimensional test problem, replace the CG linear solver with preconditioned CG. Use the level 2 block circulant preconditioner of section 5.3.3.

8.16. Prove Proposition 8.22. Use the facts that $\text{TV}(f) \le J_\beta(f) \le \text{TV}(f) + \beta\,\text{Vol}(\Omega)$ and $\text{TV}(f) - \frac{\epsilon}{2}\,\text{Vol}(\Omega) \le J_\epsilon(f) \le \text{TV}(f)$. Here $\text{Vol}(\Omega) = \int_\Omega dx$ denotes the volume of the set Ω.

Chapter 9
Nonnegativity Constraints

Imposing a priori constraints can sometimes dramatically improve the quality of solutions to inverse problems. This is particularly true in applications like astronomical imaging, where nonnegativity is important. In this chapter we present several numerical techniques to solve nonnegatively constrained optimization problems arising in image deblurring.

The methods in this chapter fall into two broad categories—variational and iterative. Iterative regularization methods that preserve nonnegativity are discussed in section 9.5. Earlier sections are devoted to variational regularization techniques. By this we mean the regularized solution is obtained by solving a (nonnegatively constrained) minimization problem.

9.1 An Illustrative Example

Consider a one-dimensional model that is similar to that presented in section 1.1. We take the same matrix K, with entries given in (1.3), but f_{true} and the model for data error are different. In particular, we take

$$f_{\text{true}}(x) = \begin{cases} 750, & 0.1 < x < 0.25, \\ 250, & 0.3 < x < 0.32, \\ 5 \times 10^5 (x - 0.75)(0.85 - x), & 0.75 < x < 0.85, \\ 0 & \text{otherwise.} \end{cases}$$

To generate the data, we take realizations of a random vector with independent components

$$d_i \sim \text{Poisson}([K\mathbf{f}_{\text{true}}]_i) + \text{Normal}(0, \sigma^2); \tag{9.1}$$

i.e., d_i is the sum of a Gaussian random variable with mean zero and variance σ^2 and a Poisson random variable with parameter $\lambda_i = [K\mathbf{f}_{\text{true}}]_i$. See section 5.1 for details. The variance of the Gaussian is taken to be $\sigma^2 = 1$.

Both $\mathbf{f}_{\text{true}}$ and the data $\mathbf{d}$ are shown in Figure 9.1. For a Poisson random variable with parameter λ, the mean and variance are both λ; see Exercise 4.4 in Chapter 4. Hence the Poisson error, with variance $\lambda_i = [K\mathbf{f}_{\text{true}}]_i$, dominates when $[K\mathbf{f}_{\text{true}}]_i$ is large. On the other hand, when $[K\mathbf{f}_{\text{true}}]_i$ is close to zero, the Gaussian error, with variance $\sigma^2 = 1$, is dominant.

We apply three different techniques to reconstruct $\mathbf{f}_{\text{true}}$. The first approach is simply to

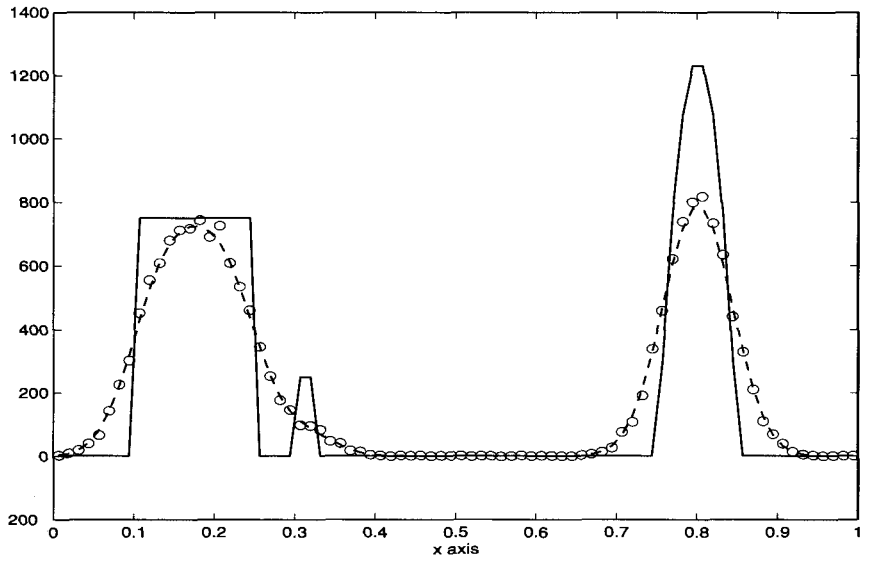


Figure 9.1. *One-dimensional test data for nonnegatively constrained regularization. The solid line represents* $\mathbf{f}_{\text{true}}$*, the dashed line represents* $K\mathbf{f}_{\text{true}}$*, and the circles denote the simulated data.*

minimize the regularized least squares functional

$$J_{\text{ls}}(\mathbf{f}) = \frac{1}{2}||K\mathbf{f} - \mathbf{d}||^2 + \frac{\alpha}{2}||\mathbf{f}||^2 \tag{9.2}$$

without any constraints. The (unconstrained) minimizer is given by

$$\mathbf{f}_\alpha^{\text{ls}} = (K^T K + \alpha I)^{-1} K^T \mathbf{d}.$$

The regularization parameter α is selected to minimize $||\mathbf{f}_\alpha^{\text{ls}} - \mathbf{f}_{\text{true}}||$, and the corresponding reconstruction is plotted in Figure 9.2.

The second approach is to solve the nonnegatively constrained least squares minimization problem

$$\min_{\mathbf{f}\in\mathbb{R}^n} J_{\text{ls}}(\mathbf{f}) \quad \text{subject to} \quad \mathbf{f} \geq \mathbf{0}. \tag{9.3}$$

By $\mathbf{f} \geq \mathbf{0}$ we mean $f_i \geq 0$ for $i = 1, \ldots, n$. The Hessian of the functional in (9.2),

$$H_{\text{ls}} = K^T K + \alpha I,$$

is positive definite, so J_{ls} is strictly convex; see Theorem 2.42. J_{ls} is also coercive (see Definition 2.29) since $J_{\text{ls}}(\mathbf{f}) \geq \alpha||\mathbf{f}||^2/2$. The constraint set, $C = \{\mathbf{f} \in \mathbb{R}^n \mid \mathbf{f} \geq \mathbf{0}\}$, is convex, so by Theorem 2.30, the constrained problem (9.3) has a unique global minimum, which we denote by $\mathbf{f}_\alpha^{\text{nnls}}$. No closed form representation exists for $\mathbf{f}_\alpha^{\text{nnls}}$. A very accurate numerical approximation, computed using the techniques of section 9.3, is shown in Figure 9.3.

Our third approach is to solve the nonnegatively constrained Poisson likelihood minimization problem,

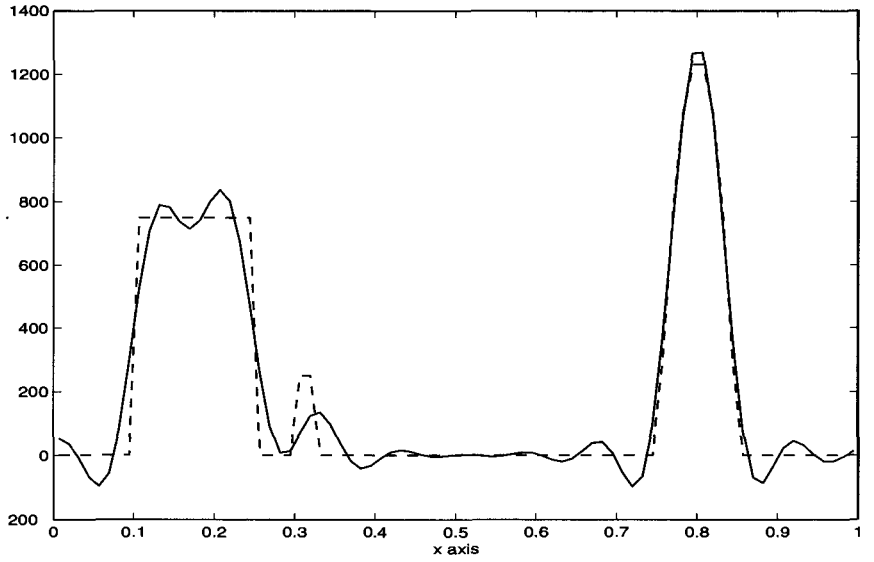


Figure 9.2. *Unconstrained least squares reconstruction from one-dimensional test data. The dashed line represents* $\mathbf{f}_{\text{true}}$*, and the solid line represents the unconstrained regularized least squares solution, obtained by minimizing the functional* (9.2).

$$\min_{\mathbf{f}\in\mathbb{R}^n} J_{\text{lhd}}(\mathbf{f}) \quad \text{subject to} \quad \mathbf{f} \geq \mathbf{0}, \tag{9.4}$$

where

$$J_{\text{lhd}}(\mathbf{f}) = \sum_{i=1}^{n}([K\mathbf{f}]_i + \sigma^2) + \sum_{i=1}^{n}\{(\overline{d}_i + \sigma^2)\ \log([K\mathbf{f}]_i + \sigma^2)\} + \frac{\alpha}{2}||\mathbf{f}||^2. \tag{9.5}$$

Here $\overline{d}_i = \max\{d_i, 0\}$. The two sums on the right-hand side of (9.5) comprise the negative log likelihood function for data $\overline{d}_i + \sigma^2$ from a Poisson distribution with mean $\lambda_i = [K\mathbf{f}]_i + \sigma^2$. The elimination of negative values of d_i and the addition of σ^2 have a stabilizing effect on the minimization when $[K\mathbf{f}]_i$ is small. When $[K\mathbf{f}]_i$ is large, the effect of adding σ^2 is insignificant.

The gradient of the likelihood functional is

$$\text{grad}\, J_{\text{lhd}}(\mathbf{f}) = K^T\,[(K\mathbf{f} - \overline{\mathbf{d}})\ ./\ (K\mathbf{f} + \sigma^2)] + \alpha\mathbf{f}, \tag{9.6}$$

where ./ denotes component-wise quotient of vectors, and the Hessian is

$$\text{Hess}\, J_{\text{lhd}}(\mathbf{f}) = K^T\, D(\mathbf{f})\, K + \alpha I, \tag{9.7}$$

where

$$D(\mathbf{f}) = \text{diag}\left(\frac{\overline{d}_i + \sigma^2}{([K\mathbf{f}]_i + \sigma^2)^2}\right). \tag{9.8}$$

Since this Hessian is positive definite whenever $\mathbf{f} \geq \mathbf{0}$, J_{lhd} is strictly convex. J_{lhd} is also coercive, so problem (9.4)–(9.5) has a unique global minimizer, which we denote by $\mathbf{f}_\alpha^{\text{lhd}}$.

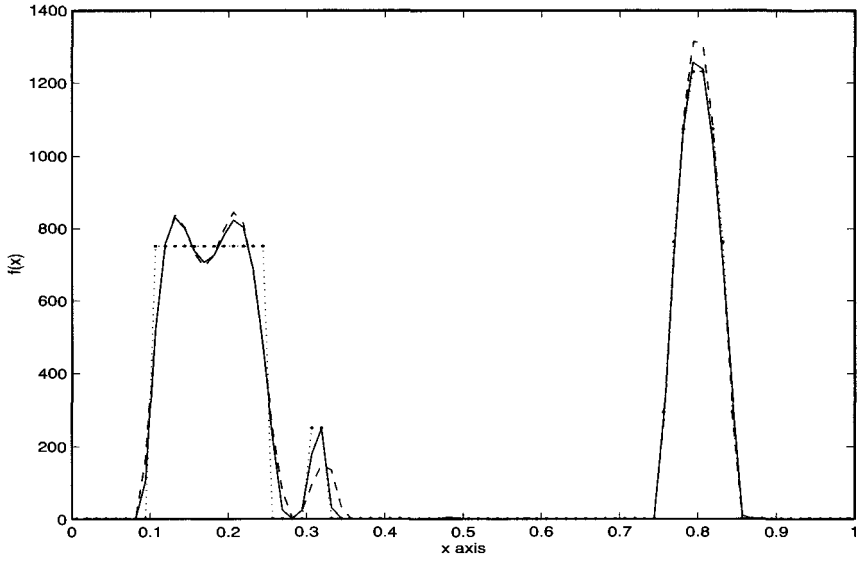


Figure 9.3. *Nonnegatively constrained reconstructions from one-dimensional test data. The dotted line represents* $\mathbf{f}_{\text{true}}$*; the dashed line represents the nonnegative least squares reconstruction* $\mathbf{f}_\alpha^{\text{nnls}}$*, obtained by solving problem* (9.3)*; and the solid line represents the nonnegative likelihood reconstruction* $\mathbf{f}_\alpha^{\text{lhd}}$*, obtained by solving problem* (9.4)–(9.5).

The regularization parameter α is selected to minimize $||\mathbf{f}_\alpha^{\text{lhd}} - \mathbf{f}_{\text{true}}||$. The resulting solution is plotted in Figure 9.3.

A comparison of the three reconstructions in Figures 9.2 and 9.3 clearly shows that imposing nonnegativity constraints can dramatically improve reconstruction quality. In addition, by incorporating the statistics of the noise in the data via the Poisson likelihood fit-to-data functional, we can obtain some improvement over the nonnegatively constrained least squares reconstruction.

We next provide some background material from finite-dimensional constrained optimization theory.

9.2 Theory of Constrained Optimization

The following definitions and the proofs of the theorems can be found in Nocedal and Wright [91] or Nash and Sofer [89]. Let $J : \mathbb{R}^n \to \mathbb{R}$ and $\mathbf{c} : \mathbb{R}^n \to \mathbb{R}^m$. Consider the inequality constrained minimization problem

$$\min_{\mathbf{f}\in\mathbb{R}^n} J(\mathbf{f}) \quad \text{subject to} \quad \mathbf{c}(\mathbf{f}) \geq \mathbf{0}. \tag{9.9}$$

By $\mathbf{c}(\mathbf{f}) \geq \mathbf{0}$ we mean $c_i(\mathbf{f}) \geq 0$ for $i = 1, \ldots, m$, where c_i is the ith component of $\mathbf{c}$. Problem (9.9) is a special case of the constrained minimization problem considered at the beginning of section 2.3. Here the constraint set, or feasible set, is $C = \{\mathbf{f} \in \mathbb{R}^n \mid \mathbf{c}(\mathbf{f}) \geq \mathbf{0}\}$.

Definition 9.1. $\mathbf{f} \in \mathbb{R}^n$ is called a feasible point for (9.9) if $\mathbf{c}(\mathbf{f}) \geq \mathbf{0}$.

Definition 9.2. Let $\mathbf{f}$ be a feasible point for (9.9). An index i is called active if $c_i(\mathbf{f}) = 0$. If $c_i(\mathbf{f}) > 0$, the index i is called inactive. The active set at $\mathbf{f}$, which we denote by $\mathcal{A}(\mathbf{f})$, consists of the active indices, i.e.,

$$\mathcal{A}(\mathbf{f}) = \{i \mid c_i(\mathbf{f}) = 0\}.$$

The set of inactive indices is denoted by $\mathcal{I}(\mathbf{f})$. $\mathcal{I}(\mathbf{f})$ is the complement of $\mathcal{A}(\mathbf{f})$ relative to the index set $\{1, \ldots, m\}$. The components f_i for which $i \in \mathcal{I}(\mathbf{f})$ are called free variables.

Definition 9.3. A feasible point $\mathbf{f}$ for (9.9) satisfies the linear independence constraint qualification (LICQ) if the set of gradient vectors corresponding to the active constraints, $\{\text{grad}\, c_i(\mathbf{f}) \mid i \in \mathcal{A}(\mathbf{f})\}$, is linearly independent. Alternatively, $\mathbf{f}$ is called a regular point.

Theorem 9.4 (First Order Necessary Conditions for Inequality Constrained Minimization). *Assume that both J and $\mathbf{c}$ are Fréchet differentiable, and let $\mathbf{f}^*$ be a local minimizer for (9.9). If $\mathbf{f}^*$ satisfies the linear independence constraint qualification, then there exists a vector $\boldsymbol{\lambda}^* = (\lambda_1^*, \ldots, \lambda_m^*)$ such that*

$$\text{grad}\, J(\mathbf{f}^*) - \sum_{i=1}^{m} \lambda_i^* \text{grad}\, c_i(\mathbf{f}^*) = \mathbf{0}, \tag{9.10}$$

and for $i = 1, \ldots, m$,

$$\lambda_i^* \geq 0, \tag{9.11}$$

$$c_i(\mathbf{f}^*) \geq 0, \tag{9.12}$$

$$\lambda_i^* c_i(\mathbf{f}^*) = 0. \tag{9.13}$$

Equations (9.10)–(9.13) are known as the Karush–Kuhn–Tucker (KKT) conditions, equation (9.13) is called a complementarity condition, and the vector $\boldsymbol{\lambda}^*$ is called the Lagrange multiplier. Conditions (9.11)–(9.13) imply that λ_i^* can be strictly positive only if the corresponding constraint is active, i.e., only if $c_i(\mathbf{f}^*) = 0$.

Definition 9.5. Let $\mathbf{f}^* \in \mathbb{R}^n$ and $\boldsymbol{\lambda}^* \in \mathbb{R}^m$ satisfy the KKT conditions (9.10)–(9.13). $\mathbf{f}^*$ and $\boldsymbol{\lambda}^*$ satisfy the strict complementarity condition if, in addition,

$$\lambda_i^* > 0 \quad \text{whenever} \quad c_i(\mathbf{f}^*) = 0. \tag{9.14}$$

The projection operator in the following definition will play an important role in many of the computational methods in this chapter.

Definition 9.6. Let $\mathcal{C}$ be a closed, convex subset of $\mathbb{R}^n$. Given $\mathbf{f} \in \mathbb{R}^n$, the (Euclidean) projection of $\mathbf{f}$ onto $\mathcal{C}$ is defined to be

$$P_{\mathcal{C}}(\mathbf{f}) = \arg\min_{\mathbf{v} \in \mathcal{C}} ||\mathbf{f} - \mathbf{v}||. \tag{9.15}$$

In other words, $P_{\mathcal{C}}(\mathbf{f})$ is the point in $\mathcal{C}$ that is closest in Euclidean distance to $\mathbf{f}$.

Theorem 9.7. *Given that $\mathcal{C}$ is closed and convex, the operator $P_{\mathcal{C}} : \mathbb{R}^n \to \mathbb{R}^n$ in (9.15) is well defined and continuous.*

See Exercise 9.2 for a proof.

9.2.1 Nonnegativity Constraints

Consider the nonnegatively constrained minimization problem

$$\min J(\mathbf{f}) \quad \text{subject to} \quad \mathbf{f} \geq \mathbf{0}. \tag{9.16}$$

This is a special case of (9.9) where $\mathbf{c}(\mathbf{f}) = \mathbf{f}$. In this case, grad $c_i = \mathbf{e}_i$, the ith standard unit vector. Thus the linear independence constraint qualification holds for any feasible point for (9.16). If $\mathbf{f}^*$ is a local minimizer for (9.16), then from (9.10) the corresponding Lagrange multiplier $\boldsymbol{\lambda}^* = \text{grad } J(\mathbf{f}^*)$. From Theorem 9.4 we obtain the following result.

Proposition 9.8. *Suppose that J is continuously differentiable and $\mathbf{f}^* = (f_1^*, \ldots, f_n^*)$ is a local minimizer for* (9.16). *Then for $i = 1, \ldots, n$,*

$$\frac{\partial J}{\partial f_i}(\mathbf{f}^*) \geq 0, \tag{9.17}$$

$$f_i^* \geq 0, \tag{9.18}$$

$$f_i^* \frac{\partial J}{\partial f_i}(\mathbf{f}^*) = 0. \tag{9.19}$$

This result could also be proved directly from Theorem 2.38. See Exercise 9.4.

Definition 9.9. A point $\mathbf{f}^*$ that satisfies (9.17)–(9.19) is called a critical point for the nonnegatively constrained minimization problem (9.16).

Remark 9.10. A critical point for problem (9.16) need not be a minimizer. However, if J is strictly convex, then there can be at most one critical point, and if such a critical point exists, it is the global minimizer for (9.16). See Exercise 9.5.

Remark 9.11. As a consequence of (9.17)–(9.19), if $\frac{\partial J}{\partial f_i}(\mathbf{f}^*) > 0$, then $f_i^* = 0$. The strict complementarity condition (9.14) for nonnegatively constrained minimization takes the following form:

$$\text{If} \quad f_i^* = 0, \quad \text{then} \quad \frac{\partial J}{\partial f_i}(\mathbf{f}^*) > 0. \tag{9.20}$$

Definition 9.12. A critical point $\mathbf{f}^*$ for (9.16) is called nondegenerate if the strict complementarity condition (9.20) holds for each $i \in \mathcal{A}(\mathbf{f}^*)$. Conversely, $\mathbf{f}^*$ is degenerate if there exists some index $i \in \mathcal{A}(\mathbf{f}^*)$ for which $f_i^* = 0$ and $\frac{\partial J}{\partial f_i}(\mathbf{f}^*) = 0$.

The feasible set in (9.16) is the closed convex set

$$\mathcal{C} = \{\mathbf{f} \in \mathbb{R}^n \mid \mathbf{f} \geq \mathbf{0}\}. \tag{9.21}$$

The projected gradient, which we define next, is useful for measuring the performance of computational methods for nonnegatively constrained optimization.

Definition 9.13. Let $\mathcal{C}$ be given in (9.21). The projected gradient, $\nabla_{\mathcal{C}} J$, is the mapping from $\mathcal{C}$ into $\mathbb{R}^n$ with components

$$[\nabla_{\mathcal{C}} J(\mathbf{f})]_i = \begin{cases} \frac{\partial J}{\partial f_i}(\mathbf{f}) & \text{if} \quad f_i > 0, \\ \min\{0, \frac{\partial J}{\partial f_i}(\mathbf{f})\} & \text{if} \quad f_i = 0. \end{cases} \tag{9.22}$$

Proposition 9.14. $\mathbf{f}^*$ *is a critical point for* (9.16) *if and only if* $\nabla_C J(\mathbf{f}^*) = \mathbf{0}$.

To simplify notation, we drop the subscript C from the convex projection operator P_C in Definition 9.6. Hence, with C given in (9.21),

$$P(\mathbf{f}) = \arg\min_{\mathbf{v}\geq\mathbf{0}} ||\mathbf{v} - \mathbf{f}||. \tag{9.23}$$

This projection operator is easy to evaluate. The ith component of $P(\mathbf{f})$ is simply

$$[P(\mathbf{f})]_i = \max(f_i, 0) = \begin{cases} f_i & \text{if } f_i \geq 0, \\ 0 & \text{if } f_i < 0. \end{cases} \tag{9.24}$$

The following proposition provides an alternative characterization of a local minimizer for problem (9.16).

Proposition 9.15. *If* $\mathbf{f}^*$ *is a local minimizer for* (9.16), *then*

$$\mathbf{f}^* = P(\mathbf{f}^* - \tau \operatorname{grad} J(\mathbf{f}^*)) \quad \textit{for any} \quad \tau > 0. \tag{9.25}$$

9.3 Numerical Methods for Nonnegatively Constrained Minimization

The methods presented in this section pertain to the nonnegatively constrained minimization problem (9.16).

Remark 9.16. Nonnegativity constraints are a special case of bound constraints, or box constraints. These take the form $a_i \leq f_i \leq b_i$, where $-\infty \leq a_i < b_i \leq +\infty$ are fixed. A great deal of effort has gone into the development of efficient numerical methods for bound-constrained optimization. The algorithms presented below are adaptations of more general bound-constrained optimization techniques.

9.3.1 The Gradient Projection Method

The gradient projection method can be viewed as a generalization of the steepest descent Algorithm 3.1 for unconstrained optimization. Alternatively, it can be viewed as a fixed point iteration for (9.25).

Algorithm 9.3.1. Gradient Projection.
To minimize $J(\mathbf{f})$ subject to $\mathbf{f} \geq \mathbf{0}$,

$\nu := 0;$
$\mathbf{f}_0 :=$ nonnegative initial guess;
begin Gradient Projection iterations
  $\mathbf{p}_\nu := -\operatorname{grad} J(\mathbf{f}_\nu);$ *% negative gradient*
  $\tau_\nu := \arg\min_{\tau>0} J(P(\mathbf{f}_\nu + \tau \mathbf{p}_\nu));$ *% projected line search*
  $\mathbf{f}_{\nu+1} := P(\mathbf{f}_\nu + \tau_\nu \mathbf{p}_\nu);$ *% update solution*
  $\nu := \nu + 1;$
end Gradient Projection iterations

An analysis of this algorithm was given by Kelley [68, section 5.4]. The following theorem summarizes two of the key results. In the case of nonnegativity constraints, the active set from Definition 9.2 reduces to

$$\mathcal{A}(\mathbf{f}) = \{i \mid f_i = 0\}.$$

Theorem 9.17. *Assume that* grad J *is Lipschitz continuous. Then*

(i) *Any limit point of the sequence* $\{\mathbf{f}_\nu\}$ *of gradient projection iterates is a critical point for* (9.16).

(ii) *If the gradient projection iterates converge to a local minimizer* $\mathbf{f}^*$ *for* (9.16), *and the strict complementarity condition* (9.20) *is satisfied, then the optimal active set* $\mathcal{A}(\mathbf{f}^*)$ *is identified in finitely many iterations; i.e., there exists* ν_0 *such that* $\mathcal{A}(\mathbf{f}_\nu) = \mathcal{A}(\mathbf{f}^*)$ *whenever* $\nu \geq \nu_0$.

Corollary 9.18. *If* J *is strictly convex, coercive, and Lipschitz continuous, then the gradient projection iterates* $\mathbf{f}_\nu$ *converge to the global minimizer* $\mathbf{f}^*$ *of* J.

Proof. A unique minimizer $\mathbf{f}^*$ for (9.16) exists by Theorem 2.30. The sequence $\{J(\mathbf{f}_\nu)\}$ is nonincreasing and bounded below by $J(\mathbf{f}^*)$. Since J is coercive, $\{\mathbf{f}_\nu\}$ is bounded and hence must have a limit point $\hat{\mathbf{f}}$. By part (i) of Theorem 9.17, $\hat{\mathbf{f}}$ is a critical point for (9.16). But by Remark 9.10, $\hat{\mathbf{f}} = \mathbf{f}^*$. □

Remark 9.19. In a manner analogous to unconstrained optimization (see section 3.4), one can replace the (exact) projected line search with an inexact projected line search. The sufficient decrease condition (3.28) can be replaced by

$$\phi(\tau) \leq \phi(0) + c_1 \langle \operatorname{grad} J(\mathbf{f}_\nu), P(\mathbf{f}_\nu + \tau \mathbf{p}_\nu) - \mathbf{f}_\nu \rangle, \tag{9.26}$$

where $\phi(\tau) = J(P(\mathbf{f}_\nu + \tau \mathbf{p}_\nu))$. The implementation of interpolating backtracking schemes like the quadratic scheme in section 3.4 is complicated by fact that $\phi(\tau)$ may have jumps in its derivative, because the path $\{P(\mathbf{f}_\nu + \tau \mathbf{p}) \mid \tau \geq 0\}$ is only piecewise linear.

Once the active set has been identified, the gradient projection method behaves like the steepest descent method on the inactive constraints. This results in an asymptotically linear convergence rate. To improve this rate, one needs to incorporate second order derivative information.

9.3.2 A Projected Newton Method

At first glance, it appears that one can simply replace the search direction $\mathbf{p} = -\operatorname{grad} J(\mathbf{f})$ in the gradient projection algorithm with the Newton direction $\mathbf{s} = -[\operatorname{Hess} J(\mathbf{f})]^{-1} \operatorname{grad} J(\mathbf{f})$ to obtain a quadratically convergent algorithm. Unfortunately, this is not the case. In fact, one can construct examples for which a feasible point $\mathbf{f}$ is not a local minimizer for (9.16), but $J(P(\mathbf{f} + \tau \mathbf{s})) \geq J(\mathbf{f})$ for any $\tau \geq 0$. See, for example, Bertsekas [8, p. 77] or [68, section 5.5.1]. This difficulty can be overcome by replacing the Hessian with the reduced Hessian, defined as follows.

Definition 9.20. The reduced Hessian, $H_R = H_R(\mathbf{f})$, has entries

$$[H_R]_{ij} = \begin{cases} \delta_{ij} & \text{if} \quad i \in \mathcal{A}(\mathbf{f}) \quad \text{or} \quad j \in \mathcal{A}(\mathbf{f}), \\ \frac{\partial^2 J}{\partial f_i \partial f_j} & \text{otherwise.} \end{cases} \tag{9.27}$$

Let H denote the ordinary, unreduced Hessian of J. Then

$$H_R = D_{\mathcal{I}} H D_{\mathcal{I}} + D_{\mathcal{A}}, \tag{9.28}$$

where $D_{\mathcal{A}}$ is the diagonal matrix whose ith diagonal entry is 1 if i lies in the active set $\mathcal{A}(\mathbf{f})$, and zero otherwise, and $D_{\mathcal{I}} = I - D_{\mathcal{A}}$.

To obtain a framework for a projected Newton method, we replace the descent direction $\mathbf{p} = -\text{grad}\, J(\mathbf{f})$ in Algorithm 9.3.1 with the projected Newton step, $\mathbf{s} = -H_R(\mathbf{f})^{-1}\, \text{grad}\, J(\mathbf{f})$. If the active set can be correctly identified, this will yield a locally quadratically convergent algorithm; see Theorem 3.11.

Algorithm 9.3.2. Projected Newton Method.
To minimize $J(\mathbf{f})$ subject to $\mathbf{f} \geq \mathbf{0}$,

$\nu := 0;$
$\mathbf{f}_0 :=$ nonnegative initial guess;
begin Projected Newton iterations
  $\mathbf{g}_\nu := \text{grad}\, J(\mathbf{f}_\nu);$ *% gradient*
  Identify active set $\mathcal{A}_\nu$;
  $H_R :=$ reduced Hessian (9.27) at $\mathbf{f}_\nu$;
  $\mathbf{s} := -H_R^{-1}\mathbf{g}_\nu;$ *% projected Newton step*
  $\tau_\nu := \arg\min_{\tau>0} J(P(\mathbf{f}_\nu + \tau\mathbf{s}));$ *% proj. line search*
  $\mathbf{f}_{\nu+1} := P(\mathbf{f}_\nu + \tau_\nu \mathbf{s});$ *% update solution*
  $\nu := \nu + 1;$
end Projected Newton iterations

Unfortunately, difficulties can sometimes arise in the identification of the active set. To make the identification of the active set more robust, one can define for $\epsilon > 0$ the ϵ-active set of indices,

$$\mathcal{A}_\epsilon(\mathbf{f}) = \{i \mid 0 \leq [\mathbf{f}]_i < \epsilon\}. \tag{9.29}$$

One varies the parameter $\epsilon > 0$ with the iteration count ν in a manner for which $\epsilon(\nu) \to 0$ as the size of the projected gradient norm, $||\nabla_{\mathcal{C}} J(\mathbf{f}_\nu)||$, tends to zero. One also replaces the reduced Hessian (9.27) by the ϵ-reduced Hessian,

$$[H_R(\mathbf{f};\epsilon)]_{ij} = \begin{cases} \delta_{ij} & \text{if} \quad i \in \mathcal{A}_\epsilon(\mathbf{f}) \quad \text{or} \quad j \in \mathcal{A}_\epsilon(\mathbf{f}), \\ \frac{\partial^2 J}{\partial f_i \partial f_j}(\mathbf{f}) & \text{otherwise.} \end{cases} \tag{9.30}$$

See [8] or [68] for details.

9.3.3 A Gradient Projection-Reduced Newton Method

We now present an alternative to the above projected Newton method, which is guaranteed to converge at a locally quadratic rate, provided that the conditions of Theorem 9.17 and

Corollary 9.18 hold and the Hessian is Lipschitz continuous. Each iteration of this method has two stages. The computations in the first stage are identical to those carried out in a gradient projection iteration; see Algorithm 9.3.1. The second stage can be viewed as the application of Newton's method to solve a version of (9.16) restricted to the free variables.

The reduced gradient $\mathbf{g}_R$ in the gradient projection-reduced Newton (GPRN) algorithm below has components

$$[\mathbf{g}_R(\mathbf{f})]_i = \begin{cases} 0 & \text{if} \quad i \in \mathcal{A}(\mathbf{f}), \\ \frac{\partial J}{\partial f_i}(\mathbf{f}) & \text{otherwise.} \end{cases} \tag{9.31}$$

Algorithm 9.3.3. GPRN.
To minimize $J(\mathbf{f})$ subject to $\mathbf{f} \geq \mathbf{0}$,

$\nu := 0$;
$\mathbf{f}_0 :=$ nonnegative initial guess;
begin GPRN iterations
Gradient Projection Stage
$\mathbf{p}^{GP} := -\text{grad}\, J(\mathbf{f}_\nu)$; *% negative gradient*
$\tau^{GP} := \arg\min_{\tau>0} J(P(\mathbf{f}_\nu + \tau \mathbf{p}^{GP}))$; *% projected line search*
$\mathbf{f}_\nu^{GP} := P(\mathbf{f}_\nu + \tau^{GP}\mathbf{p}^{GP})$;
Reduced Newton Stage
Identify active set $\mathcal{A}(\mathbf{f}_\nu^{GP})$;
$\mathbf{g}_R :=$ reduced gradient (9.31) at $\mathbf{f}_\nu^{GP}$;
$H_R :=$ reduced Hessian (9.27) at $\mathbf{f}_\nu^{GP}$;
$\mathbf{s} := -H_R^{-1}\mathbf{g}_R$; *% reduced Newton step*
$\tau^{RN} := \arg\min_{\tau>0} J(P(\mathbf{f}_\nu^{GP} + \tau\mathbf{s}))$; *% projected line search*
$\mathbf{f}_{\nu+1} := P(\mathbf{f}_\nu^{GP} + \tau^{RN}\mathbf{s})$;
$\nu := \nu + 1$;
end GPRN iterations

Remark 9.21. Assume that the strict complementarity condition (9.20) holds. Then

1. The convergence properties of the gradient projection Algorithm 9.3.1 described in Theorem 9.17 and Corollary 9.18 are retained, since $J(\mathbf{f}_{\nu+1}^{GP}) \leq J(\mathbf{f}_{\nu+1}) \leq J(\mathbf{f}_\nu^{GP})$. In particular, the optimal active set is identified in finitely many iterations, i.e., for sufficiently large ν, $\mathcal{A}(\mathbf{f}_\nu) = \mathcal{A}(\mathbf{f}^*)$.

2. The solution at the end of the reduced Newton stage, $\mathbf{f}_{\nu+1}$, gives an approximation to the problem

$$\min_{\mathbf{f}} J(\mathbf{f}) \quad \text{subject to} \quad \mathbf{f} \in \mathcal{F}(\mathbf{f}_\nu^{GP}), \tag{9.32}$$

where

$$\mathcal{F}(\mathbf{f}) \stackrel{\text{def}}{=} \{\mathbf{v} \in C \mid v_i = 0 \quad \text{whenever} \quad i \in \mathcal{A}(\mathbf{f})\}. \tag{9.33}$$

The set $\mathcal{F}(\mathbf{f})$ is called the face associated with $\mathbf{f}$. We say informally that the Newton stage "explores" the face associated with $\mathbf{f}_\nu^{GP}$. If $\mathcal{A}(\mathbf{f}_\nu^{GP})$ contains m indices, then (9.32) can be viewed as a minimization problem in the $n - m$ free (i.e., inactive) variables.

3. Eventually $\mathcal{F}(\mathbf{f}_\nu^{GP}) = \mathcal{F}(\mathbf{f}^*)$, the optimal face. This is a consequence of item 1. When this occurs, the solution to (9.32) is $\mathbf{f}^*$ and it lies on the interior of the face (because of strict complementarity). In effect, we then have an unconstrained problem in the free variables. Newton's method for (9.32) will then converge locally at a quadratic rate, provided certain mild conditions are met; see Theorem 3.11. Alternating gradient projection steps with Newton steps will not change this asymptotic rate.

Remark 9.22. Several modifications can be made to the GPRN Algorithm 9.3.3 to increase its efficiency.

1. More than one iteration can be taken in the gradient projection stage. This is cost effective if either the gradient projection steps significantly decrease the functional J or the active set changes rapidly. To quantify this, let $\mathbf{f}_{\nu,j}^{GP}$ denote the jth iterate in the projected gradient stage. One stops the iteration if either

$$J(\mathbf{f}_{\nu,j-1}^{GP}) - J(\mathbf{f}_{\nu,j}^{GP}) \leq \gamma \max_{i<j} J(\mathbf{f}_{\nu,i-1}^{GP}) - J(\mathbf{f}_{\nu,i}^{GP}), \tag{9.34}$$

where $\gamma > 0$ is fixed, or

$$\mathcal{A}(\mathbf{f}_{\nu,j}^{GP}) = \mathcal{A}(\mathbf{f}_{\nu,j-1}^{GP}). \tag{9.35}$$

2. An iterative method like the conjugate gradient method is used to explore the νth face $\mathcal{F}(\mathbf{f}_\nu^{GP})$; see item 2 in Remark 9.21. This forms the basis of the method presented next.

9.3.4 A Gradient Projection-CG Method

For many large-scale problems, it is not feasible to directly solve the linear system $H_R\mathbf{s} = -\mathbf{g}_R$ in the reduced Newton stage of the GPRN Algorithm 9.3.3. An obvious alternative is to then apply an iterative method like the CG Algorithm 3.2 to this system. It is natural to view CG in this context as a tool with which to solve the reduced minimization problem (9.32)–(9.33) instead of as a linear system solver. In other words, at each outer iteration ν, we use CG to explore the face $\mathcal{F}(\mathbf{f}_\nu^{PG})$.

What follows is a sketch of a modified version of the gradient projection-CG (GPCG) algorithm for bound-constrained quadratic minimization due to Moré and Toraldo [85]. See also Friedlander and Martinez [39]. As in [85], at each outer iteration ν we first employ a gradient projection stage. We perform projected gradient iterations (Algorithm 9.3.2) with initial guess $\mathbf{f}_\nu$ and stopping criteria (9.34)–(9.35) to obtain a new approximation $\mathbf{f}_\nu^{GP}$.

In the second stage, we form a quadratic approximation to J about $\mathbf{f}_\nu^{GP}$,

$$Q_\nu(\mathbf{p}) = J(\mathbf{f}_\nu^{GP}) + \langle \operatorname{grad} J(\mathbf{f}_\nu^{GP}), \mathbf{p} \rangle + \frac{1}{2} \langle \operatorname{Hess} J(\mathbf{f}_\nu^{GP})\, \mathbf{p}, \mathbf{p} \rangle \tag{9.36}$$

(no approximation is needed if J is quadratic), and we apply CG iteration, with initial guess $\mathbf{p}_0 = \mathbf{0}$, to solve

$$\min Q_\nu(\mathbf{p}) \quad \text{subject to} \quad p_i = 0 \text{ whenever } i \in \mathcal{A}(\mathbf{f}_\nu^{GP}). \tag{9.37}$$

Note that this can be viewed as an unconstrained quadratic minimization problem in the $n - m$ free variables, where m is the number of indices in $\mathcal{A}(\mathbf{f}_\nu^{GP})$.

At each CG iteration j, we test for insufficient decrease in Q_ν with a condition analogous to (9.34). If insufficient decrease is detected at iteration j, we perform a projected line search to obtain

$$\mathbf{f} = \arg\min_{\tau>0} J\left(P(\mathbf{f}_\nu^{GP} + \tau \mathbf{p}_j)\right). \tag{9.38}$$

If $\mathcal{A}(\mathbf{f}) = \mathcal{A}(\mathbf{f}_\nu^{GP})$, we resume the CG iterations. Otherwise, we check if $\mathcal{B}(\mathbf{f}) = \mathcal{B}(\mathbf{f}_\nu^{GP})$, where

$$\mathcal{B}(\mathbf{f}) \stackrel{\text{def}}{=} \left\{ i \mid f_i = 0 \text{ and } \frac{\partial J}{\partial f_i}(\mathbf{f}) > 0 \right\} \tag{9.39}$$

is called the binding set. If the binding sets are equal, we reinitialize CG and then proceed with the CG iterations. Otherwise, we take $\mathbf{f}_{\nu+1}$ equal to the $\mathbf{f}$ in (9.38) and proceed to the gradient projection stage of the next GPCG iteration.

9.3.5 Other Methods

A number of other large-scale bound-constrained optimization techniques can be applied to minimize functionals like (9.2) and (9.5) with nonnegativity constraints. These include interior point methods [23], trust region methods [24, 40, 76], and bound constrained limited memory BFGS methods [11].

It should be noted that other fit-to-data functionals and other regularization functionals can be employed. For example, astronomers commonly use a discrete version of negative entropy (2.50) as a regularization functional.

9.4 Numerical Test Results

We now apply some of the nonnegatively constrained minimization techniques of the previous section to several test problems. As in previous chapters, we define the relative iterative solution error norm

$$e_\nu = \frac{||\mathbf{f}_\nu - \mathbf{f}^*||}{||\mathbf{f}^*||}, \tag{9.40}$$

where $\mathbf{f}_\nu$ is an approximate solution to a nonnegatively constrained minimization problem of the form (9.16) obtained after ν iterations of a particular method, and $\mathbf{f}^*$ is a numerically exact solution, by which we mean an extremely accurate approximation obtained numerically. In all cases, the initial guess is taken to be the zero vector, $\mathbf{f}_0 = \mathbf{0}$.

9.4.1 Results for One-Dimensional Test Problems

Consider the penalized least squares functional J_{ls} in (9.2) and the penalized Poisson likelihood functional J_{lhd} in (9.5). The data and the values of the parameters α and σ^2 are the same as in section 9.1.

We first examine the behavior of the gradient projection Algorithm 9.3.1. Figure 9.4 illustrates the performance of this method for the minimization of the Poisson likelihood functional (9.5). The jumps in the graph are associated with the changes in the active set $\mathcal{A}(\mathbf{f}_\nu)$. As might be expected, the convergence rate for this method is quite slow. Somewhat faster convergence is observed when the gradient projection method is applied to the nonnegatively constrained minimization of the least squares functional (9.2).

We next examine the behavior of the GPRN Algorithm 9.3.3. The algorithm was slightly modified to allow more than one step in the projected gradient stage; see item 1 of

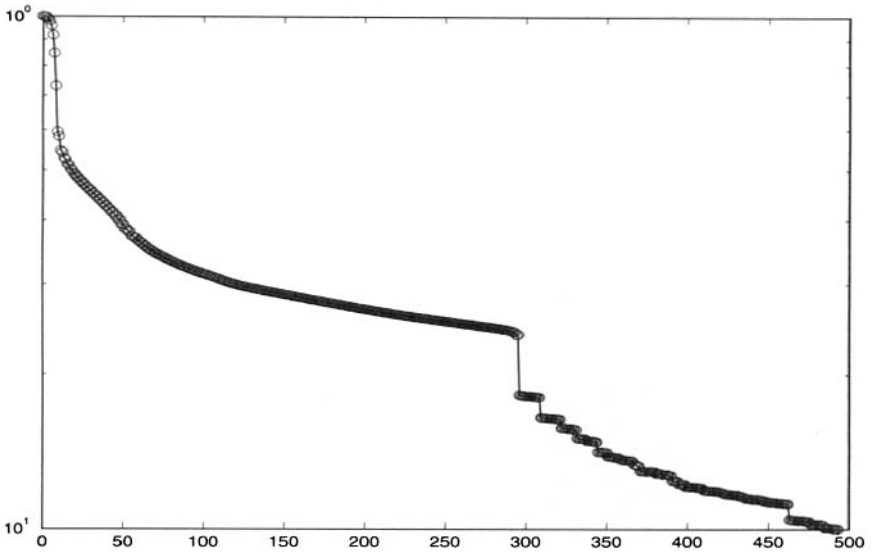


Figure 9.4. *Performance of the gradient projection method for a one-dimensional test problem. Plotted on a log scale against iteration count ν is the relative iterative solution error norm* (9.40) *for the solution of problem* (9.4)–(9.5).

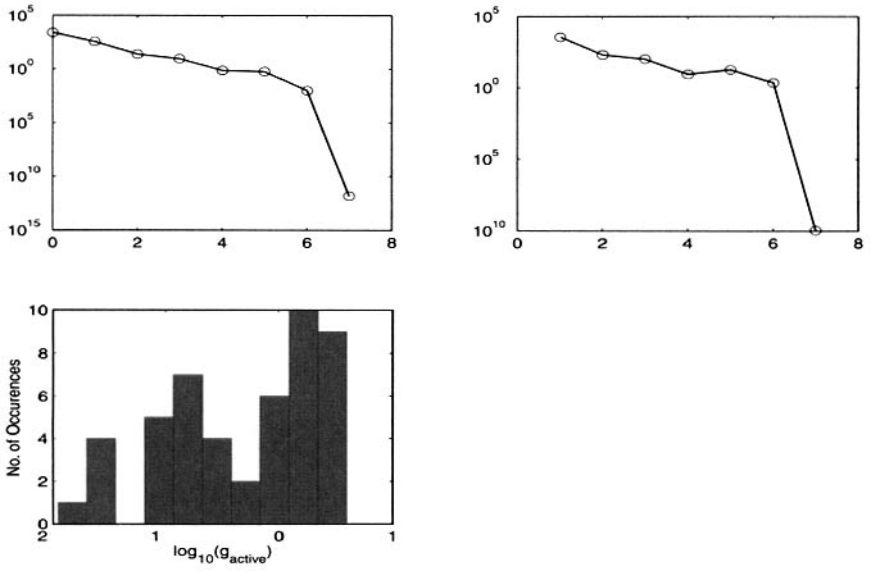


Figure 9.5. *Performance of the GPRN method for a one-dimensional least squares minimization problem. The upper left subplot shows the norm of the projected gradient* (9.22) *versus iteration count ν; the upper right subplot shows step norm $||\mathbf{f}_{\nu+1} - \mathbf{f}_\nu||$ versus ν; and the lower left subplot shows the distribution of the active components of the gradient.*

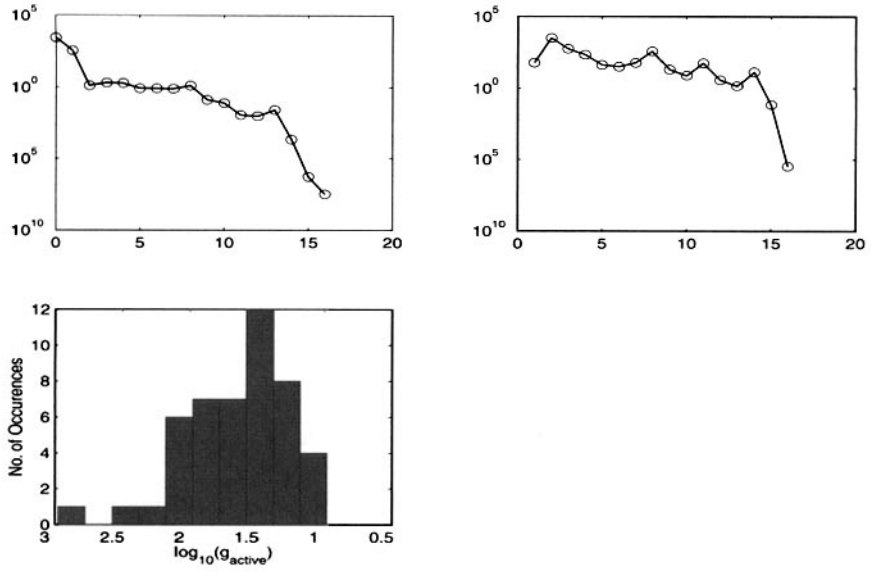


Figure 9.6. *Performance of the GPRN method for a one-dimensional Poisson likelihood minimization problem. The upper left subplot shows the norm of the projected gradient* (9.22) *versus iteration count ν; the upper right subplot shows step norm $||\mathbf{f}_{\nu+1} - \mathbf{f}_\nu||$ versus ν; and the lower left subplot shows the distribution of the active components of the gradient.*

Remark 9.22. We minimized both the least squares functional (9.2) and the Poisson likelihood functional (9.5). The associated numerical performance results are presented in Figures 9.5 and 9.6, respectively.

The lower left subplots in these figures give histograms of the log transform of the active components of the gradient, $\log_{10} \frac{\partial J}{\partial f_i}(\mathbf{f}^*)$, $i \in \mathcal{A}(\mathbf{f}^*)$. This allows the visualization and quantification of the degree of degeneracy of the constrained minimization problems; see Definition 9.12.

A comparison of Figures 9.5 and 9.6 shows that minimization of the Poisson likelihood functional (9.5) required twice as many GPRN iterations as were needed for the least squares functional (9.2). This is consistent with the fact that the Poisson likelihood minimization problem is more nearly degenerate. Moreover, the least squares functional is quadratic in $\mathbf{f}$. Hence, once the optimal active set has been identified, convergence follows in a single iteration.

As should be expected, the GPRN method converges much more rapidly than does the gradient projection method; compare Figures 9.6 and 9.4.

9.4.2 Results for a Two-Dimensional Test Problem

We generated two-dimensional blurred image data according to the model described in section 5.1.1. These data can be represented as (9.1), where now K is BTTB; see section 5.2.5. $\mathbf{f}_{\text{true}}$ is displayed in the top subplot in Figure 5.2, and the data $\mathbf{d}$ look very similar to that shown in the bottom subplot in this figure. (The data noise in the bottom subplot has a Gaussian distribution; the data noise here has a Poisson component as well as a Gaussian component whose variance is $\sigma^2 = 25$.)

We solved the corresponding nonnegatively constrained least squares problem (9.3), (9.2) using an implementation of the GPCG method sketched in section 9.3.4. As in the one-dimensional case, the regularization parameter α was selected to minimize $||\mathbf{f}_\alpha - \mathbf{f}_{\text{true}}||$. The resulting reconstruction and a numerical performance indicator (the projected gradient norm) are shown in Figure 9.7. The steady linear convergence rate is a consequence of the fact that the linear systems in the CG stage are solved inexactly. This is in contrast to the one-dimensional GPRN results, where the linear systems in the reduced Newton stage were solved exactly.

The dominant costs of the GPCG algorithm for this problem consist of matrix-vector multiplications involving the matrix K in evaluation of the cost functional and matrix-vector multiplications involving both K and K^T in evaluation of the gradient and in the CG iterations. Each such multiplication was carried out using a two-dimensional forward and inverse FFT pair; see Algorithm 5.2.5. A total of 409 FFTs were required for the 15 outer GPCG iterations. However, the active set remained unchanged and the reconstruction changed very little after outer iteration 7. To carry out 7 iterations required 269 FFTs. It is likely that this FFT count can be reduced by fine-tuning our implementation of the algorithm.

9.5 Iterative Nonnegative Regularization Methods

The methods presented earlier in this chapter can be classified as variational regularization methods. Iterative regularization methods were briefly introduced in section 1.4. Recall that by "iterative regularization" we mean that the iteration count plays the role of the regularization parameter.

The two iterative techniques presented in this section, the EM algorithm and the modified residual norm steepest descent (MRNSD) algorithm, both impose nonnegativity constraints and use iteration count as the regularization parameter.

9.5.1 Richardson–Lucy Iteration

Recall that in section 4.5.1 we derived an EM iteration (4.66) to maximize

$$J(\mathbf{f}) = \sum_{i=1}^{m} d_i \log[K\mathbf{f}]_i \tag{9.41}$$

subject to the constraints $f_i \geq 0$ and $\sum_{i=1}^{n} f_i = 1$. Note that d_i plays the role of g_i. This derivation required assumptions (4.51)–(4.57) needed to guarantee that $k_{ij} f_j$ is a probability mass function.

If the scaling used to enforce the condition (4.54) (see Exercise 4.19) is inserted in (4.66), we obtain the Richardson–Lucy (R-L) iteration [96, 78]:

$$f_j^{\nu+1} = \frac{f_j^\nu}{k_j} \sum_{i=1}^{m} k_{ij} \left(\frac{d_i}{\sum_{l=1}^{n} k_{il} f_l^\nu} \right), \qquad \text{where} \quad k_j = \sum_{l=1}^{m} k_{lj}. \tag{9.42}$$

If we multiply each side of (9.42) by k_j and sum over j, we obtain

$$\sum_{i=1}^{m} [K\mathbf{f}]_i = \sum_{i=1}^{m} d_i. \tag{9.43}$$

Hence, (9.42) yields a sequence of approximations to the maximizer of (9.41) subject to the constraints (9.43) and $f_j \geq 0,\ j = 1, \ldots, n$. One can show (see Exercise 9.14) that any maximizer of (9.41) subject to the equality constraint (9.43) is a maximizer of the Poisson log likelihood function

$$l(\mathbf{f}; \mathbf{d}) = \sum_{i=1}^{m} \{-[K\mathbf{f}]_i + d_i \log[K\mathbf{f}]_i\} \tag{9.44}$$

without this equality constraint.

We applied the R-L iteration (9.42) to the two-dimensional test problem of section 9.4.2. Because the Gaussian term in the data error may lead to negative components (see (9.1)), some data preprocessing is required. We replaced negative data components with zeros to obtain a new vector $\bar{\mathbf{d}}$. We also added the Gaussian error variance, $\sigma^2 = 25$, to the denominator term in the iteration (9.42). The resulting iteration can be expressed in matrix-vector form as

$$\mathbf{f}_{\nu+1} = \mathbf{f}_\nu \,./\, (K^T\mathbf{1}) \,.*\, K^T(\bar{d} \,./[\, K\mathbf{f}_\nu + \sigma^2\mathbf{1}]), \tag{9.45}$$

where $\mathbf{1}$ denotes the vector of 1's, and .* and ./ denote component-wise multiplication and division, respectively. Each iteration requires one matrix-vector multiplication involving K and one multiplication involving K^T. As with GPCG, these were carried out using two-dimensional FFTs, and they dominated the computational cost of the algorithm. The total number of FFTs required for each R-L iteration is 4.

The bottom subplot in Figure 9.8 shows the relative solution error norm, $||\mathbf{f}_\nu - \mathbf{f}_{\text{true}}||/||\mathbf{f}_{\text{true}}||$, versus the iteration count ν. The minimum error occurred at iteration $\nu = 57$. Ignoring setup costs, R-L required 228 FFTs. This is slightly less than the number of FFTs required of the GPCG algorithm; see section 9.4.2.

The minimum error R-L reconstruction is shown at the top of Figure 9.8. Qualitatively, this looks very similar to the nonnegative least squares (NNLS) reconstruction, shown at the top of Figure 9.7.

9.5.2 A Modified Steepest Descent Algorithm

The method presented in this section was introduced by Kaufman [66], who referred to it as the EM-least squares (EMLS) algorithm. Our derivation follows that in [88] and is based on ideas in [54]. This approach also provides an alternative derivation of R-L iteration (9.42). See Exercise 9.18.

The squared residual norm for the equation $K\mathbf{f} = \mathbf{d}$ is proportional to

$$J(\mathbf{f}) = \frac{1}{2}||K\mathbf{f} - \mathbf{d}||^2. \tag{9.46}$$

The substitution

$$\mathbf{f} = e^{\mathbf{z}} \tag{9.47}$$

(i.e., $f_i = \exp(z_i),\ i = 1, \ldots, n$) yields

$$\tilde{J}(\mathbf{z}) = \frac{1}{2}||Ke^{\mathbf{z}} - \mathbf{d}||^2. \tag{9.48}$$

To compute the gradient of $\tilde{J}$, we take the approach in Example 2.36. The directional

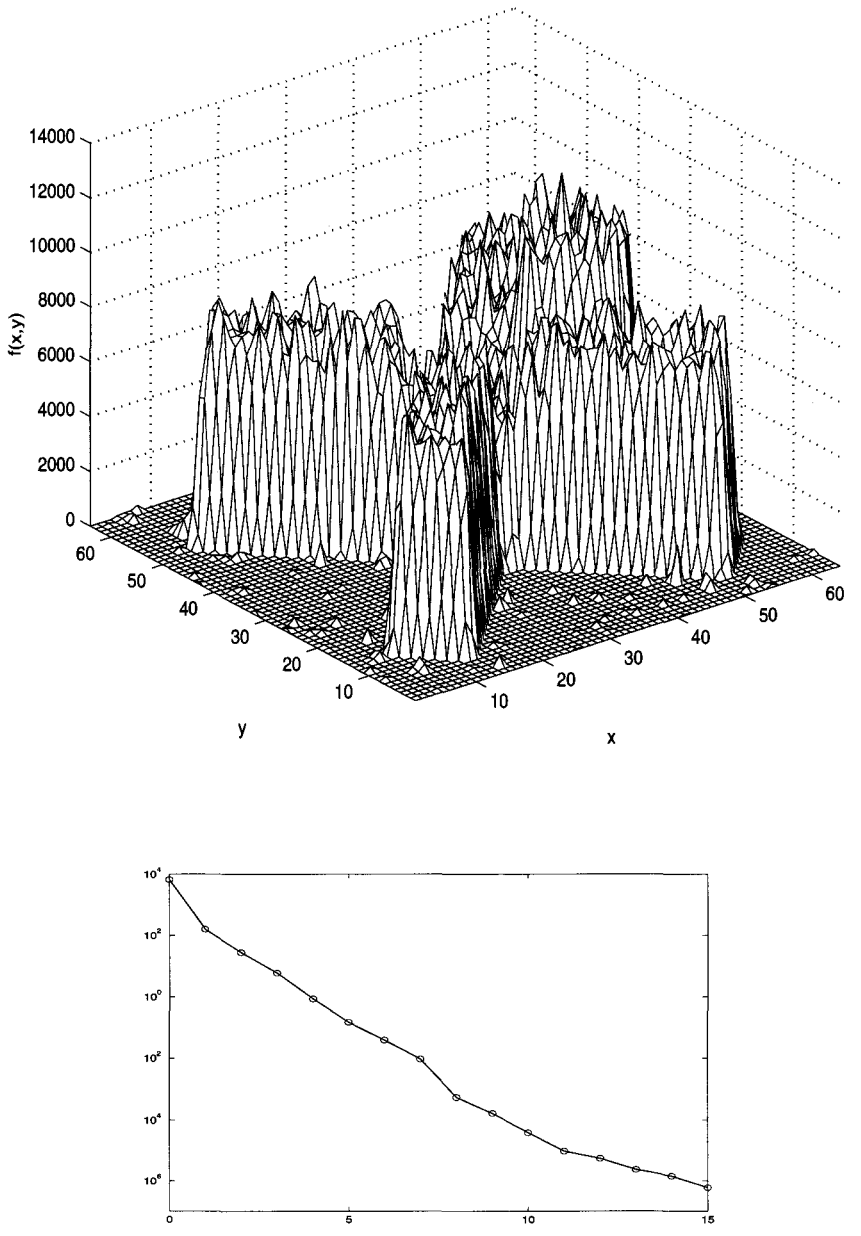


Figure 9.7. *Results for nonnegatively constrained regularized least squares minimization using a GPCG method. The top plot shows the reconstructed image. The bottom plot shows the norm of the projected gradient versus outer GPCG iteration.*

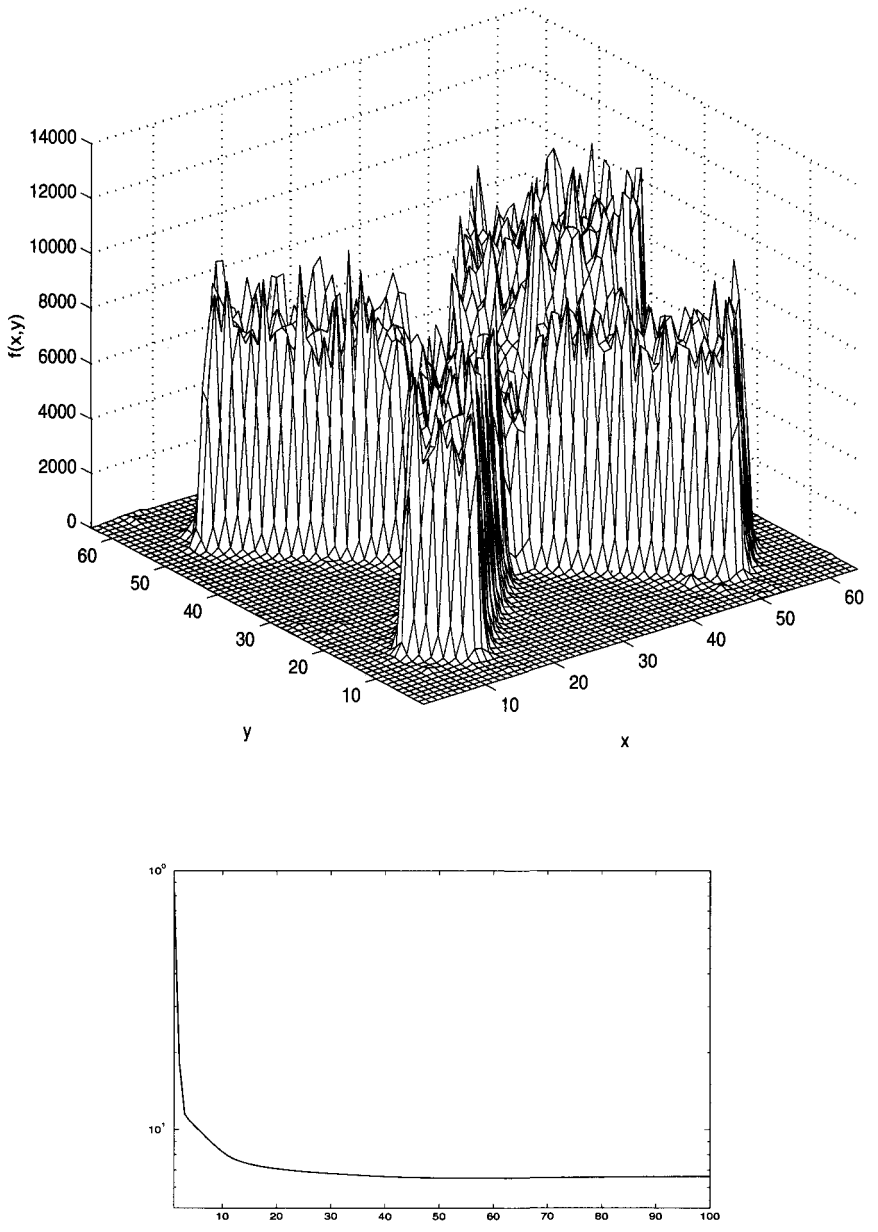


Figure 9.8. *Results for R–L iteration. The top plot shows the reconstructed image. The bottom plot shows the relative norm of the iterative solution error.*

derivative in the direction $\mathbf{v}$ is given by

$$\begin{aligned}\frac{d}{d\tau}\tilde{J}(\mathbf{z}+\tau\mathbf{v})|_{\tau=0} &= \left\langle Ke^{\mathbf{z}}-\mathbf{d}, K\left(\frac{d}{d\tau}e^{\mathbf{z}+\tau\mathbf{v}}|_{\tau=0}\right)\right\rangle \\ &= \langle K^T(Ke^{\mathbf{z}}-\mathbf{d}), e^{\mathbf{z}}.*\mathbf{v}\rangle \\ &= \langle e^{\mathbf{z}}.*[K^T(Ke^{\mathbf{z}}-\mathbf{d})], \mathbf{v}\rangle,\end{aligned}$$

where $\langle\cdot,\cdot\rangle$ denotes Euclidean inner product and $.*$ denotes the component-wise product of vectors. To obtain the last equality, we used the fact that $\langle \mathbf{x}, \mathbf{u}.*\mathbf{v}\rangle = \langle \mathbf{u}.*\mathbf{x}, \mathbf{v}\rangle$. Consequently,

$$\begin{aligned}\operatorname{grad}\tilde{J}(\mathbf{z}) &= e^{\mathbf{z}}.*[K^T(Ke^{\mathbf{z}}-\mathbf{d})] \\ &= \mathbf{f}.*[K^T(K\mathbf{f}-\mathbf{d})] \\ &= \mathbf{f}.*\operatorname{grad}J(\mathbf{f}).\end{aligned} \tag{9.49}$$

A necessary condition for a minimizer of J subject to $\mathbf{f}\geq\mathbf{0}$ is then

$$\mathbf{f}.*\operatorname{grad}J(\mathbf{f})=\mathbf{0}. \tag{9.50}$$

Note that this is the KKT condition (9.19). Equation (9.50) is equivalent to

$$\mathbf{f}=\mathbf{f}-\tau\mathbf{f}.*\operatorname{grad}J(\mathbf{f}), \qquad \tau>0. \tag{9.51}$$

A corresponding fixed point iteration is

$$\mathbf{f}_{\nu+1}=\mathbf{f}_\nu+\tau\mathbf{p}_\nu, \qquad \text{where} \quad \mathbf{p}_\nu=-\mathbf{f}_\nu.*\operatorname{grad}J(\mathbf{f}_\nu). \tag{9.52}$$

This resembles standard steepest descent iteration (see Section 3.1), except that the negative gradient is weighted by the components of the approximate solution. An unconstrained line search to determine τ yields

$$\tau_{\text{uncnstr}}=-\frac{\langle\mathbf{p}_\nu, K^T(K\mathbf{f}_\nu-\mathbf{d})\rangle}{\langle\mathbf{p}_\nu, K^TK\mathbf{p}_\nu\rangle}. \tag{9.53}$$

However, $\mathbf{f}_\nu+\tau_{\text{uncnstr}}\mathbf{p}_\nu$ may not be a feasible point, i.e., it may have negative components. Assume that $\mathbf{f}_\nu$ is feasible. To maintain feasibility, we take $\mathbf{f}_{\nu+1}=\mathbf{f}_\nu+\tau_\nu\mathbf{p}_\nu$, where

$$\tau_\nu=\min\{\tau_{\text{uncnstr}}, \tau_{\text{bndry}}\}, \tag{9.54}$$

with

$$\tau_{\text{bndry}}\overset{\text{def}}{=}\max\{\tau>0\mid\mathbf{f}_\nu+\tau\mathbf{p}_\nu\geq\mathbf{0}\}=\min\left\{-\frac{[\mathbf{f}_\nu]_i}{[\mathbf{p}_\nu]_i}\mid[\mathbf{p}_\nu]_i<0\right\}. \tag{9.55}$$

Following [88], the resulting algorithm is the MRNSD algorithm.

Algorithm 9.5.1. MRNSD.
To minimize $J(\mathbf{f})=\frac{1}{2}||K\mathbf{f}-\mathbf{d}||^2$ subject to $\mathbf{f}\geq\mathbf{0}$,

$\nu := 0;$
$\mathbf{f}_0 :=$ nonnegative initial guess;
$\mathbf{g}_0 := K^T(K\mathbf{f}_0-\mathbf{d});$ *% initial gradient*
$\gamma := \langle\mathbf{g}_0, \mathbf{f}_0.*\mathbf{g}_0\rangle;$

```
begin MRNSD iterations
    p_ν := −f_ν. * g_ν;        % descent direction
    u := K p_ν;
    τ_bndry := min{−[f_ν]_i/[p_ν]_i | [p_ν]_i < 0};
    τ_ν = min(γ/||u||^2, τ_bndry);        % line search parameter
    f_{ν+1} := f_ν + τ_ν p_ν;        update solution
    g_{ν+1} := g_ν + τ_ν K^T u;
    γ := ⟨g_{ν+1}, f_{ν+1}. * g_{ν+1}⟩;
    ν := ν + 1;
end MRNSD iterations
```

Remark 9.23. In comparing the results from section 9.5 with those from earlier sections of this chapter, an obvious question arises: Which regularization approach, variational or iterative, is better? The answer depends on several factors:

1. In terms of computational cost, if a good a priori value of the regularization parameter is not available, one may have to solve several variational problems and then select the best one using an a posteriori regularization parameter selection technique from Chapter 7. Iterative regularization techniques are then likely to be much cheaper, assuming the cost of determining when to stop the iteration is relatively low.

2. Variational regularization techniques are more flexible. They allow incorporation of prior information about the solution and constraints in a straightforward manner. One can also incorporate statistical information about noise in the data via a fit-to-data functional of likelihood form.

3. When the cost functional and the constraints are convex, the solution to a variational regularization problem is independent of the initial guess. This is not the case with iterative regularization schemes and can be a significant shortcoming. For example, for both the R-L iteration (9.42) and the MRNSD iteration (9.52), if a component f_i is initially set to zero, it will remain zero for all subsequent iterations.

Exercises

9.1. Verify that the gradient and Hessian of J_{lhd} in (9.5) are given by (9.6) and (9.7)–(9.8), respectively.

9.2. Prove Theorem 9.7.

9.3. Prove Proposition 9.8 using Theorem 9.4.

9.4. Prove Proposition 9.8 using Theorem 2.38.

9.5. Confirm the first statement in Remark 9.10. Provide a counterexample to show that a critical point for (9.16) need not be a minimizer.

9.6. Prove the second statement in Remark 9.10. Show that if J is strictly convex and $\mathbf{f}^*$ is a critical point, then $\mathbf{f}^*$ is the unique global constrained minimizer for (9.16).

9.7. Show that (9.23) implies (9.24).

9.8. Prove Proposition 9.14.

9.9. Prove Proposition 9.15.

9.10. Implement the gradient projection Algorithm 9.3.1 to minimize the regularized least squares functional (9.2) using the test data and parameters discussed in section 9.4.1.

9.11. Replace the gradient projection algorithm in Exercise 9.10 with a projected Newton algorithm. Then apply projected Newton with the least squares functional (9.2) replaced with the Poisson likelihood functional (9.5).

9.12. Apply the GPCG algorithm to minimize the Poisson likelihood functional (9.5) for the two-dimensional test problem of section 9.4.2.

9.13. Show that if $K\mathbf{f} = \mathbf{d}$, then $\mathbf{f}$ minimizes $J(\mathbf{f}) = -\ell(\mathbf{f}; \mathbf{d})$ in (9.44).

9.14. Show that if $\mathbf{f} \geq 0$ maximizes (9.41) subject to the equality constraint (9.43), then $\mathbf{f}$ is a critical point for $J(\mathbf{f}) = -\ell(\mathbf{f}; \mathbf{d})$ in (9.44). Since $J(\mathbf{f})$ is convex, this critical point is a minimizer for J. *Hint:* Show that at a constrained minimizer,

$$\sum_{i=1}^{m} d_i \frac{t_{ij}}{[K\mathbf{f}]_i} - \lambda \sum_{i=1}^{m} k_{ij} = 0, \qquad j = 1, \ldots, n. \tag{9.56}$$

Multiply by f_j, sum over j, and use the constraint (9.43) to show that the Lagrange multiplier $\lambda = 1$. But then (9.56) is the negative gradient of $J(\mathbf{f})$.

9.15. Apply the R-L iteration to the one-dimensional test problem of section 9.4.1.

9.16. Verify that $\langle \mathbf{x}, \mathbf{u}. * \mathbf{v} \rangle = \langle \mathbf{u}. * \mathbf{x}, \mathbf{v} \rangle$, where $\langle \cdot, \cdot \rangle$ denotes the Euclidean inner product.

9.17. Verify the equality in (9.55).

9.18. Derive the R-L iteration (9.42) using ideas from section 9.5.2. *Hint:* Rewrite the negative of the Poisson log likelihood function (9.44) as $J(\mathbf{f}) = \langle K\mathbf{f}, \mathbf{1} \rangle - \langle \mathbf{d}, \log(K\mathbf{f}) \rangle$.

9.19. Apply the MRNSD algorithm to the two-dimensional test problem of section 9.4.2.

This page intentionally left blank

Bibliography

[1] M. Abramowitz and I. Stegun, eds., *Handbook of Mathematical Functions*, Dover, New York, 1965.

[2] R. Acar and C. R. Vogel, *Analysis of total variation penalty methods*, Inverse Problems, **10** (1994), pp. 1217–1229.

[3] K. E. Atkinson, *An Introduction to Numerical Analysis*, 2nd Edition, Wiley, New York, 1989.

[4] G. Aubert, M. Barlaud, L. Blanc-Feraud, and P. Charbonnier, *Deterministic edge-preserving regularization in computed imaging*, Tech. report 94-01, Informatique, Signaux et Systemes de Sophia Antipolis, France, 1994.

[5] O. Axelsson and V. A. Barker, *Finite Element Solution of Boundary Value Problems*, Academic Press, New York, 1984.

[6] H. T. Banks and K. Kunisch, *Estimation Techniques for Distributed Parameter Systems*, Birkhäuser Boston, Boston, 1989.

[7] M. Bertero and P. Boccacci, *Introduction to Inverse Problems in Imaging*, IOP Publishing, Bristol, UK, 1998.

[8] D. Bertsekas, *Projected Newton methods for optimization problems with simple constraints*, SIAM Journal on Control and Optimization, **20** (1982), pp. 221–246.

[9] A. Bjork, *Numerical Methods for Least Squares Problems*, SIAM, Philadelphia, 1996.

[10] W. E. Boyce and R. C. DiPrima, *Elementary Differential Equations and Boundary Value Problems*, 7th Edition, Wiley, New York, 2000.

[11] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu, *A limited memory algorithm for bound constrained optimization*, SIAM Journal on Scientific Computing, **16** (1995), pp. 1190–1208.

[12] G. Casella and R. L. Berger, *Statistical Inference*, Brooks/Cole, Pacific Grove, CA, 1990.

[13] A. Chambolle and P.-L. Lions, *Image recovery via total variation minimization and related problems*, Numerische Mathematik, **76** (1997), pp. 167–188.

[14] R. H. Chan and M. K. Ng, *Conjugate gradient method for Toeplitz systems*, SIAM Review, **38** (1996), pp. 427–482.

[15] R. H. Chan, T. F. Chan, and C. K. Wong, *Cosine transform based preconditioners for total variation deblurring*, IEEE Transactions on Image Processing, **8** (1999), pp. 1472–1478.

[16] R. H. Chan, M. K. Ng, and C. Wong, *Sine transform based preconditioners for symmetric Toeplitz systems*, Linear Algebra and Its Applications, **232** (1996), pp. 237–260.

[17] R. H. Chan, J. G. Nagy, and R. J. Plemmons, *FFT-based preconditioners for Toeplitz-block least squares problems*, SIAM Journal on Numerical Analysis, **30** (1993), pp. 1740–1768.

[18] R. H. Chan, M. K. Ng, and R. J. Plemmons, *Generalization of Strang's preconditioner with applications to Toeplitz least squares problems*, Numerical Linear Algebra and Applications, **3** (1996), pp. 45–64.

[19] T. Chan and P. Mulet, *On the convergence of the lagged diffusivity fixed point method in total variation image restoration*, SIAM Journal on Numerical Analysis, **36** (1999), pp. 354–367.

[20] T. Chan and J. Olkin, *Circulant preconditioners for Toeplitz-block matrices*, Numerical Algorithms, **6** (1994), pp. 89–101.

[21] T. F. Chan, G. H. Golub, and P. Mulet, *A nonlinear primal-dual method for TV-based image restoration*, SIAM Journal on Scientific Computing, **20** (1999), pp. 1964–1977.

[22] G. Chavent and P. Lemonnier, *Identification de la Non-Linéarité D'Une Équation Parabolique Quasilinéare*, Applied Mathematics and Optimization, **1** (1974), pp. 121–162.

[23] T. F. Coleman and Y. Li, *An interior trust region approach for nonlinear minimization subject to bounds*, SIAM Journal on Optimization, **6** (1996), pp. 418–445.

[24] A. R. Conn, N. I. M. Gould, and P. L. Toint, *Global convergence of a class of trust region algorithms for optimization with simple bounds*, SIAM Journal on Numerical Analysis, **25** (1988), pp. 433–460.

[25] A. R. Conn, N. I. M. Gould, and P. L. Toint, *Trust Region Methods*, SIAM, Philadelphia, 2000.

[26] J. W. Cooley and J. W. Tukey, *An algorithm for the machine calculation of complex Fourier series*, Mathematics of Computation, **19** (1965), pp. 297–301.

[27] J. W. Daniel, *The conjugate gradient method for linear and nonlinear operator equations*, SIAM Journal on Numerical Analysis, **4** (1967), pp. 10–26.

[28] A. R. Davies and R. S. Anderssen, *Optimization in the regularization of ill-posed problems*, Journal of the Australian Mathematical Society Series B, **28** (1986), pp. 114–133.

[29] P. J. Davis, *Circulant Matrices*, Wiley, New York, 1979.

[30] K. Deimling, *Nonlinear Functional Analysis*, Springer-Verlag, Berlin, 1985.

[31] R. S. Dembo, S. C. Eisenstat, and T. Steihaug, *Inexact Newton methods*, SIAM Journal on Numerical Analysis, **16** (1982), pp. 400–408.

[32] J. E. Dennis and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, SIAM, Philadelphia, 1996.

[33] D. Dobson and F. Santosa, *An image enhancement technique for electrical impedance tomography*, Inverse Problems, **10** (1994), pp. 317–334.

[34] D. Dobson and F. Santosa, *Recovery of blocky images from noisy and blurred data*, SIAM Journal on Applied Mathematics, **56** (1996), pp. 1181–1198.

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

[36] R. Fletcher, *Practical Methods of Optimization*, 2nd Edition, Wiley, New York, 1987.

[37] R. Fletcher and C. M. Reeves, *Function minimization by conjugate gradients*, Computer Journal, **7** (1964), pp. 149–154.

[38] J. N. Franklin, *Well-posed stochastic extensions of ill-posed linear problems*, Journal of Mathematical Analysis and Applications, **31** (1970), pp. 682–716.

[39] A. Friedlander and J. M. Martinez, *On the maximization of a concave quadratic function with box constraints*, SIAM Journal on Optimization, **4** (1994), pp. 177–192.

[40] A. Friedlander, J. M. Martinez, and S. A. Santos, *A new trust region algorithm for bound constrained minimization*, Applied Mathematics and Optimization, **30** (1994), pp. 235–266.

[41] D. Geman and C. Yang, *Nonlinear image recovery with half-quadratic regularization*, IEEE Transactions on Image Processing, **4** (1995), pp. 932–945.

[42] H. Gfrerer, *An a posteriori parameter choice for ordinary and iterated Tikhonov regularization of ill-posed problems leading to optimal convergence rates*, Mathematics of Computation, **49** (1987), pp. 507–522.

[43] D. Gilliam, J. Lund, and C. R. Vogel, *Quantifying information content for ill-posed problems*, Inverse Problems, **6** (1990), pp. 205–217.

[44] D. A. Girard, *The fast Monte-Carlo cross-validation and C_L procedures: Comments, new results, and application to image recovery problems*, Computational Statistics, **10** (1995), pp. 205–231.

[45] E. Giusti, *Minimal Surfaces and Functions of Bounded Variation*, Birkhäuser Boston, Boston, 1984.

[46] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd Edition, Johns Hopkins University Press, Baltimore, 1996.

[47] G. H. Golub and U. von Matt, *Generalized cross-validation for large-scale problems*, Journal of Computational and Graphical Statistics, **6** (1997), pp. 1–34.

[48] C. W. Groetsch, *The Theory of Tikhonov Regularization for Fredholm Equations of the First Kind*, Pitman, Boston, 1984.

[49] C. W. Groetsch, *Inverse Problems in the Mathematical Sciences*, Vieweg, Braunschweig, 1993.

[50] M. Hanke, *Conjugate Gradient Type Methods for Ill-Posed Problems*, Longman Scientific & Technical, Essex, UK, 1995.

[51] M. Hanke, *Limitations of the L-curve method in ill-posed problems*, BIT, **36** (1996), pp. 287–301.

[52] M. Hanke and P. C. Hansen, *Regularization methods for large-scale problems*, Surveys on Mathematics for Industry, **3** (1993), pp. 253–315.

[53] M. Hanke and J. G. Nagy, *Toeplitz approximate inverse preconditioner for banded Toeplitz matrices*, Numerical Algorithms, **7** (1994), pp. 183–199.

[54] M. Hanke, J. Nagy, and C. R. Vogel, *Quasi-Newton approach to nonnegative image restoration*, Linear Algebra and Its Applications, **316** (2000), pp. 223–236.

[55] M. Hanke and C. R. Vogel, *Two-level preconditioners for regularized inverse problems* I: *Theory*, Numerische Mathematik, **83** (1999), pp. 385–402.

[56] P. C. Hansen, *Numerical tools for analysis and solution of Fredholm integral equations of the first kind*, Inverse Problems, **8** (1992), pp. 956–972.

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

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

[59] P. C. Hansen, *Regularization tools version* 3.0 *for Matlab* 5.2, Numerical Algorithms, **20** (1999), pp. 195–196.

[60] P. C. Hansen and D. P. O'Leary, *The use of the L-curve in the regularization of discrete ill-posed problems*, SIAM Journal on Scientific Computing, **14** (1993), pp. 1487–1503.

[61] M. F. Hutchinson, *A stochastic estimator of the trace of the influence matrix for Laplacian smoothing splines*, Communications in Statistics, Simulation, and Computation, **19** (1990), pp. 433–450.

[62] K. Ito and K. Kunisch, *An active set strategy based on the augmented Lagrangian formulation for image reconstruction*, RAIRO, Mathematical Modeling and Numerical Analysis, **33** (1999), pp. 1–21.

[63] M. Jacobsen, *Two-Grid Iterative Methods for Ill-Posed Problems*, MSc. Thesis, Report IMM-EKS-2000-27, Informatics and Mathematical Modelling, Technical University of Denmark, Copenhagen, Denmark, 2000.

[64] A. K. Jain, *Fundamentals of Digital Image Processing*, Prentice-Hall, New York, 1989.

[65] T. Kailath and A. H. Sayed, *Fast Reliable Algorithms for Matrices with Structure*, SIAM, Philadelphia, 1999.

[66] L. Kaufman, *Maximum likelihood, least squares, and penalized least squares for PET*, IEEE Transactions on Medical Imaging, **12** (1993), pp. 200–214.

[67] J. Kay, *Asymptotic comparison factors for smoothing parameter choices in regression problems*, Statistics and Probability Letters, **15** (1992), pp. 329–335.

[68] C. T. Kelley, *Iterative Methods for Optimization*, SIAM, Philadelphia, 1999.

[69] M. E. Kilmer and D. P. O'Leary, *Choosing regularization parameters in iterative methods for ill-posed problems*, SIAM Journal on Matrix Analysis and Applications, **22** (2001), pp. 1204–1221.

[70] A. Kirsch, *An Introduction to the Mathematical Theory of Inverse Problems*, Springer-Verlag, New York, 1996.

[71] E. Kreyszig, *Introduction to Functional Analysis with Applications*, Wiley, New York, 1978.

[72] L. Landweber, *An iteration formula for Fredholm integral equations of the first kind*, American Journal of Mathematics, **73** (1951), pp. 615–624.

[73] A. S. Leonov, *Numerical piecewise-uniform regularization for two-dimensional ill-posed problems*, Inverse Problems, **15** (1999), pp. 1165–1176.

[74] Y. Li and F. Santasa, *A computational algorithm for minimizing total variation in image restoration*, IEEE Transactions on Image Processing, **5** (1996), pp. 987–995.

[75] C. C. Lin and L. A. Segel, *Mathematics Applied to Deterministic Problems in the Natural Sciences*, SIAM, Philadelphia, 1988.

[76] C.-J. Lin and J. J. Moré, *Newton's method for large bound-constrained optimization problems*, SIAM Journal on Optimization, **9** (1999), pp. 1100–1127.

[77] J. Llacer and E. Veklerov, *Feasible images and practical stopping rules for iterative algorithms in emission tomography*, IEEE Transactions on Medical Imaging, **8** (1989), pp. 186–193.

[78] B. Lucy, *An iterative method for the rectification of observed distributions*, Astronomical Journal, **79** (1974), pp. 745–754.

[79] D. G. Luenberger, *Optimization by Vector Space Methods*, Wiley, New York, 1969.

[80] D. G. Luenberger, *Introduction to Linear and Nonlinear Programming*, Addison-Wesley, Reading, MA, 1973.

[81] M. A. Lukas, *Asymptotic behavior of the minimum bound method for choosing the regularization parameter*, Inverse Problems, **14** (1998), pp. 149–159.

[82] M. A. Lukas, *Comparisons of parameter choice methods for regularization with discrete noisy data*, Inverse Problems, **14** (1998), pp. 161–184.

[83] G. J. McLachlan and T. Krishnan, *The EM Algorithm and Extensions*, Wiley, New York, 1997.

[84] C. L. Mallows, *Some comments on C_P*, Technometrics, **15** (1973), pp. 661–675.

[85] J. J. Moré and G. Toraldo, *On the solution of large quadratic programming problems with bound constraints*, SIAM Journal on Optimization, **1** (1991), pp. 93–113.

[86] V. A. Morozov, *On the solution of functional equations by the method of regularization*, Soviet Mathematics Doklady, **7** (1966), pp. 414–417.

[87] V. A. Morozov, *Regularization Methods for Ill-Posed Problems*, CRC Press, Boca Raton, FL, 1993.

[88] J. Nagy and Z. Strakos, *Enforcing nonnegativity in image reconstruction algorithms*, in Proceedings of the SPIE International Conference on Mathematical Modeling, Estimation, and Imaging **4121**, David C. Wilson, et. al., eds., SPIE, Bellingham, WA, 2000, pp. 182–190.

[89] S. G. Nash and A. Sofer, *Linear and Nonlinear Programming*, McGraw-Hill, New York, 1996.

[90] F. Natterer, *The Mathematics of Computerized Tomography*, Wiley, New York, 1986.

[91] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer-Verlag, New York, 1999.

[92] F. O'Sullivan and G. Wahba, *A cross validated Bayesian retrieval algorithm for nonlinear remote sensing*, Journal of Computational Physics, **59** (1985), pp. 441–455.

[93] D. L. Phillips, *A technique for the numerical solution of certain integral equations of the first kind*, Journal of the Association for Computing Machinery, **9** (1962), pp. 84–97.

[94] E. Polak, *Optimization. Algorithms and Consistent Approximations*, Springer-Verlag, New York, 1997.

[95] T. Raus, *On the discrepancy principle for the solution of ill-posed problems*, Acta et Commentationes Universitatis Tartuensis de Mathematica, **672** (1984), pp. 16–26 (in Russian).

[96] W. H. Richardson, *Bayesian-based iterative methods for image restoration*, Journal of the Optical Society of America, **62** (1972), pp. 55–59.

[97] A. Rieder, *A wavelet multilevel method for ill-posed problems stabilized by Tikhonov regularization*, Numerische Mathematik, **75** (1997), pp. 501–522.

[98] K. Riley and C. R. Vogel, *Preconditioners for linear systems arising in image reconstruction*, in Advanced Signal Processing Algorithms, Architectures, and Implementations VIII, Proceedings of SPIE **3461–37**, 1998, pp. 372–380.

[99] M. C. Roggemann and B. Welsh, *Imaging Through Turbulence*, CRC Press, Boca Raton, FL, 1996.

[100] R. L. Royden, *Real Analysis,* 2nd Edition, MacMillan, New York, 1968.

[101] L. I. Rudin, S. Osher, and E. Fatemi, *Nonlinear total variation based noise removal algorithms*, Physica D, **60** (1992), pp. 259–268.

[102] Y. Saad, *Iterative Methods for Sparse Linear Systems*, PWS Publishing, Boston, 1996.

[103] A. Spence, *Error bounds and estimates for eigenvalues of integral equations*, Numerische Mathematik, **29** (1978), pp. 133–147.

[104] T. Steihaug, *The conjugate gradient method and trust regions in large scale optimization*, SIAM Journal on Numerical Analysis, **20** (1983), pp. 626–637.

[105] G. Strang, *A proposal for Toeplitz matrix calculations*, Studies in Applied Mathematics, **74** (1986), pp. 171–176.

[106] A. N. Tikhonov, *Regularization of incorrectly posed problems*, Soviet Mathematics Doklady, **4** (1963), pp. 1624–1627.

[107] A. N. Tikhonov and V. Arsenin, *Solutions of Ill-Posed Problems*, Wiley, New York, 1977.

[108] A. N. Tikhonov, A. S. Leonov, and A. G. Yagola, *Nonlinear Ill-Posed Problems, Volumes* I *and* II, Chapman and Hall, London, 1998.

[109] A. M. Thompson, J. C. Brown, J. W. Kay, and D. M. Titterington, *A comparison of methods of choosing the smoothing parameter in image restoration by regularization*, IEEE Transactions on Pattern Analysis and Machine Intelligence, **13** (1991), pp. 326–339.

[110] C. F. Van Loan, *Computational Frameworks for the Fast Fourier Transform*, SIAM, Philadelphia, 1992.

[111] Y. Vardi and D. Lee, *From image deblurring to optimal investments: Maximum likelihood solutions for positive linear inverse problems*, Journal of the Royal Statistical Society B, **55** (1993), pp. 569–612.

[112] E. Veklerov and J. Llacer, *Stopping rule for the MLE algorithm based on statistical hypothesis testing*, IEEE Transactions on Medical Imaging, **6** (1987), pp. 313–319.

[113] C. R. Vogel, *Optimal choice of a truncation level for the truncated SVD solution of linear first kind integral equations when the data are noisy*, SIAM Journal on Numerical Analysis, **23** (1986), pp. 109–117.

[114] C. R. Vogel, *Non-convergence of the L-curve regularization parameter selection method*, Inverse Problems, **12** (1996), pp. 535–547.

[115] C. R. Vogel, *Sparse matrix equations arising in distributed parameter identification*, SIAM Journal on Matrix Analysis, **20** (1999), pp. 1027–1037.

[116] C. R. Vogel, *A limited memory BFGS method for an inverse problem in atmospheric imaging*, in Methods and Applications of Inversion, P.C. Hansen, B.H. Jacobsen, and K. Mosegaard, eds., Lecture Notes in Earth Sciences **92**, 2000, Springer-Verlag, New York, pp. 292–304.

[117] C. R. Vogel, *Negative results for multilevel preconditioners in image deblurring*, in Scale-Space Theories in Computer Vision, M. Nielsen, P. Johansen, O.F. Olsen, and J. Weickert, eds., Springer-Verlag, New York, 1999.

[118] C. R. Vogel and M. Hanke, *Two-Level Preconditioners for Regularized Inverse Problems* II: *Implementation and Numerical Results*, technical report, Department of Mathematical Sciences, Montana State University, Bozeman, MT, 1999.

[119] C. R. Vogel and M. E. Oman, *Iterative methods for total variation denoising*, SIAM Journal on Scientific Computing, **17** (1996), pp. 227–238.

[120] C. R. Vogel and M. E. Oman, *A fast, robust algorithm for total variation based reconstruction of noisy, blurred images*, IEEE Transactions on Image Processing, **7** (1998), pp. 813–824.

[121] C. R. Vogel and J. G. Wade, *Analysis of costate discretizations in parameter identification for linear evolution equations*, SIAM Journal on Control and Optimization, **33** (1995), pp. 227–254.

[122] G. Wahba, *Practical approximate solutions to linear operator equations when the data are noisy*, SIAM Journal on Numerical Analysis, **14** (1977), pp. 651–667.

[123] G. Wahba, *Spline Models for Observational Data*, SIAM, Philadelphia, 1990.

[124] J. Weickert, *Anisotropic Diffusion in Image Processing*, Teubner, Stuttgart, 1998.

[125] E. Zeidler, *Nonlinear Functional Analysis and Its Applications.* I. *Fixed-Point Theorems*, Springer-Verlag, New York, 1986.

[126] E. Zeidler, *Nonlinear Functional Analysis and Its Applications.* II. *Variational Methods and Optimization*, Springer-Verlag, New York, 1985.

[127] E. Zeidler, *Applied Functional Analysis: Applications to Mathematical Physics*, Springer-Verlag, New York, 1995.

[128] E. Zeidler, *Applied Functional Analysis: Main Principles and their Applications*, Springer-Verlag, New York, 1995.

Index