Parameter Estimation and Inverse Problems

Second Editon

Solution Semimorm $||Lm||_2$

Residual Norm $||G(m)-d||_2$

Richard C. Aster
Brian Borchers
Clifford H. Thurber

PARAMETER ESTIMATION AND INVERSE PROBLEMS

Second Edition

This page intentionally left blank

PARAMETER ESTIMATION AND INVERSE PROBLEMS

Second Edition

RICHARD C. ASTER

BRIAN BORCHERS

CLIFFORD H. THURBER

Notices

Knowledge and best practice in this field are constantly changing. As new research and experience broaden our understanding, changes in research methods, professional practices, or medical treatment may become necessary.

Practitioners and researchers must always rely on their own experience and knowledge in evaluating and using any information, methods, compounds, or experiments described herein. In using such information or methods they should be mindful of their own safety and the safety of others, including parties for whom they have a professional responsibility.

To the fullest extent of the law, neither the Publisher nor the authors, contributors, or editors, assume any liability for any injury and/or damage to persons or property as a matter of products liability, negligence or otherwise, or from any use or operation of any methods, products, instructions, or ideas contained in the material herein.

> For information on all Academic Press publications
> visit our website at *www.elsevierdirect.com*

CONTENTS

PREFACE

This textbook evolved from a course in geophysical inverse methods taught during the past two decades at New Mexico Tech, first by Rick Aster and, subsequently, jointly between Rick Aster and Brian Borchers. The audience for the course has included a broad range of first- or second-year graduate students (and occasionally advanced undergraduates) from geophysics, hydrology, mathematics, astrophysics, and other disciplines. Cliff Thurber joined this collaboration during the production of the first edition and has taught a similar course at the University of Wisconsin-Madison.

Our principal goal for this text is to promote fundamental understanding of parameter estimation and inverse problem philosophy and methodology, specifically regarding such key issues as uncertainty, ill-posedness, regularization, bias, and resolution. We emphasize theoretical points with illustrative examples, and MATLAB codes that implement these examples are provided on a companion website. Throughout the examples and exercises, a web icon indicates that there is additional material on the website. Exercises include a mix of applied and theoretical problems.

This book has necessarily had to distill a tremendous body of mathematics and science going back to (at least) Newton and Gauss. We hope that it will continue to find a broad audience of students and professionals interested in the general problem of estimating physical models from data. Because this is an introductory text surveying a very broad field, we have not been able to go into great depth. However, each chapter has a "notes and further reading" section to help guide the reader to further exploration of specific topics. Where appropriate, we have also directly referenced research contributions to the field.

Some advanced topics have been deliberately left out of this book because of space limitations and/or because we expect that many readers would not be sufficiently familiar with the required mathematics. For example, readers with a strong mathematical background may be surprised that we primarily consider inverse problems with discrete data and discretized models. By doing this we avoid much of the technical complexity of functional analysis. Some advanced applications and topics that we have omitted include inverse scattering problems, seismic diffraction tomography, wavelets, data assimilation, simulated annealing, and expectation maximization methods.

We expect that readers of this book will have prior familiarity with calculus, differential equations, linear algebra, probability, and statistics at the undergraduate level. In our experience, many students can benefit from at least a review of these topics, and we commonly spend the first two to three weeks of the course reviewing material from Appendices A, B, and C.

Chapters 1 through 4 form the heart of the book, and should be covered in sequence. Chapters 5 through 8 are mostly independent of each other, but draw heavily on the material in Chapters 1 through 4. As such, they may be covered in any order. Chapters 9 and 10 are independent of Chapters 5 through 8, but are most appropriately covered in sequence. Chapter 11 is independent of the material in Chapters 5 through 10, and provides an introduction to the Bayesian perspective on inverse problems and Bayesian solution methods.

If significant time is allotted for review of linear algebra, vector calculus, probability, and statistics in the appendices, there will probably not be time to cover the entire book in one semester. However, it should be possible for teachers to cover substantial material following Chapter 4.

Rick Aster, Brian Borchers, and Cliff Thurber
June 2011

Introduction

Synopsis

General issues associated with parameter estimation and inverse problems are introduced through the concepts of the forward problem and its inverse solution. Scaling and superposition properties that characterize linear systems are given, and common situations leading to linear and nonlinear mathematical models are discussed. Examples of discrete and continuous linear and nonlinear parameter estimation problems to be revisited in later chapters are shown. Mathematical demonstrations highlighting the key issues of solution existence, uniqueness, and instability are presented and discussed.

1.1. CLASSIFICATION OF PARAMETER ESTIMATION AND INVERSE PROBLEMS

Scientists and engineers frequently wish to relate physical parameters characterizing a **model**, m, to collected observations making up some set of **data**, d. We will commonly assume that the fundamental physics are adequately understood, so a function, G, may be specified relating m and d such that

$$G(m) = d. \tag{1.1}$$

In practice, d may be a function of time and/or space, or may be a collection of discrete observations. An important issue is that actual observations always contain some amount of noise. Two common ways that noise may arise are unmodeled influences on instrument readings and numerical round-off. We can thus envision data as generally consisting of noiseless observations from a "perfect" experiment, d_{true}, plus a noise component η,

$$d = G(m_{\text{true}}) + \eta \tag{1.2}$$

$$= d_{\text{true}} + \eta, \tag{1.3}$$

where d_{true} exactly satisfies (1.1) for m equal to the true model, m_{true}, and we assume that the forward modeling is exact. We will see that it is commonly mathematically possible, although practically undesirable, to also fit all or part of η by (1.1). It may seem remarkable, but it is often the case that a solution for m that is influenced by even a small

noise amplitude η can have little or no correspondence to m_{true}. Another key issue that may seem astounding at first is that commonly there are an infinite number of models aside from m_{true} which fit the perfect data, d_{true}.

When m and d are functions, we typically refer to G as an **operator**. G will be called a function when m and d are vectors. The operator G can take on many forms. In some cases G is an ordinary differential equation (ODE) or partial differential equation (PDE). In other cases, G is a linear or nonlinear system of algebraic equations.

Note that there is some inconsistency between mathematicians and other scientists in modeling terminology. Applied mathematicians usually refer to $G(m) = d$ as the "mathematical model" and to m as the "parameters." On the other hand, scientists often refer to G as the "forward operator" and to m as the "model." We will adopt the scientific parlance and refer to m as the "the model" while referring to the equation $G(m) = d$ as the "mathematical model."

The **forward problem** is to find d given m. Computing $G(m)$ might involve solving an ODE or PDE, evaluating an integral, or applying an algorithm for which there is no explicit analytical formula for $G(m)$. Our focus in this text is on the **inverse problem** of finding m given d. A third problem, not addressed here, is the **model identification problem** of determining G given examples of m and d.

In many cases, we will want to determine a finite number of parameters, n, to define a model. The parameters may define a physical entity directly (e.g., density, voltage, seismic velocity), or may be coefficients or other constants in a functional relationship that describes a physical process. In this case, we can express the model parameters as an n element vector $\mathbf{m}$. Similarly, if there are a finite number of data points then we can express the data as an m element vector $\mathbf{d}$. (Note that the use of the integer m here for the number of data points is easily distinguishable from the model m by its context.) Such problems are called **discrete inverse problems** or **parameter estimation problems**. A general parameter estimation problem can be written as a system of equations

$$G(\mathbf{m}) = \mathbf{d}. \tag{1.4}$$

In other cases, where the model and data are functions of continuous variables, such as time or space, the associated task of estimating m from d is called a **continuous inverse problem**. A central theme of this book is that continuous inverse problems can often be well–approximated by discrete inverse problems.

We will generally refer to problems with small numbers of parameters as "parameter estimation problems." Problems with a larger number of parameters, and which will often require the application of stabilizing constraints, will be referred to as "inverse problems." A key aspect of many inverse problems is that they are ill–conditioned in a sense that will be discussed later in this chapter. In both parameter estimation and inverse problems we solve for a set of parameters that characterize a model, and a key point of this text is that the treatment of all such problems can be sufficiently generalized so

that the distinction is largely irrelevant. In practice, what is important is the distinction between ill–conditioned and well–conditioned parameter estimation problems.

A type of mathematical model for which many useful results exist is the class of **linear systems.** Linear systems obey superposition

$$G(m_1 + m_2) = G(m_1) + G(m_2) \tag{1.5}$$

and scaling

$$G(\alpha m) = \alpha G(m). \tag{1.6}$$

In the case of a discrete linear inverse problem, (1.4) can always be written in the form of a linear system of algebraic equations (see Exercise 1.1).

$$G(\mathbf{m}) = \mathbf{Gm} = \mathbf{d}. \tag{1.7}$$

In a continuous linear inverse problem, G can often be expressed as a linear integral operator, where (1.1) has the form

$$\int_a^b g(x, \xi) m(\xi) d\xi = d(x) \tag{1.8}$$

and the function $g(x, \xi)$ is called the **kernel**. The linearity of (1.8) is easily seen because

$$\int_a^b g(x, \xi)(m_1(\xi) + m_2(\xi)) d\xi = \int_a^b g(x, \xi) m_1(\xi) d\xi + \int_a^b g(x, \xi) m_2(x) d\xi \tag{1.9}$$

and

$$\int_a^b g(x, \xi) \alpha m(\xi) d\xi = \alpha \int_a^b g(x, \xi) m(\xi) d\xi. \tag{1.10}$$

Equations in the form of (1.8), where $m(x)$ is the unknown, are called **Fredholm integral equations of the first kind (IFK)**. IFKs arise in a surprisingly large number of inverse problems. A key characteristic of these equations is that they have mathematical properties which make it difficult to obtain useful solutions by straightforward methods.

In many cases the kernel in (1.8) can be written to depend explicitly on $x - \xi$, producing a **convolution equation**,

$$\int_{-\infty}^{\infty} g(x - \xi) m(\xi) d\xi = d(x). \tag{1.11}$$

Here we have written the interval of integration as extending from minus infinity to plus infinity, but other intervals can easily be accommodated by having $g(x - \xi) = 0$ outside of the interval of interest. When a forward problem has the form of (1.11), determining $d(x)$ from $m(x)$ is called **convolution**, and the inverse problem of determining $m(x)$ from $d(x)$ is called **deconvolution**.

Another IFK arises in the problem of inverting a **Fourier transform**

$$\Phi(f) = \int\limits_{-\infty}^{\infty} e^{-\iota 2\pi f x} \phi(x) dx \qquad (1.12)$$

(where ι is $\sqrt{-1}$) to obtain $\phi(x)$. Although there are many tables and analytic methods of obtaining Fourier transforms and their inverses, numerical estimates of $\phi(x)$ may be of interest, such as where there is no analytic inverse, or where we wish to estimate $\phi(x)$ from spectral data collected at discrete frequencies.

It is an intriguing question as to why linearity appears in many interesting physical problems. A principal reason is that many such systems are associated with only small departures from equilibrium. One important geophysical example is seismic wave propagation, where the stresses associated with elastic fields are often very small relative to the elastic moduli of the medium. This situation leads to small strains and to a very nearly linear stress–strain relationship. Because of this, seismic wave field problems in many useful circumstances obey superposition and scaling. Other fields such as gravity and magnetism, at field strengths typically encountered in geophysics, also show effectively linear physics.

Because many important inverse problems are linear, and because linear theory is a key component in solving nonlinear problems, Chapters 2 through 8 of this book cover theory and methods for the solution of linear inverse problems. Nonlinear mathematical models arise when the parameters of interest have an inherently nonlinear relationship to the observables. This situation commonly occurs, for example, in electromagnetic field problems where we wish to relate geometric model parameters such as layer thicknesses to observed field properties. We discuss methods for nonlinear parameter estimation and inverse problems in Chapters 9 and 10, respectively.

1.2. EXAMPLES OF PARAMETER ESTIMATION PROBLEMS

Example 1.1

A canonical parameter estimation problem is the fitting of a function, defined by a collection of parameters, to a data set. In cases where this function fitting procedure can be cast as a linear inverse problem, the procedure is referred to as **linear regression**. An ancient example of linear regression is the characterization of a ballistic trajectory. In a basic take on this problem, the data, $\mathbf{y}$, are altitude observations of a ballistic body at a set of times $\mathbf{t}$
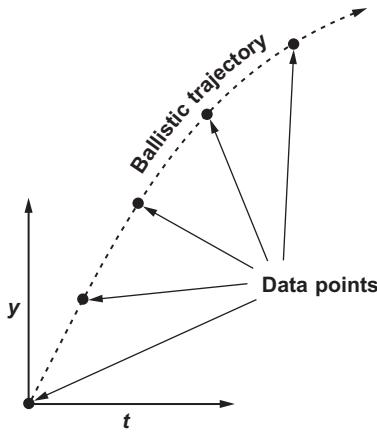
Figure 1.1 The parabolic trajectory problem.

(Figure 1.1). We wish to solve for a model, **m**, that contains the initial altitude (m_1), initial vertical velocity (m_2), and effective gravitational acceleration (m_3) experienced by the body during its trajectory. This and related problems are naturally of practical interest in aeronautics and weaponry, but are also of fundamental geophysical interest, for example, in absolute gravity meters capable of estimating g from the acceleration of a falling object in a vacuum to accuracies on the order of one part in 10^9 [91].

The mathematical model is a quadratic function in the (t, y) plane

$$y(t) = m_1 + m_2 t - (1/2)m_3 t^2 \tag{1.13}$$

that we expect to apply at all times along the trajectory, not just at the times t_i when we happen to have observations. The data will consist of m observations of the height of the body y_i at corresponding times t_i. Assuming that the t_i are measured precisely, and applying (1.13) to each observation, we obtain a system of equations with m rows and $n = 3$ columns that relates the data y_i to the model parameters, m_j

$$\begin{bmatrix} 1 & t_1 & -\frac{1}{2}t_1^2 \\ 1 & t_2 & -\frac{1}{2}t_2^2 \\ 1 & t_3 & -\frac{1}{2}t_3^2 \\ \vdots & \vdots & \vdots \\ 1 & t_m & -\frac{1}{2}t_m^2 \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ m_3 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_m \end{bmatrix}. \tag{1.14}$$

Although the mathematical model of (1.13) is quadratic, the equations for the three parameters m_i in (1.14) are linear, so solving for $\mathbf{m} = [m_1, m_2, m_3]^T$ is a linear parameter estimation problem.

If there are more data points than model parameters in (1.14) ($m > n$), then the m constraint equations in (1.14) will likely be inconsistent, and it will be impossible to find

a model **m** that satisfies every equation exactly. The nonexistence of a model exactly satisfying the observations in such a case can be seen geometrically because no parabola will exist that goes through all of the (t_i, y_i) points (Exercise 1.2). Such a situation could arise in practice because of noise in the determinations of the y_i, and/or because the forward model of (1.13) is approximate. For example, we have neglected the physics of atmospheric drag, so a true trajectory will not be exactly parabolic and thus exactly modeled by (1.13).

In elementary linear algebra, where an exact solution is typically expected for a system of linear equations, we might throw up our hands at this point and simply state that no solution exists. However, useful solutions to such systems may be found in practice by solving for model parameters that satisfy the data in an approximate, or "best-fit," sense.

A reasonable approach to finding the "best" approximate solution to an inconsistent system of linear equations is to find an **m** that minimizes some misfit measure, calculated from the differences between the observations and the theoretical predictions of the forward problem, commonly called **residuals**. A traditional and very widely applied strategy is to find the model that minimizes the **2–norm** (Euclidean length) of the residual vector

$$\|\mathbf{y} - \mathbf{Gm}\|_2 = \sqrt{\sum_{i=1}^{m} (y_i - (\mathbf{Gm})_i)^2}. \tag{1.15}$$

However, (1.15) is not the only, or necessarily the best, misfit measure that can be applied to approximate solve systems of equations. An alternative misfit measure that is superior in many situations is the **1–norm**

$$\|\mathbf{y} - \mathbf{Gm}\|_1 = \sum_{i=1}^{m} |y_i - (\mathbf{Gm})_i|. \tag{1.16}$$

We shall see in Chapter 2 that a solution that minimizes (1.16) is less sensitive to data points that are wildly discordant with the mathematical model than one that minimizes (1.15). Solution techniques that are resistant to such data **outliers** are called **robust estimation procedures**.

⬤──

Example 1.2

A classic nonlinear parameter estimation problem in geophysics is determining the space and time coordinates of an earthquake nucleation, the hypocenter, which is specified by
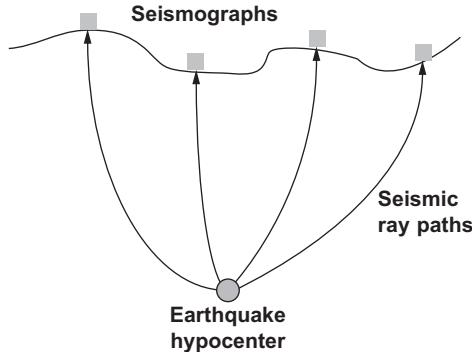
Figure 1.2 The earthquake location problem.

the four-vector

$$\mathbf{m} = \begin{bmatrix} \boldsymbol{\xi} \\ \tau \end{bmatrix} \tag{1.17}$$

where $\boldsymbol{\xi}$ is the three-dimensional earthquake location and the fourth element, τ, is the earthquake origin time (Figure 1.2). The hypocenter model we seek best fits a vector of seismic phase **arrival times**, $\mathbf{t}$, observed at an m-station seismic network. The mathematical model is

$$G(\mathbf{m}) = \mathbf{t} \tag{1.18}$$

where G models the physics of seismic wave propagation to map a hypocenter into a vector of predicted seismic arrival times at m stations. G depends on the seismic velocity structure, $\nu(\mathbf{x})$, which we assume here to be known.

The earthquake location problem is nonlinear even if $\nu(\mathbf{x})$ is a constant, c. In this case, all of the ray paths in Figure 1.2 would be straight, and the arrival time of the seismic signal at station i would be

$$t_i = \frac{\|\mathbf{S}_{\cdot,i} - \boldsymbol{\xi}\|_2}{c} + \tau \tag{1.19}$$

where the *ith* column of the matrix $\mathbf{S}$, $\mathbf{S}_{\cdot,i}$, specifies the coordinates for station i. Equation (1.19) is nonlinear with respect to the spatial parameters ξ_i in $\mathbf{m}$, and thus the problem cannot be expressed as a linear system of equations.

In a few special cases, a change of variables can convert a nonlinear problem to a linear one. More generally, nonlinear parameter estimation problems can often be solved by choosing a starting model and then iteratively improving it until a good solution is obtained. General methods for solving nonlinear parameter estimation problems are discussed in Chapter 9.

1.3. EXAMPLES OF INVERSE PROBLEMS

Example 1.3

In **vertical seismic profiling**, we wish to know the vertical seismic velocity of the material surrounding a borehole. A downward-propagating seismic wavefront is generated at the surface by a source, and seismic waves are sensed by a string of seismometers in the borehole (Figure 1.3).

The arrival times of the seismic wavefront at each instrument, which reflects the seismic velocity for vertically traveling waves as a function of depth, are measured from the recorded seismograms. The problem is nonlinear if it is expressed in terms of seismic velocities. However, we can linearize this problem via a simple change of variables where we parameterize the seismic structure in terms of **slowness**, $s(z)$, the reciprocal of the velocity $v(z)$. The observed travel time at depth z can then be expressed as the definite integral of the vertical slowness, s, from the surface to z:

$$t(z) = \int_0^z s(\xi)\,d\xi \tag{1.20}$$

$$= \int_0^\infty s(\xi)H(z-\xi)\,d\xi \tag{1.21}$$

where the kernel function H is the **Heaviside step function**, which is equal to one when its argument is nonnegative and zero when its argument is negative. The explicit dependence of the kernel on $z - \xi$ shows that (1.21) is a convolution.
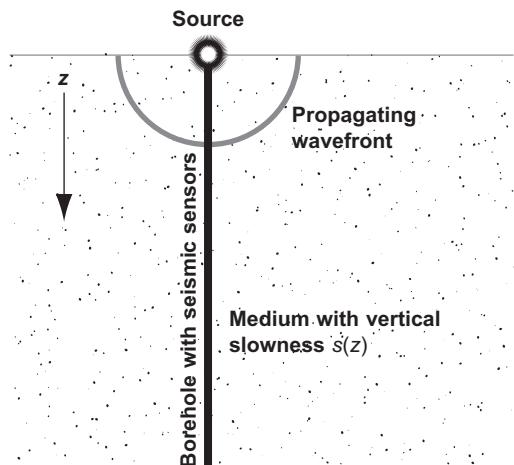
Figure 1.3 The vertical seismic profiling problem.

In theory, we can solve (1.21) quite easily because, by the fundamental theorem of calculus,

$$s(z) = \frac{dt(z)}{dz}. \tag{1.22}$$

In practice, simply differentiating the observations to obtain a solution may not be useful. This is because there will generally be noise present in the observed times $t(z)$, and applying (1.22) may result in a solution that includes unphysical values of $s(z)$ or other unrealistic model features.

Example 1.4

A further instructive linear inverse problem is the inversion of a vertical gravity anomaly, $d(x)$, observed at some height, h, to estimate an unknown buried line mass density distribution deviation from a background model, $m(x) = \Delta\rho(x)$ (Figure 1.4). The mathematical model for this problem can be written as an IFK, because the data are a superposition of the vertical gravity contributions from the differential elements comprising the line mass

$$d(x) = \Gamma \int_{-\infty}^{\infty} \frac{h}{\left((\xi - x)^2 + h^2\right)^{3/2}} m(\xi) d\xi \tag{1.23}$$

$$= \int_{-\infty}^{\infty} g(\xi - x) m(\xi) d\xi \tag{1.24}$$

where Γ is Newton's gravitational constant. Note that the kernel has the form $g(\xi - x)$, and (1.24) is thus a convolution. Because the kernel is a smooth function, $d(x)$ will be a smoothed and scaled transformation of $m(x)$. Conversely, solutions for $m(x)$ will be a
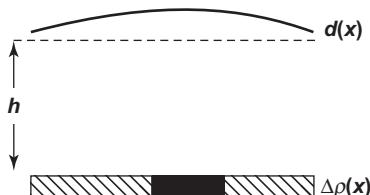
Figure 1.4 A linear inverse problem: determine the density deviation of a buried line mass, $\Delta\rho(x)$, relative to a background model, from gravity anomaly observations $d(x)$.

roughened transformation of $d(x)$. For this reason we again need to be wary of possibly severe deleterious effects of noise in the data.

Example 1.5

Consider a variation on Example 1.4, where the depth of the line density perturbation varies, rather than the density contrast. The gravity anomaly is now attributable to a variation in the burial depth, $m(x) = h(x)$, of an assumed known density perturbation, $\Delta\rho$ (Figure 1.5). The physics is the same as in Example 1.4, so the data are still given by the superposition of density perturbation contributions to the gravitational anomaly field, but the mathematical model now takes the form

$$d(x) = \Gamma \int_{-\infty}^{\infty} \frac{m(\xi)}{\left((\xi - x)^2 + m^2(\xi)\right)^{3/2}} \Delta\rho\, d\xi. \tag{1.25}$$

This problem is nonlinear in $m(x)$ because (1.25) does not follow the superposition and scaling rules (1.5) and (1.6).

Nonlinear inverse problems are generally significantly more difficult to solve than linear ones. In special cases, they may be solvable by coordinate transformations that globally linearize the problem or by other clever special–case methods. In other cases, the problem cannot be globally linearized, so nonlinear optimization techniques must be applied. The essential differences in the treatment of linear and nonlinear problems arise because, as we shall see, all linear problems can be generalized to be the "same" in an important sense, so that a single set of solution methods can be applied to all. In contrast, nonlinear problems tend to be nonlinear in mathematically different ways and
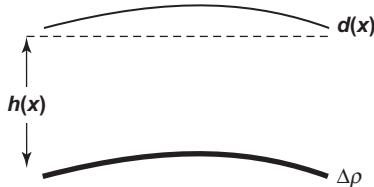
Figure 1.5 A nonlinear inverse problem: determine the depth to a buried line mass density anomaly $h(x)$ from observed gravity anomaly observations $d(x)$.

either require specific strategies or, more commonly, can by solved by iterative methods that may rely on local linearization.

Example 1.6

A classic pedagogical inverse problem is an experiment in which an angular distribution of illumination passes through a thin slit and produces a diffraction pattern, for which the intensity is observed (Figure 1.6; [141]).

The data, $d(s)$, are measurements of diffracted light intensity as a function of the outgoing angle $-\pi/2 \leq s \leq \pi/2$. Our goal is to find the intensity of the incident light on the slit, $m(\theta)$, as a function of the incoming angle $-\pi/2 \leq \theta \leq \pi/2$.

The forward problem relating d and m can be expressed as the linear mathematical model,

$$d(s) = \int_{-\pi/2}^{\pi/2} (\cos(s) + \cos(\theta))^2 \left(\frac{\sin(\pi(\sin(s) + \sin(\theta)))}{\pi(\sin(s) + \sin(\theta))} \right)^2 m(\theta) d\theta. \tag{1.26}$$

Example 1.7

Consider the problem of recovering the history of groundwater pollution at a source site from later measurements of the contamination at downstream wells to which the contaminant plume has been transported by advection and diffusion (Figure 1.7). This
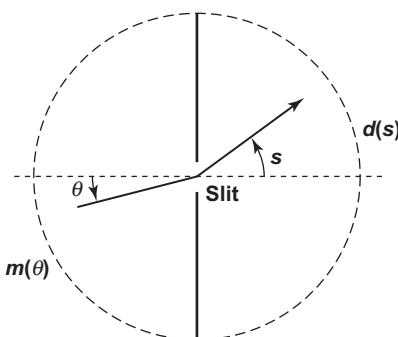
Figure 1.6 The Shaw diffraction intensity problem (1.26).
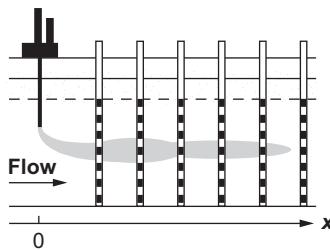
Figure 1.7 The contaminant plume source history reconstruction problem.

"source history reconstruction problem" has been considered by a number of authors [117, 144, 145].

   The mathematical model for contaminant transport is an advection–diffusion equation

$$\frac{\partial C}{\partial t} = D\frac{\partial^2 C}{\partial x^2} - v\frac{\partial C}{\partial x} \tag{1.27}$$

$$C(0, t) = C_{in}(t)$$

$$C(x, t) \to 0 \ \text{ as } \ x \to \infty$$

where D is the diffusion coefficient and v is the velocity of groundwater flow. The solution to (1.27) at time T is the convolution

$$C(x, T) = \int_0^T C_{in}(t)f(x, T-t)dt, \tag{1.28}$$

where $C_{in}(t)$ is the time history of contaminant injection at $x = 0$, and the kernel is

$$f(x, T-t) = \frac{x}{2\sqrt{\pi D(T-t)^3}}e^{-\frac{[x-v(T-t)]^2}{4D(T-t)}}. \tag{1.29}$$

Example 1.8

An important and instructive inverse problem is **tomography**, from the Greek roots *tomos*, "to section" or "to cut" (the ancient concept of an *atom* was that of an irreducible, uncuttable object) and *graphein*, "to write." Tomography is the general technique of determining models that are consistent with path-integrated properties such as attenu-ation (e.g., X-ray, radar, muon, seismic), travel time (e.g., electromagnetic, seismic, or

acoustic), or source intensity (e.g., positron emission). Here, we will use examples from seismic tomography. Although tomography problems originally involved determining models that were two-dimensional slices of three-dimensional objects, the term is now commonly used in situations where the model is two- or three-dimensional. Tomography has many applications in medicine, engineering, acoustics, and Earth science. One important geophysical example is cross-well tomography, where seismic sources are installed in a borehole, and the signals are received by sensors in another borehole. Another example is joint earthquake location/velocity structure inversion carried out on scales ranging from a fraction of a cubic kilometer to global [158, 159, 160].

The physical model for tomography in its most basic form (Figure 1.8) assumes that geometric ray theory (essentially the high-frequency limiting case of the wave equation) is valid, so that wave energy traveling between a source and receiver can be considered to be propagating along infinitesimally narrow ray paths. The density of ray path coverage in a tomographic problem may vary significantly throughout a section or volume, and provide much better constraints on physical properties in densely sampled regions than in sparsely sampled ones.

In seismic tomography, if the slowness at a point $\mathbf{x}$ is $s(\mathbf{x})$, and the ray path ℓ is known, then the travel time for seismic energy transiting along that ray path is given by the line integral along ℓ

$$t = \int_{\ell} s(\mathbf{x}(l))\,dl. \tag{1.30}$$
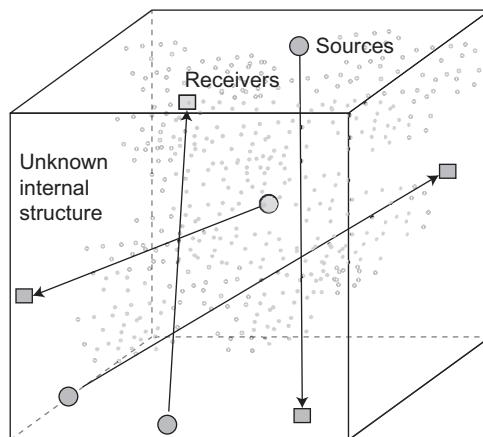
Figure 1.8 Conceptual depiction of ray path tomography. Sources and receivers may, in general, be either at the edges or within the volume, and ray paths may be either straight, as depicted, or curved.

Note that (1.30) is just a higher-dimensional generalization of (1.21), the forward problem for the vertical seismic profiling example. In general, seismic ray paths will be bent due to refraction and/or reflection. In cases where such effects are negligible, ray paths can be usefully approximated as straight lines (e.g., as depicted in Figure 1.8), and the forward and inverse problems can be cast in a linear form. However, if the ray paths are significantly bent by slowness variations, the resulting inverse problem will be nonlinear.

1.4. DISCRETIZING INTEGRAL EQUATIONS

Consider problems of the form

$$d(x) = \int_a^b g(x, \xi) m(\xi) d\xi. \tag{1.31}$$

Here $d(x)$ is a known function, typically representing observed data. The kernel $g(x, \xi)$ is considered to be given, and encodes the physics relating an unknown model $m(x)$ to observed data $d(s)$. The interval $[a, b]$ may be finite or infinite. The function $d(x)$ might in theory be known over an entire interval, but in practice we will only have measurements of $d(x)$ at a finite set of points.

We wish to solve for $m(x)$. This type of linear equation is called a Fredholm integral equation of the first kind, or IFK. A surprisingly large number of inverse problems, including all of the examples from the previous section, can be written as IFKs. Unfortunately, IFKs have properties that can make them very challenging to solve.

To obtain useful numerical solutions to IFKs, we will frequently discretize them into forms that are tractably solvable using the methods of linear algebra. We first assume that $d(x)$ is known at a finite number of points $x_1, x_2, \ldots, x_m$. We can then write the forward problem as

$$d_i = d(x_i) = \int_a^b g(x_i, \xi) m(\xi) d\xi \quad i = 1, 2, \ldots, m \tag{1.32}$$

or as

$$d_i = \int_a^b g_i(x) m(x) dx \quad i = 1, 2, \ldots, m, \tag{1.33}$$

where $g_i(x) = g(x_i, \xi)$. The functions $g_i(t)$ are referred to as **representers** or **data kernels**.

Here, we will apply a **quadrature rule** to approximate (1.33) numerically. Note that, although quadrature methods are applied in this section to linear integral equations, they also have utility in the discretization of nonlinear problems. The simplest quadrature approach is the **midpoint rule**, where we divide the interval $[a, b]$ into n subintervals, and pick points $x_1, x_2, \ldots, x_n$ in the middle of each interval. The points are given by

$$x_j = a + \frac{\Delta x}{2} + (j - 1)\Delta x \qquad (1.34)$$

where

$$\Delta x = \frac{b - a}{n}. \qquad (1.35)$$

The integral (1.33) is then approximated by (Figure 1.9):

$$d_i = \int_a^b g_i(x)m(x)\,dx \approx \sum_{j=1}^n g_i(x_j)m(x_j)\Delta x, \qquad i = 1, 2, \ldots, m. \qquad (1.36)$$

If we let

$$G_{i,j} = g_i(x_j)\Delta x \qquad \begin{pmatrix} i = 1, 2, \ldots, m \\ j = 1, 2, \ldots, n \end{pmatrix} \qquad (1.37)$$

and

$$m_j = m(x_j) \quad j = 1, 2, \ldots, n, \qquad (1.38)$$

then we obtain a linear system of equations $\mathbf{Gm} = \mathbf{d}$.

The approach of using the midpoint rule to approximate the integral is known as **simple collocation**. Of course, there are also more sophisticated quadrature rules for numerically approximating integrals (e.g., the trapezoidal rule, or Simpson's rule). In each case, we end up with an m by n linear system of equations, but the formulas for evaluating the elements of $\mathbf{G}$ will be different.
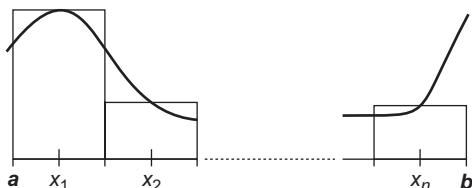
Figure 1.9 Grid for the midpoint rule.

Example 1.9

Consider vertical seismic profiling (Example 1.3), where we wish to estimate vertical seismic slowness using travel time measurements of downward propagating seismic waves (Figure 1.10). We discretize the forward problem (1.21) for m travel time data points, t_i, and n model depths, z_j, spaced at constant intervals of Δz. This discretization is depicted in Figure 1.10.

The discretized problem has

$$t_i = \sum_{j=1}^{n} H(\gamma_i - z_j) s_j \Delta z, \tag{1.39}$$

where $n/m = \Delta\gamma/\Delta z$ is an integer. The rows, $\mathbf{G}_{i,\cdot}$, of the matrix each consist of $i \cdot n/m$ elements equal to Δz on the left and $n - (i \cdot n/m)$ zeros on the right. For $n = m$, $\mathbf{G}$ is a lower triangular matrix with each nonzero entry equal to Δz.

Example 1.10

To discretize the Shaw problem (1.26), we apply the method of simple collocation with m and n intervals for the data and model functions, respectively. We additionally define
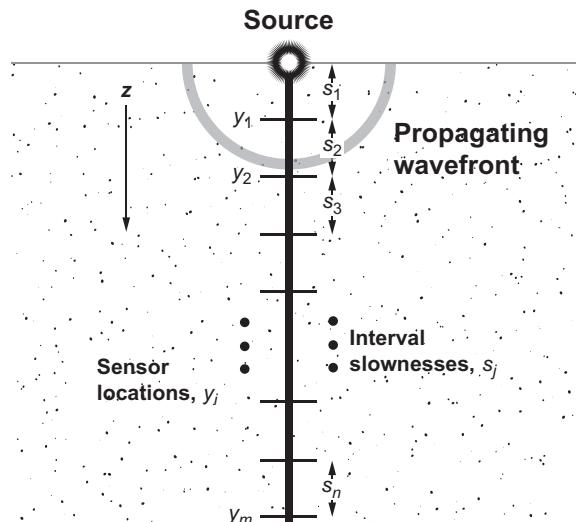
Figure 1.10 Discretization of the vertical seismic profiling problem ($n = 2m$) into uniform intervals.

the data and model points at m and n equally spaced respective angles given by

$$s_i = \frac{(i - 0.5)\pi}{m} - \frac{\pi}{2} \quad i = 1, 2, \ldots, m \tag{1.40}$$

and

$$\theta_j = \frac{(j - 0.5)\pi}{n} - \frac{\pi}{2} \quad j = 1, 2, \ldots, n. \tag{1.41}$$

Correspondingly, discretizing the data and model into m- and n-length vectors

$$d_i = d(s_i) \quad i = 1, 2, \ldots, m \tag{1.42}$$

and

$$m_j = m(\theta_j) \quad j = 1, 2, \ldots, n \tag{1.43}$$

leads to a discrete linear system $\mathbf{Gm} = \mathbf{d}$, where

$$G_{i,j} = (\cos(s_i) + \cos(\theta_j))^2 \left(\frac{\sin(\pi(\sin(s_i) + \sin(\theta_j)))}{\pi(\sin(s_i) + \sin(\theta_j))} \right)^2 \Delta\theta \tag{1.44}$$

and

$$\Delta\theta = \frac{\pi}{n}. \tag{1.45}$$

Example 1.11

We discretize the advection–diffusion problem (1.27), assuming that the parameters D and v in (1.29) are known. We wish to estimate $C_{in}(t)$ from simultaneous concentration observations at the locations x_i at some later time T. The convolution (1.28) for $C(x, T)$ is discretized as

$$\mathbf{d} = \mathbf{Gm} \tag{1.46}$$

where $\mathbf{d}$ is a vector of sampled concentrations at different well locations, $\mathbf{x}$, at time T, $\mathbf{m}$ is a vector of C_{in} values to be estimated, and

$$G_{i,j} = f(x_i, T - t_j)\Delta t \tag{1.47}$$

$$= \frac{x_i}{2\sqrt{\pi D(T - t_j)^3}} e^{-\frac{[x_i - v(T - t_j)]^2}{4D(T - t_j)}} \Delta t. \tag{1.48}$$

Example 1.12

A common way of discretizing the model in a tomographic problem is as uniform blocks (Figure 1.11). This approach is essentially applying the midpoint rule to the travel-time forward problem (1.30).

In this parameterization, the elements of **G** are just the lengths of the ray paths within corresponding blocks. Consider the example of Figure 1.11, where nine homogeneous blocks with sides of unit length and unknown slowness are crossed by eight ray paths. For straight ray paths, we map the two-dimensional slowness grid to a model vector using a row-by-row indexing convention. The constraint equations in the mathematical model are then

$$
\mathbf{Gm} = \begin{bmatrix}
1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\
1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\
\sqrt{2} & 0 & 0 & 0 & \sqrt{2} & 0 & 0 & 0 & \sqrt{2} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sqrt{2}
\end{bmatrix}
\begin{bmatrix}
s_{1,1} \\ s_{1,2} \\ s_{1,3} \\ s_{2,1} \\ s_{2,2} \\ s_{2,3} \\ s_{3,1} \\ s_{3,2} \\ s_{3,3}
\end{bmatrix}
= \begin{bmatrix}
t_1 \\ t_2 \\ t_3 \\ t_4 \\ t_5 \\ t_6 \\ t_7 \\ t_8
\end{bmatrix}. \tag{1.49}
$$

Because there are nine unknown parameters $s_{i,j}$ in the model, but only eight constraints, the **G** matrix is clearly rank deficient. In fact, rank(**G**) is only seven. In addition,
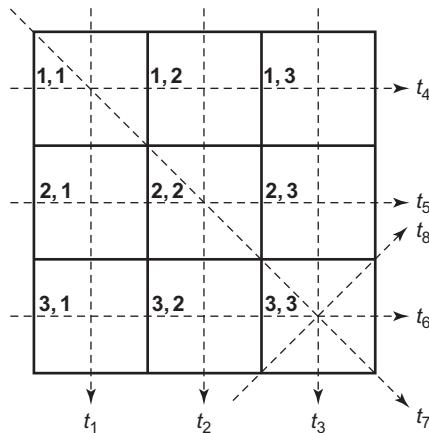
Figure 1.11   Discretization of a tomography problem into uniform blocks. Ray paths correspond to the constraint equations in (1.49).

there is clearly redundant information in (1.49), in that the slowness $s_{3,3}$ is completely determined by t_8, yet $s_{3,3}$ also influences the observations t_3, t_6, and t_7.

1.5. WHY INVERSE PROBLEMS ARE DIFFICULT

Scientists and engineers need to be concerned with far more than simply finding mathematically acceptable answers to parameter estimation and inverse problems. One reason is that there may be many models that adequately fit the data. It is essential to characterize just what solution has been obtained, how "good" it is in terms of physical plausibility, its ability to predict the data, and perhaps how consistent it is with other constraints. Essential issues that must be considered include **solution existence**, **solution uniqueness**, and **instability** of the solution process.

1. *Existence*. There may be no model that exactly fits the data. This can occur in practice because the mathematical model of the system's physics is approximate (or perhaps simply incorrect) or because the data contain noise.
2. *Uniqueness*. If exact solutions do exist, they may not be unique, even for an infinite number of exact data points. That is, there may be other solutions besides m_{true} that exactly satisfy $G(m) = d_{\text{true}}$. This situation commonly occurs in potential field problems. A classic example is the external gravitational field from a spherically symmetric mass distribution, which depends only on the total mass, and not on the radial density distribution.

   Nonuniqueness is a characteristic of rank deficient discrete linear inverse problems because the matrix $\mathbf{G}$ in this case has a nontrivial **null space**. In linear inverse problems, models ($\mathbf{m}_0$) that lie in the null space of $\mathbf{G}$ are solutions to $\mathbf{Gm}_0 = \mathbf{0}$. By superposition, any linear combination of these **null space models** can be added to a particular model that satisfies (1.7) and not change the fit to the data. There are thus an infinite number of mathematically acceptable models in such situations. In practical terms, suppose that there exists a nonzero model $\mathbf{m}_0$ that results in an instrument reading of zero. We cannot discriminate this situation from the situation where $\mathbf{m}_0$ is truly zero.

   An important and thorny issue with problems that have nonunique solutions is that an estimated model may be significantly smoothed or otherwise biased relative to the true situation. Characterizing such bias is essential to interpreting models in terms of their possible correspondence to reality. This issue falls under the general topic of **model resolution analysis**.
3. *Instability*. The process of computing an inverse solution can be, and often is, extremely unstable in that a small change in measurement (e.g., a small η in (1.3)) can lead to an enormous change in the estimated model. Inverse problems where this

situation arises are referred to as **ill-posed** in the case of continuous systems, or **ill-conditioned** in the case of discrete linear systems. A key point is that it is commonly possible to stabilize the inversion process by imposing additional constraints that bias the solution, a process that is generally referred to as **regularization**. Regularization is frequently essential to producing a usable solution to an otherwise intractable ill-posed or ill-conditioned inverse problem.

To examine existence, uniqueness, and instability issues, let us consider some simple mathematical examples using an IFK,

$$\int_0^1 g(x, \xi) m(\xi) d\xi = \gamma(x). \tag{1.50}$$

First, consider the trivial case where the kernel is a constant, such as

$$g(x, \xi) = 1, \tag{1.51}$$

which produces the integral equation

$$\int_0^1 m(\xi) d\xi = \gamma(x). \tag{1.52}$$

Because the left-hand side of (1.52) is independent of x, this system has no solution unless $\gamma(x)$ is a constant. Thus, there are an infinite number of mathematically conceivable data sets $\gamma(x)$ that are not constant and for which no exact solution exists. This is a simple illustration of a solution existence issue.

Where a solution to (1.52) does exist, the solution is nonunique because there are an infinite number of functions that, when integrated over the unit interval, produce the same constant and thus satisfy the IFK exactly. This demonstrates a uniqueness issue.

A more subtle example of nonuniqueness can be seen for

$$g(x, \xi) = x \cdot \sin(\pi \xi) \tag{1.53}$$

in (1.50), so that the IFK becomes

$$\int_0^1 x \cdot \sin(\pi \xi) m(\xi) d\xi = \gamma(x). \tag{1.54}$$

The functions $\sin(k\pi x)$ for integer values of k are orthogonal in the sense that

$$\int_0^1 \sin(k\pi x)\, \sin(l\pi x)\, dx = -\frac{1}{2}\int_0^1 \cos(\pi(k+l)x) - \cos(\pi(k-l)x)\, dx =$$

$$-\frac{1}{2\pi}\left(\frac{\sin(\pi(k+l))}{k+l} - \frac{\sin(\pi(k-l))}{k-l}\right) = 0 \quad (k \neq \pm l;\ k,\ l \neq 0). \tag{1.55}$$

Thus, in (1.54), for models of the form $m(x) = \sin(k\pi x)$, for $k = \pm 2, \pm 3, \ldots$, we have

$$\int_0^1 g(x,\ \xi)m(\xi)\, d\xi = \int_0^1 g(x,\ \xi)\, \sin(k\pi\xi)\, d\xi = 0. \tag{1.56}$$

Furthermore, because (1.54) is a linear system, we can add any function of the form

$$m_0(x) = \sum_{k=2}^\infty \alpha_k\, \sin(k\pi x) \tag{1.57}$$

to a solution, $m(x)$, and obtain a new model that fits the data equally well.

$$\int_0^1 x \cdot \sin(\pi\xi)(m(\xi) + m_0(\xi))\, d\xi = \int_0^1 x \cdot \sin(\pi\xi)m(\xi)\, d\xi + \int_0^1 s \cdot \sin(\pi\xi)m_0(\xi)\, d\xi$$

$$= \int_0^1 x \cdot \sin(\pi\xi)m(\xi)\, d\xi + 0. \tag{1.58}$$

There are thus an infinite number of very different solutions that fit the data equally well.

Finally, even if we do not encounter existence or uniqueness issues, instability is a fundamental feature of IFKs. It can be shown that, in the limit as k goes to infinity,

$$\lim_{k\to\infty} \int_{-\infty}^\infty g(x,\ \xi)\, \sin\, k\pi\xi\, d\xi = 0 \tag{1.59}$$

for *all* square-integrable functions $g(x,\ \xi)$. This result is known as the **Riemann–Lebesgue lemma** [134]. Examining (1.59) in more detail, we can better understand why this occurs. The oscillatory sine function is smoothed by integration with the kernel $g(\xi,\ x)$. For sufficiently large values of the sine frequency, k, the positive and negative excursions of the sine function will average out to zero. The inverse problem has this situation reversed, so that an inferred model can be extremely sensitive to small changes

in the data. If such small data changes are created by random noise that has nothing to do with the physical system that we are studying, then an inferred model from solving an inverse problem to fit this noise can easily have essentially no correspondence to the true model.

The unstable character of IFK solutions is similar to the situation encountered in solving linear systems of equations where the condition number of the matrix is very large, or equivalently, where the matrix is nearly singular. In both cases, the difficulty lies in the mathematical model itself, and not in the particular algorithm used to solve the problem. Fundamentally, most inverse problems are ill-posed.

1.6. EXERCISES

1. Consider a mathematical model of the form $G(\mathbf{m}) = \mathbf{d}$, where $\mathbf{m}$ is a vector of length n, and $\mathbf{d}$ is a vector of length m. Suppose that the model obeys the superposition and scaling laws and is thus linear. Show that $G(\mathbf{m})$ can be written in the form

$$G(\mathbf{m}) = \mathbf{\Gamma m} \tag{1.60}$$

where $\mathbf{\Gamma}$ is an m by n matrix. What are the elements of $\mathbf{\Gamma}$? Hint: Consider the standard basis, and write $\mathbf{m}$ as a linear combination of the vectors in the standard basis. Apply the superposition and scaling laws. Finally, recall the definition of matrix–vector multiplication.

2. Can (1.14) be inconsistent, even with only $m = 3$ data points? How about just $m = 2$ data points? If the system can be inconsistent, give an example. If not, explain why.

3. Consider the borehole vertical seismic profile problem of Examples 1.3 and 1.9 for $n = 100$ equally spaced seismic sensors located at depths of $z = 0.2, 0.4, \ldots , 20$ m, and for a model $\mathbf{m}$ describing n corresponding equal-length seismic slowness values for 0.2 m intervals having midpoints at $z - 0.1$ m.

   a. Calculate the appropriate system matrix, $\mathbf{G}$, for discretizing the integral equation (1.21) using the midpoint rule.

   b. For a seismic velocity model having a linear depth gradient specified by

$$v = v_0 + kz, \tag{1.61}$$

where the velocity at $z = 0$ is $v_0 = 1$ km/s and the gradient is $k = 40$ m/s per m, calculate the true slowness values, $\mathbf{m}_{\text{true}}$, at the midpoints of the n intervals. Integrate the corresponding slowness function for (1.61) using (1.21) to calculate a noiseless synthetic data vector, $\mathbf{d}$, of predicted seismic travel times at the sensor depths.

   c. Solve for the slowness, $\mathbf{m}$, as a function of depth using your $\mathbf{G}$ matrix and analytically calculated noiseless travel times using the MATLAB backslash operator. Compare your result graphically with $\mathbf{m}_{\text{true}}$.

   d. Generate a noisy travel time vector where independent normally distributed noise with a standard deviation of 0.05 ms is added to the elements of **d**. Resolve the system for **m** and again compare your result graphically with $\mathbf{m}_{\text{true}}$. How has the model changed?

   e. Repeat the problem, but for just $n = 4$ sensor depths and corresponding equal length slowness intervals. Is the recovery of the true model improved? Explain in terms of the condition numbers of your **G** matrices.

4. Find a journal article that discusses the solution of an inverse problem in a discipline of special interest to you. What are the data? Are the data discrete or continuous? Have the authors discussed possible sources of noise in the data? What is the model? Is the model continuous or discrete? What physical laws determine the forward operator G? Is G linear or nonlinear? Do the authors discuss any issues associated with existence, uniqueness, or instability of solutions?

1.7. NOTES AND FURTHER READING

Some important references on inverse problems in geophysics and remote sensing include [28, 60, 99, 126, 166]. Instructive examples of ill–posed problems and their solutions can be found in the book edited by Tikhonov and Goncharsky [162]. More mathematically oriented references on inverse problems include [8, 45, 59, 65, 67, 90, 94, 108, 112, 155, 161]. Tomography, particularly in medical imaging and seismology, is a very large field. Some general references on tomography are [70, 79, 84, 98, 100, 114, 121].

This page intentionally left blank

Linear Regression

Synopsis

Linear regression is introduced as a parameter estimation problem, and least squares solutions are derived. Maximum likelihood is defined, and its association with least squares solutions under normally distributed data errors is demonstrated. Statistical tests based on χ^2 that provide insight into least squares solutions are discussed. The mapping of data errors into model errors in the context of least squares is described. The determination of confidence intervals using the model covariance matrix and the meaning of model parameter correlations is discussed. The problems of estimating unknown data standard deviations and recognizing proportional data errors are addressed. The issue of data outliers and the concept of robust estimation are introduced, and 1–norm minimization is introduced as a robust estimation technique. General propagation of errors between data and model using Monte Carlo methods is discussed in the context of the iteratively reweighted least squares 1–norm minimization algorithm.

2.1. INTRODUCTION TO LINEAR REGRESSION

The problem of finding a parameterized curve that approximately fits a set of data is referred to as **regression**. When the regression model is linear in the fitted parameters, then we have a **linear regression** problem. In this chapter, linear regression problems are analyzed as discrete linear inverse problems.

Consider a discrete linear inverse problem. We begin with a data vector, $\mathbf{d}$, of m observations, and a vector, $\mathbf{m}$, of n model parameters that we wish to determine. The forward problem can be expressed as a linear system of equations using an m by n matrix $\mathbf{G}$

$$\mathbf{Gm} = \mathbf{d}. \tag{2.1}$$

Recall that if $\text{rank}(\mathbf{G}) = n$, then the matrix has full column rank. In this chapter we will assume that the matrix $\mathbf{G}$ has full column rank. In Chapter 3 we will consider rank deficient problems.

For a full column rank matrix, it is frequently the case that there is no solution $\mathbf{m}$ that satisfies (2.1) exactly. This happens because the dimension of the range of $\mathbf{G}$ is smaller than m and a noisy data vector, $\mathbf{d}$, will generally lie outside of the range of $\mathbf{G}$ ($\mathbf{d}$ will lie in R^n for typical noise scenarios).

A useful approximate solution may still be found by finding a particular model $\mathbf{m}$ that minimizes some measure of the misfit between the actual data and $\mathbf{Gm}$. The **residual vector** is the vector of differences between observed data and corresponding model predictions

$$\mathbf{r} = \mathbf{d} - \mathbf{Gm}. \tag{2.2}$$

and the elements of $\mathbf{r}$ are frequently referred to simply as residuals. One commonly used measure of the misfit is the 2-norm of the residual vector, and a model that minimizes this 2-norm is called a **least squares solution**. The least squares or 2-norm solution is of special interest both because it is readily amenable to analysis and geometric intuition, and because it turns out to be statistically the most likely solution if data errors are normally distributed.

The least squares solution is, from the normal equations (A.73),

$$\mathbf{m}_{L_2} = (\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T\mathbf{d}. \tag{2.3}$$

It can be shown that if $\mathbf{G}$ is of full column rank, then $(\mathbf{G}^T\mathbf{G})^{-1}$ exists (Exercise A.13f).

A classic linear regression problem is finding parameters m_1 and m_2 for a line, $y_i = m_1 + m_2 x_i$, that best fits a set of $m > 2$ data points. The system of equations in this case is

$$\mathbf{Gm} = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ 1 & x_m \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ \cdot \\ \cdot \\ \cdot \\ d_m \end{bmatrix} = \mathbf{d}, \tag{2.4}$$

where the d_i are observations of y at each corresponding position x_i. Applying (2.3) to the $\mathbf{G}$ and $\mathbf{m}$ specified in (2.4) gives the least squares solution

$$\mathbf{m}_{L_2} = (\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T\mathbf{d} \tag{2.5}$$

$$= \left(\begin{bmatrix} 1 & \cdots & 1 \\ x_1 & \cdots & x_m \end{bmatrix} \begin{bmatrix} 1 & x_1 \\ \vdots & \vdots \\ 1 & x_m \end{bmatrix} \right)^{-1} \begin{bmatrix} 1 & \cdots & 1 \\ x_1 & \cdots & x_m \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_m \end{bmatrix}$$

$$= \begin{bmatrix} m & \sum_{i=1}^{m} x_i \\ \sum_{i=1}^{m} x_i & \sum_{i=1}^{m} x_i^2 \end{bmatrix}^{-1} \begin{bmatrix} \sum_{i=1}^{m} d_i \\ \sum_{i=1}^{m} x_i d_i \end{bmatrix}$$

$$= \frac{1}{m \sum_{i=1}^{m} x_i^2 - (\sum_{i=1}^{m} x_i)^2} \begin{bmatrix} \sum_{i=1}^{m} x_i^2 & -\sum_{i=1}^{m} x_i \\ -\sum_{i=1}^{m} x_i & m \end{bmatrix} \begin{bmatrix} \sum_{i=1}^{m} d_i \\ \sum_{i=1}^{m} x_i d_i \end{bmatrix}.$$

2.2. STATISTICAL ASPECTS OF LEAST SQUARES

If we consider data points to be imperfect measurements that include random errors, then we are faced with the problem of finding the solution that is best from a statistical point of view. One approach, **maximum likelihood estimation**, considers the question from the following perspective. Given that we observed a particular data set, that we know the statistical characteristics of these observations, and that we have a mathematical model for the forward problem, what is the model from which these observations would most likely arise?

Maximum likelihood estimation is a general method that can be applied to any estimation problem where a joint probability density function (B.26) can be assigned to the observations. The essential problem is to find the most likely model, as characterized by the elements of the parameter vector $\mathbf{m}$, for the set of observations contained in the vector $\mathbf{d}$. We will assume that the observations are independent so that we can use the product form of the joint probability density function (B.28).

Given a model $\mathbf{m}$, we have a probability density function $f_i(d_i|\mathbf{m})$ for each of the observations. In general, these probability density functions will vary depending on $\mathbf{m}$, so probability densities are conditional on $\mathbf{m}$. The joint probability density for a vector of independent observations $\mathbf{d}$ will be

$$f(\mathbf{d}|\mathbf{m}) = f_1(d_1|\mathbf{m}) \cdot f_2(d_2|\mathbf{m}) \cdot \cdot \cdot f_m(d_m|\mathbf{m}). \tag{2.6}$$

Note that the $f(d_i|\mathbf{m})$ are probability densities, not probabilities. We can only compute the probability of observing data in some range for a given model $\mathbf{m}$ by integrating $f(\mathbf{d}|\mathbf{m})$ over that range. In fact, the probability of getting any particular set of data exactly is precisely zero! This conceptual conundrum can be resolved by considering the probability of getting a data set that lies within a small m-dimensional box around a particular data set $\mathbf{d}$. This probability will be nearly proportional to the probability density $f(\mathbf{d}|\mathbf{m})$.

In practice, we measure a particular data vector and wish to find the "best" model to match it in the maximum likelihood sense. That is, $\mathbf{d}$ will be a fixed set of observations, and $\mathbf{m}$ will be a vector of parameters to be estimated. The **likelihood function**, L, is the probability of $\mathbf{m}$ given an observed $\mathbf{d}$, which is identical to (2.6), the joint probability density function of $\mathbf{d}$ given $\mathbf{m}$

$$L(\mathbf{m}|\mathbf{d}) = f(\mathbf{d}|\mathbf{m}). \tag{2.7}$$

For many possible models, $\mathbf{m}$, (2.7) will be extremely close to zero because such models would be extremely unlikely to produce the observed data set $\mathbf{d}$. The likelihood would be much larger for any models that, conversely, would be relatively likely to produce the observed data. According to the **maximum likelihood principle** we should select the

model $\mathbf{m}$ that maximizes the likelihood function (2.7). Model estimates obtained in this manner have many desirable statistical properties [29, 40].

It is particularly insightful that when we have a discrete linear inverse problem and the data errors are independent and normally distributed, then the maximum likelihood principle solution is the least squares solution. To show this, assume that the data have independent random errors that are normally distributed with expected value zero, and where the standard deviation of the ith observation, d_i, is σ_i. The probability density for d_i then takes the form of (B.6),

$$f_i(d_i|\mathbf{m}) = \frac{1}{\sigma_i\sqrt{2\pi}}e^{-\frac{1}{2}(d_i-(\mathbf{Gm})_i)^2/\sigma_i^2}. \tag{2.8}$$

The likelihood function for the complete data set is the product of the individual likelihoods,

$$L(\mathbf{m}|\mathbf{d}) = \frac{1}{(2\pi)^{m/2}\prod_{i=1}^{m}\sigma_i}\prod_{i=1}^{m}e^{-\frac{1}{2}(d_i-(\mathbf{Gm})_i)^2/\sigma_i^2}. \tag{2.9}$$

The constant factor does not affect the maximization of L, so we can solve

$$\max\prod_{i=1}^{m}e^{-\frac{1}{2}(d_i-(\mathbf{Gm})_i)^2/\sigma_i^2}. \tag{2.10}$$

The natural logarithm is a monotonically increasing function, so we can equivalently solve

$$\max\log\prod_{i=1}^{m}e^{-\frac{1}{2}(d_i-(\mathbf{Gm})_i)^2/\sigma_i^2} = \max\left[-\frac{1}{2}\sum_{i=1}^{m}\frac{(d_i-(\mathbf{Gm})_i)^2}{\sigma_i^2}\right]. \tag{2.11}$$

Finally, if we turn the maximization into a minimization by changing sign and ignore the constant factor of $1/2$, the problem becomes

$$\min\sum_{i=1}^{m}\frac{(d_i-(\mathbf{Gm})_i)^2}{\sigma_i^2}. \tag{2.12}$$

Aside from the distinct $1/\sigma_i^2$ factors in each term of the sum, this is identical to the least squares problem for $\mathbf{Gm}=\mathbf{d}$.

To incorporate the data standard deviations into this solution, we scale the system of equations to obtain a weighted system of equations. Let a diagonal weighting matrix be

$$\mathbf{W} = \mathrm{diag}(1/\sigma_1, \ 1/\sigma_2, \ \ldots, \ 1/\sigma_m). \tag{2.13}$$

Then let

$$\mathbf{G}_w = \mathbf{W}\mathbf{G} \tag{2.14}$$

and

$$\mathbf{d}_w = \mathbf{W}\mathbf{d}. \tag{2.15}$$

The weighted system of equations is then

$$\mathbf{G}_w\mathbf{m} = \mathbf{d}_w. \tag{2.16}$$

The normal equations (A.73) solution to (2.16) is

$$\mathbf{m}_{L_2} = (\mathbf{G}_w^T\mathbf{G}_w)^{-1}\mathbf{G}_w^T\mathbf{d}_w. \tag{2.17}$$

Now,

$$\|\mathbf{d}_w - \mathbf{G}_w\mathbf{m}_{L_2}\|_2^2 = \sum_{i=1}^m (d_i - (\mathbf{G}\mathbf{m}_{L_2})_i)^2/\sigma_i^2. \tag{2.18}$$

Thus, the least squares solution to $\mathbf{G}_w\mathbf{m} = \mathbf{d}_w$ is the maximum likelihood solution.

The sum of the squares of the residuals also provides useful statistical information about the quality of model estimates obtained with least squares. The **chi–square statistic** is

$$\chi_{\text{obs}}^2 = \sum_{i=1}^m (d_i - (\mathbf{G}\mathbf{m}_{L_2})_i)^2/\sigma_i^2. \tag{2.19}$$

Since χ_{obs}^2 depends on the random measurement errors in $\mathbf{d}$, it is itself a random variable. It can be shown that under our assumptions, χ_{obs}^2 has a χ^2 distribution with $\nu = m - n$ degrees of freedom [29, 40].

The probability density function for the χ^2 distribution is

$$f_{\chi^2}(x) = \frac{1}{2^{\nu/2}\Gamma(\nu/2)} x^{\frac{1}{2}\nu-1} e^{-x/2} \tag{2.20}$$

(Figure B.4). The χ^2 **test** provides a statistical assessment of the assumptions that we used in finding the least squares solution. In this test, we compute χ_{obs}^2 and compare it to the theoretical χ^2 distribution with $\nu = m - n$ degrees of freedom.

The probability of obtaining a χ^2 value as large or larger than the observed value (and hence a worse misfit between data and model data predictions than that obtained) is called the **p-value** of the test, and is given by

$$p = \int_{\chi_{\text{obs}}^2}^{\infty} f_{\chi^2}(x)\,dx. \tag{2.21}$$

When data errors are independent and normally distributed, and the mathematical model is correct, it can be shown that the p-value will be uniformly distributed between zero and one (Exercise 2.4). In practice, particular p-values that are very close to either extreme indicate that one or more of these assumptions are incorrect.

There are three general cases.

1. The p-value is not too small and not too large. Our least squares solution produces an acceptable data fit and our statistical assumptions of data errors are consistent. Practically, p does not actually have to be very large to be deemed marginally "acceptable" in many cases (e.g., $p \approx 10^{-2}$), as truly "wrong" models will typically produce extraordinarily small p-values (e.g., 10^{-12}) because of the short-tailed nature of the normal distribution.

   Because the p-value will be uniformly distributed when we have a correct mathematical model and our statistical data assumptions are valid, it is inappropriate to conclude anything based on the differences between p-values in this range. For example, one should not conclude that a p-value of 0.7 is "better" than a p-value of 0.2.

2. The p-value is very small. We are faced with three nonexclusive possibilities, but something is clearly wrong.

   a. The data truly represent an extremely unlikely realization. This is easy to rule out for p-values very close to zero. For example, suppose an experiment produced a data realization where the probability of a worse fit was 10^{-9}. If the model was correct, then we would have to perform on the order of a billion experiments to get a comparably poor fit to the data. It is far more likely that something else is wrong.

   b. The mathematical model $\mathbf{Gm} = \mathbf{d}$ is incorrect. Most often this happens because we have left some important aspect of the physics out of the mathematical model.

   c. The data errors are underestimated or not normally distributed. In particular, we may have underestimated the σ_i.

3. The p-value is very close to one. The fit of the model predictions to the data is almost exact. We should investigate the possibility that we have overestimated the data errors. A more sinister possibility is that a very high p-value is indicative of data fraud, such as might happen if data were cooked up ahead of time to fit a particular model!

A rule of thumb for problems with a large number of degrees of freedom, ν, is that the expected value of χ^2 approaches ν. This arises because, by the central limit theorem (Section B.6), the χ^2 random variable, which is itself a sum of random variables, will become normally distributed as the number of terms in the sum becomes large. The mean of the resulting distribution will approach ν and the standard deviation will approach $(2\nu)^{1/2}$.

In addition to examining χ^2_{obs}, it is important to examine the individual weighted residuals corresponding to a model, $r_{w,i} = (\mathbf{d} - \mathbf{Gm})_i/\sigma_i = (\mathbf{d}_w - \mathbf{G}_w\mathbf{m})_i$. The elements of $\mathbf{r}_w$ should be roughly normally distributed with standard deviation one and should show no obvious patterns. In some cases where an incorrect model has been fitted to the data, the residuals will reveal the nature of the modeling error. For example, in linear regression to a line, it might be that all of the residuals are negative for small and large values of the independent variable x but positive for intermediate values of x. This would suggest that an additional quadratic term is required in the regression model.

Parameter estimates obtained via linear regression are linear combinations of the data (2.17). If the data errors are normally distributed, then the parameter estimates will also be normally distributed because a linear combination of normally distributed random variables is normally distributed [4, 29]. To derive the mapping between data and model covariances, consider the covariance of a data vector, $\mathbf{d}$, of normally distributed, independent random variables, operated on by a general linear transformation specified by a matrix, $\mathbf{A}$. The appropriate covariance mapping is (B.65)

$$\text{Cov}(\mathbf{Ad}) = \mathbf{A}\text{Cov}(\mathbf{d})\mathbf{A}^T. \tag{2.22}$$

The least squares solution has $\mathbf{A} = (\mathbf{G}_w^T\mathbf{G}_w)^{-1}\mathbf{G}_w^T$. The general covariance matrix of the model parameters for a least squares solution is thus

$$\text{Cov}(\mathbf{m}_{L_2}) = (\mathbf{G}_w^T\mathbf{G}_w)^{-1}\mathbf{G}_w^T\text{Cov}(\mathbf{d}_w)\mathbf{G}_w(\mathbf{G}_w^T\mathbf{G}_w)^{-1}. \tag{2.23}$$

If the weighted data are independent, and thus have an identity covariance matrix, this simplifies to

$$\text{Cov}(\mathbf{m}_{L_2}) = (\mathbf{G}_w^T\mathbf{G}_w)^{-1}\mathbf{G}_w^T\mathbf{I}_m\mathbf{G}_w(\mathbf{G}_w^T\mathbf{G}_w)^{-1} = (\mathbf{G}_w^T\mathbf{G}_w)^{-1}. \tag{2.24}$$

In the case of independent and identically distributed normal data errors, so that the data covariance matrix $\text{Cov}(\mathbf{d})$ is simply the variance σ^2 times the m by m identity matrix, $\mathbf{I}_m$, (2.24) can be written in terms of the unweighted system matrix as

$$\text{Cov}(\mathbf{m}_{L_2}) = \sigma^2(\mathbf{G}^T\mathbf{G})^{-1}. \tag{2.25}$$

Note that even for a diagonal data covariance matrix, the model covariance matrix is typically not diagonal, and the model parameters are thus correlated. Because elements of least squares models are each constructed from linear combinations of the data vector elements, this statistical dependence between the elements of $\mathbf{m}$ should not be surprising.

The expected value of the least squares solution is

$$E[\mathbf{m}_{L_2}] = (\mathbf{G}_w^T\mathbf{G}_w)^{-1}\mathbf{G}_w^T E[\mathbf{d}_w]. \tag{2.26}$$

Because $E[\mathbf{d}_w] = \mathbf{d}_{\text{true},w}$, and $\mathbf{G}_w \mathbf{m}_{\text{true}} = \mathbf{d}_{\text{true},w}$, we have

$$\mathbf{G}_w^T \mathbf{G}_w \mathbf{m}_{\text{true}} = \mathbf{G}_w^T \mathbf{d}_{\text{true},w}. \tag{2.27}$$

Thus

$$E[\mathbf{m}_{L_2}] = (\mathbf{G}_w^T \mathbf{G}_w)^{-1} \mathbf{G}_w^T \mathbf{G}_w \mathbf{m}_{\text{true}} \tag{2.28}$$

$$= \mathbf{m}_{\text{true}}. \tag{2.29}$$

In statistical terms, the least squares solution is said to be **unbiased**.

We can compute 95% confidence intervals for individual model parameters using the fact that each model parameter m_i has a normal distribution with mean given by the corresponding element of $\mathbf{m}_{\text{true}}$ and variance $\text{Cov}(\mathbf{m}_{L_2})_{i,i}$. The 95% confidence intervals are given by

$$\mathbf{m}_{L_2} \pm 1.96 \, \text{diag}(\text{Cov}(\mathbf{m}_{L_2}))^{1/2}, \tag{2.30}$$

where the 1.96 factor arises from

$$\frac{1}{\sigma \sqrt{2\pi}} \int\limits_{-1.96\sigma}^{1.96\sigma} e^{-\frac{x^2}{2\sigma^2}} \, dx \approx 0.95. \tag{2.31}$$

Example 2.1

Let us recall Example 1.1 of linear regression of ballistic observations to a quadratic model, where the regression model is

$$y(t) = m_1 + m_2 t - (1/2) m_3 t^2. \tag{2.32}$$

Here y is measured in the upward direction, and the minus sign is applied to the third term because gravitational acceleration is downward. Consider a synthetic data set with $m = 10$ observations and independent normal data errors ($\sigma = 8$ m), generated using

$$\mathbf{m}_{\text{true}} = [10 \text{ m}, \ 100 \text{ m/s}, \ 9.8 \text{ m/s}^2]^T \tag{2.33}$$

(Table 2.1).

To obtain the least squares solution, we construct the **G** matrix. The *ith* row of **G** is given by

$$\mathbf{G}_{i,\cdot} = [1, \ t_i, \ -(1/2) t_i^2], \tag{2.34}$$

Table 2.1 Data for the ballistics example.

t (s)	1	2	3	4	5	6	7	8	9	10
y (m)	109.4	187.5	267.5	331.9	386.1	428.4	452.2	498.1	512.3	513.0

so that

$$
\mathbf{G} = \begin{bmatrix}
1 & 1 & -0.5 \\
1 & 2 & -2.0 \\
1 & 3 & -4.5 \\
1 & 4 & -8.0 \\
1 & 5 & -12.5 \\
1 & 6 & -18.0 \\
1 & 7 & -24.5 \\
1 & 8 & -32.0 \\
1 & 9 & -40.5 \\
1 & 10 & -50.0
\end{bmatrix}.
\tag{2.35}
$$

We solve for the parameters using the weighted normal equations, (2.17), to obtain a model estimate,

$$
\mathbf{m}_{L_2} = [16.4 \text{ m}, \ 97.0 \text{ m/s}, \ 9.4 \text{ m/s}^2]^T.
\tag{2.36}
$$

Figure 2.1 shows the observed data and the fitted curve. The model covariance matrix associated with $\mathbf{m}_{L_2}$ is

$$
\text{Cov}(\mathbf{m}_{L_2}) = \begin{bmatrix}
88.53 & -33.60 & -5.33 \\
-33.60 & 15.44 & 2.67 \\
-5.33 & 2.67 & 0.48
\end{bmatrix}.
\tag{2.37}
$$
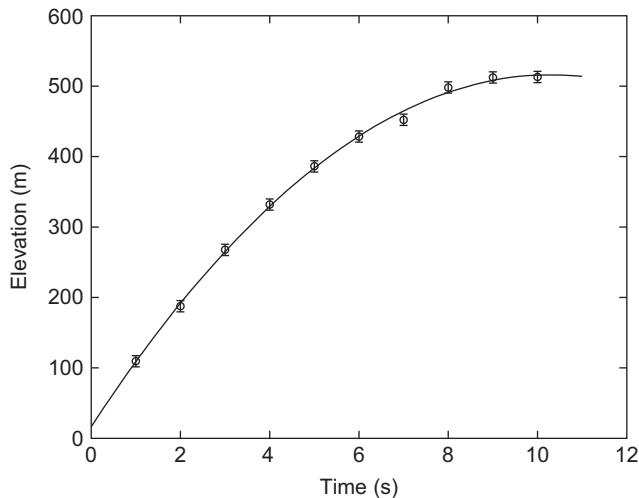
Figure 2.1 Data and model predictions for the ballistics example, with one standard deviation data error bounds indicated.

In our example, $m = 10$ and $n = 3$, so (2.30) gives the following parameter estimates with 95% confidence intervals:

$$\mathbf{m}_{L_2} = [16.42 \pm 18.44\text{m},\ 96.97 \pm 7.70\text{m/s},\ 9.41 \pm 1.36\text{m/s}^2]^T. \tag{2.38}$$

The χ^2 value for this regression is approximately 4.2, and the number of degrees of freedom is $\nu = m - n = 10 - 3 = 7$, so the p-value, (2.21), is

$$p = \int_{4.20}^{\infty} \frac{1}{2^{7/2}\Gamma(7/2)} x^{\frac{5}{2}} e^{-\frac{x}{2}}\, dx \approx 0.76, \tag{2.39}$$

which is in the realm of plausibility. This means that the fitted model is consistent with the modeling and data uncertainty assumptions.

If we consider combinations of model parameters, the interpretation of the uncertainty in the model parameters becomes more complicated. To characterize model uncertainty more effectively, we can examine 95% **confidence regions** for pairs or larger sets of parameters. When joint parameter confidence regions are projected onto the coordinate axes, m_i, we obtain intervals for parameters that may be significantly larger than we would estimate when considering parameters individually, as in (2.38).

For a vector of estimated model parameters $\mathbf{m}_{L_2}$ characterized by an n-dimensional multivariate normal distribution with mean $\mathbf{m}_{\text{true}}$ and covariance matrix $\mathbf{C} = \text{Cov}(\mathbf{m}_{L_2})$,

$$(\mathbf{m}_{\text{true}} - \mathbf{m}_{L_2})^T \mathbf{C}^{-1} (\mathbf{m}_{\text{true}} - \mathbf{m}_{L_2}), \tag{2.40}$$

can be shown to have a χ^2 distribution with n degrees of freedom [89]. Thus if Δ^2 is the 95th percentile of the χ^2 distribution with n degrees of freedom, the 95% confidence region is defined by the inequality

$$(\mathbf{m}_{\text{true}} - \mathbf{m}_{L_2})^T \mathbf{C}^{-1} (\mathbf{m}_{\text{true}} - \mathbf{m}_{L_2}) \leq \Delta^2. \tag{2.41}$$

The confidence region defined by this inequality is an n-dimensional ellipsoid.

If we wish to find an error ellipsoid for a lower dimensional subset of the model parameters, we can project the n-dimensional error ellipsoid onto the lower dimensional subspace by taking only those rows and columns of $\mathbf{C}$ and elements of $\mathbf{m}$ that correspond to the dimensions that we want to keep [1]. In this case, the number of degrees of freedom in the associated Δ^2 calculation should be correspondingly reduced from n to match the number of model parameters in the projected error ellipsoid.

Since the covariance matrix and its inverse are symmetric and positive definite, we can diagonalize $\mathbf{C}^{-1}$ using (A.77) as

$$\mathbf{C}^{-1} = \mathbf{P}^T \boldsymbol{\Lambda} \mathbf{P}, \tag{2.42}$$

where $\mathbf{\Lambda}$ is a diagonal matrix of positive eigenvalues and the columns of $\mathbf{P}$ are orthonormal eigenvectors. The semiaxes defined by the columns of $\mathbf{P}$ and associated eigenvalues are referred to as the **principal axes** of the error ellipsoid. The ith semimajor error ellipsoid axis direction is defined by $\mathbf{P}_{\cdot,i}$ and its corresponding length is $\Delta/\sqrt{\Lambda_{i,i}}$.

Because the model covariance matrix is typically not diagonal, the principal axes are typically not aligned in the m_i axis directions. However, we can project the appropriate confidence ellipsoid onto the m_i axes to obtain a "box" that includes the entire 95% error ellipsoid, along with some additional external volume. Such a box provides a conservative confidence interval for a joint collection of model parameters.

Correlations for parameter pairs (m_i, m_j) are measures of the inclination of the error ellipsoid with respect to the parameter axes. A correlation approaching $+1$ means the projection is needle-like with its long principal axis having a positive slope, a zero correlation means that the projection has principal axes that are aligned with the axes of the (m_i, m_j) plane, and a correlation approaching -1 means that the projection is needle-like with its long principal axis having a negative slope.

● ───

Example 2.2

The parameter correlations for Example 2.1 are

$$\rho_{m_i,m_j} = \frac{\mathrm{Cov}(m_i,\ m_j)}{\sqrt{\mathrm{Var}(m_i) \cdot \mathrm{Var}(m_j)}}, \tag{2.43}$$

which give

$$\rho_{m_1,m_2} = -0.91 \tag{2.44}$$

$$\rho_{m_1,m_3} = -0.81 \tag{2.45}$$

$$\rho_{m_2,m_3} = 0.97. \tag{2.46}$$

The three model parameters are highly statistically dependent, and the error ellipsoid is thus inclined in model space. Figure 2.2 shows the 95% confidence ellipsoid.

Diagonalization of $\mathbf{C}^{-1}$ (2.42) shows that the directions of the semiaxes for the error ellipsoid are

$$\mathbf{P} = [\mathbf{P}_{\cdot,1},\ \mathbf{P}_{\cdot,2},\ \mathbf{P}_{\cdot,3}] \approx \begin{bmatrix} 0.93 & 0.36 & -0.03 \\ -0.36 & 0.90 & -0.23 \\ -0.06 & 0.23 & 0.97 \end{bmatrix}, \tag{2.47}$$

with corresponding eigenvalues

$$[\lambda_1,\ \lambda_2,\ \lambda_3] \approx [0.0098,\ 0.4046,\ 104.7]. \tag{2.48}$$

Figure 2.2 Projections of the 95% error ellipsoid onto model axes. (a) Projections in perspective; (b) projections onto the parameter axis planes.

The corresponding 95% confidence ellipsoid semiaxis lengths are

$$\sqrt{F_{\chi^2,3}^{-1}(0.95)}[1/\sqrt{\lambda_1},\ 1/\sqrt{\lambda_2},\ 1/\sqrt{\lambda_3}] \approx [28.2,\ 4.4,\ 0.27] \tag{2.49}$$

where $F_{\chi^2,3}^{-1}(0.95) \approx 7.80$ is the 95th percentile of the χ^2 distribution with three degrees of freedom.

Projecting the 95% confidence ellipsoid defined by (2.47) and (2.49) into the $(m_1,\ m_2,\ m_3)$ coordinate system, and selecting maximum absolute values in the m_i directions to define an ellipsoid-bounding box, we obtain 95% confidence intervals for the parameters considered jointly,

$$[m_1,\ m_2,\ m_3] = [16.42 \pm 26.25 \text{ m},\ 96.97 \pm 10.24 \text{ m/s},\ 9.41 \pm 1.65 \text{ m/s}^2], \tag{2.50}$$

which are appreciably broader than the single parameter confidence estimates obtained using only the diagonal covariance matrix terms in (2.38). Note that there is actually a greater than 95% probability that the box defined by (2.50) will include the true values of the parameters. The reason is that these intervals, considered together as a rectangular prism–shaped region, include a significant parameter space volume that lies outside of the 95% confidence ellipsoid.

It is insightful to note that the model covariance matrix (2.23) does not depend on the estimated model, but depends solely on the system matrix and data covariance. Model covariance is thus exclusively a characteristic of experimental design that reflects

how much influence the noise in a *general* data set will have on a model estimate, not on *particular* data values from an individual experiment. It is essential to evaluate the *p*-value, or another "goodness-of-fit" measure, in assessing a model because an examination of the solution parameters and the covariance matrix alone does *not* reveal whether we are actually fitting the data adequately.

2.3. AN ALTERNATIVE VIEW OF THE 95% CONFIDENCE ELLIPSOID

Recall (2.29) that in linear regression, the least squares solution $\mathbf{m}_{L_2}$ for zero–mean multivariate distributed normal data errors itself has a multivariate normal distribution with

$$E[\mathbf{m}_{L_2}] = \mathbf{m}_{\text{true}}. \tag{2.51}$$

By (2.24), the model covariance matrix is

$$\mathbf{C} = \text{Cov}(\mathbf{m}_{L_2}) = (\mathbf{G}_w^T \mathbf{G}_w)^{-1}, \tag{2.52}$$

where the rows of $\mathbf{G}_w$ are those of $\mathbf{G}$ that have been weighted by respective reciprocal data standard deviations (2.14), and we assume that the data errors are independent and that $(\mathbf{G}_w^T \mathbf{G}_w)$ is nonsingular. By Theorem B.6,

$$(\mathbf{m}_{\text{true}} - \mathbf{m}_{L_2})^T \mathbf{C}^{-1} (\mathbf{m}_{\text{true}} - \mathbf{m}_{L_2}) \tag{2.53}$$

has a χ^2 distribution with degrees of freedom equal to the number of model parameters, n. Let Δ^2 be the 95th percentile of the χ^2 distribution with n degrees of freedom. Then the probability

$$P\left((\mathbf{m}_{\text{true}} - \mathbf{m}_{L_2})^T \mathbf{C}^{-1} (\mathbf{m}_{\text{true}} - \mathbf{m}_{L_2}) \leq \Delta^2\right) \tag{2.54}$$

will be 0.95.

Although (2.54) describes an ellipsoid centered at $\mathbf{m}_{\text{true}}$, the inequality is symmetric in $\mathbf{m}_{\text{true}}$ and $\mathbf{m}_{L_2}$, and can also therefore be thought of as defining an ellipsoid centered around $\mathbf{m}_{L_2}$. Thus, there is a 95% probability that when we gather our data and compute $\mathbf{m}_{L_2}$, the true model $\mathbf{m}_{\text{true}}$ will lie within the model space ellipsoid defined by

$$(\mathbf{m} - \mathbf{m}_{L_2})^T \mathbf{C}^{-1} (\mathbf{m} - \mathbf{m}_{L_2}) \leq \Delta^2. \tag{2.55}$$

Since $\mathbf{C} = (\mathbf{G}_w^T \mathbf{G}_w)^{-1}$, $\mathbf{C}^{-1} = \mathbf{G}_w^T \mathbf{G}_w$, and the 95th percentile confidence ellipsoid can also be written as

$$(\mathbf{m} - \mathbf{m}_{L_2})^T \mathbf{G}_w^T \mathbf{G}_w (\mathbf{m} - \mathbf{m}_{L_2}) \leq \Delta^2. \tag{2.56}$$

If we let

$$\chi^2(\mathbf{m}) = \|\mathbf{G}_w \mathbf{m} - \mathbf{d}_w\|_2^2 = (\mathbf{G}_w \mathbf{m} - \mathbf{d}_w)^T (\mathbf{G}_w \mathbf{m} - \mathbf{d}_w), \tag{2.57}$$

then

$$\chi^2(\mathbf{m}) - \chi^2(\mathbf{m}_{L_2}) = (\mathbf{G}_w\mathbf{m} - \mathbf{d}_w)^T(\mathbf{G}_w\mathbf{m} - \mathbf{d}_w) - (\mathbf{G}_w\mathbf{m}_{L_2} - \mathbf{d}_w)^T(\mathbf{G}_w\mathbf{m}_{L_2} - \mathbf{d}_w)$$

$$= \mathbf{m}^T\mathbf{G}_w^T\mathbf{G}_w\mathbf{m} - \mathbf{d}_w^T\mathbf{G}_w\mathbf{m} - \mathbf{m}^T\mathbf{G}_w^T\mathbf{d}_w + \mathbf{d}_w^T\mathbf{d}_w$$

$$- \mathbf{m}_{L_2}^T\mathbf{G}_w^T\mathbf{G}_w\mathbf{m}_{L_2} + \mathbf{d}_w^T\mathbf{G}_w\mathbf{m}_{L_2} + \mathbf{m}_{L_2}^T\mathbf{G}_w^T\mathbf{d}_w - \mathbf{d}_w^T\mathbf{d}_w$$

$$= \mathbf{m}^T\mathbf{G}_w^T\mathbf{G}_w\mathbf{m} - \mathbf{d}_w^T\mathbf{G}_w\mathbf{m} - \mathbf{m}^T\mathbf{G}_w^T\mathbf{d}_w$$

$$- \mathbf{m}_{L_2}^T\mathbf{G}_w^T\mathbf{G}_w\mathbf{m}_{L_2} + \mathbf{d}_w^T\mathbf{G}_w\mathbf{m}_{L_2} + \mathbf{m}_{L_2}^T\mathbf{G}_w^T\mathbf{d}_w. \qquad (2.58)$$

Since $\mathbf{m}_{L_2}$ is a least squares solution to the weighted system of equations, it satisfies the corresponding normal equations. We can thus replace all occurrences of $\mathbf{G}_2^T\mathbf{d}_w$ with $\mathbf{G}_w^T\mathbf{G}_w\mathbf{m}_{L_2}$ using (2.3) to obtain

$$\chi^2(\mathbf{m}) - \chi^2(\mathbf{m}_{L_2}) = \mathbf{m}^T\mathbf{G}_w^T\mathbf{G}_w\mathbf{m} - \mathbf{m}_{L_2}^T\mathbf{G}_w^T\mathbf{G}_w\mathbf{m} - \mathbf{m}^T\mathbf{G}_w^T\mathbf{G}_w\mathbf{m}_{L_2}$$

$$- \mathbf{m}_{L_2}^T\mathbf{G}_w^T\mathbf{G}_w\mathbf{m}_{L_2} + \mathbf{m}_{L_2}^T\mathbf{G}_w^T\mathbf{G}_w\mathbf{m}_{L_2} + \mathbf{m}_{L_2}^T\mathbf{G}_w^T\mathbf{G}_w\mathbf{m}_{L_2}$$

$$= \mathbf{m}^T\mathbf{G}_w^T\mathbf{G}_w\mathbf{m} - \mathbf{m}_{L_2}^T\mathbf{G}_w^T\mathbf{G}_w\mathbf{m}$$

$$- \mathbf{m}^T\mathbf{G}_w^T\mathbf{G}_w\mathbf{m}_{L_2} + \mathbf{m}_{L_2}^T\mathbf{G}_w^T\mathbf{G}_w\mathbf{m}_{L_2}, \qquad (2.59)$$

and, finally,

$$\chi^2(\mathbf{m}) - \chi^2(\mathbf{m}_{L_2}) = (\mathbf{m} - \mathbf{m}_{L_2})^T\mathbf{G}_w^T\mathbf{G}_w(\mathbf{m} - \mathbf{m}_{L_2}). \qquad (2.60)$$

Thus our 95% confidence ellipsoid can also be written as

$$\chi^2(\mathbf{m}) - \chi^2(\mathbf{m}_{L_2}) \le \Delta^2 \qquad (2.61)$$

and the contour of the $\chi^2(\mathbf{m})$ function at $\chi^2(\mathbf{m}_{L_2}) + \Delta^2$ gives the boundary of the 95th percentile confidence ellipsoid.

2.4. UNKNOWN MEASUREMENT STANDARD DEVIATIONS

Suppose that we do not know the standard deviations of the measurement errors a priori. In this case, if we assume that the measurement errors are independent and normally distributed with expected value of zero and standard deviation σ, then we can perform the linear regression and estimate σ from the residuals.

First, we find the least squares solution to the unweighted problem $\mathbf{Gm} = \mathbf{d}$, and let

$$\mathbf{r} = \mathbf{d} - \mathbf{Gm}_{L_2}. \qquad (2.62)$$

To estimate the standard deviation from the residuals, let

$$s = \frac{||\mathbf{r}||_2}{\sqrt{\nu}}, \qquad (2.63)$$

where $\nu = m - n$ is the number of degrees of freedom [40].

As you might expect, there is a statistical cost associated with not knowing the true standard deviation. If the data standard deviations are known ahead of time, then each

$$m_i' = \frac{m_i - m_{\text{true}_i}}{\sqrt{\mathbf{C}_{i,i}}}, \tag{2.64}$$

where $\mathbf{C}$ is the covariance matrix (2.25), has a standard normal distribution. However, if instead of a known standard deviation we have an estimate, s, obtained using (2.63), then, if $\mathbf{C}'$ is given by the covariance matrix formula (2.25), but with $\sigma = s$, each

$$m_i' = \frac{m_i - m_{\text{true}_i}}{\sqrt{\mathbf{C}_{i,i}'}} \tag{2.65}$$

has a Student's t distribution (B.7) with $\nu = m - n$ degrees of freedom. For smaller numbers of degrees of freedom this produces appreciably broader confidence intervals than the standard normal distribution. As ν becomes large, (2.63) becomes an increasingly better estimate of σ as the two distributions converge. Confidence ellipsoids corresponding to this case can also be computed, but the formula is somewhat more complicated than in the case of known standard deviations [40].

In assessing goodness-of-fit in this case, a problem arises in that we can no longer apply the χ^2 test. Recall that the χ^2 test was based on the assumption that the data errors were normally distributed with known standard deviations σ_i. If the actual residuals were too large relative to the σ_i, then χ^2 would be large, and we would reject the linear regression fit based on a very small p-value. However, if we substitute (2.63) into (2.19), we find that $\chi^2_{\text{obs}} = \nu$, so such a model will always pass the χ^2 test.

●————————————————————————————

Example 2.3

Consider the analysis of a linear regression problem in which the measurement errors are assumed to be independent and normally distributed, with equal but unknown standard deviations, σ. We are given data y_i collected at points x_i (Figure 2.3) that appear to follow a linear relationship.

In this case, the system matrix for the forward problem is

$$\mathbf{G} = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ . & . \\ . & . \\ . & . \\ 1 & x_m \end{bmatrix}. \tag{2.66}$$

Figure 2.3 Data for Example 2.3, and corresponding linear regression line.

The least squares solution to

$$\mathbf{Gm} = \mathbf{y} \tag{2.67}$$

has

$$y_i = -1.03 + 10.09x_i. \tag{2.68}$$

Figure 2.3 shows the data and the linear regression line. Our estimate of the standard deviation of the measurement errors from (2.63) is $s = 30.74$. The estimated covariance matrix for the fitted parameters is

$$\mathbf{C}' = s^2(\mathbf{G}^T\mathbf{G})^{-1} = \begin{bmatrix} 338.24 & -4.93 \\ -4.93 & 0.08 \end{bmatrix}. \tag{2.69}$$

The parameter confidence intervals, evaluated for each parameter separately, are

$$m_1 = -1.03 \pm \sqrt{338.24}\, t_{m-2,0.975} = -1.03 \pm 38.05 \tag{2.70}$$

and

$$m_2 = 10.09 \pm \sqrt{0.08}\, t_{m-2,0.975} = 10.09 \pm 0.59. \tag{2.71}$$

Since the actual standard deviation of the measurement errors is unknown, we cannot perform a χ^2 test of goodness-of-fit. However, we can still examine the residuals. Figure 2.4 shows the residuals. It is clear that although they appear to be random, the standard deviation seems to increase as x and y increase. This is a common phenomenon
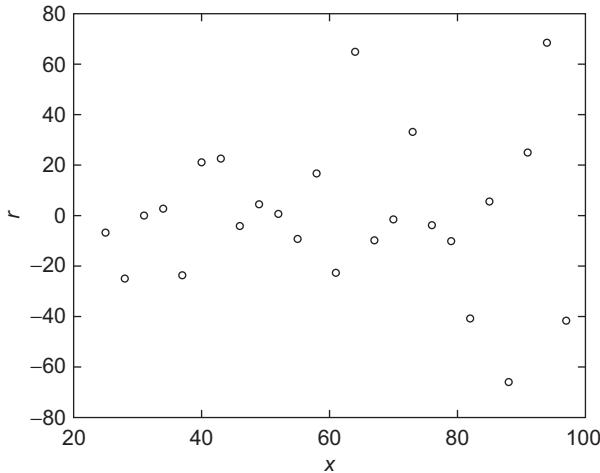
Figure 2.4 Unweighted residuals for Example 2.3.

in linear regression, called a **proportional effect**. One possible way that such an effect might occur is if the sizes of the measurement errors were proportional to the measurement magnitude due to characteristics of the instrument used.

For independent errors where the standard deviations of the data points are proportional to the observation, we can rescale the system of equations (2.67) by dividing each equation by y_i, to obtain

$$\mathbf{G}_w\mathbf{m} = \mathbf{y}_w. \tag{2.72}$$

If statistical assumptions are correct, (2.72) has a least squares solution that approximates (2.17). We obtain a revised least squares estimate of

$$y_i = -12.24 + 10.25x_i \tag{2.73}$$

with 95% parameter confidence intervals, evaluated as in (2.70) and (2.71), of

$$m_1 = -12.24 \pm 22.39 \tag{2.74}$$

and

$$m_2 = 10.25 \pm 0.47. \tag{2.75}$$

Figure 2.5 shows the data and least squares fit. Figure 2.6 shows the scaled residuals. Note that there is now no obvious trend in the magnitude of the residuals as x and y increase, as there was in Figure 2.4. The estimated standard deviation is 0.045, or 4.5% of the y value. In fact, these data were generated according to the true model
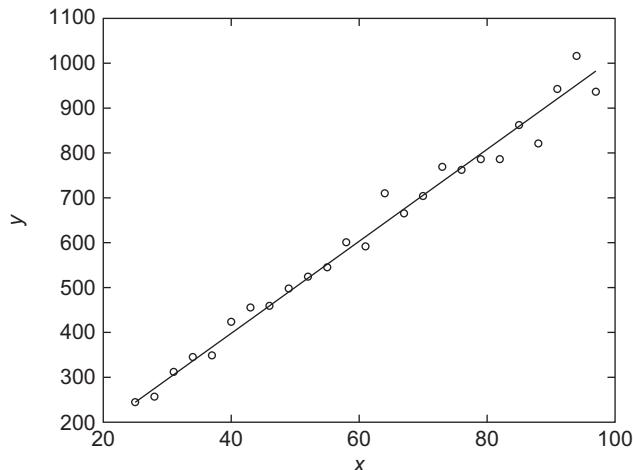
Figure 2.5 Data for Example 2.3, and corresponding linear regression line, weighted system.

Figure 2.6 Weighted residuals for Example 2.3.

$y_i = 10x_i + 0$, using standard deviations for the measurement errors that were 5% of the y value.

2.5. L_1 **REGRESSION**

Least squares solutions are highly susceptible to even small numbers of discordant observations, or **outliers**. Outliers are data points that are highly inconsistent with the other

data. Outliers may arise from procedural measurement error, for example from incorrectly recording the position of a decimal point in an observation or from instrumental glitches. Outliers should be investigated carefully, since the data may actually be showing us that the form of the mathematical model that we are trying to fit is incorrect. However, if we conclude that there are only a small number of outliers in the data due to incorrect measurements, we need to analyze the data in a way that minimizes their effect on the estimated model.

We can readily appreciate the strong effect of outliers on least squares solutions from a maximum likelihood perspective by noting the very rapid fall-off of the tails of the normal distribution. For example, the probability of a single data point drawn from a normal distribution being more than five standard deviations away from its expected value is less than 1 in 1 million:

$$P(|X - E[X]| \geq 5\sigma) = \frac{2}{\sqrt{2\pi}} \int_5^\infty e^{-\frac{1}{2}x^2} \, dx \approx 6 \times 10^{-7}. \tag{2.76}$$

If an outlier occurs in the data set due to a non–normal error process, the least squares solution will go to great lengths to accommodate it, and thus prevent its corresponding factor in the total likelihood product (2.9) from being vanishingly small.

As an alternative to least squares, consider the solution that minimizes the 1-norm of the residual vector,

$$\mu^{(1)} = \sum_{i=1}^m \frac{|d_i - (\mathbf{Gm})_i|}{\sigma_i} = \|\mathbf{d}_w - \mathbf{G}_w \mathbf{m}\|_1. \tag{2.77}$$

The **1–norm solution**, $\mathbf{m}_{L_1}$, will be more outlier resistant, or **robust**, than the least squares solution, $\mathbf{m}_{L_2}$, because (2.77) does not square each of the terms in the misfit measure, as (2.12) does. The 1-norm solution $\mathbf{m}_{L_1}$ also has a maximum likelihood interpretation; it is the maximum likelihood estimator for data with errors distributed according to a double-sided exponential distribution (Appendix B),

$$f(x) = \frac{1}{\sigma\sqrt{2}} e^{-\sqrt{2}|x-\mu|/\sigma}. \tag{2.78}$$

Data sets distributed as (2.78) are unusual. Nevertheless, it is often worthwhile to consider a solution where (2.77) is minimized rather than (2.12), even if most of the measurement errors are normally distributed, should there be reason to suspect the presence of outliers. This solution strategy may be useful if the data outliers occur for reasons that do not undercut our belief that the mathematical model is otherwise correct.

Example 2.4

We can demonstrate the advantages of 1-norm minimization using the quadratic regression example discussed earlier. Figure 2.7 shows the original sequence of independent data points with unit standard deviations, except one of the points (d_4) is now an outlier with respect to a mathematical model of the form (2.32). It is the original data point with 200 m subtracted from it. The least squares model for this data set is

$$\mathbf{m}_{L_2} = [26.4 \text{ m, } 75.6 \text{ m/s, } 4.86 \text{ m/s}^2]^T. \tag{2.79}$$

The least squares solution is skewed away from the majority of data points in trying to accommodate the outlier and is a poor estimate of the true model. We can also see that (2.79) fails to fit these data acceptably because of its huge χ^2 value (≈ 489). This is clearly astronomically out of bounds for a problem with 7 degrees of freedom, where the χ^2 value should not be far from 7. The corresponding p-value for $\chi^2 = 489$ is effectively zero.

The upper curve in Figure 2.7

$$\mathbf{m}_{L_1} = [17.6 \text{ m, } 96.4 \text{ m/s, } 9.31 \text{ m/s}^2]^T \tag{2.80}$$

is obtained using the 1-norm solution that minimizes (2.77). The data prediction from (2.80) faithfully fits the quadratic trend for the majority of the data points and is only slightly influenced by the outlier at $t = 4$. It is also much closer than (2.79) to the true model (2.33), and to the least squares model for the data set without the outlier (2.36).
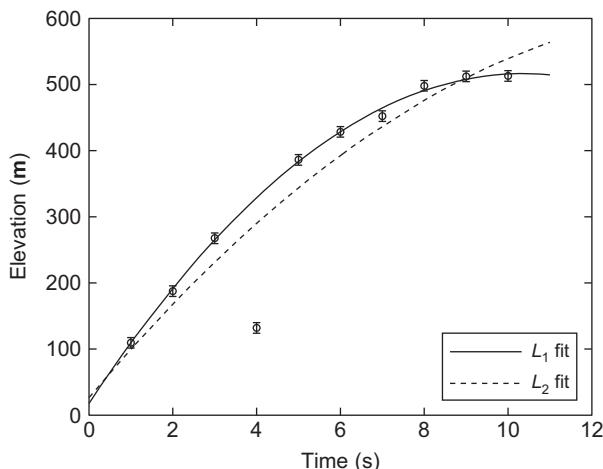
Figure 2.7 L_1 (solid) and L_2 (dashed) solutions for a parabolic data set with an outlier at $t = 4$ s.

In examining the differences between 2- and 1-norm models, it is instructive to consider the almost trivial regression problem of estimating the value of a single parameter from repeated measurements. The system of equations $\mathbf{Gm} = \mathbf{d}$ is

$$
\begin{bmatrix} 1 \\ 1 \\ 1 \\ . \\ . \\ . \\ 1 \end{bmatrix} \mathbf{m} = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ . \\ . \\ . \\ d_m \end{bmatrix}, \tag{2.81}
$$

where $\mathbf{m}$ is the 1 by 1 vector containing the parameter that we wish to estimate.

The least squares solution to (2.81) can be seen from the normal equations (A.73) to be simply the observational average

$$
\mathbf{m}_{L_2} = (\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T\mathbf{d} = m^{-1}\sum_{i=1}^{m} d_i. \tag{2.82}
$$

Finding the 1-norm solution is more complicated. The problem is that the 1-norm of the residual vector,

$$
f(\mathbf{m}) = \|\mathbf{d} - \mathbf{Gm}\|_1 = \sum_{i=1}^{m} |d_i - (\mathbf{Gm})_i|, \tag{2.83}
$$

is a nondifferentiable function of $\mathbf{m}$ at each point where $(\mathbf{Gm})_i = d_i$. The good news is that $f(\mathbf{m})$ is a convex function of $\mathbf{m}$. Thus any local minimum point is also a global minimum point. We can proceed by finding $f'(\mathbf{m})$ at those points where it is defined, and then separately consider the points at which the derivative is not defined. Every minimum point must either have $f'(\mathbf{m})$ undefined or $f'(\mathbf{m}) = 0$.

At those points where $f'(\mathbf{m})$ is defined, it is given by

$$
f'(\mathbf{m}) = \sum_{i=1}^{m} \text{sgn}(d_i - \mathbf{m}). \tag{2.84}
$$

where the **signum function**, $\text{sgn}(x)$, is -1 if its argument is negative, 1 if its argument is positive, and 0 if its argument is zero. The derivative (2.84) is zero when exactly half of the data are less than $\mathbf{m}$ and half of the data are greater than $\mathbf{m}$. Of course, this can only happen when the number of observations, m, is even. In this case, any value of $\mathbf{m}$ lying between the two middle observations is a 1-norm solution. When there are an odd number of data, the median data point is the unique 1-norm solution. Even an extreme outlier will not have a large effect on the median of an otherwise clustered set of observations. This illuminates the robustness of the 1-norm solution.

The general problem of finding solutions that minimize $\|\mathbf{d} - \mathbf{Gm}\|_1$ is a bit compli-cated. One practical method is **iteratively reweighted least squares**, or **IRLS** [138]. The IRLS algorithm solves a sequence of weighted least squares problems whose solu-tions converge to a 1-norm minimizing solution. Beginning with the residual vector

$$\mathbf{r} = \mathbf{d} - \mathbf{Gm}, \tag{2.85}$$

we want to minimize

$$f(\mathbf{m}) = \|\mathbf{r}\|_1 = \sum_{i=1}^{m} |r_i|. \tag{2.86}$$

The function in (2.86), like the function in (2.83), is nondifferentiable at any point where one of the elements of $\mathbf{r}$ is zero. Ignoring this issue for a moment, we can go ahead and compute the derivatives of f at other points:

$$\frac{\partial f(\mathbf{m})}{\partial m_k} = \sum_{i=1}^{m} \frac{\partial |r_i|}{\partial m_k} = -\sum_{i=1}^{m} G_{i,k} \, \mathrm{sgn}(r_i). \tag{2.87}$$

Writing $\mathrm{sgn}(r_i)$ as $r_i / |r_i|$ gives

$$\frac{\partial f(\mathbf{m})}{\partial m_k} = -\sum_{i=1}^{m} G_{i,k} \frac{1}{|r_i|} r_i. \tag{2.88}$$

The gradient of f is

$$\nabla f(\mathbf{m}) = -\mathbf{G}^T \mathbf{Rr} = -\mathbf{G}^T \mathbf{R}(\mathbf{d} - \mathbf{Gm}), \tag{2.89}$$

where $\mathbf{R}$ is a diagonal weighting matrix with diagonal elements that are the absolute values of the reciprocals of the residuals, so that

$$R_{i,i} = 1/|r_i|. \tag{2.90}$$

To find the 1-norm minimizing solution, we solve $\nabla f(\mathbf{m}) = \mathbf{0}$, which gives

$$\mathbf{G}^T \mathbf{R}(\mathbf{d} - \mathbf{Gm}) = \mathbf{0} \tag{2.91}$$

or

$$\mathbf{G}^T \mathbf{RGm} = \mathbf{G}^T \mathbf{Rd}. \tag{2.92}$$

Because $\mathbf{R}$ is a nonlinear function of $\mathbf{m}$, (2.92) is a nonlinear system of equations that we cannot solve directly. IRLS is an iterative algorithm to find the appropriate weights and solve the system. The algorithm begins with the least squares solution $\mathbf{m}^0 = \mathbf{m}_{L_2}$. We calculate the corresponding residual vector $\mathbf{r}^0 = \mathbf{d} - \mathbf{Gm}^0$ to construct the weighting matrix $\mathbf{R}$ using (2.90). We then solve (2.92) to obtain a new model $\mathbf{m}^1$ and

associated residual vector $\mathbf{r}^1$. The process is repeated until the model and residual vectors converge. A typical rule is to stop the iteration when

$$\frac{\|\mathbf{m}^{k+1} - \mathbf{m}^k\|_2}{1 + \|\mathbf{m}^{k+1}\|_2} < \tau \tag{2.93}$$

for some tolerance τ.

If any element of the residual vector becomes zero, then (2.90) becomes undefined. However, this problem can be easily addressed by selecting a tolerance ϵ below which we consider the residuals to be effectively zero. If $|r_i| < \epsilon$, then we set $R_{i,i} = 1/\epsilon$. With this modification it can be shown that this procedure will always converge to an approximate 1-norm minimizing solution.

As with the χ^2 misfit measure for least squares solutions, there is a corresponding p-value that can be evaluated under the assumption of normal data errors for the assessment of 1-norm solutions [127]. Let

$$\mu_{\mathrm{obs}}^{(1)} = \|\mathbf{Gm}_{L_1} - \mathbf{d}\|_1. \tag{2.94}$$

For an observed 1-norm misfit measure (2.94), the probability that a worse misfit could have occurred, given independent and normally distributed data and ν degrees of freedom, is approximately given by

$$p^{(1)}(x, \nu) = P\left(\mu^{(1)} > \mu_{\mathrm{obs}}^{(1)}\right) = 1 - S(x) + \frac{\gamma Z^{(2)}(x)}{6}, \tag{2.95}$$

where

$$S(x) = \frac{1}{\sigma_1 \sqrt{2\pi}} \int_{-\infty}^{x} e^{-\frac{\xi^2}{2\sigma_1^2}} d\xi \tag{2.96}$$

$$\sigma_1 = \sqrt{(1 - 2/\pi)\nu} \tag{2.97}$$

$$\gamma = \frac{2 - \pi/2}{(\pi/2 - 1)^{3/2} \nu^{\frac{1}{2}}} \tag{2.98}$$

$$Z^{(2)}(x) = \frac{x^2 - 1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} \tag{2.99}$$

$$x = \frac{\mu^{(1)} - \sqrt{2/\pi}\,\nu}{\sigma_1}. \tag{2.100}$$

2.6. MONTE CARLO ERROR PROPAGATION

For solution techniques that are nonlinear and/or algorithmic, such as IRLS, there is typically no simple way to propagate uncertainties in the data to uncertainties in the

estimated model parameters. In such cases, one can apply **Monte Carlo error prop-agation** techniques, in which we simulate a collection of noisy data vectors and then examine the statistics of the resulting ensemble of models.

For L_1 minimizing solutions, we can obtain an approximate covariance matrix by first forward-propagating the solution into an assumed noise-free baseline data vector

$$\mathbf{Gm}_{L_1} = \mathbf{d}_b. \tag{2.101}$$

We next re-solve the IRLS problem many times for independent data realizations, obtaining a suite of q 1-norm solutions to

$$\mathbf{Gm}_{L_1,i} = \mathbf{d}_b + \boldsymbol{\eta}_i \tag{2.102}$$

where $\boldsymbol{\eta}_i$ is the ith noise vector realization. If $\mathbf{A}$ is the q by n matrix where the ith row contains the difference between the ith model estimate and the average model,

$$\mathbf{A}_{i,\cdot} = \mathbf{m}_{L_1,i}^T - \bar{\mathbf{m}}_{L_1}^T, \tag{2.103}$$

then an empirical estimate of the covariance matrix is

$$\text{Cov}(\mathbf{m}_{L_1}) = \frac{\mathbf{A}^T \mathbf{A}}{q}. \tag{2.104}$$

Example 2.5

Recall Example 2.4. An estimate of $\text{Cov}(\mathbf{m}_{L_1})$ using 10,000 iterations of the Monte Carlo procedure is

$$\text{Cov}(\mathbf{m}_{L_1}) = \begin{bmatrix} 124.1 & -45.47 & -7.059 \\ -45.47 & 20.54 & 3.501 \\ -7.059 & 3.501 & 0.6316 \end{bmatrix}, \tag{2.105}$$

which contains elements that are about 1.4 times as large as those of the least squares solution (2.37). Unlike least squares solutions, model parameters obtained with the IRLS algorithm will not generally be normally distributed. However, we can compute approximate confidence intervals for the parameters from the covariance matrix diagonal, provided that analysis of the obtained Monte Carlo solution parameter distributions reveals that they are approximately normally distributed. Such an analysis can be performed by examining the parameter distributions with a **Q–Q** plot (Appendix B) and/or generating an ellipsoidal confidence boundary under normal assumptions and counting the proportion of points within the ellipsoid to check for consistency. In this example, Q–Q plots reveal the estimates to be approximately normally distributed, and calculating corresponding 95% confidence intervals from (2.105) using (2.30) gives

$$\mathbf{m}_{L_1} = [17.6 \pm 21.8 \text{ m}, \ 96.4 \pm 8.88 \text{ m/s}, \ 9.31 \pm 1.56 \text{ m/s}^2]^T. \tag{2.106}$$

2.7. EXERCISES

1. A seismic profiling experiment is performed where the first arrival times of seismic energy from a mid-crustal refractor are observed at distances (in kilometers) of

$$
\mathbf{x} = \begin{bmatrix} 6.0000 \\ 10.1333 \\ 14.2667 \\ 18.4000 \\ 22.5333 \\ 26.6667 \end{bmatrix}
\tag{2.107}
$$

from the source, and are found to be (in seconds after the source origin time)

$$
\mathbf{t} = \begin{bmatrix} 3.4935 \\ 4.2853 \\ 5.1374 \\ 5.8181 \\ 6.8632 \\ 8.1841 \end{bmatrix}.
\tag{2.108}
$$

These vectors can also be found in the MATLAB data file **profile.mat**. A two-layer, flat Earth structure gives the mathematical model

$$
t_i = t_0 + s_2 x_i,
\tag{2.109}
$$

where the intercept time, t_0, depends on the thickness and slowness of the upper layer, and s_2 is the slowness of the lower layer. The estimated noise in the first arrival time measurements is believed to be independent and normally distributed with expected value 0 and standard deviation $\sigma = 0.1$ s.

a. Find the least squares solution for the model parameters t_0 and s_2. Plot the data, the fitted model, and the residuals.

b. Calculate and comment on the model parameter correlation matrix (e.g., 2.43). How are the correlations manifested in the general appearance of the error ellipsoid in (t_0, s_2) space?

c. Plot the error ellipsoid in the (t_0, s_2) plane and calculate conservative 95% confidence intervals for t_0 and s_2 for the appropriate value of Δ^2. Hint: The following MATLAB function will plot a two-dimensional covariance ellipse about the model parameters, where C is the covariance matrix, DELTA2 is Δ^2, and m is the 2-vector of model parameters.

```
%set the number of points on the ellipse to generate and plot
function plot_ellipse(DELTA2,C,m)
n=100;
%construct a vector of n equally-spaced angles from (0,2*pi)
```

```
theta=linspace(0,2*pi,n)';
%corresponding unit vector
xhat=[cos(theta),sin(theta)];
Cinv=inv(C);
%preallocate output array
r=zeros(n,2);
for i=1:n
%store each (x,y) pair on the confidence ellipse
%in the corresponding row of r
r(i,:)=sqrt(DELTA2/(xhat(i,:)*Cinv*xhat(i,:)'))*xhat(i,:);
end
plot(m(1)+r(:,1), m(2)+r(:,2));
axis equal
```

  d. Evaluate the *p*-value for this model. You may find the library function **chi2cdf**
     to be useful here.
  e. Evaluate the value of χ^2 for 1000 Monte Carlo simulations using the data pre-
     diction from your model perturbed by noise that is consistent with the data
     assumptions. Compare a histogram of these χ^2 values with the theoretical χ^2
     distribution for the correct number of degrees of freedom. You may find the
     library function **chi2pdf** to be useful here.
  f. Are your *p*-value and Monte Carlo χ^2 distribution consistent with the theoretical
     modeling and the data set? If not, explain what is wrong.
  g. Use IRLS to find 1–norm estimates for t_0 and s_2. Plot the data predictions from
     your model relative to the true data and compare with (a).
  h. Use Monte Carlo error propagation and IRLS to estimate symmetric 95%
     confidence intervals on the 1–norm solution for t_0 and s_2.
  i. Examining the contributions from each of the data points to the 1–norm misfit
     measure, can you make a case that any of the data points are statistical outliers?

2. In this chapter we have largely assumed that the data errors are independent. Suppose
   instead that the data errors have an MVN distribution with expected value **0** and a
   covariance matrix $\mathbf{C}_D$. It can be shown that the likelihood function is then

$$L(\mathbf{m}|\mathbf{d}) = \frac{1}{(2\pi)^{m/2}} \frac{1}{\sqrt{\det(\mathbf{C}_D)}} e^{-(\mathbf{Gm}-\mathbf{d})^T \mathbf{C}_D^{-1}(\mathbf{Gm}-\mathbf{d})/2}. \qquad (2.110)$$

  a. Show that the maximum likelihood estimate can be obtained by solving the
     minimization problem,

$$\min(\mathbf{Gm} - \mathbf{d})^T \mathbf{C}_D^{-1}(\mathbf{Gm} - \mathbf{d}). \qquad (2.111)$$

  b. Show that (2.111) can be solved using the system of equations

$$\mathbf{G}^T \mathbf{C}_D^{-1} \mathbf{Gm} = \mathbf{G}^T \mathbf{C}_D^{-1} \mathbf{d}. \qquad (2.112)$$

   c. Show that (2.111) is equivalent to the linear least squares problem

$$\min \|\mathbf{C}_D^{-1/2}\mathbf{Gm} - \mathbf{C}_D^{-1/2}\mathbf{d}\|_2, \qquad (2.113)$$

where $\mathbf{C}_D^{-1/2}$ is the matrix square root of $\mathbf{C}_D^{-1}$.

   d. The Cholesky factorization of $\mathbf{C}_D^{-1}$ can also be used instead of the matrix square root. Show that (2.111) is equivalent to the linear least squares problem

$$\min \|\mathbf{RGm} - \mathbf{Rd}\|_2, \qquad (2.114)$$

where $\mathbf{R}$ is the Cholesky factor of $\mathbf{C}_D^{-1}$.

3. Use MATLAB to generate 10,000 realizations of a data set of $m = 5$ points $\mathbf{d} = a + b\mathbf{x} + \boldsymbol{\eta}$, where $\mathbf{x} = [1, 2, 3, 4, 5]^T$, the $n = 2$ true model parameters are $a = b = 1$, and $\boldsymbol{\eta}$ is an m-element vector of independent $N(0, 1)$ noise.

   a. Assuming that the noise standard deviation is known *a priori* to be 1, solve for the parameters a and b using least squares for each realization and histogram them in 100 bins.

   b. Calculate the parameter covariance matrix, $\mathbf{C} = \sigma^2(\mathbf{G}^T\mathbf{G})^{-1}$, assuming independent $N(0, 1)$ data errors, and give standard deviations, σ_a and σ_b, for your estimates of a and b estimated from $\mathbf{C}$.

   c. Calculate standardized parameter estimates

$$a' = \frac{a - 1}{\sqrt{C_{1,1}}} \qquad (2.115)$$

and

$$b' = \frac{b - 1}{\sqrt{C_{2,2}}} \qquad (2.116)$$

for your solutions for a and b. Demonstrate using a Q–Q plot (Appendix B) that your estimates for a' and b' are distributed as $N(0, 1)$.

   d. Show, using a Q–Q plot, that the squared residual lengths

$$\|\mathbf{r}\|_2^2 = \|\mathbf{d} - \mathbf{Gm}\|_2^2 \qquad (2.117)$$

for your solutions in (a) are distributed as χ^2 with $m - n = \nu = 3$ degrees of freedom.

   e. Assume that the noise standard deviation for the synthetic data set is not known, and instead estimate it for each realization, k, as

$$s_k = \sqrt{\frac{1}{n - m} \sum_{i=1}^{m} r_i^2}. \qquad (2.118)$$

Histogram your standardized solutions

$$a' = \frac{a - \bar{a}}{\sqrt{C'_{1,1}}} \tag{2.119}$$

and

$$b' = \frac{b - \bar{b}}{\sqrt{C'_{2,2}}}, \tag{2.120}$$

where $\mathbf{C}' = s_k^2 (\mathbf{G}^T \mathbf{G})^{-1}$ is the covariance matrix estimation for the kth realization.

f. Demonstrate using a Q–Q plot that your estimates for a' and b' are distributed as the Student's t distribution with $\nu = 3$ degrees of freedom.

4. Suppose that we analyze a large number of data sets $\mathbf{d}$ in a linear regression problem and compute p-values for each data set. The χ^2_{obs} values should be distributed according to a χ^2 distribution with $m - n$ degrees of freedom. Show that the corresponding p-values will be uniformly distributed between 0 and 1.

5. Use linear regression to fit a polynomial of the form

$$y_i = a_0 + a_1 x_i + a_2 x_i^2 + \ldots + a_{19} x_i^{19} \tag{2.121}$$

to the noise-free data points

$$(x_i, \ y_i) = (-0.95, \ -0.95), \ (-0.85, \ -0.85), \ \ldots, \ (0.95, \ 0.95). \tag{2.122}$$

Use the normal equations to solve the least squares problem.

Plot the data and your fitted model, and list the parameters, a_i, obtained in your regression. Clearly, the correct solution has $a_1 = 1$, and all other $a_i = 0$. Explain why your answer differs.

2.8. NOTES AND FURTHER READING

Linear regression is a major subfield within statistics, and there are literally hundreds of associated textbooks. Many of these references focus on applications of linear regression in the social sciences. In such applications, the primary focus is often on determining which variables have an effect on response variables of interest (rather than on estimating parameter values for a predetermined model). In this context it is important to compare alternative regression models and to test the hypothesis that a predictor variable has a nonzero coefficient in the regression model. Since we normally know which predictor variables are important in the physical sciences, the approach commonly differs. Useful linear regression references from the standpoint of estimating parameters in the context considered here include [40, 113].

Robust statistical methods are an important topic. Huber discusses a variety of robust statistical procedures [77]. The computational problem of computing a 1–norm solution has been extensively researched. Techniques for 1-norm minimization include methods based on the simplex method for linear programming, interior point methods, and iteratively reweighted least squares [6, 32, 130, 138]. The IRLS method is the simplest to implement, but interior point methods can be the most efficient approaches for very large problems. Watson reviews the history of methods for finding p-norm solutions including the 1-norm case [171].

We have assumed that **G** is known exactly. In some cases elements of this matrix might be subject to measurement error. This problem has been studied as the **total least squares problem** [78]. An alternative approach to least squares problems with uncertainties in **G** that has recently received considerable attention is called **robust least squares** [11, 43].

This page intentionally left blank

Rank Deficiency and Ill-Conditioning

Synopsis

The characteristics of rank-deficient and ill-conditioned linear systems of equations are explored using the singular value decomposition. The connection between model and data null spaces and solution uniqueness and ability to fit data is examined. Model and data resolution matrices are defined. The relationship between singular value size and singular vector roughness and its connection to the effect of noise on solutions are discussed in the context of the fundamental trade-off between model resolution and instability. Specific manifestations of these issues in rank-deficient and ill-conditioned discrete problems are shown in several examples.

3.1. THE SVD AND THE GENERALIZED INVERSE

A method of analyzing and solving least squares problems that is of particular interest in ill-conditioned and/or rank-deficient systems is the **singular value decomposition**, or **SVD**. In the SVD [53, 93, 152], an m by n matrix $\mathbf{G}$ is factored into

$$\mathbf{G} = \mathbf{U}\mathbf{S}\mathbf{V}^T, \tag{3.1}$$

where

- $\mathbf{U}$ is an m by m orthogonal matrix with columns that are unit basis vectors spanning the **data space**, R^m.
- $\mathbf{V}$ is an n by n orthogonal matrix with columns that are basis vectors spanning the **model space**, R^n.
- $\mathbf{S}$ is an m by n diagonal matrix with diagonal elements called **singular values**.

The SVD matrices can be computed in MATLAB with the **svd** command. It can be shown that every matrix has a singular value decomposition [53].

The singular values along the diagonal of $\mathbf{S}$ are customarily arranged in decreasing size, $s_1 \geq s_2 \geq \ldots \geq s_{\min(m,n)} \geq 0$. Note that some of the singular values may be zero. If only the first p singular values are nonzero, we can partition $\mathbf{S}$ as

$$\mathbf{S} = \begin{bmatrix} \mathbf{S}_p & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{3.2}$$

where $\mathbf{S}_p$ is a p by p diagonal matrix composed of the positive singular values. Expanding the SVD representation of $\mathbf{G}$ in terms of the columns of $\mathbf{U}$ and $\mathbf{V}$ gives

$$\mathbf{G} = \begin{bmatrix} \mathbf{U}_{\cdot,1}, \ \mathbf{U}_{\cdot,2}, \ \ldots, \ \mathbf{U}_{\cdot,m} \end{bmatrix} \begin{bmatrix} \mathbf{S}_p & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{V}_{\cdot,1}, \ \mathbf{V}_{\cdot,2}, \ \ldots, \ \mathbf{V}_{\cdot,n} \end{bmatrix}^T \tag{3.3}$$

$$= \begin{bmatrix} \mathbf{U}_p, \ \mathbf{U}_0 \end{bmatrix} \begin{bmatrix} \mathbf{S}_p & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{V}_p, \ \mathbf{V}_0 \end{bmatrix}^T \tag{3.4}$$

where $\mathbf{U}_p$ denotes the first p columns of $\mathbf{U}$, $\mathbf{U}_0$ denotes the last $m - p$ columns of $\mathbf{U}$, $\mathbf{V}_p$ denotes the first p columns of $\mathbf{V}$, and $\mathbf{V}_0$ denotes the last $n - p$ columns of $\mathbf{V}$. Because the last $m - p$ columns of $\mathbf{U}$ and the last $n - p$ columns of $\mathbf{V}$ in (3.4) are multiplied by zeros in $\mathbf{S}$, we can simplify the SVD of $\mathbf{G}$ into its **compact form**

$$\mathbf{G} = \mathbf{U}_p \mathbf{S}_p \mathbf{V}_p^T. \tag{3.5}$$

For any vector $\mathbf{y}$ in the range of $\mathbf{G}$, applying (3.5) gives

$$\mathbf{y} = \mathbf{G}\mathbf{x} \tag{3.6}$$

$$= \mathbf{U}_p \left(\mathbf{S}_p \mathbf{V}_p^T \mathbf{x} \right). \tag{3.7}$$

Thus every vector in $R(\mathbf{G})$ can be written as $\mathbf{y} = \mathbf{U}_p \mathbf{z}$ where $\mathbf{z} = \mathbf{S}_p \mathbf{V}_p^T \mathbf{x}$. Writing out this matrix–vector multiplication, we see that any vector $\mathbf{y}$ in $R(\mathbf{G})$ can be written as a linear combination of the columns of $\mathbf{U}_p$

$$\mathbf{y} = \sum_{i=1}^{p} z_i \mathbf{U}_{\cdot,i}. \tag{3.8}$$

The columns of $\mathbf{U}_p$ span $R(\mathbf{G})$, are linearly independent, and form an orthonormal basis for $R(\mathbf{G})$. Because this orthonormal basis has p vectors, $\mathrm{rank}(\mathbf{G}) = p$.

Since $\mathbf{U}$ is an orthogonal matrix, the columns of $\mathbf{U}$ form an orthonormal basis for R^m. By Theorem A.5, $N(\mathbf{G}^T) + R(\mathbf{G}) = R^m$, so the remaining $m - p$ columns of $\mathbf{U}_0$ form an orthonormal basis for the null space of $\mathbf{G}^T$. Note that because the null space basis is nonunique, and because basis vectors contain an inherent sign ambiguity, basis vectors calculated and illustrated in this chapter and elsewhere may not match ones calculated locally using the provided MATLAB code. We will sometimes refer to $N(\mathbf{G}^T)$ as the **data null space**. Similarly, because $\mathbf{G}^T = \mathbf{V}_p \mathbf{S}_p \mathbf{U}_p^T$, the columns of $\mathbf{V}_p$ form an orthonormal basis for $R(\mathbf{G}^T)$ and the columns of $\mathbf{V}_0$ form an orthonormal basis for $N(\mathbf{G})$. We will sometimes refer to $N(\mathbf{G})$ as the **model null space**.

Two other important SVD properties are similar to properties of eigenvalues and eigenvectors. See Section A.6. Because the columns of $\mathbf{V}$ are orthonormal,

$$\mathbf{V}^T \mathbf{V}_{\cdot,i} = \mathbf{e}_i. \tag{3.9}$$

Thus

$$\mathbf{G}\mathbf{V}_{\cdot,i} = \mathbf{U}\mathbf{S}\mathbf{V}^T\mathbf{V}_{\cdot,i} \tag{3.10}$$

$$= \mathbf{U}\mathbf{S}\mathbf{e}_i \tag{3.11}$$

$$= s_i\mathbf{U}_{\cdot,i}, \tag{3.12}$$

and, similarly,

$$\mathbf{G}^T\mathbf{U}_{\cdot,i} = \mathbf{V}\mathbf{S}^T\mathbf{U}^T\mathbf{U}_{\cdot,i} \tag{3.13}$$

$$= \mathbf{V}\mathbf{S}^T\mathbf{e}_i \tag{3.14}$$

$$= s_i\mathbf{V}_{\cdot,i}. \tag{3.15}$$

There is a connection between the singular values of $\mathbf{G}$ and the eigenvalues of $\mathbf{G}\mathbf{G}^T$ and $\mathbf{G}^T\mathbf{G}$.

$$\mathbf{G}\mathbf{G}^T\mathbf{U}_{\cdot,i} = \mathbf{G}s_i\mathbf{V}_{\cdot,i} \tag{3.16}$$

$$= s_i\mathbf{G}\mathbf{V}_{\cdot,i} \tag{3.17}$$

$$= s_i^2\mathbf{U}_{\cdot,i}. \tag{3.18}$$

Similarly,

$$\mathbf{G}^T\mathbf{G}\mathbf{V}_{\cdot,i} = s_i^2\mathbf{V}_{\cdot,i}. \tag{3.19}$$

These relations show that we could, in theory, compute the SVD by finding the eigenvalues and eigenvectors of $\mathbf{G}^T\mathbf{G}$ and $\mathbf{G}\mathbf{G}^T$. In practice, more efficient specialized algorithms are used [38, 53, 164].

The SVD can be used to compute a generalized inverse of $\mathbf{G}$, called the **Moore–Penrose pseudoinverse**, because it has desirable inverse properties originally identified by Moore and Penrose [110, 129]. The generalized inverse is

$$\mathbf{G}^\dagger = \mathbf{V}_p\mathbf{S}_p^{-1}\mathbf{U}_p^T. \tag{3.20}$$

MATLAB has a **pinv** command that generates $\mathbf{G}^\dagger$. This command allows the user to select a tolerance such that singular values smaller than the tolerance are not included in the computation.

Using (3.20), we define the pseudoinverse solution to be

$$\mathbf{m}_\dagger = \mathbf{G}^\dagger\mathbf{d} \tag{3.21}$$

$$= \mathbf{V}_p\mathbf{S}_p^{-1}\mathbf{U}_p^T\mathbf{d}. \tag{3.22}$$

Among the desirable properties of (3.22) is that $\mathbf{G}^\dagger$, and hence $\mathbf{m}_\dagger$, always exist. In contrast, the inverse of $\mathbf{G}^T\mathbf{G}$ that appears in the normal equations (2.3) does not exist

when $\mathbf{G}$ is not of full column rank. We will shortly show that $\mathbf{m}_\dagger$ is a least squares solution.

To encapsulate what the SVD tells us about our linear system, $\mathbf{G}$, and the corresponding generalized inverse system $\mathbf{G}^\dagger$, consider four cases:

1. $m = n = p$. Both the model and data null spaces, $N(\mathbf{G})$ and $N(\mathbf{G}^T)$, respectively, are trivial. $\mathbf{U}_p = \mathbf{U}$ and $\mathbf{V}_p = \mathbf{V}$ are square orthogonal matrices, so that $\mathbf{U}_p^T = \mathbf{U}_p^{-1}$, and $\mathbf{V}_p^T = \mathbf{V}_p^{-1}$. Equation (3.22) gives

$$\mathbf{G}^\dagger = \mathbf{V}_p \mathbf{S}_p^{-1} \mathbf{U}_p^T \tag{3.23}$$

$$= (\mathbf{U}_p \mathbf{S}_p \mathbf{V}_p^T)^{-1} \tag{3.24}$$

$$= \mathbf{G}^{-1} \tag{3.25}$$

which is the matrix inverse for a square full rank matrix. The solution is unique, and the data are fit exactly.

2. $m = p$ and $p < n$. $N(\mathbf{G})$ is nontrivial because $p < n$, but $N(\mathbf{G}^T)$ is trivial. $\mathbf{U}_p^T = \mathbf{U}_p^{-1}$ and $\mathbf{V}_p^T \mathbf{V}_p = \mathbf{I}_p$. $\mathbf{G}$ applied to the generalized inverse solution gives

$$\mathbf{G}\mathbf{m}_\dagger = \mathbf{G}\mathbf{G}^\dagger \mathbf{d} \tag{3.26}$$

$$= \mathbf{U}_p \mathbf{S}_p \mathbf{V}_p^T \mathbf{V}_p \mathbf{S}_p^{-1} \mathbf{U}_p^T \mathbf{d} \tag{3.27}$$

$$= \mathbf{U}_p \mathbf{S}_p \mathbf{I}_p \mathbf{S}_p^{-1} \mathbf{U}_p^T \mathbf{d} \tag{3.28}$$

$$= \mathbf{d}. \tag{3.29}$$

The data are fit exactly but the solution is nonunique because of the existence of the nontrivial model null space $N(\mathbf{G})$.

If $\mathbf{m}$ is any least squares solution, then it satisfies the normal equations. This is shown in Exercise C.5.

$$(\mathbf{G}^T \mathbf{G})\mathbf{m} = \mathbf{G}^T \mathbf{d}. \tag{3.30}$$

Since $\mathbf{m}_\dagger$ is a least squares solution, it also satisfies the normal equations.

$$(\mathbf{G}^T \mathbf{G})\mathbf{m}_\dagger = \mathbf{G}^T \mathbf{d}. \tag{3.31}$$

Subtracting (3.30) from (3.31), we find that

$$(\mathbf{G}^T \mathbf{G})(\mathbf{m}_\dagger - \mathbf{m}) = \mathbf{0}. \tag{3.32}$$

Thus $\mathbf{m}_\dagger - \mathbf{m}$ lies in $N(\mathbf{G}^T \mathbf{G})$. It can be shown (see Exercise 13f in Appendix A) that $N(\mathbf{G}^T \mathbf{G}) = N(\mathbf{G})$. This implies that $\mathbf{m}_\dagger - \mathbf{m}$ lies in $N(\mathbf{G})$.

The general solution is thus the sum of $\mathbf{m}_\dagger$ and an arbitrary model null space vector, $\mathbf{m}_0$, that can be written as a linear combination of a set of basis vectors for

$N(\mathbf{G})$. In terms of the columns of $\mathbf{V}$, we can thus write

$$\mathbf{m} = \mathbf{m}_\dagger + \mathbf{m}_0$$

$$= \mathbf{m}_\dagger + \sum_{i=p+1}^{n} \alpha_i \mathbf{V}_{\cdot,i} \tag{3.33}$$

for any coefficients, α_i. Because the columns of $\mathbf{V}$ are orthonormal, the square of the 2-norm of a general solution always equals or exceeds that of $\mathbf{m}_\dagger$

$$\|\mathbf{m}\|_2^2 = \|\mathbf{m}_\dagger\|_2^2 + \sum_{i=p+1}^{n} \alpha_i^2 \geq \|\mathbf{m}_\dagger\|_2^2 \tag{3.34}$$

where we have equality only if all of the α_i are zero. The generalized inverse solution is thus a **minimum length least squares solution**.

We can also write this solution in terms of $\mathbf{G}$ and $\mathbf{G}^T$:

$$\mathbf{m}_\dagger = \mathbf{V}_p \mathbf{S}_p^{-1} \mathbf{U}_p^T \mathbf{d} \tag{3.35}$$

$$= \mathbf{V}_p \mathbf{S}_p \mathbf{U}_p^T \mathbf{U}_p \mathbf{S}_p^{-2} \mathbf{U}_p^T \mathbf{d} \tag{3.36}$$

$$= \mathbf{G}^T (\mathbf{U}_p \mathbf{S}_p^{-2} \mathbf{U}_p^T) \mathbf{d} \tag{3.37}$$

$$= \mathbf{G}^T (\mathbf{G}\mathbf{G}^T)^{-1} \mathbf{d}. \tag{3.38}$$

In practice it is better to compute a solution using the SVD than to use (3.38) because of numerical accuracy issues.

3. $n = p$ and $p < m$. $N(\mathbf{G})$ is trivial but $N(\mathbf{G}^T)$ is nontrivial. Because $p < m$, $R(\mathbf{G})$ is a subspace of R^m. Here

$$\mathbf{G}\mathbf{m}_\dagger = \mathbf{U}_p \mathbf{S}_p \mathbf{V}_p^T \mathbf{V}_p \mathbf{S}_p^{-1} \mathbf{U}_p^T \mathbf{d} \tag{3.39}$$

$$= \mathbf{U}_p \mathbf{U}_p^T \mathbf{d}. \tag{3.40}$$

The product $\mathbf{U}_p \mathbf{U}_p^T \mathbf{d}$ gives the projection of $\mathbf{d}$ onto $R(\mathbf{G})$. Thus $\mathbf{G}\mathbf{m}_\dagger$ is the point in $R(\mathbf{G})$ that is closest to $\mathbf{d}$, and $\mathbf{m}_\dagger$ is a least squares solution to $\mathbf{G}\mathbf{m} = \mathbf{d}$. Only if $\mathbf{d}$ is actually in $R(\mathbf{G})$ will $\mathbf{m}_\dagger$ be an exact solution to $\mathbf{G}\mathbf{m} = \mathbf{d}$.

We can see that this solution is exactly that obtained from the normal equations because

$$(\mathbf{G}^T \mathbf{G})^{-1} = (\mathbf{V}_p \mathbf{S}_p \mathbf{U}_p^T \mathbf{U}_p \mathbf{S}_p \mathbf{V}_p^T)^{-1} \tag{3.41}$$

$$= (\mathbf{V}_p \mathbf{S}_p^2 \mathbf{V}_p^T)^{-1} \tag{3.42}$$

$$= \mathbf{V}_p \mathbf{S}_p^{-2} \mathbf{V}_p^T \tag{3.43}$$

and

$$\mathbf{m}_\dagger = \mathbf{G}^\dagger \mathbf{d} \tag{3.44}$$

$$= \mathbf{V}_p \mathbf{S}_p^{-1} \mathbf{U}_p^T \mathbf{d} \tag{3.45}$$

$$= \mathbf{V}_p \mathbf{S}_p^{-2} \mathbf{V}_p^T \mathbf{V}_p \mathbf{S}_p \mathbf{U}_p^T \mathbf{d} \tag{3.46}$$

$$= (\mathbf{G}^T \mathbf{G})^{-1} \mathbf{G}^T \mathbf{d}. \tag{3.47}$$

This solution is unique, but cannot fit general data exactly. As with (3.38), it is better in practice to use the generalized inverse solution than to use (3.47) because of numerical accuracy issues.

4. $p < m$ and $p < n$. Both $N(\mathbf{G}^T)$ and $N(\mathbf{G})$ are nontrivial. In this case, the generalized inverse solution encapsulates the behavior of both of the two previous cases, minimizing both $\|\mathbf{Gm} - \mathbf{d}\|_2$ and $\|\mathbf{m}\|_2$.

As in case 3,

$$\mathbf{Gm}_\dagger = \mathbf{U}_p \mathbf{S}_p \mathbf{V}_p^T \mathbf{V}_p \mathbf{S}_p^{-1} \mathbf{U}_p^T \mathbf{d} \tag{3.48}$$

$$= \mathbf{U}_p \mathbf{U}_p^T \mathbf{d} \tag{3.49}$$

$$= \text{proj}_{R(\mathbf{G})} \mathbf{d}. \tag{3.50}$$

Thus $\mathbf{m}_\dagger$ is a least squares solution to $\mathbf{Gm} = \mathbf{d}$.

As in case 2 we can write the model and its norm using (3.33) and (3.34). Thus $\mathbf{m}_\dagger$ is the least squares solution of minimum length.

We have shown that the generalized inverse provides an inverse solution (3.22) that always exists, is both least squares and minimum length, and properly accommodates the rank and dimensions of $\mathbf{G}$. Relationships between the subspaces $R(\mathbf{G})$, $N(\mathbf{G}^T)$, $R(\mathbf{G}^T)$, $N(\mathbf{G})$, and the operators $\mathbf{G}$ and $\mathbf{G}^\dagger$, are schematically depicted in Figure 3.1. Table 3.1 summarizes the SVD and its properties.

The existence of a nontrivial model null space (one that includes more than just the zero vector) is at the heart of solution nonuniqueness. There are an infinite number of solutions that will fit the data equally well, because model components in $N(\mathbf{G})$ have
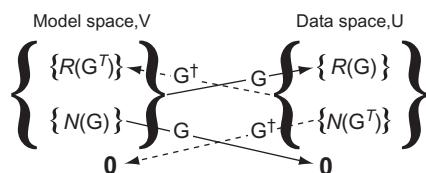

Figure 3.1 SVD model and data space mappings, where $\mathbf{G}^\dagger$ is the generalized inverse. $N(\mathbf{G}^T)$ and $N(\mathbf{G})$ are the data and model null spaces, respectively.

Table 3.1 Summary of the SVD and its associated scalars and matrices.

Object	Size	Properties
p	scalar	$\mathrm{rank}(\mathbf{G}) = p$; Number of nonzero singular values
m	scalar	Dimension of the data space
n	scalar	Dimension of the model space
$\mathbf{G}$	m by n	Forward problem matrix; $\mathbf{G} = \mathbf{U}\mathbf{S}\mathbf{V}^T = \mathbf{U}_p\mathbf{S}_p\mathbf{V}_p^T$
$\mathbf{U}$	m by m	Orthogonal matrix; $\mathbf{U} = [\mathbf{U}_p, \mathbf{U}_0]$
s_i	scalar	ith singular value
$\mathbf{S}$	m by n	Diagonal matrix of singular values; $\mathbf{S}_{i,i} = s_i$
$\mathbf{V}$	n by n	Orthogonal matrix; $\mathbf{V} = [\mathbf{V}_p, \mathbf{V}_0]$
$\mathbf{U}_p$	m by p	Columns form a basis for $R(\mathbf{G})$
$\mathbf{S}_p$	p by p	Diagonal matrix of nonzero singular values
$\mathbf{V}_p$	n by p	Columns form an orthonormal basis for $R(\mathbf{G}^T)$
$\mathbf{U}_0$	m by $m-p$	Columns form an orthonormal basis for $N(\mathbf{G}^T)$
$\mathbf{V}_0$	n by $n-p$	Columns form an orthonormal basis for $N(\mathbf{G})$
$\mathbf{U}_{\cdot,i}$	m by 1	Eigenvector of $\mathbf{G}\mathbf{G}^T$ with eigenvalue s_i^2
$\mathbf{V}_{\cdot,i}$	n by 1	Eigenvector of $\mathbf{G}^T\mathbf{G}$ with eigenvalue s_i^2
$\mathbf{G}^\dagger$	n by m	Pseudoinverse of $\mathbf{G}$; $\mathbf{G}^\dagger = \mathbf{V}_p\mathbf{S}_p^{-1}\mathbf{U}_p^T$
$\mathbf{m}_\dagger$	m by 1	Generalized inverse solution; $\mathbf{m}_\dagger = \mathbf{G}^\dagger\mathbf{d}$

no effect on data fit. To select a particular preferred solution from this infinite set thus requires more constraints (such as minimum length or smoothing constraints) than are encoded in the matrix $\mathbf{G}$.

To see the significance of the $N(\mathbf{G}^T)$ subspace, consider an arbitrary data vector, $\mathbf{d}_0$, which lies in $N(\mathbf{G}^T)$:

$$\mathbf{d}_0 = \sum_{i=p+1}^{m} \beta_i \mathbf{U}_{\cdot,i}. \tag{3.51}$$

The generalized inverse operating on such a data vector gives

$$\mathbf{m}_\dagger = \mathbf{V}_p\mathbf{S}_p^{-1}\mathbf{U}_p^T\mathbf{d}_0 \tag{3.52}$$

$$= \mathbf{V}_p\mathbf{S}_p^{-1}\sum_{i=p+1}^{n} \beta_i \mathbf{U}_p^T\mathbf{U}_{\cdot,i} = \mathbf{0} \tag{3.53}$$

because the $\mathbf{U}_{\cdot,i}$ are orthogonal. $N(\mathbf{G}^T)$ is a subspace of R^m consisting of all vectors $\mathbf{d}_0$ that have no influence on the generalized inverse model, $\mathbf{m}_\dagger$. If $p < n$, there are an infinite number of potential data sets that will produce the same model when (3.22) is applied.

3.2. COVARIANCE AND RESOLUTION OF THE GENERALIZED INVERSE SOLUTION

The generalized inverse always gives us a solution, $\mathbf{m}_\dagger$, with well–determined properties, but it is essential to investigate how faithful a representation any model is likely to be of the true situation.

In Chapter 2, we found that under the assumption of independent and normally distributed measurement errors with constant standard deviation, the least squares solution was an unbiased estimator of the true model, and that the estimated model parameters had a multivariate normal distribution with covariance

$$\text{Cov}(\mathbf{m}_{L_2}) = \sigma^2 (\mathbf{G}^T \mathbf{G})^{-1}. \tag{3.54}$$

We can attempt the same analysis for the generalized inverse solution $\mathbf{m}_\dagger$. The covariance matrix would be given by

$$\text{Cov}(\mathbf{m}_\dagger) = \mathbf{G}^\dagger \text{Cov}(\mathbf{d})(\mathbf{G}^\dagger)^T \tag{3.55}$$

$$= \sigma^2 \mathbf{G}^\dagger (\mathbf{G}^\dagger)^T \tag{3.56}$$

$$= \sigma^2 \mathbf{V}_p \mathbf{S}_p^{-2} \mathbf{V}_p^T \tag{3.57}$$

$$= \sigma^2 \sum_{i=1}^{p} \frac{V_{\cdot,i} V_{\cdot,i}^T}{s_i^2}. \tag{3.58}$$

Since the s_i are decreasing, successive terms in this sum make larger and larger contributions to the covariance. If we were to truncate (3.58), we could actually decrease the variance in our model estimate! This is discussed further in Section 3.3.

Unfortunately, unless $p = n$, the generalized inverse solution is *not* an unbiased estimator of the true solution. This occurs because the true solution may have nonzero projections onto those basis vectors in $\mathbf{V}$ that are unused in the generalized inverse solution. In practice, the bias introduced by restricting the solution to the subspace spanned by the columns of $\mathbf{V}_p$ may be far larger than the uncertainty due to measurement error.

The concept of **model resolution** is an important way to characterize the bias of the generalized inverse solution. In this approach we see how closely the generalized inverse solution matches a given model, assuming that there are no errors in the data. We begin with a model $\mathbf{m}_{\text{true}}$. By multiplying $\mathbf{G}$ times $\mathbf{m}_{\text{true}}$, we can find a corresponding data vector $\mathbf{d}_{\text{true}}$. If we then multiply $\mathbf{G}^\dagger$ times $\mathbf{d}_{\text{true}}$, we obtain a generalized inverse solution

$$\mathbf{m}_\dagger = \mathbf{G}^\dagger \mathbf{G} \mathbf{m}_{\text{true}}. \tag{3.59}$$

We would obviously like to recover the original model, so that $\mathbf{m}^\dagger = \mathbf{m}_{\text{true}}$. However, because $\mathbf{m}_{\text{true}}$ may have had a nonzero projection onto the model null space

$N(\mathbf{G})$, $\mathbf{m}_\dagger$ will not in general be equal to $\mathbf{m}_{\text{true}}$. The **model resolution matrix** that characterizes this effect is

$$\mathbf{R}_{\text{m}} = \mathbf{G}^\dagger \mathbf{G} \tag{3.60}$$

$$= \mathbf{V}_p \mathbf{S}_p^{-1} \mathbf{U}_p^T \mathbf{U}_p \mathbf{S}_p \mathbf{V}_p^T \tag{3.61}$$

$$= \mathbf{V}_p \mathbf{V}_p^T. \tag{3.62}$$

If $N(\mathbf{G})$ is trivial, then $\text{rank}(\mathbf{G}) = p = n$, and $\mathbf{R}_{\text{m}}$ is the n by n identity matrix. In this case the original model is recovered exactly and we say that the resolution is perfect. If $N(\mathbf{G})$ is a nontrivial subspace of R^n, then $p = \text{rank}(\mathbf{G}) < n$, so that $\mathbf{R}_{\text{m}}$ is not the identity matrix. The model resolution matrix is instead a nonidentity symmetric matrix that characterizes how the generalized inverse solution smears out the original model, $\mathbf{m}_{\text{true}}$, into a recovered model, $\mathbf{m}_\dagger$. The trace of $\mathbf{R}_{\text{m}}$ is often used as a simple quantitative measure of the resolution. If $\text{Tr}(\mathbf{R}_{\text{m}})$ is close to n, then $\mathbf{R}_{\text{m}}$ is relatively close to the identity matrix.

The model resolution matrix can be used to quantify the bias introduced by the pseudoinverse when $\mathbf{G}$ does not have full column rank. We begin by showing that the expected value of $\mathbf{m}_\dagger$ is $\mathbf{R}_{\text{m}} \mathbf{m}_{\text{true}}$.

$$E[\mathbf{m}_\dagger] = E[\mathbf{G}^\dagger \mathbf{d}] \tag{3.63}$$

$$= \mathbf{G}^\dagger E[\mathbf{d}] \tag{3.64}$$

$$= \mathbf{G}^\dagger \mathbf{G} \mathbf{m}_{\text{true}} \tag{3.65}$$

$$= \mathbf{R}_{\text{m}} \mathbf{m}_{\text{true}}. \tag{3.66}$$

Thus the bias in the generalized inverse solution is

$$E[\mathbf{m}_\dagger] - \mathbf{m}_{\text{true}} = \mathbf{R}_{\text{m}} \mathbf{m}_{\text{true}} - \mathbf{m}_{\text{true}} \tag{3.67}$$

$$= (\mathbf{R}_{\text{m}} - \mathbf{I}) \mathbf{m}_{\text{true}} \tag{3.68}$$

where

$$\mathbf{R}_{\text{m}} - \mathbf{I} = \mathbf{V}_p \mathbf{V}_p^T - \mathbf{V} \mathbf{V}^T \tag{3.69}$$

$$= -\mathbf{V}_0 \mathbf{V}_0^T. \tag{3.70}$$

We can formulate a bound on the norm of the bias using (3.68) as

$$\|E[\mathbf{m}_\dagger] - \mathbf{m}_{\text{true}}\| \le \|\mathbf{R}_{\text{m}} - \mathbf{I}\| \|\mathbf{m}_{\text{true}}\|. \tag{3.71}$$

$\|\mathbf{R}_{\text{m}} - \mathbf{I}\|$ thus characterizes the bias introduced by the generalized inverse solution. However, the detailed effects of limited resolution on the recovered model will depend on $\|\mathbf{m}_{\text{true}}\|$, about which we may have quite limited a priori knowledge.

In practice, the model resolution matrix is commonly used in two different ways. First, we can examine diagonal elements of $\mathbf{R}_m$. Diagonal elements that are close to one correspond to parameters for which we can claim good resolution. Conversely, if any of the diagonal elements are small, then the corresponding model parameters will be poorly resolved. Second, we can multiply $\mathbf{R}_m$ times a particular test model $\mathbf{m}$ to see how that model would be resolved by the inverse solution. This strategy is called a **resolution test**. One commonly used test model is a **spike model**, which is a vector with all zero elements, except for a single entry which is one. Multiplying $\mathbf{R}_m$ times a spike model effectively picks out the corresponding column of the resolution matrix. These columns of the resolution matrix are called **resolution kernels**. Such functions are also similar to the averaging kernels in the method of Backus and Gilbert discussed in Chapter 5.

We can multiply $\mathbf{G}^\dagger$ and $\mathbf{G}$ in the opposite order from (3.62) to obtain the **data space resolution matrix**, $\mathbf{R}_d$

$$\mathbf{d}_\dagger = \mathbf{Gm}_\dagger \tag{3.72}$$

$$= \mathbf{GG}^\dagger \mathbf{d} \tag{3.73}$$

$$= \mathbf{R}_d \mathbf{d} \tag{3.74}$$

where

$$\mathbf{R}_d = \mathbf{U}_p \mathbf{S}_p \mathbf{V}_p^T \mathbf{V}_p \mathbf{S}_p^{-1} \mathbf{U}_p^T \tag{3.75}$$

$$= \mathbf{U}_p \mathbf{U}_p^T. \tag{3.76}$$

If $N(\mathbf{G}^T)$ contains only the zero vector, then $p = m$, and $\mathbf{R}_d = \mathbf{I}$. In this case, $\mathbf{d}_\dagger = \mathbf{d}$, and the generalized inverse solution $\mathbf{m}_\dagger$ fits the data exactly. However, if $N(\mathbf{G}^T)$ is nontrivial, then $p < m$, and $\mathbf{R}_d$ is not the identity matrix. In this case $\mathbf{m}_\dagger$ does not exactly fit the data.

Note that model and data space resolution matrices (3.62) and (3.76) do *not* depend on specific data or models, but are exclusively properties of $\mathbf{G}$. They reflect the physics and geometry of a problem, and can thus be assessed during the design phase of an experiment.

3.3. INSTABILITY OF THE GENERALIZED INVERSE SOLUTION

The generalized inverse solution $\mathbf{m}_\dagger$ has zero projection onto $N(\mathbf{G})$. However, it may include terms involving column vectors in $\mathbf{V}_p$ with very small nonzero singular values. In analyzing the generalized inverse solution it is useful to examine the **singular value spectrum**, which is simply the range of singular values. Small singular values cause the generalized inverse solution to be extremely sensitive to small amounts of noise in the

data. As a practical matter, it can also be difficult to distinguish between zero singular values and extremely small singular values. We can quantify the instabilities created by small singular values by recasting the generalized inverse solution to make the effect of small singular values explicit. We start with the formula for the generalized inverse solution,

$$\mathbf{m}_\dagger = \mathbf{V}_p \mathbf{S}_p^{-1} \mathbf{U}_p^T \mathbf{d}. \tag{3.77}$$

The elements of the vector $\mathbf{U}_p^T \mathbf{d}$ are the dot products of the first p columns of $\mathbf{U}$ with $\mathbf{d}$

$$\mathbf{U}_p^T \mathbf{d} = \begin{bmatrix} (\mathbf{U}_{.,1})^T \mathbf{d} \\ (\mathbf{U}_{.,2})^T \mathbf{d} \\ \cdot \\ \cdot \\ \cdot \\ (\mathbf{U}_{.,p})^T \mathbf{d} \end{bmatrix}. \tag{3.78}$$

When we left-multiply $\mathbf{S}_p^{-1}$ times (3.78), we obtain

$$\mathbf{S}_p^{-1} \mathbf{U}_p^T \mathbf{d} = \begin{bmatrix} \frac{(\mathbf{U}_{.,1})^T \mathbf{d}}{s_1} \\ \frac{(\mathbf{U}_{.,2})^T \mathbf{d}}{s_2} \\ \cdot \\ \cdot \\ \cdot \\ \frac{(\mathbf{U}_{.,p})^T \mathbf{d}}{s_p} \end{bmatrix}. \tag{3.79}$$

Finally, when we left-multiply $\mathbf{V}_p$ times (3.79), we obtain a linear combination of the columns of $\mathbf{V}_p$ that can be written as

$$\mathbf{m}_\dagger = \mathbf{V}_p \mathbf{S}_p^{-1} \mathbf{U}_p^T \mathbf{d} = \sum_{i=1}^p \frac{\mathbf{U}_{.,i}^T \mathbf{d}}{s_i} \mathbf{V}_{.,i}. \tag{3.80}$$

In the presence of random noise, $\mathbf{d}$ will generally have a nonzero projection onto each of the directions specified by the columns of $\mathbf{U}$. The presence of a very small s_i in the denominator of (3.80) can thus give us a very large coefficient for the corresponding model space basis vector $\mathbf{V}_{.,i}$, and these basis vectors can dominate the solution. In the worst case, the generalized inverse solution is just a noise amplifier, and the answer is practically useless.

A measure of the instability of the solution is the **condition number**. Note that the condition number considered here for an m by n matrix is a generalization of the condition number for an n by n matrix (A.107), and the two formulations are equivalent when $m = n$.

Suppose that we have a data vector $\mathbf{d}$ and an associated generalized inverse solution $\mathbf{m}_\dagger = \mathbf{G}^\dagger \mathbf{d}$. If we consider a slightly perturbed data vector $\mathbf{d}'$ and its associated generalized inverse solution $\mathbf{m}'_\dagger = \mathbf{G}^\dagger \mathbf{d}'$, then

$$\mathbf{m}_\dagger - \mathbf{m}'_\dagger = \mathbf{G}^\dagger(\mathbf{d} - \mathbf{d}') \tag{3.81}$$

and

$$\|\mathbf{m}_\dagger - \mathbf{m}'_\dagger\|_2 \leq \|\mathbf{G}^\dagger\|_2 \|\mathbf{d} - \mathbf{d}'\|_2. \tag{3.82}$$

From (3.80), it is clear that the largest difference in the inverse models will occur when $\mathbf{d} - \mathbf{d}'$ is in the direction $\mathbf{U}_{\cdot,p}$. If

$$\mathbf{d} - \mathbf{d}' = \alpha \mathbf{U}_{\cdot,p}, \tag{3.83}$$

then

$$\|\mathbf{d} - \mathbf{d}'\|_2 = \alpha. \tag{3.84}$$

We can then compute the effect on the generalized inverse solution as

$$\mathbf{m}_\dagger - \mathbf{m}'_\dagger = \frac{\alpha}{s_p} \mathbf{V}_{\cdot,p} \tag{3.85}$$

with

$$\|\mathbf{m}_\dagger - \mathbf{m}'_\dagger\|_2 = \frac{\alpha}{s_p}. \tag{3.86}$$

Thus, we have a bound on the instability of the generalized inverse solution

$$\|\mathbf{m}_\dagger - \mathbf{m}'_\dagger\|_2 \leq \frac{1}{s_p} \|\mathbf{d} - \mathbf{d}'\|_2. \tag{3.87}$$

Similarly, we can see that the generalized inverse model is smallest in norm when $\mathbf{d}$ points in a direction parallel to $\mathbf{V}_{\cdot,1}$. Thus

$$\|\mathbf{m}_\dagger\|_2 \geq \frac{1}{s_1} \|\mathbf{d}\|_2. \tag{3.88}$$

Combining these inequalities, we obtain

$$\frac{\|\mathbf{m}_\dagger - \mathbf{m}'_\dagger\|_2}{\|\mathbf{m}_\dagger\|_2} \leq \frac{s_1}{s_p} \frac{\|\mathbf{d} - \mathbf{d}'\|_2}{\|\mathbf{d}\|_2}. \tag{3.89}$$

The bound (3.89) is applicable to pseudoinverse solutions, regardless of what value of p we use. If we decrease p and thus eliminate model space vectors associated with small singular values, then the solution becomes more stable. However, this stability comes at the expense of reducing the dimension of the subspace of R^n where the solution lies.

As a result, the model resolution matrix for the stabilized solution obtained by decreasing p becomes less like the identity matrix, and the fit to the data worsens.

The condition number of $\mathbf{G}$ is the coefficient in (3.89)

$$\text{cond}(\mathbf{G}) = \frac{s_1}{s_k} \tag{3.90}$$

where $k = \min(m, n)$. The MATLAB command **cond** can be used to compute (3.90). If $\mathbf{G}$ is of full rank, and we use all of the singular values in the pseudoinverse solution ($p = k$), then the condition number is exactly (3.90). If $\mathbf{G}$ is of less than full rank, then the condition number is effectively infinite. As with the model and data resolution matrices ((3.62) and (3.76)), the condition number is a property of $\mathbf{G}$ that can be computed in the design phase of an experiment before any data are collected.

A condition that insures solution stability and arises naturally from consideration of (3.80) is the **discrete Picard condition** [67]. The discrete Picard condition is satisfied when the dot products of the columns of $\mathbf{U}$ and the data vector decay to zero more quickly than the singular values, s_i. Under this condition, we should not see instability due to small singular values. The discrete Picard condition can be assessed by plotting the ratios of $\mathbf{U}_{\cdot,i}^T\mathbf{d}$ to s_i across the singular value spectrum.

If the discrete Picard condition is not satisfied, we may still be able to recover a useful model by truncating the series for $\mathbf{m}_\dagger$ (3.80) at term $p' < p$, to produce a **truncated SVD**, or **TSVD** solution. One way to decide where to truncate the series is to apply the **discrepancy principle**. Under the discrepancy principle, we choose p' so that the model fits the data to a specified tolerance, δ, that is,

$$\|\mathbf{G}_w\mathbf{m} - \mathbf{d}_w\|_2 \leq \delta, \tag{3.91}$$

where $\mathbf{G}_w$ and $\mathbf{d}_w$ are the weighted system matrix and data vector, respectively.

How should we select δ? We discussed in Chapter 2 that when we estimate the solution to a full column rank least squares problem, $\|\mathbf{G}_w\mathbf{m}_{L_2} - \mathbf{d}_w\|_2^2$ has a χ^2 distribution with $m - n$ degrees of freedom, so we could set δ equal to $m - n$ if $m > n$. However, when the number of model parameters n is greater than or equal to the number of data m, this formulation fails because there is no χ^2 distribution with fewer than one degree of freedom. In practice, a common heuristic is to require $\|\mathbf{G}_w\mathbf{m} - \mathbf{d}_w\|_2^2$ to be smaller than m, because the approximate median of a χ^2 distribution with m degrees of freedom is m (Figure B.4).

A TSVD solution will not fit the data as well as solutions that include the model space basis vectors with small singular values. However, fitting the data vector too precisely in ill-posed problems (sometimes referred to as **over-fitting**) will allow data noise to control major features, or even completely dominate, the model.

The TSVD solution is but one example of **regularization**, where solutions are selected to sacrifice fit to the data in exchange for solution stability. Understanding the trade-off between fitting the data and solution stability involved in regularization is of fundamental importance.

3.4. A RANK DEFICIENT TOMOGRAPHY PROBLEM

A linear least squares problem is said to be **rank deficient** if there is a clear distinction between the nonzero and zero singular values and rank(G) is less than n. Numerically computed singular values will often include some that are extremely small but not quite zero, because of round-off errors. If there is a substantial gap between the largest of these tiny singular values and the first truly nonzero singular value, then it can be easy to distinguish between the two populations. Rank deficient problems can often be solved in a straightforward manner by applying the generalized inverse solution. After truncating the effectively zero singular values, a least squares model of limited resolution will be produced, and stability will seldom be an issue.

Example 3.1
Using the SVD, let us revisit the straight ray path tomography example that we considered earlier in Examples 1.6 and 1.12 (Figure 3.2). We introduced a rank deficient system in which we were constraining an $n = 9$-parameter slowness model with $m = 8$ travel time observations. We map the two-dimensional grid of slownesses into a model
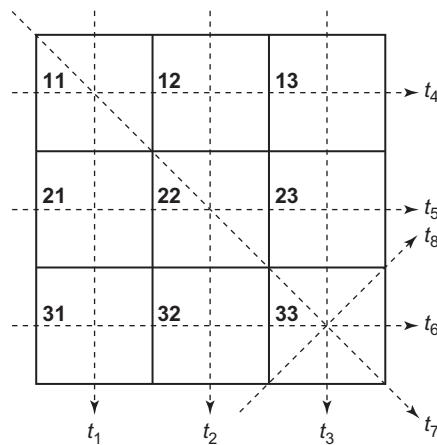

Figure 3.2 A simple tomography example (revisited).

vector by using a row-by-row indexing convention to obtain

$$
\mathbf{Gm} =
\begin{bmatrix}
1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\
1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\
\sqrt{2} & 0 & 0 & 0 & \sqrt{2} & 0 & 0 & 0 & \sqrt{2} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sqrt{2}
\end{bmatrix}
\begin{bmatrix}
s_{11} \\ s_{12} \\ s_{13} \\ s_{21} \\ s_{22} \\ s_{23} \\ s_{31} \\ s_{32} \\ s_{33}
\end{bmatrix}
=
\begin{bmatrix}
t_1 \\ t_2 \\ t_3 \\ t_4 \\ t_5 \\ t_6 \\ t_7 \\ t_8
\end{bmatrix} .
\tag{3.92}
$$

The eight singular values of $\mathbf{G}$ are, numerically evaluated,

$$
\operatorname{diag}(\mathbf{S}) =
\begin{bmatrix}
3.180 \\
2.000 \\
1.732 \\
1.732 \\
1.732 \\
1.607 \\
0.553 \\
1.429 \times 10^{-16}
\end{bmatrix} .
\tag{3.93}
$$

The smallest singular value, s_8, is nonzero in numerical evaluation only because of round-off errors in the SVD algorithm. It is zero in an analytical solution and is effectively zero in numerical computation relative to the other singular values. The ratio of the largest to smallest of the other singular values is about 6, and the generalized inverse solution (3.80) will thus be stable in the presence of noise. Because $\operatorname{rank}(\mathbf{G}) = p = 7$ is less than both m and n, the problem is both rank deficient and will in general have no exact solution. The model null space, $N(\mathbf{G})$, is spanned by the two orthonormal vectors that form the 8th and 9th columns of $\mathbf{V}$. An orthonormal basis for the null space is

$$
\mathbf{V}_0 = \begin{bmatrix} \mathbf{V}_{.,8}, & \mathbf{V}_{.,9} \end{bmatrix} =
\begin{bmatrix}
-0.0620 & -0.4035 \\
-0.4035 & 0.0620 \\
0.4655 & 0.3415 \\
0.4035 & -0.0620 \\
0.0620 & 0.4035 \\
-0.4655 & -0.3415 \\
-0.3415 & 0.4655 \\
0.3415 & -0.4655 \\
0.0000 & 0.0000
\end{bmatrix} .
\tag{3.94}
$$

To obtain a geometric appreciation for the two model null space vectors in this example, we can reshape them into 3 by 3 matrices corresponding to the geometry of the blocks (e.g., by using the MATLAB **reshape** command) to plot their elements in proper physical positions. Here, we have

$$
\text{reshape}(\mathbf{V}_{.,8},\ 3,\ 3)' =
\begin{bmatrix}
-0.0620 & -0.4035 & 0.4655 \\
0.4035 & 0.0620 & -0.4655 \\
-0.3415 & 0.3415 & 0.0000
\end{bmatrix}
\tag{3.95}
$$

$$
\text{reshape}(\mathbf{V}_{.,9},\ 3,\ 3)' =
\begin{bmatrix}
-0.4035 & 0.0620 & 0.3415 \\
-0.0620 & 0.4035 & -0.3415 \\
0.4655 & -0.4655 & 0.0000
\end{bmatrix}
\tag{3.96}
$$

(Figures 3.3 and 3.4).

Recall that if $\mathbf{m}_0$ is in the model null space, then (because $\mathbf{Gm}_0 = \mathbf{0}$) we can add such a model to any solution and not change the fit to the data (3.33). When mapped to their physical locations, three common features of the model null space basis vector elements in this example stand out:

1. The sums along all rows and columns are zero.
2. The upper left to lower right diagonal sum is zero.
3. There is no projection in the $m_9 = s_{33}$ model space direction.

The zero sum conditions (1) and (2) arise because paths passing through any three horizontal or vertical sets of blocks can only constrain the sum of those block values. The condition (3) that $m_9 = 0$ occurs because that model element is uniquely constrained by the 8th ray, which passes exclusively through the $s_{3,3}$ block. Thus, any variation in m_9
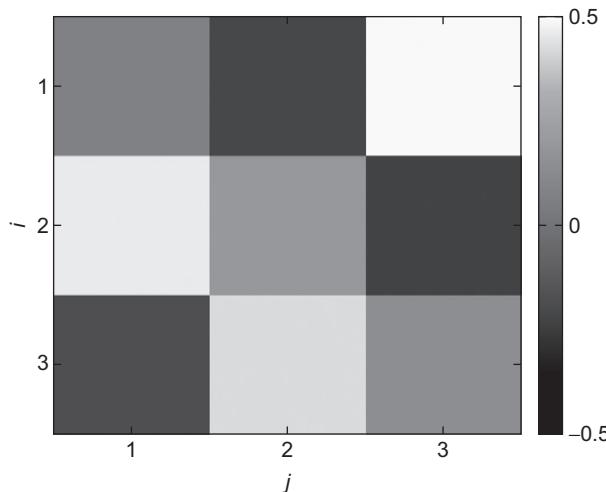

Figure 3.3 Image of the null space model $\mathbf{V}_{.,8}$.

Figure 3.4 Image of the null space model $\mathbf{V}_{.,9}$.

will clearly affect the predicted data, and any vector in the model null space must have a value of zero in m_9.

The single basis vector spanning the data null space in this example is

$$\mathbf{U}_0 = \mathbf{U}_{.,8} = \begin{bmatrix} -0.408 \\ -0.408 \\ -0.408 \\ 0.408 \\ 0.408 \\ 0.408 \\ 0.000 \\ 0.000 \end{bmatrix}. \qquad (3.97)$$

This indicates that increasing the times t_1, t_2, and t_3 and decreasing the times t_4, t_5, and t_6 by equal amounts will result in no change in the pseudoinverse solution.

Recall that, even for noise-free data, we will not recover a general $\mathbf{m}_{\text{true}}$ in a rank deficient problem using (3.22), but will instead recover a "smeared" model $\mathbf{R}_m\mathbf{m}_{\text{true}}$. Because $\mathbf{R}_m$ for a rank deficient problem is itself rank deficient, this smearing is irreversible. The full $\mathbf{R}_m$ matrix dictates precisely how this smearing occurs. The elements of $\mathbf{R}_m$ for this example are shown in Figure 3.5.

Examining the entire n by n model resolution matrix becomes cumbersome in large problems. The n diagonal elements of $\mathbf{R}_m$ can be examined more easily to provide basic information on how well recovered each model parameter will be. The reshaped
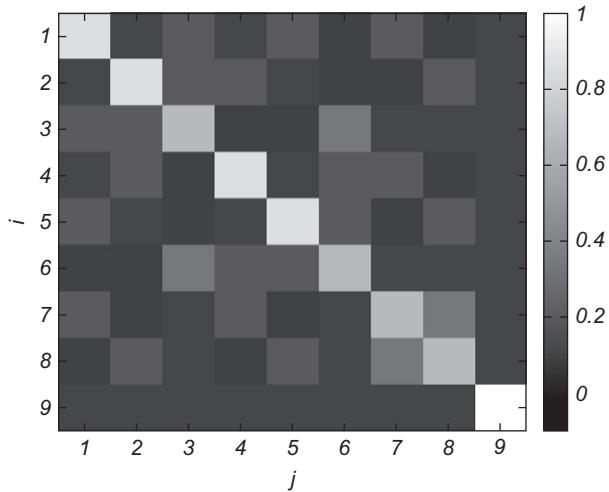
Figure 3.5 Elements of the model resolution matrix, $\mathbf{R}_m$ (3.62), for the generalized inverse solution.
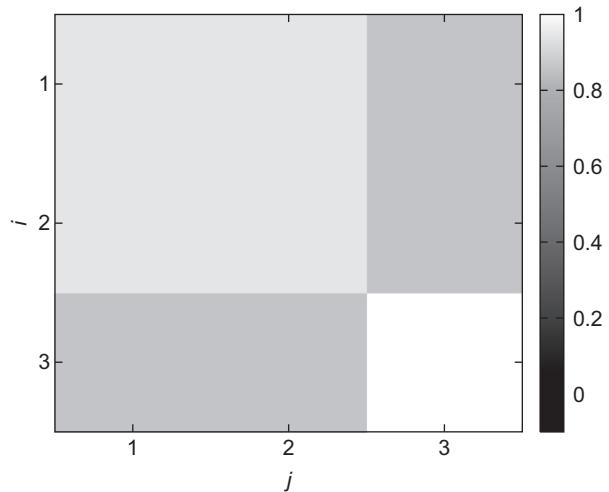

Figure 3.6 Diagonal elements of the resolution matrix plotted in their respective geometric model locations.

diagonal of $\mathbf{R}_m$ from Figure 3.5 is

$$\text{reshape}(\text{diag}(\mathbf{R}_m),\ 3,\ 3)' = \begin{bmatrix} 0.833 & 0.833 & 0.667 \\ 0.833 & 0.833 & 0.667 \\ 0.667 & 0.667 & 1.000 \end{bmatrix}. \tag{3.98}$$

These values are plotted in Figure 3.6.

Figure 3.6 and (3.98) tell us that m_9 is perfectly resolved, but that we can expect loss of resolution (and hence smearing of the true model into other blocks) for all of the other solution parameters.

We next assess the smoothing effects of limited model resolution by performing a resolution test using synthetic data for a test model of interest, and assessing the recovery of the test model by examining the corresponding inverse solution. Consider a spike model consisting of the vector with its 5th element equal to one and zeros elsewhere (Figure 3.7). Forward modeling gives the predicted data set for $\mathbf{m}_{\text{test}}$

$$\mathbf{d}_{\text{test}} = \mathbf{G}\mathbf{m}_{\text{test}} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ \sqrt{2} \\ 0 \end{bmatrix} \tag{3.99}$$

and the corresponding (reshaped) generalized inverse model is the fifth column of $\mathbf{R}_{\text{m}}$, which is

$$\text{reshape}(\mathbf{m}_\dagger, 3, 3)' = \begin{bmatrix} 0.167 & 0 & -0.167 \\ 0 & 0.833 & 0.167 \\ -0.167 & 0.167 & 0 \end{bmatrix} \tag{3.100}$$
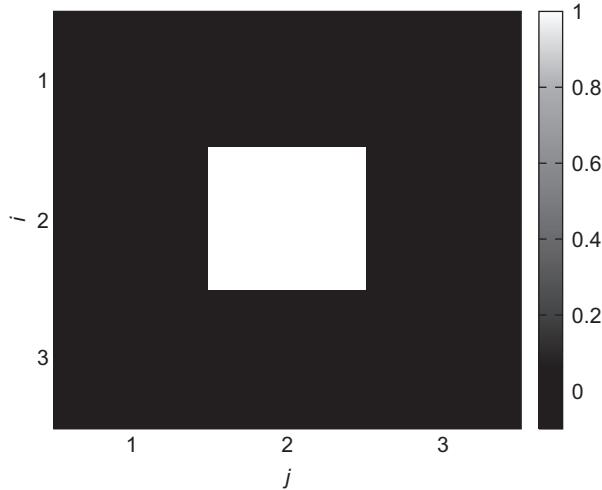

Figure 3.7 A spike test model.

Figure 3.8　The generalized inverse solution for the noise-free spike test.

(Figure 3.8). The recovered model in this spike test shows that limited resolution causes information about the central block slowness to smear into some, but not all, of the adjacent blocks *even for noise-free data*, with the exact form of the smearing dictated by the model resolution matrix.

It is important to reemphasize that the ability to recover the true model in practice is affected *both* by the bias caused by limited resolution, which is a characteristic of the matrix **G** and hence applies even to noise-free data, and by the mapping of any data noise into the model parameters. In specific cases one effect or the other may dominate.

3.5. DISCRETE ILL-POSED PROBLEMS

In many problems the singular values decay gradually toward zero and do not show an obvious jump between nonzero and zero singular values. This happens frequently when we discretize Fredholm integral equations of the first kind as in Chapter 1. In particular, as we increase the number of points in the discretization, we typically find that **G** becomes more and more poorly conditioned. We will refer to these as **discrete ill-posed problems**.

The rate of singular value spectrum decay can be used to characterize a discrete ill-posed problem as mildly, moderately, or severely ill-posed. If $s_j = O(j^{-\alpha})$ for $\alpha \leq 1$ (where O means "on the order of") then we call the problem mildly ill-posed. If $s_j = O(j^{-\alpha})$ for $\alpha > 1$, then the problem is moderately ill-posed. If $s_j = O(e^{-\alpha j})$ for some $\alpha > 0$, then the problem is severely ill-posed.

In discrete ill–posed problems, singular vectors $\mathbf{V}_{.j}$ associated with large singular values are typically smooth, while those corresponding to smaller singular values are highly oscillatory [67]. The influence of rough basis functions becomes increasingly apparent in the character of the generalized inverse solution as more singular values and vectors are included. When we attempt to solve such a problem with the TSVD in the presence of data noise, it is critical to decide where to truncate (3.80). If we truncate the sum too early, then our solution will lack details that require model vectors associated with the smaller singular values for their representation. However, if we include too many terms, then the solution becomes dominated by the influence of the data noise.

⬤——

Example 3.2

Consider an inverse problem where we have a physical process (e.g., seismic ground motion) recorded by a linear instrument of limited bandwidth (e.g., a vertical seismometer; Figure 3.9). The response of such a device is commonly characterized by an **instrument impulse response**, which is the response of the system to a delta function input. Consider the instrument impulse response

$$g(t) = \begin{cases} g_0 t e^{-t/T_0} & (t \geq 0) \\ 0 & (t < 0). \end{cases} \tag{3.101}$$

Figure 3.9 shows the displacement response of a critically damped seismometer with a characteristic time constant T_0 and gain, g_0, to a unit area (1 m/s$^2 \cdot$ s) impulsive ground acceleration input.

Figure 3.9 Example instrument response; seismometer output voltage in response to a unit area ground acceleration impulse (delta function).

Assuming that the displacement of the seismometer is electronically converted to output volts, we conveniently choose g_0 to be $T_0 e^{-1}$ V/m·s to produce a 1-V maximum output value for the impulse response, and $T_0 = 10$ s.

The seismometer output (or seismogram), $v(t)$, is a voltage record given by the convolution of the true ground acceleration, $m_{\text{true}}(t)$, with (3.101)

$$v(t) = \int_{-\infty}^{\infty} g(\tau - t) m_{\text{true}}(\tau) d\tau. \tag{3.102}$$

We are interested in the inverse **deconvolution** operation that will remove the smoothing effect of $g(t)$ in (3.102) and allow us to recover the true ground acceleration m_{true}.

Discretizing (3.102) using the midpoint rule with a time interval Δt, we obtain

$$\mathbf{d} = \mathbf{Gm} \tag{3.103}$$

where

$$G_{i,j} = \begin{cases} (t_i - t_j) e^{-(t_i - t_j)/T_0} \Delta t & (t_i \geq t_j) \\ 0 & (t_i < t_j). \end{cases} \tag{3.104}$$

The rows of $\mathbf{G}$ in (3.104) are time-reversed, and the columns of $\mathbf{G}$ are non–time-reversed, sampled representations of the impulse response $g(t)$, lagged by i and j, respectively. Using a time interval of $[-5, 100]$ s, outside of which (3.101) and any model, $\mathbf{m}$, of interest are assumed to be very small or zero, and a discretization interval of $\Delta t = 0.5$ s, we obtain a discretized m by n system matrix $\mathbf{G}$ with $m = n = 210$.

The singular values of $\mathbf{G}$ are all nonzero and range from about 25.3 to 0.017, giving a condition number of ≈ 1480, and showing that this discretization has produced a discrete system that is mildly ill-posed (Figure 3.10). However, adding noise at the level of 1 part in 1000 will be sufficient to make the generalized inverse solution unstable. The reason for this can be seen by examining successive rows of $\mathbf{G}$, which are nearly but not quite identical, with

$$\frac{\mathbf{G}_{i,\cdot} \mathbf{G}_{i+1,\cdot}^T}{\|\mathbf{G}_{i,\cdot}\|_2 \|\mathbf{G}_{i+1,\cdot}\|_2} \approx 0.999. \tag{3.105}$$

This near-identicalness of the rows of $\mathbf{G}$ makes the system of equations nearly singular, hence resulting in a large condition number.

Now, consider a true ground acceleration signal that consists of two acceleration pulses with widths of $\sigma = 2$ s, centered at $t = 8$ s and $t = 25$ s

$$m_{\text{true}}(t) = e^{-(t-8)^2/(2\sigma^2)} + 0.5 e^{-(t-25)^2/(2\sigma^2)}. \tag{3.106}$$

Figure 3.10 Singular values for the discretized convolution matrix.

We sample $\mathbf{m}_{\text{true}}(t)$ on the time interval $[-5, 100]$ s to obtain a 210-element vector $\mathbf{m}_{\text{true}}$, and generate the noise-free data set

$$\mathbf{d}_{\text{true}} = \mathbf{G}\mathbf{m}_{\text{true}} \tag{3.107}$$

and a second data set with independent $N(0, (0.05 \text{ V})^2)$ noise added. The data set with noise is shown in Figure 3.12.

The recovered least squares model from the full $(p = 210)$ generalized inverse solution,

$$\mathbf{m} = \mathbf{V}\mathbf{S}^{-1}\mathbf{U}^T\mathbf{d}_{\text{true}} \tag{3.108}$$

is shown in Figure 3.13. The model fits its noiseless data vector, $\mathbf{d}_{\text{true}}$, perfectly, and is essentially identical to the true model (Figure 3.11).

The least squares solution for the noisy data vector, $\mathbf{d}_{\text{true}} + \boldsymbol{\eta}$,

$$\mathbf{m} = \mathbf{V}\mathbf{S}^{-1}\mathbf{U}^T(\mathbf{d}_{\text{true}} + \boldsymbol{\eta}) \tag{3.109}$$

is shown in Figure 3.14.

Although this solution fits its particular data vector, $\mathbf{d}_{\text{true}} + \boldsymbol{\eta}$, exactly, it is worthless in divining information about the true ground motion. Information about $\mathbf{m}_{\text{true}}$ is overwhelmed by the small amount of added noise, amplified enormously by the inversion process.

Can a useful model be recovered by the TSVD? Using the discrepancy principle as our guide and selecting a range of solutions by varying p', we can in fact obtain

Figure 3.11 The true model.

Figure 3.12 Predicted data from the true model plus independent $N(0, (0.05 \text{ V})^2)$ noise.

an appropriate solution when we use just $p' = 26$ columns of $\mathbf{V}$ to obtain a solution (Figure 3.15).

Essential features of the true model are resolved in the solution of Figure 3.15, but the solution technique introduces oscillations and loss of resolution. Specifically, we see that the widths of the inferred pulses are somewhat wider, and the inferred amplitudes somewhat less, than those of the true ground acceleration. These effects are both hall–marks of limited resolution, as characterized by a nonidentity model resolution matrix.

Figure 3.13 Generalized inverse solution using all 210 singular values for the noise-free data.

Figure 3.14 Generalized inverse solution using all 210 singular values for the noisy data of Figure 3.12.

An image of the model resolution matrix in Figure 3.16 shows a finite–width central band and oscillatory side lobes.

A typical (80th) column of the model resolution matrix displays the smearing of the true model into the recovered model for the choice of the $p = 26$ inverse operator (Figure 3.17). The smoothing is over a characteristic width of about 5 s, which is why our recovered model, although it does a decent job of rejecting noise, underestimates

**Figure 3.15** TSVD solution using $p' = 26$ singular values for the noisy data shown in Figure 3.12.
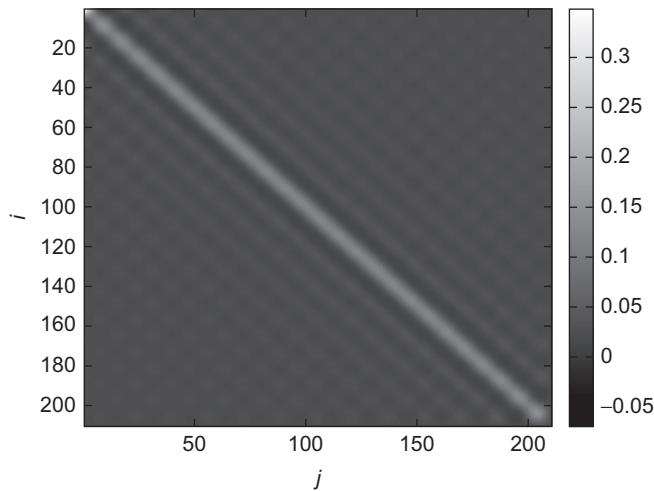
Figure 3.16 The model resolution matrix elements, $\mathbf{R}_{\mathrm{m}i,j}$, for the TSVD solution including $p' = 26$ singular values.

the amplitude and narrowness of the pulses in the true model (Figure 3.11). The oscillatory behavior of the resolution matrix is attributable to our abrupt truncation of the model space.

Each of the n columns of $\mathbf{V}$ is an oscillatory model basis function, with $j - 1$ zero crossings, where j is the column number. In truncating (3.80) at 26 terms to stabilize the inverse solution, we place a limit on the most oscillatory model space basis vectors that we will allow. This truncation gives us a model, and model resolution, that contain
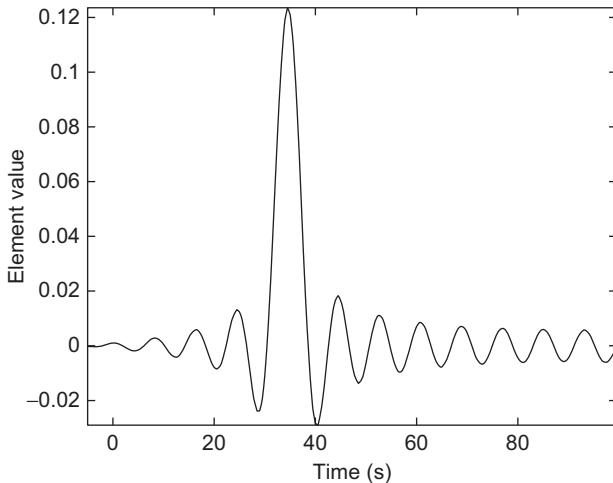
Figure 3.17 The (80th) column of the model resolution matrix, $\mathbf{R}_{\mathrm{m}}$, for the TSVD solution including $p' = 26$ singular values.

oscillatory structure with up to around $p - 1 = 25$ zero crossings. We will examine this perspective further in Chapter 8, where issues associated with oscillatory model basis functions will be revisited in the context of Fourier theory.

Example 3.3

Recall the Shaw problem from Examples 1.6 and 1.10. Figure 3.18 shows the singular value spectrum for the corresponding $\mathbf{G}$ matrix with $n = m = 20$, which is characterized by very rapid singular value decay to zero.

This is a severely ill-posed problem, and there is no obvious break point above which the singular values can reasonably be considered to be nonzero and below which the singular values can be considered to be 0. The MATLAB rank command gives $p' = 18$, suggesting that the last two singular values are effectively 0. The condition number of this problem is enormous (larger than 10^{14}).

The 18th column of $\mathbf{V}$, which corresponds to the smallest nonzero singular value, is shown in Figure 3.19. In contrast, the first column of $\mathbf{V}$, which corresponds to the largest singular value, represents a much smoother model (Figure 3.20). This behavior is typical of discrete ill-posed problems.

Next, we will perform a simple resolution test. Suppose that the input to the system is given by

$$m_i = \begin{cases} 1 & i = 10 \\ 0 & \text{otherwise.} \end{cases} \tag{3.110}$$
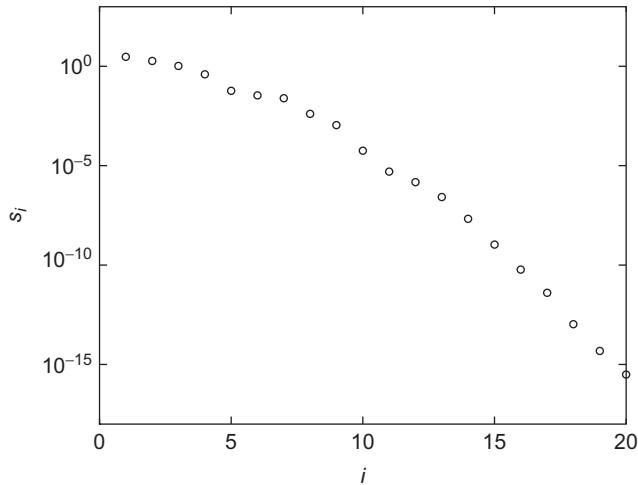
Figure 3.18 Singular values of **G** for the Shaw example ($n = m = 20$).
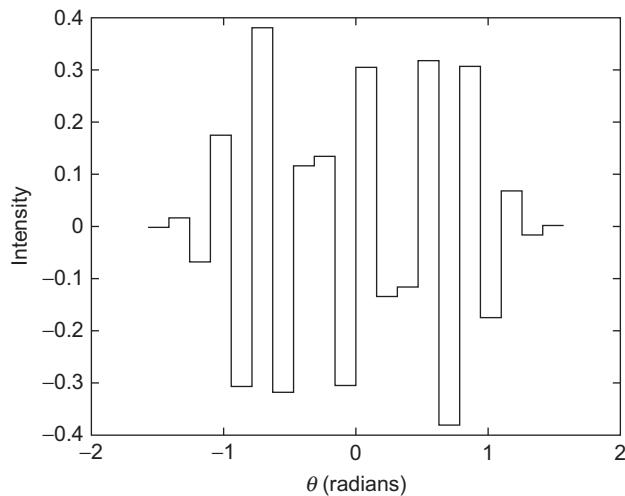
Figure 3.19 $\mathbf{V}_{\cdot,18}$ for the Shaw example problem.

(Figure 3.21). We use the model to obtain noise-free data and then apply the generalized inverse (3.22) with various values of p to obtain TSVD inverse solutions. The corresponding data are shown in Figure 3.22. If we compute the generalized inverse from these data using MATLAB's double-precision algorithms, we get fairly good recovery of (3.110), although there are still some lower amplitude negative intensity values (Figure 3.23).
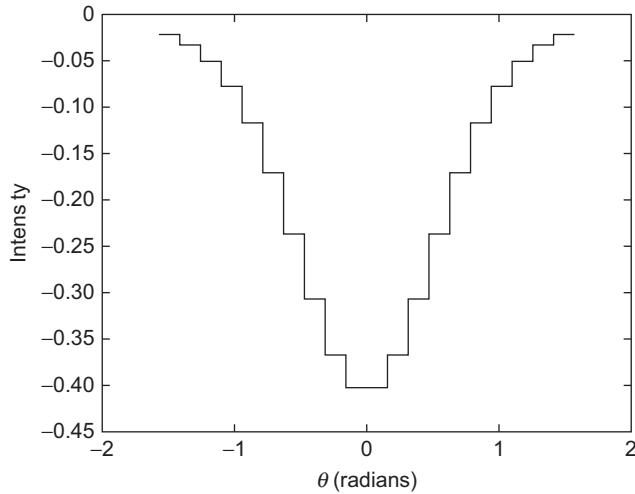
**Figure 3.20** $V_{\cdot,1}$ for the Shaw example problem.
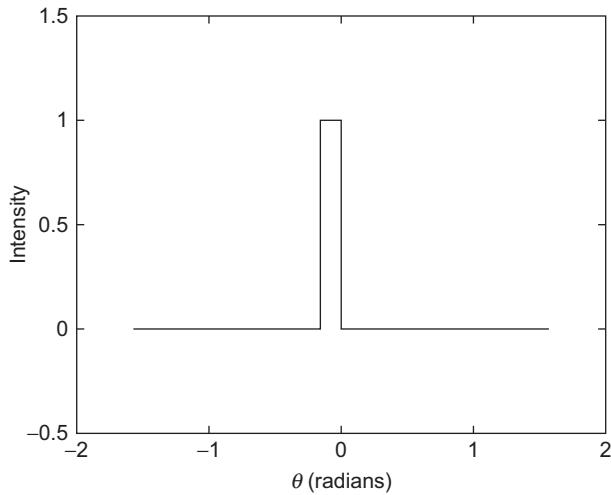
Figure 3.21 The spike model.

However, if we add a very small amount of noise to the data in Figure 3.22, things change dramatically. Adding $N(0, (10^{-6})^2)$ noise to the data of Figure 3.22 and computing a generalized inverse solution using $p' = 18$ produces the wild solution of Figure 3.24, which bears no resemblance to the true model. Note that the vertical scale in Figure 3.24 is multiplied by 10^6! Furthermore, the solution includes negative intensities, which are not physically possible. This inverse solution is even more sensitive to noise
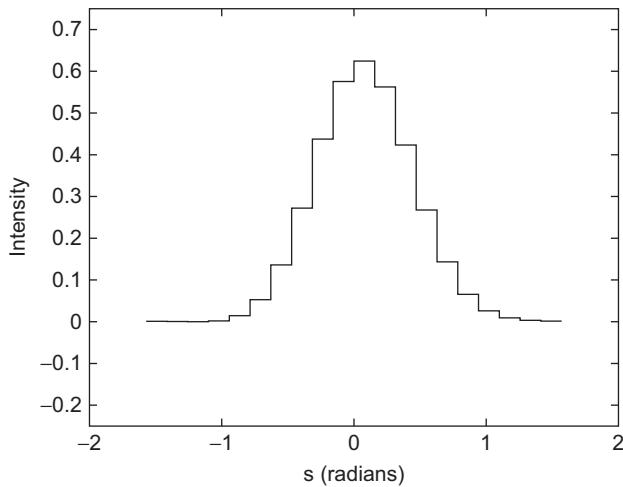
**Figure 3.22**   Noise-free data predicted for the spike model.
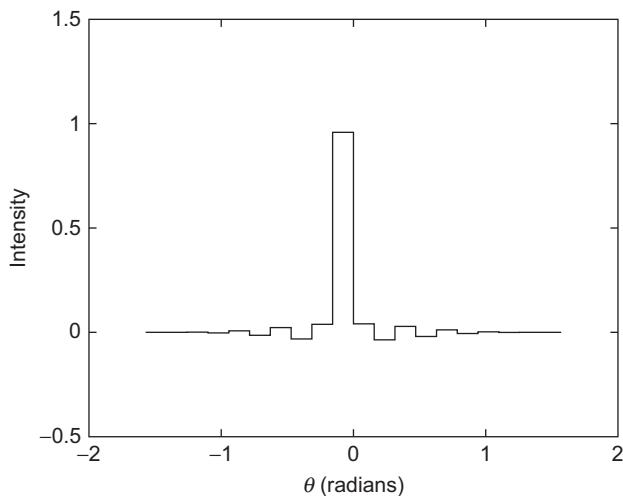
Figure 3.23   The generalized inverse solution for the spike model, no noise.

than that of the previous deconvolution example, to the extent that even noise on the order of 1 part in 10^6 will destabilize the solution.

Next, we consider what happens when we use only the 10 largest singular values and their corresponding model space vectors to construct a TSVD solution. Figure 3.25 shows the solution using 10 singular values with the same noise as Figure 3.24. Because we have cut off a number of singular values, we have reduced the model resolution. The inverse solution is smeared out, but it is still possible to conclude that there is some
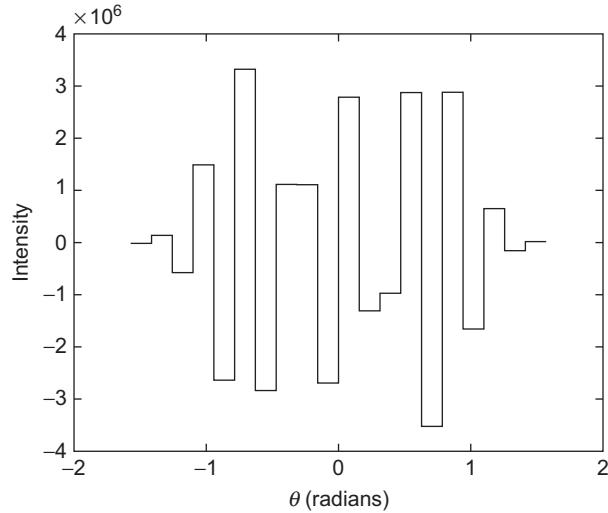
Figure 3.24 Recovery of the spike model with $N(0, (10^{-6})^2)$ noise using the TSVD method ($p' = 18$). Note that the intensity scale ranges from -4×10^6 to 4×10^6.
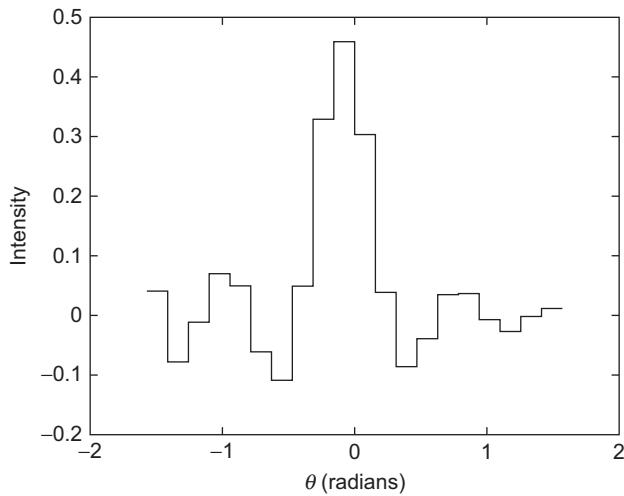
Figure 3.25 Recovery of the spike model with noise using the TSVD method ($p' = 10$).

significant spike-like feature near $\theta = 0$. In contrast to the situation that we observed in Figure 3.24, the model recovery is not visibly affected by the noise. The trade-off is that we must now accept the imperfect resolution of this solution and its attendant bias towards smoother models.

What happens if we discretize the problem with a larger number of intervals? Figure 3.26 shows the singular values for the **G** matrix with $n = m = 100$ intervals. The first 20

Figure 3.26 Singular values of **G** for the Shaw example ($n = m = 100$).
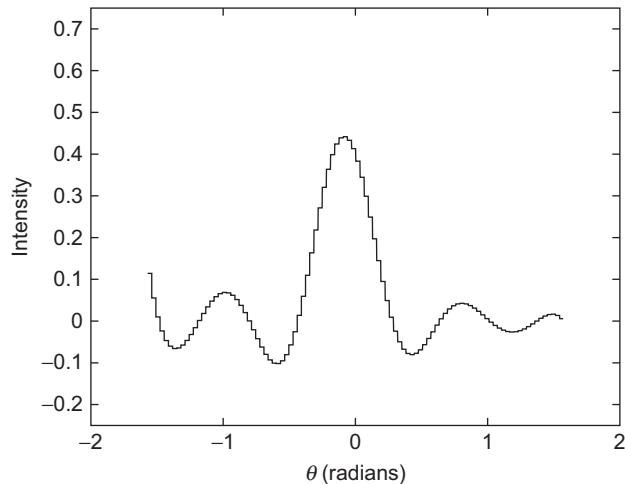
Figure 3.27 Recovery of the spike model with noise using the TSVD method ($n = m = 100$, $p' = 10$).

or so singular values are apparently nonzero, while the last 80 or so singular values are effectively zero.

Figure 3.27 shows the inverse solution for the spike model with $n = m = 100$ and $p = 10$. This solution is very similar to the solution shown in Figure 3.24. In general, discretizing over more intervals does not hurt as long as the solution is appropriately regularized and the additional computation time is acceptable.
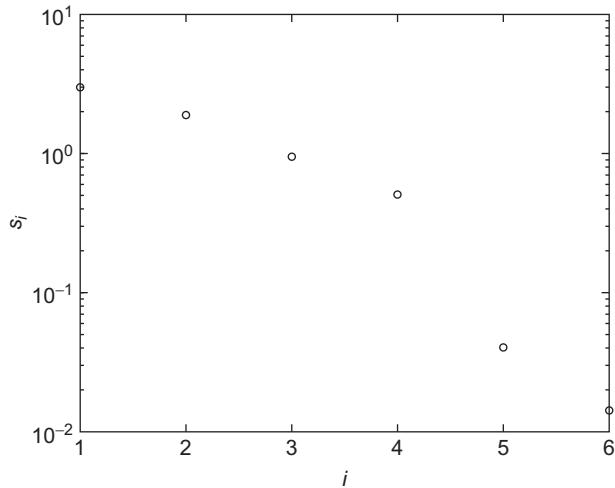
Figure 3.28 Singular values of **G** for the Shaw example ($n = m = 6$).

What about a smaller number of intervals? Figure 3.28 shows the singular values of the **G** matrix with $n = m = 6$. In this case there are no terribly small singular values. However, with only 6 elements in the model vector, we cannot hope to resolve the details of a source intensity distribution with a complex structure. This is an example of regularization by discretization (see also Exercise 1.3).

This example again demonstrates a fundamental dilemma. If we include small singular values in the series solution (3.80), then our solution becomes unstable in the presence of data noise. If we do not include these terms, our solution is less sensitive to data noise, but we sacrifice resolution and introduce bias.

3.6. EXERCISES

1. The pseudoinverse of a matrix **G** was originally defined by Moore and Penrose as the unique matrix $\mathbf{G}^\dagger$ with the properties
   a. $\mathbf{GG}^\dagger\mathbf{G} = \mathbf{G}$.
   b. $\mathbf{G}^\dagger\mathbf{GG}^\dagger = \mathbf{G}^\dagger$.
   c. $(\mathbf{GG}^\dagger)^T = \mathbf{GG}^\dagger$.
   d. $(\mathbf{G}^\dagger\mathbf{G})^T = \mathbf{G}^\dagger\mathbf{G}$.
   Show that $\mathbf{G}^\dagger$ as given by (3.20) satisfies these four properties.
2. Another resolution test commonly performed in tomography studies is a **checker-board test**, which consists of using a test model composed of alternating positive and negative perturbations. Perform a checkerboard test on the tomography problem

in Example 3.1 using the test model,

$$\mathbf{m}_{\text{true}} = \begin{bmatrix} -1 & 1 & -1 \\ 1 & -1 & 1 \\ -1 & 1 & -1 \end{bmatrix}. \tag{3.111}$$

Evaluate the difference between the true (checkerboard) model and the recovered model in your test, and interpret the pattern of differences. Are any block values recovered exactly? If so, does this imply perfect resolution for these model parameters?

3. Using the parameter estimation problem described in Example 1.1 for determining the three parameters defining a ballistic trajectory, construct synthetic examples that demonstrate the following four cases using the SVD. In each case, display and interpret the SVD components **U**, **V**, and **S** in terms of the rank, p, of your forward problem **G** matrix. Display and interpret any model and data null space vector(s) and calculate model and data space resolution matrices.

   a. Three data points that are exactly fit by a unique model. Plot your data points and the predicted data for your model.

   b. Two data points that are exactly fit by an infinite suite of parabolas. Plot your data points and the predicted data for a suite of these models.

   c. Four data points that are only approximately fit by a parabola. Plot your data points and the predicted data for the least squares model.

   d. Two data points that are only approximately fit by any parabola, and for which there are an infinite number of least squares solutions. Plot your data points and the predicted data for a suite of least squares models.

4. A large north–south by east–west–oriented, nearly square plan view, sandstone quarry block (16 m by 16 m) with a bulk compressional wave seismic velocity of approximately 3000 m/s is suspected of harboring higher-velocity dinosaur remains. An ultrasonic tomography scan is performed in a horizontal plane bisecting the boulder, producing a data set consisting of 16 E→W, 16 N→S, 31 NE→SW, and 31 NW→SE travel times (see Figure 3.29). The travel time data (units of s) have statistically independent errors, and the travel time contribution for a uniform background model (with a velocity of 3000 m/s) has been subtracted from each travel time measurement.

   The MATLAB data files that you will need to load containing the travel time data follow: **rowscan.mat**, **colscan.mat**, **diag1scan.mat**, and **diag2scan.mat**. The standard deviations of all data measurements are 1.5×10^{-5} s. Because the travel time contributions for a uniform background model (with a velocity of 3000 m/s) have been subtracted from each travel time measurement, you will be solving for slowness and velocity perturbations relative to a uniform slowness model of $1/3000$ s/m. Use
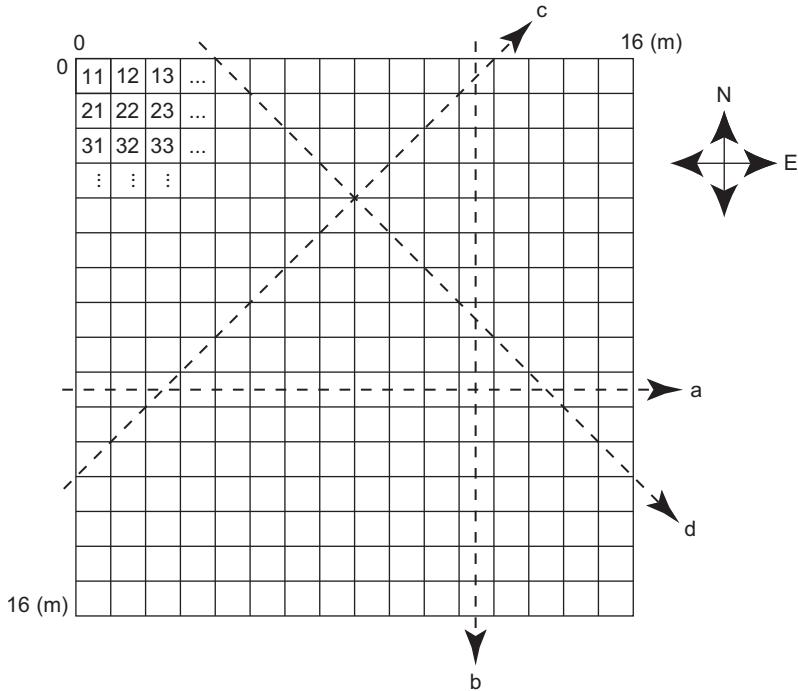
Figure 3.29 Tomography exercise, showing block discretization, block numbering convention, and representative ray paths going east-west (a), north-south (b), southwest-northeast (c), and northwest-southeast (d).

a row-by-row mapping between the slowness grid and the model vector (e.g., Example 1.12). The row format of each data file is (x_1, y_1, x_2, y_2, t) where the starting point coordinate of each source is (x_1, y_1), the end point coordinate is (x_2, y_2), and the travel time along a ray path between the source and receiver points is a path integral (in seconds).

Parameterize the slowness structure in the plane of the survey by dividing the boulder into a 16×16 grid of 256 1-m-square, north-by-east blocks and construct a linear system for the forward problem (Figure 3.29). Assume that the ray paths through each homogeneous block can be represented by straight lines, so that the travel time expression is

$$t = \int_{\ell} s(\mathbf{x}) d\ell \tag{3.112}$$

$$= \sum_{\text{blocks}} s_{\text{block}} \cdot \Delta l_{\text{block}}, \tag{3.113}$$

where Δl_{block} is 1 m for the row and column scans and $\sqrt{2}$ m for the diagonal scans.

Use the SVD to find a minimum–length/least squares solution, $m_\dagger$, for the 256 block slowness perturbations that fit the data as exactly as possible. Perform two inversions in this manner:

1. Use the row and column scans only.
2. Use the complete data set.

For each inversion:

a. Note the rank of your **G** matrix relating the data and model.

b. State and discuss the general solution and/or data fit significance of the elements and dimensions of the data and model null spaces. Plot and interpret an element of each space and contour or otherwise display a nonzero model that fits the trivial data set $\mathbf{Gm} = \mathbf{d} = \mathbf{0}$ exactly.

c. Note whether there are any model parameters that have perfect resolution.

d. Produce a 16 by 16 element contour or other plot of your slowness perturbation model, displaying the maximum and minimum slowness perturbations in the title of each plot. Interpret any internal structures geometrically and in terms of seismic velocity (in m/s).

e. Show the model resolution by contouring or otherwise displaying the 256 diagonal elements of the model resolution matrix, reshaped into an appropriate 16 by 16 grid.

f. Describe how one could use solutions to $\mathbf{Gm} = \mathbf{d} = \mathbf{0}$ to demonstrate that very rough models exist that will fit any data set just as well as a generalized inverse model. Show one such wild model.

5. Consider the data in Table 3.2 (also found in the file **ifk.mat**).

The function $d(y)$, $0 \leq y \leq 1$, is related to an unknown function $m(x)$, $0 \leq x \leq 1$, by the mathematical model

$$d(y) = \int_0^1 x e^{-xy} m(x)\,dx. \qquad (3.114)$$

a. Using the data provided, discretize the integral equation using simple collocation to create a square **G** matrix and solve the resulting system of equations.

Table 3.2 Data for Exercise 3.5.

y	0.0250	0.0750	0.1250	0.1750	0.2250	0.2750	0.3250	0.3750	0.4250	0.4750
$d(y)$	0.2388	0.2319	0.2252	0.2188	0.2126	0.2066	0.2008	0.1952	0.1898	0.1846
y	0.5250	0.5750	0.6250	0.6750	0.7250	0.7750	0.8250	0.8750	0.9250	0.9750
$d(y)$	0.1795	0.1746	0.1699	0.1654	0.1610	0.1567	0.1526	0.1486	0.1447	0.1410

    b. What is the condition number for this system of equations? Given that the data $d(\gamma)$ are only accurate to about four digits, what does this tell you about the accuracy of your solution?

    c. Use the TSVD to compute a solution to this problem. You may find a plot of the Picard ratios $\mathbf{U}_{.,i}^{T}\mathbf{d}/s_i$ to be especially useful in deciding how many singular values to include.

3.7. NOTES AND FURTHER READING

The Moore–Penrose generalized inverse was independently discovered by Moore in 1920 and Penrose in 1955 [110, 129]. Penrose is generally credited with first showing that the SVD can be used to compute the generalized inverse [129]. Books that discuss the linear algebra of the generalized inverse in more detail include [10, 23].

There was significant early work on the SVD in the 19th century by Beltrami, Jordan, Sylvester, Schmidt, and Weyl [151]. However, the singular value decomposition in matrix form is typically credited to Eckart and Young [41]. Some books that discuss the properties of the SVD and prove its existence include [53, 109, 152]. Lanczos presents an alternative derivation of the SVD [93]. Algorithms for the computation of the SVD are discussed in [38, 53, 164]. Books that discuss the use of the SVD and truncated SVD in solving discrete linear inverse problems include [65, 67, 108, 153].

Resolution tests with spike and checkerboard models as in Example 3.1 are commonly used in practice. However, Leveque, Rivera, and Wittlinger discuss some serious problems with such resolution tests [97]. Complementary information can be acquired by examining the diagonal elements of the resolution matrix, which can be efficiently calculated in isolation from off-diagonal elements even for very large inverse problems [9, 103] (Chapter 6).

Matrices like those in Example 3.2 in which the elements along diagonals are constant are called **Toeplitz matrices** [74]. Specialized methods for regularization of problems involving Toeplitz matrices are available [66].

It is possible to effectively regularize the solution to a discretized version of a continuous inverse problem by selecting a coarse discretization (e.g., Exercise 1.3 and Example 3.3). This approach is analyzed in [45]. However, in doing so we lose the ability to analyze the bias introduced by the regularization. In general, we prefer to use a fine discretization that is consistent with the physics of the forward problem and explicitly regularize the resulting discretized problem.

This page intentionally left blank

Tikhonov Regularization

Synopsis

The method of Tikhonov regularization for stabilizing inverse problem solutions is introduced and illustrated with examples. Zeroth-order Tikhonov regularization is explored, including its resolution, bias, and uncertainty properties. The concepts of filter factors (which control the contribution of singular values and their corresponding singular vectors to the solution) and the discrepancy and L-curve criteria (strategies for selecting the regularization parameter) are presented. Higher-order Tikhonov regularization techniques and their computation by the generalized singular value decomposition (GSVD) and truncated GSVD are discussed.

4.1. SELECTING GOOD SOLUTIONS TO ILL-POSED PROBLEMS

We saw in Chapter 3 that, given the SVD of $\mathbf{G}$ (3.1), we can express a generalized inverse solution by a series (3.81)

$$\mathbf{m}_\dagger = \mathbf{V}_p \mathbf{S}_p^{-1} \mathbf{U}_p^T \mathbf{d} = \sum_{i=1}^{p} \frac{\mathbf{U}_{\cdot,i}^T \mathbf{d}}{s_i} \mathbf{V}_{\cdot,i}. \tag{4.1}$$

We also saw that the generalized inverse solution can become extremely unstable when one or more of the singular values, s_i, is small. One approach for dealing with this difficulty, the truncated singular value decomposition (TSVD), was to truncate the series to remove singular vectors, $\mathbf{V}_{\cdot,i}$, associated with smaller singular values, s_i. This stabilized, or regularized, the solution in the sense that it made the result less sensitive to data noise. The cost of this stabilization approach is that the regularized solution had reduced resolution and was no longer unbiased.

   In this chapter we will introduce and discuss Tikhonov regularization, a very widely applied and easily implemented technique for regularizing discrete ill-posed problems. We will show a series formula for the Tikhonov solution that is a modified version of the generalized inverse series (3.81). The Tikhonov series solution has coefficients that are functions of the regularization parameter controlling the degree of regularization and which give greater weight to model elements associated with larger singular values.

For a general linear least squares problem there may be infinitely many least squares solutions. If we consider that the data contain noise, and that there is no point in fitting such noise exactly, it becomes evident that there can be many solutions that can adequately fit the data in the sense that $\|\mathbf{Gm} - \mathbf{d}\|_2$ is small enough.

In zeroth-order Tikhonov regularization, we consider all solutions with $\|\mathbf{Gm} - \mathbf{d}\|_2 \leq \delta$, and select the one that minimizes the norm of $\mathbf{m}$

$$\begin{aligned} \min & \|\mathbf{m}\|_2 \\ & \|\mathbf{Gm} - \mathbf{d}\|_2 \leq \delta. \end{aligned} \tag{4.2}$$

The fundamental motivation for minimizing the norm of $\mathbf{m}$ is to arrive at a solution that contains just sufficient feature complexity, as quantified by a norm measure, to adequately fit the data. Note that as δ increases, the set of feasible models expands, and the minimum value of $\|\mathbf{m}\|_2$ decreases. In other words, as we allow a poorer fit to the data, a smaller norm model will suffice to fit the data. We can thus trace out a curve of minimum values of $\|\mathbf{m}\|_2$ versus δ (Figure 4.1). It is also possible to trace out this curve by considering problems of the form

$$\begin{aligned} \min & \|\mathbf{Gm} - \mathbf{d}\|_2 \\ & \|\mathbf{m}\|_2 \leq \epsilon. \end{aligned} \tag{4.3}$$

As ϵ decreases, the set of feasible solutions becomes smaller, and the minimum value of $\|\mathbf{Gm} - \mathbf{d}\|_2$ increases. Again, as we adjust ϵ we trace out the curve of optimal values of $\|\mathbf{m}\|_2$ and $\|\mathbf{Gm} - \mathbf{d}\|_2$ (Figure 4.2).

A third option is to consider the **damped least squares** problem

$$\min \|\mathbf{Gm} - \mathbf{d}\|_2^2 + \alpha^2 \|\mathbf{m}\|_2^2, \tag{4.4}$$

which arises when we apply the method of Lagrange multipliers to (4.2), where α is a **regularization parameter**. It can be shown that for appropriate choices of δ, ϵ, and α, the three problems (4.2), (4.3), and (4.4) yield the same solution [65]. We will
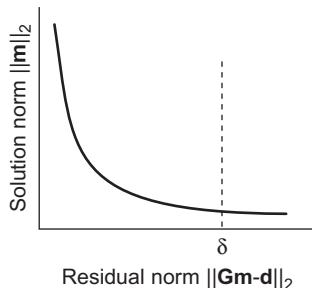
Figure 4.1 A particular misfit norm, δ, and its position on the trade-off curve between residual misfit, $\|\mathbf{Gm} - \mathbf{d}\|_2$, and model norm, $\|\mathbf{m}\|_2$.
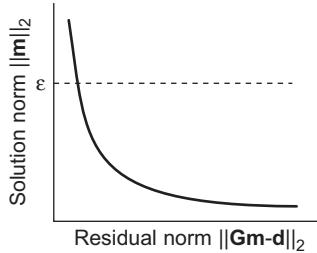
Figure 4.2 A particular model norm, ϵ, and its position on the trade-off curve between residual misfit, $\|\mathbf{Gm} - \mathbf{d}\|_2$, and model norm, $\|\mathbf{m}\|_2$.

concentrate on solving the damped least squares form of the problem (4.4). Solutions to (4.2) and (4.3) can be obtained using (4.4) by adjusting the regularization parameter α until the constraints are just satisfied.

When plotted on a log–log scale, the curve of optimal values of $\|\mathbf{m}\|_2$ versus $\|\mathbf{Gm} - \mathbf{d}\|_2$ often takes on a characteristic "L" shape in linear problems. This happens because $\|\mathbf{m}\|_2$ is a strictly decreasing function of α and $\|\mathbf{Gm} - \mathbf{d}\|_2$ is a strictly increasing function of α. The sharpness of the "corner" varies from problem to problem, but it is frequently well-defined. For this reason, the curve is called an **L–curve** [63]. In addition to the discrepancy principle, another popular criterion for picking the value of α is the **L–curve criterion**, in which the value of α that gives the solution closest to the corner of the L–curve is selected.

4.2. SVD IMPLEMENTATION OF TIKHONOV REGULARIZATION

The damped least squares problem (4.4) is equivalent to the ordinary least squares problem obtained by augmenting the least squares problem for $\mathbf{Gm} = \mathbf{d}$ in the following manner:

$$\min \left\| \begin{bmatrix} \mathbf{G} \\ \alpha\mathbf{I} \end{bmatrix} \mathbf{m} - \begin{bmatrix} \mathbf{d} \\ \mathbf{0} \end{bmatrix} \right\|_2^2. \tag{4.5}$$

As long as α is nonzero, the last n rows of the augmented matrix in (4.5) are obviously linearly independent. Equation (4.5) is thus a full rank least squares problem that can be solved by the method of normal equations, that is,

$$\begin{bmatrix} \mathbf{G}^T & \alpha\mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{G} \\ \alpha\mathbf{I} \end{bmatrix} \mathbf{m} = \begin{bmatrix} \mathbf{G}^T & \alpha\mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{d} \\ \mathbf{0} \end{bmatrix}. \tag{4.6}$$

Equation (4.6) simplifies to

$$(\mathbf{G}^T\mathbf{G} + \alpha^2\mathbf{I})\mathbf{m} = \mathbf{G}^T\mathbf{d}, \tag{4.7}$$

which is the set of constraint equations for a **zeroth-order Tikhonov regularization** solution of $\mathbf{Gm} = \mathbf{d}$.

Employing the SVD of $\mathbf{G}$, (4.7) can be written as

$$(\mathbf{VS}^T\mathbf{U}^T\mathbf{USV}^T + \alpha^2\mathbf{I})\mathbf{m} = \mathbf{G}^T\mathbf{d} \tag{4.8}$$

$$(\mathbf{VS}^T\mathbf{SV}^T + \alpha^2\mathbf{I})\mathbf{m} = \mathbf{VS}^T\mathbf{U}^T\mathbf{d}. \tag{4.9}$$

A simple substitution diagonalizes this system of equations and makes it straightforward to write out the solution. Let $\mathbf{x} = \mathbf{V}^T\mathbf{m}$ and $\mathbf{m} = \mathbf{Vx}$. Since $\mathbf{VV}^T = \mathbf{I}$, we can write (4.9) as

$$(\mathbf{VS}^T\mathbf{SV}^T + \alpha^2\mathbf{VV}^T)\mathbf{m} = \mathbf{VS}^T\mathbf{U}^T\mathbf{d}. \tag{4.10}$$

$$\mathbf{V}(\mathbf{S}^T\mathbf{S} + \alpha^2\mathbf{I})\mathbf{V}^T\mathbf{m} = \mathbf{VS}^T\mathbf{U}^T\mathbf{d}. \tag{4.11}$$

$$(\mathbf{S}^T\mathbf{S} + \alpha^2\mathbf{I})\mathbf{x} = \mathbf{S}^T\mathbf{U}^T\mathbf{d}. \tag{4.12}$$

The matrix on the left-hand side of this equation is diagonal, so it is trivial to solve the system of equations,

$$x_i = \frac{s_i\mathbf{U}_{\cdot,i}^T\mathbf{d}}{s_i^2 + \alpha^2}. \tag{4.13}$$

Since $\mathbf{m} = \mathbf{Vx}$, we obtain the solution

$$\mathbf{m}_\alpha = \sum_{i=1}^{k} \frac{s_i\mathbf{U}_{\cdot,i}^T\mathbf{d}}{s_i^2 + \alpha^2}\mathbf{V}_{\cdot,i} \tag{4.14}$$

where $k = \min(m, n)$ so that all of the nonzero singular values and vectors are included.

To relate this formula to (4.1), we can rewrite it slightly as

$$\mathbf{m}_\alpha = \sum_{i=1}^{k} \frac{s_i^2}{s_i^2 + \alpha^2}\frac{\mathbf{U}_{\cdot,i}^T\mathbf{d}}{s_i}\mathbf{V}_{\cdot,i} \tag{4.15}$$

or

$$\mathbf{m}_\alpha = \sum_{i=1}^{k} f_i\frac{\mathbf{U}_{\cdot,i}^T\mathbf{d}}{s_i}\mathbf{V}_{\cdot,i} \ . \tag{4.16}$$

Here, the **filter factors**

$$f_i = \frac{s_i^2}{s_i^2 + \alpha^2} \tag{4.17}$$

control the contribution to the sum from different terms. For $s_i \gg \alpha$, $f_i \approx 1$, and for $s_i \ll \alpha$, $f_i \approx 0$. For singular values between these two extremes, as the s_i decrease,

the f_i produce a monotonically decreasing contribution of corresponding model space vectors, $\mathbf{V}_{\cdot,i}$.

A related method called the **damped SVD method** [65] uses the filter factors

$$\hat{f}_i = \frac{s_i}{s_i + \alpha}. \tag{4.18}$$

This has a similar effect to using (4.17), but transitions more slowly with the index i.

Example 4.1

We revisit the severely ill-posed Shaw problem, which was previously introduced in Examples 1.6 and 1.10, and was analyzed using the SVD in Example 3.3. The true model in this synthetic example is a spike of unit amplitude in the 10th model element, and independent $N(0, (10^{-6})^2)$ noise has been added to the data vector. We begin by computing the L-curve at 1000 points and finding its corner by estimating the point of maximum curvature. Figure 4.3 shows the L-curve. The corner of the curve corresponds to $\alpha \approx 6.40 \times 10^{-6}$.

Next, we compute the Tikhonov regularization solution corresponding to this value of α. This solution is shown in Figure 4.4. Note that this solution is much better than the wild solution obtained by the TSVD with $p' = 18$ (Figure 3.24).

Alternatively, we can use the discrepancy principle to find an appropriate α to obtain a Tikhonov regularized solution. Because independent $N(0, (1 \times 10^{-6})^2)$ noise was added to these $m = 20$ data points, we search for a solution for which the square of the
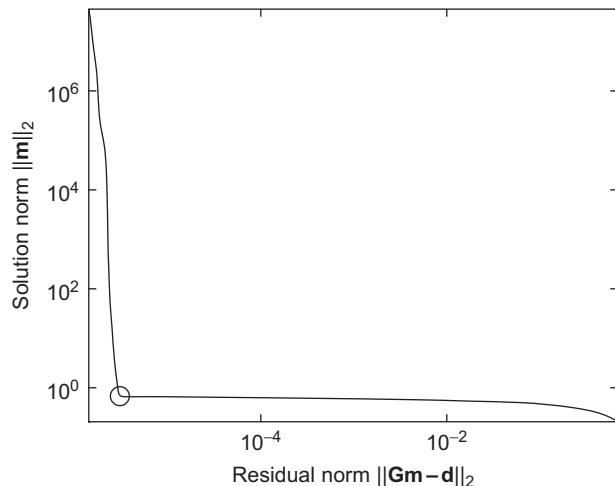

Figure 4.3 Zeroth-order Tikhonov regularization L-curve for the Shaw problem, with corner estimated using maximum functional curvature.
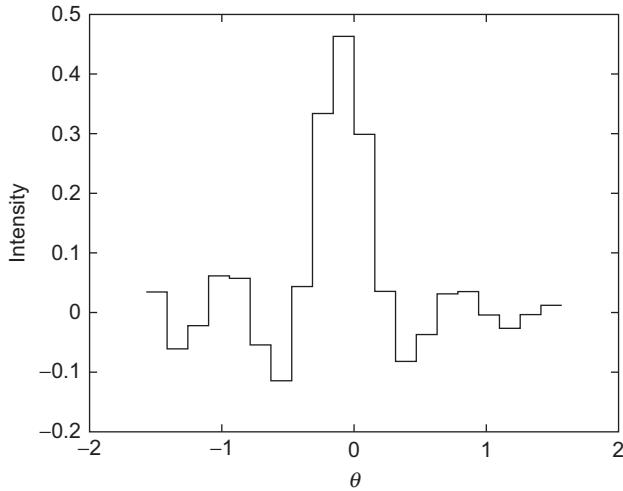
Figure 4.4 Recovery of the spike model with noise, zeroth-order Tikhonov solution ($\alpha = 6.40 \times 10^{-6}$ determined from the L-curve corner).
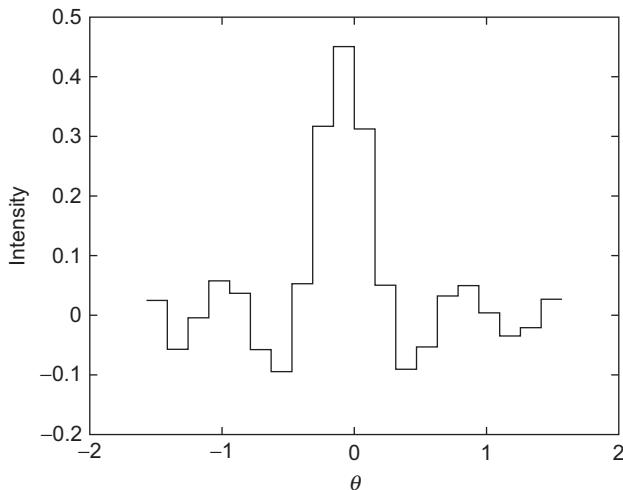
Figure 4.5 Recovery of the spike model with noise, zeroth-order Tikhonov solution ($\alpha = 4.29 \times 10^{-5}$ determined from the discrepancy principle).

norm of the residuals is 20×10^{-12}, which corresponds to a residual norm $\|\mathbf{Gm} - \mathbf{d}\|_2$ of $\sqrt{20} \times 10^{-6} \approx 4.47 \times 10^{-6}$.

The discrepancy principle results in a somewhat larger value of the regularization parameter, $\alpha = 4.29 \times 10^{-5}$, than that obtained using the L-curve technique above. The corresponding solution, shown in Figure 4.5, thus has a smaller model norm, but the two models are quite close.

Figure 4.6   Picard plot for the Shaw problem of Example 4.1.

It is interesting to note that the norm of the residual of the true (spike) model in this particular case, 3.86×10^{-6}, is actually slightly smaller than the tolerance that we specified via the discrepancy principle (4.47×10^{-6}). The discrepancy principle method did not recover the original spike model because the spike model has a norm of 1, while the solution obtained by the discrepancy principle has a norm of only 0.67.

Plotting the singular values s_i, the values of $|(\mathbf{U}_{\cdot,i})^T\mathbf{d}|$, and the ratios $|(\mathbf{U}_{\cdot,i})^T\mathbf{d}|/s_i$ allows us to examine the discrete Picard condition for this problem (Figure 4.6). $|(\mathbf{U}_{\cdot,i})^T\mathbf{d}|$ reaches a noise floor of about 1×10^{-6} after $i = 11$. The singular values continue to decay. As a consequence, the ratios increase rapidly. It is clear from this plot that we cannot expect to obtain useful information from the singular values beyond $i = 11$. The 11th singular value is $\approx 5.1 \times 10^{-6}$, which is comparable to the values of α in Figures 4.4 and 4.5.

4.3.   RESOLUTION, BIAS, AND UNCERTAINTY IN THE TIKHONOV SOLUTION

As in our earlier TSVD approach, we can compute a model resolution matrix for the Tikhonov regularization method. Using equation (4.7) and the SVD, the solution can be written as

$$\begin{aligned}
\mathbf{m}_\alpha &= (\mathbf{G}^T\mathbf{G} + \alpha^2\mathbf{I})^{-1}\mathbf{G}^T\mathbf{d} \\
&= \mathbf{G}^\sharp\mathbf{d} \\
&= \mathbf{V}\mathbf{F}\mathbf{S}^\dagger\mathbf{U}^T\mathbf{d},
\end{aligned} \tag{4.19}$$

where $\mathbf{F}$ is an n by n diagonal matrix with diagonal elements given by the filter factors f_i of (4.17), and $\mathbf{S}^\dagger$ is the pseudoinverse of $\mathbf{S}$. $\mathbf{G}^\sharp$ is a generalized inverse matrix that can be used to construct model and data resolution matrices as was done for the SVD solution in (3.63) and (3.77). The resolution matrices are

$$\mathbf{R}_m = \mathbf{G}^\sharp \mathbf{G} = \mathbf{V}\mathbf{F}\mathbf{V}^T \tag{4.20}$$

and

$$\mathbf{R}_d = \mathbf{G}\mathbf{G}^\sharp = \mathbf{U}\mathbf{F}\mathbf{U}^T. \tag{4.21}$$

Note that the resolution matrices are dependent on the particular value of α (4.19).

●————————————————————————————————————

Example 4.2

In Example 4.1, with $\alpha = 4.29 \times 10^{-5}$ as selected using the discrepancy principle, the model resolution matrix for zeroth-order Tikhonov regularization has the following diagonal elements:

$$\mathrm{diag}(\mathbf{R}_m) \approx [0.91,\ 0.49,\ 0.45,\ 0.39,\ 0.42,\ 0.41,\ 0.43,\ 0.44,\ 0.44,\ \ldots$$
$$0.45,\ 0.45,\ 0.44,\ 0.44,\ 0.43,\ 0.41,\ 0.42,\ 0.39,\ 0.45,\ 0.49,\ 0.91]^T \tag{4.22}$$

indicating that most model parameters are not well resolved. Figure 4.7 displays the effect of this limited resolution by applying $\mathbf{R}_m$ to the (true) spike model (3.63) or, equivalently, inverting noise-free spike model data using (4.6) for the regularization parameter
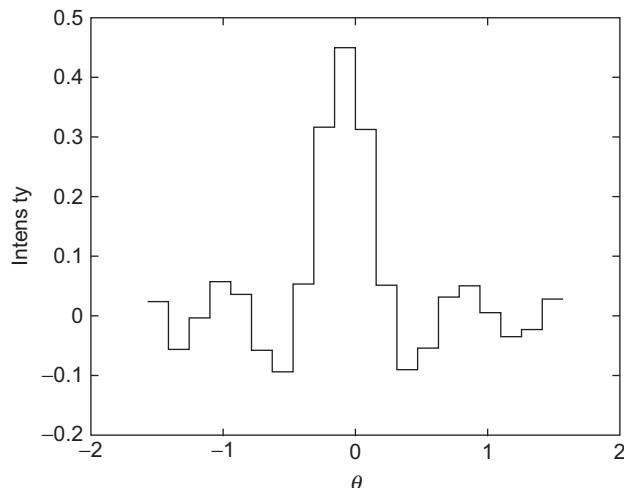

Figure 4.7 Resolution test using the spike model ($\alpha = 4.29 \times 10^{-5}$ determined from the discrepancy principle). Note that this model is nearly equivalent to that shown in Figure 4.5.

Figure 4.8 Resolution matrix for the Shaw problem, $\alpha = 4.29 \times 10^{-5}$. Note that the noise-free inversion of a spike model e_i will produce the *ith* column or row of $\mathbf{R}_m$ (e.g., Figure 4.7 is a plot of the 10th column/row).

value estimated using the discrepancy principle. Note that the result of limited resolution is that the true model "leaks" or "smears" into adjacent model parameters and is reduced in its maximum amplitude in the recovered model. In this example, the noise-free spike model recovery obtained in this resolution test is nearly identical to the recovery from the noisy spike model data using zeroth-order Tikhonov regularization (Figure 4.5), indicating that noise has only a very slight effect on model recovery accuracy. Thus, the differences between the true and recovered models here are essentially entirely due to the regularization that was necessary to stabilize the solution, rather than from noise propagation from data to model (see Example 4.3). Figure 4.7 displays just a single row or column from the (symmetric) $\mathbf{R}_m$, but effects of limited resolution can be examined more comprehensively by imaging the entire resolution matrix (Figure 4.8).

As in Chapter 2, we can compute a covariance matrix for the estimated model parameters using (B.65). Since

$$\mathbf{m}_\alpha = \mathbf{G}^\sharp \mathbf{d}, \tag{4.23}$$

the model covariance is

$$\mathrm{Cov}(\mathbf{m}_\alpha) = \mathbf{G}^\sharp \mathrm{Cov}(\mathbf{d})(\mathbf{G}^\sharp)^T. \tag{4.24}$$
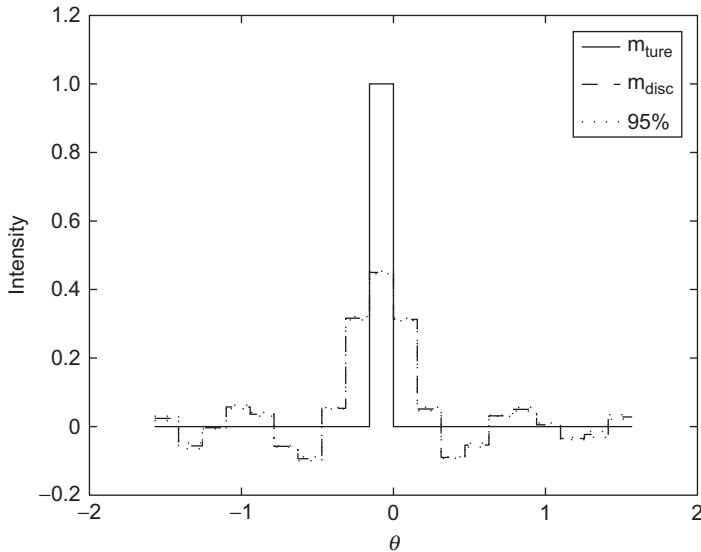
Figure 4.9 Tikhonov solution and confidence intervals for the Shaw problem, estimated using (4.24), where the true model is a spike and the data noise is independent and $N(0, (1 \times 10^{-6})^2)$. The regularization parameter $\alpha = 4.29 \times 10^{-5}$ was chosen using the discrepancy principle. The confidence interval is barely visible at this scale because inaccuracy in the model recovery is dominated by limited resolution.

Note that, as with the TSVD solution of Chapter 3, the Tikhonov regularized solution will generally be biased, and differences between the regularized solution values and the true model may actually be much larger than the confidence intervals obtained from the covariance matrix of the model parameters. Rather, the confidence intervals reflect the difference between $\mathbf{m}_\alpha$ and $\mathbf{R}_m\mathbf{m}_{\text{true}}$. See Figure 4.9.

Example 4.3

Recall our earlier example of the Shaw problem with the true spike model. Figure 4.9 shows the true model, the solution obtained using $\alpha = 4.29 \times 10^{-5}$ chosen using the discrepancy principle, and 95% confidence intervals for the estimated parameters. Note that the confidence intervals are extremely tight, and that very few of the true model parameters are included within the confidence intervals. In this case, the regularization bias, which is *not* estimated by the covariance matrix, is far larger than the propagated data uncertainty. In other words, the inaccuracy in model recovery in this case is dominated by limited resolution rather than by noise propagation from data to model. The solution shown in Figure 4.9 is essentially identical to the product of $\mathbf{R}_m$ and $\mathbf{m}_{\text{true}}$ shown in Figure 4.7 or the 10th column of the resolution matrix of Figure 4.8.

4.4. HIGHER-ORDER TIKHONOV REGULARIZATION

So far in our discussions of Tikhonov regularization we have minimized an objective function involving $\|\mathbf{m}\|_2$. In many situations, we would prefer to obtain a solution that minimizes some other measure of $\mathbf{m}$, expressed as $\mathbf{Lm}$, such as the norm of the first or second derivative of $\mathbf{m}$, reflecting a preference for a "flat" or "smooth" model. We then solve the regularized least squares problem

$$\min \ \|\mathbf{Gm} - \mathbf{d}\|_2^2 + \alpha^2 \|\mathbf{Lm}\|_2^2. \tag{4.25}$$

As with (4.5), this can be rewritten as a standard linear least squares problem:

$$\min \ \left\| \begin{bmatrix} \mathbf{G} \\ \alpha\mathbf{L} \end{bmatrix} \mathbf{m} - \begin{bmatrix} \mathbf{d} \\ \mathbf{0} \end{bmatrix} \right\|_2^2. \tag{4.26}$$

For example, if we have discretized our problem using simple collocation and our model is one-dimensional, then we can approximate, to a multiplicative constant, the first derivative of the model by a finite-difference operation $\mathbf{L}_1\mathbf{m}$, where

$$\mathbf{L}_1 = \begin{bmatrix} -1 & 1 & & & & \\ & -1 & 1 & & & \\ & & \ddots & \ddots & & \\ & & & -1 & 1 & \\ & & & & -1 & 1 \end{bmatrix}. \tag{4.27}$$

The matrices that are used to differentiate $\mathbf{m}$ for the purposes of regularization are commonly referred to as **roughening matrices**. In (4.27), $\mathbf{L}_1\mathbf{m}$ is a finite-difference approximation that is proportional to the first derivative of $\mathbf{m}$. By penalizing $\|\mathbf{L}_1\mathbf{m}\|_2$, we will favor solutions that are relatively flat (i.e., in the limit, constant). Note that $\|\mathbf{L}_1\mathbf{m}\|_2$ is a seminorm because it is zero for any constant model, not just for $\mathbf{m} = \mathbf{0}$.

In applying **second–order Tikhonov regularization** to a one-dimensional problem, we use a roughening matrix of the form

$$\mathbf{L}_2 = \begin{bmatrix} 1 & -2 & 1 & & & \\ & 1 & -2 & 1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & 1 & -2 & 1 \\ & & & & 1 & -2 & 1 \end{bmatrix}. \tag{4.28}$$

Here, $\mathbf{L}_2\mathbf{m}$ is a finite-difference approximation that is proportional to the second derivative of $\mathbf{m}$, and the seminorm term $\|\mathbf{L}_2\mathbf{m}\|_2$ in (4.25) penalizes solutions that are rough in a squared second derivative sense. We will refer to regularization strategies of the form of (4.25) that use $\mathbf{L}_1$ and $\mathbf{L}_2$ roughening matrices as **first– and second–order Tikhonov regularization**, respectively.

If our model is higher dimensional (e.g., two or three dimensions), then the roughening matrices described here would not be appropriate. In such cases, second-order Tikhonov regularization is often implemented using a finite-difference approximation to the Laplacian operator of appropriate dimensionality (see Exercise 4.3).

To ensure that the least squares problem (4.26) has a unique solution, we will require that the matrix

$$A = \begin{bmatrix} \mathbf{G} \\ \alpha \mathbf{L} \end{bmatrix} \tag{4.29}$$

has full column rank or equivalently that $N(\mathbf{G}) \cap N(\mathbf{L}) = \{\mathbf{0}\}$.

We have already seen how to apply zeroth-order Tikhonov regularization to solve (4.25), with $\mathbf{L} = \mathbf{I}$, using the singular value decomposition (4.16). To solve and analyze higher-order Tikhonov regularization problems, we employ the **generalized singular value decomposition**, or GSVD [54, 62, 65]. The GSVD enables the solution (4.25) to be expressed as a sum of filter factors times generalized singular vectors in a manner that is analogous to the series representation of the generalized inverse solution (4.16).

Unfortunately, the definition of the GSVD and associated notation are not presently standardized. In the following, we will adhere to the conventions used by the MATLAB **gsvd** command where $\mathbf{G}$ is an m by n matrix and $\mathbf{L}$ is a p by n matrix. Although MATLAB's implementation of the GSVD can handle matrices $\mathbf{G}$ and $\mathbf{L}$ that have overlapping null spaces, we assume in the following derivation that the matrix in (4.29) has full column rank so that the solution to (4.26) will be unique. To further simplify the derivation we will also assume that $\text{rank}(\mathbf{L}) = p$. This is certainly true for the $\mathbf{L}$ matrices in (4.27) and (4.28). In general we can eliminate redundant rows from $\mathbf{L}$ to make $\mathbf{L}$ have full row rank.

Under the above assumptions there exist matrices $\mathbf{U}$, $\mathbf{V}$, $\mathbf{X}$, $\mathbf{\Lambda}$, and $\mathbf{M}$ with the following properties and relationships:

- $\mathbf{U}$ is m by m and orthogonal.
- $\mathbf{V}$ is p by p and orthogonal.
- $\mathbf{X}$ is n by n and nonsingular.
- $\mathbf{\Lambda}$ is an m by n matrix with diagonal entries that may be shifted from the main diagonal of the matrix. The diagonal entries are

$$0 \leq \Lambda_{1,k+1} \leq \Lambda_{2,k+2} \leq \cdots \leq \Lambda_{m,k+m} \leq 1 \tag{4.30}$$

where $k = 0$ when $m > n$, and $k = n - m$ when $m \leq n$.

- $\mathbf{M}$ is a p by n diagonal matrix with

$$M_{1,1} \geq M_{2,2} \geq \cdots \geq M_{p,p} \geq 0, \tag{4.31}$$

and

$$\mathbf{M}^T\mathbf{M} + \mathbf{\Lambda}^T\mathbf{\Lambda} = \mathbf{I}. \tag{4.32}$$

- The matrices $\mathbf{G}$ and $\mathbf{L}$ can be written as

$$\mathbf{G} = \mathbf{U}\mathbf{\Lambda}\mathbf{X}^T \tag{4.33}$$

and

$$\mathbf{L} = \mathbf{V}\mathbf{M}\mathbf{X}^T. \tag{4.34}$$

The **generalized singular values** of $\mathbf{G}$ and $\mathbf{L}$ are

$$\gamma_i = \frac{\lambda_i}{\mu_i} \tag{4.35}$$

where

$$\lambda = \sqrt{\mathrm{diag}(\mathbf{\Lambda}^T\mathbf{\Lambda})} \tag{4.36}$$

and

$$\mu = \sqrt{\mathrm{diag}(\mathbf{M}^T\mathbf{M})}. \tag{4.37}$$

These definitions may seem somewhat odd, since the diagonal elements of $\mathbf{\Lambda}^T\mathbf{\Lambda}$ and $\mathbf{M}^T\mathbf{M}$ are simply squares of the diagonal elements of $\mathbf{\Lambda}$ and $\mathbf{M}$. The issue here is that the diagonals of $\mathbf{\Lambda}$ and $\mathbf{M}$ are not of the same length. The effect of these definitions is to create vectors λ and μ that are of length n, padding with zeros as needed.

Because of the ordering of λ and μ, the generalized singular values appear in ascending order with

$$\gamma_1 \le \gamma_2 \le \cdots \le \gamma_n. \tag{4.38}$$

Also note that if $\mu_i = 0$, then the corresponding generalized singular value γ_i is infinite or undefined.

Letting $\mathbf{Y} = \mathbf{X}^{-T}$ (the inverse transpose), we can apply (4.33) and the orthogonality of $\mathbf{U}$ to show that

$$\mathbf{Y}^T\mathbf{G}^T\mathbf{G}\mathbf{Y} = \mathbf{\Lambda}^T\mathbf{\Lambda}. \tag{4.39}$$

Note that

$$\lambda_i = \sqrt{\mathbf{Y}_{\cdot,i}^T\mathbf{G}^T\mathbf{G}\mathbf{Y}_{\cdot,i}} = \|\mathbf{G}\mathbf{Y}_{\cdot,i}\|_2. \tag{4.40}$$

Whenever λ_i is 0, this means that the corresponding column of $\mathbf{Y}$ is in $N(\mathbf{G})$. However, when λ_i is nonzero, the corresponding column of $\mathbf{Y}$ is not in $N(\mathbf{G})$. Since $\mathbf{Y}$ is nonsingular, the columns of $\mathbf{Y}$ are linearly independent. If we pick r so that $\lambda_r = 0$, but $\lambda_{r+1} \ne 0$, then r is the dimension of $N(\mathbf{G})$, and the vectors $\mathbf{Y}_{\cdot,1}, \mathbf{Y}_{\cdot,2}, \ldots, \mathbf{Y}_{\cdot,r}$ form

a basis for $N(\mathbf{G})$. Note that because $\operatorname{rank}(\mathbf{G}) + \dim(N(\mathbf{G})) = n$, we have determined that $\operatorname{rank}(\mathbf{G}) = n - r$.

Similarly, it is easy to use (4.34) and the orthogonality of $\mathbf{V}$ to show that

$$\mathbf{Y}^T \mathbf{L}^T \mathbf{L} \mathbf{Y} = \mathbf{M}^T \mathbf{M}. \tag{4.41}$$

Since $\operatorname{rank}(\mathbf{L}) = p$, the dimension of the null space of $\mathbf{L}$ is $n - p$. Thus μ_{p+1}, μ_{p+2}, . . . , μ_n are 0, and the vectors $\mathbf{Y}_{\cdot,p+1}$, $\mathbf{Y}_{\cdot,p+2}$, . . . , $\mathbf{Y}_{\cdot,n}$ form a basis for $N(\mathbf{L})$. Note that the columns of $\mathbf{Y}$ are not generally orthogonal, so we have not found orthonormal bases for the null spaces of $\mathbf{G}$ and $\mathbf{L}$.

The GSVD may be computationally expensive (or intractable for very large problems) to evaluate. However, as with the SVD, once the decomposition matrices are computed, solutions to least squares problems may be calculated very easily. We begin by introducing the change of variables

$$\mathbf{Y}\mathbf{x} = \mathbf{m}. \tag{4.42}$$

With this substitution, the normal equations for (4.26) become

$$\left(\mathbf{G}^T \mathbf{G} + \alpha^2 \mathbf{L}^T \mathbf{L}\right) \mathbf{Y}\mathbf{x} = \mathbf{G}^T \mathbf{d}. \tag{4.43}$$

Using (4.33) and (4.34), and $\mathbf{Y} = \mathbf{X}^{-T}$, we have

$$\left(\mathbf{Y}^{-T} \boldsymbol{\Lambda}^T \mathbf{U}^T \mathbf{U} \boldsymbol{\Lambda} \mathbf{Y}^{-1} + \alpha^2 \mathbf{Y}^{-T} \mathbf{M}^T \mathbf{V}^T \mathbf{V} \mathbf{M} \mathbf{Y}^{-1}\right) \mathbf{Y}\mathbf{x} = \mathbf{Y}^{-T} \boldsymbol{\Lambda}^T \mathbf{U}^T \mathbf{d}, \tag{4.44}$$

$$\left(\mathbf{Y}^{-T} \boldsymbol{\Lambda}^T \boldsymbol{\Lambda} \mathbf{Y}^{-1} + \alpha^2 \mathbf{Y}^{-T} \mathbf{M}^T \mathbf{M} \mathbf{Y}^{-1}\right) \mathbf{Y}\mathbf{x} = \mathbf{Y}^{-T} \boldsymbol{\Lambda}^T \mathbf{U}^T \mathbf{d}, \tag{4.45}$$

and, finally,

$$\left(\boldsymbol{\Lambda}^T \boldsymbol{\Lambda} + \alpha^2 \mathbf{M}^T \mathbf{M}\right) \mathbf{x} = \boldsymbol{\Lambda}^T \mathbf{U}^T \mathbf{d}. \tag{4.46}$$

The matrix on the left-hand side of this system of equations is diagonal. Multiplying both sides by the inverse of this diagonal matrix gives the solution

$$x_i = \frac{\lambda_i \mathbf{U}_{\cdot,i+k}^T \mathbf{d}}{\lambda_i^2 + \alpha^2 \mu_i^2}, \tag{4.47}$$

where, as in (4.30), $k = 0$ when $m > n$, and $k = n - m$ when $m \leq n$. In terms of the generalized singular values, this can be written as

$$x_i = \frac{\gamma_i^2}{\gamma_i^2 + \alpha^2} \frac{\mathbf{U}_{\cdot,i+k}^T \mathbf{d}}{\lambda_i}. \tag{4.48}$$

Substituting this expression for x_i into $\mathbf{m} = \mathbf{Y}\mathbf{x}$, we obtain the summation formula

$$\mathbf{m}_{\alpha,L} = \sum_{i=1}^{n} \frac{\gamma_i^2}{\gamma_i^2 + \alpha^2} \frac{\mathbf{U}_{\cdot,i+k}^T \mathbf{d}}{\lambda_i} \mathbf{Y}_{\cdot,i} \tag{4.49}$$

where

$$f_i = \frac{\gamma_i^2}{\gamma_i^2 + \alpha^2} \tag{4.50}$$

are **GSVD filter factors** that are analogous to the filter factors obtained in the series expression for the zeroth-order Tikhonov regularized solution (4.17). In evaluating this sum, we sometimes encounter situations in which γ_i is infinite. In those cases, the filter factor f_i should be set to 1. Similarly, there are situations in which $\lambda_i = 0$ and $\gamma_i = 0$, producing an expression of the form $0^2/0$ in the sum. These terms should be treated as 0.

When **G** comes from a Fredholm integral equation of the first kind, the GSVD typically has two properties that were also characteristic of the SVD. First, the m nonzero generalized singular values γ_n, γ_{n-1}, γ_{n-m+1} from (4.35) trend toward zero without any obvious break. Second, the vectors $\mathbf{U}_{\cdot,i}$, $\mathbf{V}_{\cdot,i}$, $\mathbf{X}_{\cdot,i}$, and $\mathbf{Y}_{\cdot,i}$ tend to become rougher as γ_i decreases.

● ─────────────────────────────────────

Example 4.4

We return to the vertical seismic profiling example previously discussed in Examples 1.3 and 1.9. Here, consider a 1-km deep borehole experiment discretized using $m = n = 50$ observation and model points, corresponding to sensors every 20 m, and 20 m thick, constant-slowness model layers. Figure 4.10 shows the test model that we will try to recover. A synthetic data set was generated with $N(0, (2 \times 10^{-4}\text{s})^2)$ noise added.

Figure 4.10 A smooth test model for the VSP problem.

The discretized system of equations **Gm = d** has a small condition number (64). This happens in part because we have chosen a very coarse discretization, which effectively regularizes the problem by discretization. Another reason is that the vertical seismic profiling problem is only mildly ill-posed [45]. Figure 4.11 shows the least squares solution, together with 95% confidence intervals.

From the statistical point of view, this solution is completely acceptable. However, suppose that from other information, we believe the slowness should vary smoothly with depth. We will next apply higher-order Tikhonov regularization to obtain smooth solutions to this problem.

Figure 4.12 shows the first-order Tikhonov regularization L-curve for this problem. The L-curve has a distinct corner near $\alpha \approx 122$. Figure 4.13 shows the corresponding solution. The first-order regularized solution is much smoother than the least squares solution, and is much closer to the true solution.

Figure 4.14 shows the L-curve for second-order Tikhonov regularization, which has a corner near $\alpha \approx 1965$. Figure 4.15 shows the corresponding solution. This solution is smoother still compared to the first-order regularized solution. Both the first- and second-order solutions depart most from the true solution at shallow depths where the true slowness has the greatest slope and curvature. This happens because the first- and second-order Tikhonov regularized solutions are biased towards flatness and second–derivative smoothness, respectively.

Figure 4.16 shows filter factors corresponding to these first- and second-order solutions. Higher-order terms in (4.49) are severely downweighted in both cases, particularly

Figure 4.11 Least squares solution for the VSP problem, with 95% confidence intervals.
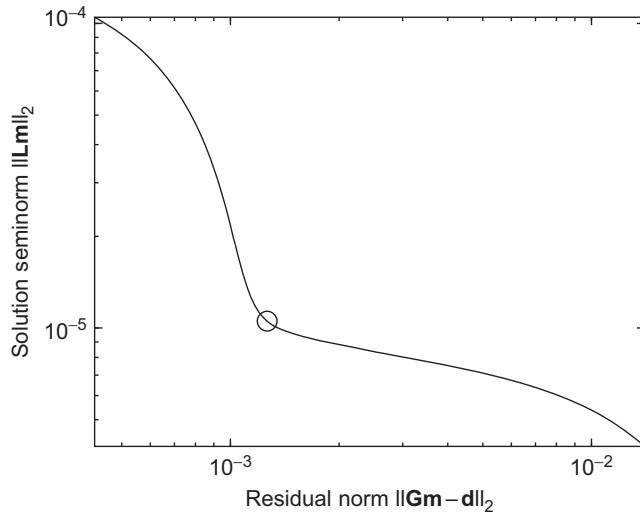
Figure 4.12 L-curve and corner for the VSP problem, first-order regularization.

Figure 4.13 Tikhonov solution for the VSP problem, first-order regularization, $\alpha = 122$, shown in comparison with the true model (Figure 4.10).

in the second–order case. Because of the smoothness of the true model, the model seminorms can be reduced considerably through the selection of relatively large regularization parameters, α, without large data misfit increases. In this example the 2–norms
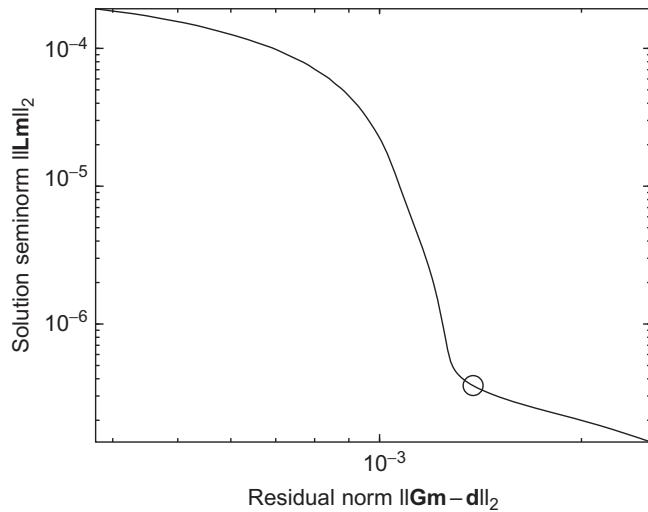
Figure 4.14 L-curve and corner for the VSP problem, second-order regularization.

Figure 4.15 Tikhonov solution for the VSP problem, second-order regularization, $\alpha = 2341$, shown in comparison with the true model (Figure 4.10).

of the difference between the first- and second-order solutions and the true model (discretized into 50 values) are approximately 1.2×10^{-5} s/km and 1.0×10^{-5} s/km, respectively.

Figure 4.16 Filter factors (4.17) for optimal first- and second-order Tikhonov solutions to the VSP problem shown in Figures 4.13 and 4.15.

4.5. RESOLUTION IN HIGHER-ORDER TIKHONOV REGULARIZATION

As with zeroth–order Tikhonov regularization, we can compute a resolution matrix for higher-order Tikhonov regularization. For a particular roughening matrix $\mathbf{L}$ and value of α, the Tikhonov regularization solution can be written as

$$\mathbf{m}_{\alpha,L} = \mathbf{G}^{\sharp}\mathbf{d} \tag{4.51}$$

where

$$\mathbf{G}^{\sharp} = (\mathbf{G}^T\mathbf{G} + \alpha^2\mathbf{L}^T\mathbf{L})^{-1}\mathbf{G}^T. \tag{4.52}$$

Using the GSVD, we can write this expression as

$$\mathbf{G}^{\sharp} = (\mathbf{X}\mathbf{\Lambda}^T\mathbf{U}^T\mathbf{U}\mathbf{\Lambda}\mathbf{X}^T + \alpha^2(\mathbf{X}\mathbf{M}^T\mathbf{V}^T\mathbf{V}\mathbf{M}\mathbf{X}^T))^{-1}\mathbf{X}\mathbf{\Lambda}^T\mathbf{U}^T \tag{4.53}$$

$$= (\mathbf{X}\mathbf{\Lambda}^T\mathbf{\Lambda}\mathbf{X}^T + \alpha^2(\mathbf{X}\mathbf{M}^T\mathbf{M}\mathbf{X}^T))^{-1}\mathbf{X}\mathbf{\Lambda}^T\mathbf{U}^T \tag{4.54}$$

$$= (\mathbf{X}(\mathbf{\Lambda}^T\mathbf{\Lambda} + \alpha^2\mathbf{M}^T\mathbf{M})\mathbf{X}^T)^{-1}\mathbf{X}\mathbf{\Lambda}^T\mathbf{U}^T \tag{4.55}$$

$$= \mathbf{X}^{-T}(\mathbf{\Lambda}^T\mathbf{\Lambda} + \alpha^2\mathbf{M}^T\mathbf{M})^{-1}\mathbf{\Lambda}^T\mathbf{U}^T. \tag{4.56}$$

The model resolution matrix is then

$$\mathbf{R}_{\mathrm{m}} = \mathbf{G}^{\sharp}\mathbf{G} \tag{4.57}$$

$$= \mathbf{X}^{-T}(\mathbf{\Lambda}^{T}\mathbf{\Lambda} + \alpha^{2}\mathbf{M}^{T}\mathbf{M})^{-1}\mathbf{\Lambda}^{T}\mathbf{U}^{T}\mathbf{U}\mathbf{\Lambda}\mathbf{X}^{T} \tag{4.58}$$

$$= \mathbf{X}^{-T}\mathbf{F}\mathbf{X}^{T}, \tag{4.59}$$

where

$$\mathbf{F} = \left(\mathbf{\Lambda}^{T}\mathbf{\Lambda} + \alpha^{2}\mathbf{M}^{T}\mathbf{M}\right)^{-1}\mathbf{\Lambda}^{T}\mathbf{\Lambda} \tag{4.60}$$

is a diagonal matrix of GSVD filter factors (4.50).

Example 4.5

To examine the resolution of the Tikhonov regularized inversions of Example 4.4, we perform a spike test using (4.59). Figure 4.17 shows the effect of multiplying $\mathbf{R}_{\mathrm{m}}$ times a unit amplitude spike model (at depth 500 m) under first- and second-order Tikhonov regularization using α values of 122 and 2341, respectively. These curves can equivalently be conceptualized as rows/columns of the full resolution matrix at the index corresponding to 500 m. The spike test results indicate that these Tikhonov regularized solutions are smoothed versions of the spike model. Under first- or second-order regularization, the resolution of various model features will depend critically on how smooth or rough these features are in the true model. In Figures 4.13 and 4.15, the higher-order solutions recover the true model better because the true model is smooth. Conversely, the spike model is not well recovered because of its rapid variation.

Figure 4.17 The model resolution matrix $\mathbf{R}_{\mathrm{m}}$ multiplied times the spike model for the first- and second-order regularized solutions of Example 4.4.

4.6. THE TGSVD METHOD

In the discussion of the SVD in Chapter 3, we examined the TSVD method of regularization, which rejects model space basis vectors associated with smaller singular values. Equivalently, this can be thought of as a damped SVD solution in which filter factors of one are used for basis vectors associated with larger singular values, and filter factors of zero are used for basis vectors associated with smaller singular values. This approach can be extended to the GSVD solution (4.49) to produce a **truncated generalized singular value decomposition** or **TGSVD** solution. In the TGSVD solution we simply assign filter factors (4.50) of one to the q largest generalized singular values terms in the sum to obtain

$$\mathbf{m}_{q,L} = \sum_{i=n-q+1}^{n} \frac{\mathbf{U}_{\cdot,i+k}^{T}\mathbf{d}}{\lambda_i} Y_{\cdot,i} \tag{4.61}$$

Example 4.6

Applying the TGSVD method to the VSP problem, we find L-curve corners near $q = 8$ in the first-order case shown in Figure 4.18, and $q = 7$ in the second-order case shown in Figure 4.19. Examining the filter factors obtained for the corresponding Tikhonov solutions shown in Figure 4.16, we find that they decay precipitously with decreasing index near these locations. Figures 4.20 and 4.21 show the corresponding TGSVD solutions.
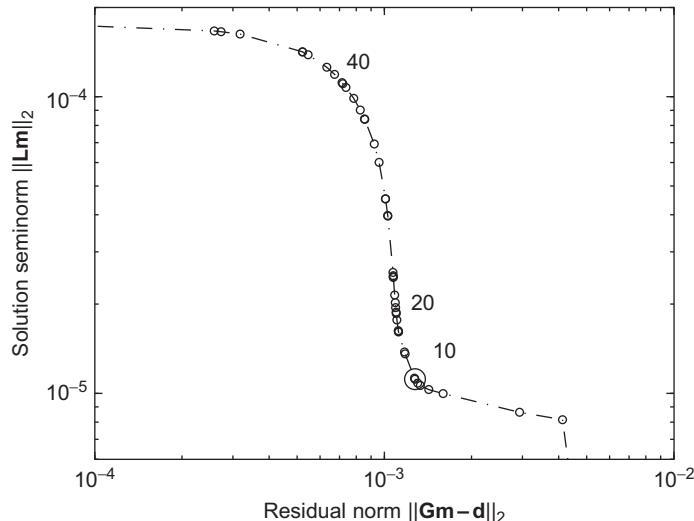
Figure 4.18 TGSVD L-curve for the VSP problem as a function of q for first-order regularization with the $q = 8$ solution indicated.
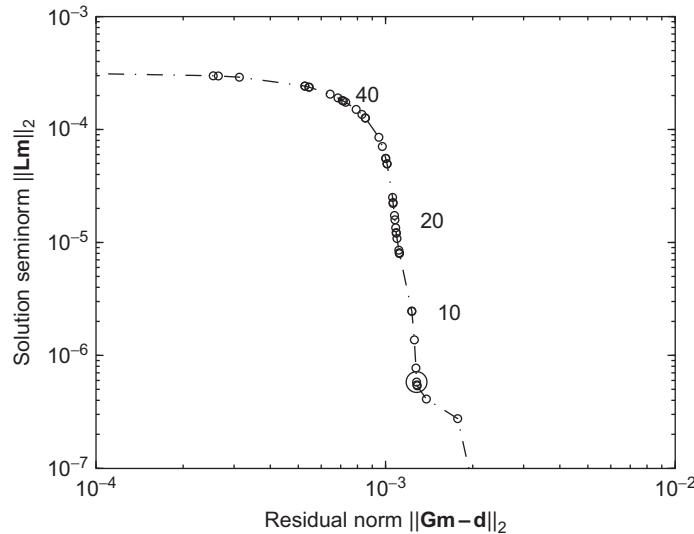
Figure 4.19 TGSVD L-curve for the VSP problem as a function of q for second-order regularization with the $q = 7$ solution indicated.

Figure 4.20 TGSVD solution of the VSP problem, $q = 8$, first-order regularization, shown in comparison with the true model.

The model recovery is comparable to that obtained with the Tikhonov method. The 2-norms of the difference between the first- and second-order solutions and the true model are approximately 1.0×10^{-2} s/km and 7.1×10^{-3} s/km, respectively, which are similar to the Tikhonov solutions in Example 4.4.

Figure 4.21 TGSVD solution of the VSP problem, $q = 7$, second-order regularization.

4.7. GENERALIZED CROSS-VALIDATION

Generalized cross-validation (GCV) is an alternative method for selecting a regularization parameter, α, that has a number of desirable statistical properties.

In ordinary or "leave-one-out" cross-validation, we consider the models that are obtained by leaving one of the m data points out of the fitting process. Consider the modified Tikhonov regularization problem in which we ignore a data point d_k,

$$\min \sum_{i \neq k} ((\mathbf{Gm})_i - d_i)^2 + \alpha^2 \|\mathbf{Lm}\|_2^2 \tag{4.62}$$

Call the solution to this problem $\mathbf{m}_{\alpha,L}^{[k]}$, where the superscript indicates that d_k was left out of the computation. Ideally, the model $\mathbf{m}_{\alpha,L}^{[k]}$ would accurately predict the missing data value d_k. In the leave-one-out approach, we select the regularization parameter α so as to minimize the predictive errors for all k:

$$\min g(\alpha) = \frac{1}{m} \sum_{k=1}^{m} ((\mathbf{Gm}_{\alpha,L}^{[k]})_k - d_k)^2. \tag{4.63}$$

Unfortunately, computing $g(\alpha)$ involves solving m problems of the form (4.62). Generalized cross-validation is a way to speed up this computation.

First, let

$$\tilde{d}_i = \begin{cases} (\mathbf{Gm}_{\alpha,L}^{[k]})_k & i = k \\ d_i & i \neq k. \end{cases} \tag{4.64}$$

Note that because $(\mathbf{Gm}_{\alpha,L}^{[k]})_k = \tilde{d}_k$, $\mathbf{m}_{\alpha,L}^{[k]}$ also solves

$$\min((\mathbf{Gm})_k - \tilde{d}_k)^2 + \sum_{i \neq k}((\mathbf{Gm})_i - \tilde{d}_i)^2 + \alpha^2\|\mathbf{Lm}\|_2^2, \tag{4.65}$$

which is equivalent to

$$\min \|\mathbf{Gm} - \tilde{\mathbf{d}}\|_2^2 + \alpha^2\|\mathbf{Lm}\|_2^2. \tag{4.66}$$

This result is known as the **leave-one-out lemma**. By this lemma,

$$\mathbf{m}_{\alpha,L}^{[k]} = \mathbf{G}^\sharp \tilde{\mathbf{d}}. \tag{4.67}$$

We will use (4.67) to eliminate $\mathbf{m}_{\alpha,L}^{[k]}$ from (4.63), because

$$\frac{(\mathbf{GG}^\sharp \tilde{\mathbf{d}})_k - (\mathbf{GG}^\sharp \mathbf{d})_k}{\tilde{d}_k - d_k} = (\mathbf{GG}^\sharp)_{k,k}, \tag{4.68}$$

where $(\mathbf{GG}^\sharp)_{k,k}$ are the diagonal elements of the data resolution matrix (4.21). Subtracting both sides of the equation from 1 gives

$$\frac{\tilde{d}_k - d_k - (\mathbf{GG}^\sharp \tilde{\mathbf{d}})_k + (\mathbf{GG}^\sharp \mathbf{d})_k}{\tilde{d}_k - d_k} = 1 - (\mathbf{GG}^\sharp)_{k,k}. \tag{4.69}$$

Since $(\mathbf{GG}^\sharp \mathbf{d})_k = (\mathbf{Gm}_{\alpha,L})_k$, $(\mathbf{GG}^\sharp \tilde{\mathbf{d}})_k = \tilde{d}_k$, and $(\mathbf{Gm}_{\alpha,L}^{[k]})_k = \tilde{d}_k$, equation (4.69) simplifies to

$$\frac{(\mathbf{Gm}_{\alpha,L})_k - d_k}{(\mathbf{Gm}_{\alpha,L}^{[k]})_k - d_k} = 1 - (\mathbf{GG}^\sharp)_{k,k}. \tag{4.70}$$

Rearranging this formula and substituting into (4.63), we obtain

$$g(\alpha) = \frac{1}{m} \sum_{k=1}^{m} \left(\frac{(\mathbf{Gm}_{\alpha,L})_k - d_k}{1 - (\mathbf{GG}^\sharp)_{k,k}} \right)^2. \tag{4.71}$$

We can simplify the formula further by replacing the $(\mathbf{GG}^\sharp)_{k,k}$ with the average value

$$(\mathbf{GG}^\sharp)_{k,k} \approx \frac{1}{m}\mathrm{Tr}(\mathbf{GG}^\sharp), \tag{4.72}$$

which gives an expression for the (4.63) that can be evaluated as a function of α:

$$g(\alpha) \approx \frac{1}{m}\sum_{k=1}^{m}\frac{((\mathbf{Gm}_{\alpha,L})_k - d_k)^2}{(\frac{1}{m}(m - \mathrm{Tr}(\mathbf{GG}^\sharp)))^2} \tag{4.73}$$

$$= \frac{m\|\mathbf{Gm}_{\alpha,L} - \mathbf{d}\|_2^2}{\mathrm{Tr}(\mathbf{I} - \mathbf{GG}^\sharp)^2}. \tag{4.74}$$

It can be shown that under reasonable assumptions regarding the noise and smoothness of $\mathbf{m}_{\mathrm{true}}$, the value of α that minimizes (4.74) approaches the value that minimizes $E[\mathbf{Gm}_{\alpha,L} - \mathbf{d}_{\mathrm{true}}]$ as the number of data points m goes to infinity, and that under the same assumptions, $E[\|\mathbf{m}_{\mathrm{true}} - \mathbf{m}_{\alpha,L}\|_2]$ goes to 0 as m goes to infinity [36, 170]. In practice, the size of the data set is fixed in advance, so the limit is not directly applicable. However, these results provide a theoretical justification for using the GCV method to select the Tikhonov regularization parameter.

Example 4.7

Figures 4.22 and 4.23 show $g(\alpha)$ for the VSP test problem, using first- and second-order Tikhonov regularization, respectively. Respective GCV function (4.74) minima occur
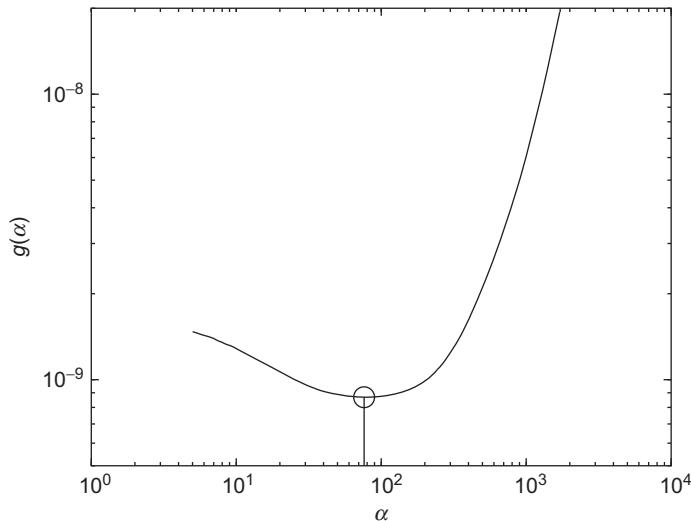
Figure 4.22 GCV curve for the VSP problem and its minimum, first-order regularization.
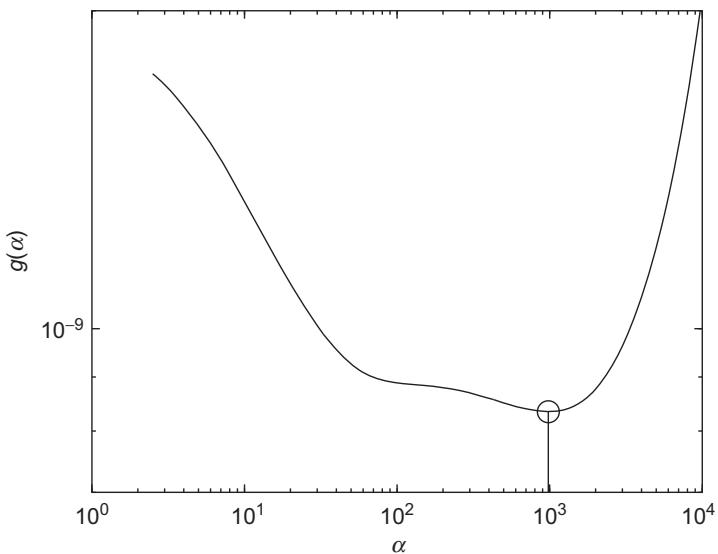
Figure 4.23 GCV curve for the VSP problem and its minimum, second-order regularization.

Figure 4.24 GCV solution for the VSP problem, first-order, $\alpha = 76.3$, shown in comparison with the true model.
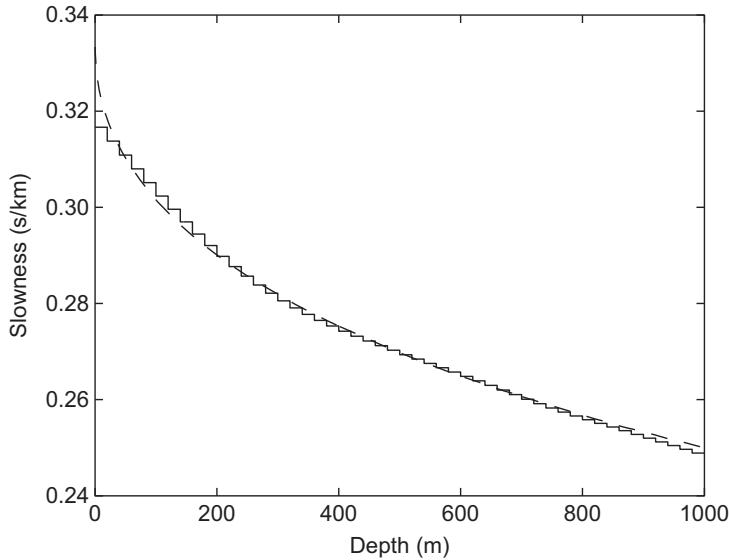
Figure 4.25 GCV solution for the VSP problem, second-order, $\alpha = 981$, shown in comparison with the true model.

near $\alpha = 76.3$ and $\alpha = 981$, which are somewhat smaller than the α values estimated previously using the L-curve (Example 4.4). The corresponding models (Figures 4.24 and 4.25) thus have somewhat larger seminorms.

4.8. ERROR BOUNDS

We next present two theoretical results that help to address the accuracy of Tikhonov regularization solutions. We will present these results in a simplified form, covering only zeroth-order Tikhonov regularization.

The first question is whether for a particular value of the regularization parameter, α, we can establish a bound on the sensitivity of the regularized solution to the noise in the observed data $\mathbf{d}$ and/or errors in the system matrix $\mathbf{G}$. This would provide a sort of condition number for the inverse problem. Note that this does not tell us how far the regularized solution is from the true model, since Tikhonov regularization has introduced a bias in the solution. Under Tikhonov regularization with a nonzero α, we would not obtain the true model even if the noise was 0.

The following theorem gives a bound for zeroth-order Tikhonov regularization [65]. A slightly more complicated formula is available for higher-order Tikhonov regularization [65].

Theorem 4.1 *Suppose that the problems*

$$\min \ \|\mathbf{Gm} - \mathbf{d}\|_2^2 + \alpha^2 \|\mathbf{m}\|_2^2 \tag{4.75}$$

and

$$\min \ \|\bar{\mathbf{G}}\mathbf{m} - \bar{\mathbf{d}}\|_2^2 + \alpha^2 \|\mathbf{m}\|_2^2 \tag{4.76}$$

are solved to obtain $\mathbf{m}_\alpha$ *and* $\bar{\mathbf{m}}_\alpha$. *Then*

$$\frac{\|\mathbf{m}_\alpha - \bar{\mathbf{m}}_\alpha\|_2}{\|\mathbf{m}_\alpha\|_2} \leq \frac{\bar{\kappa}_\alpha}{1 - \epsilon \bar{\kappa}_\alpha} \left(2\epsilon + \frac{\|\mathbf{e}\|_2}{\|\mathbf{d}_\alpha\|_2} + \epsilon \bar{\kappa}_\alpha \frac{\|\mathbf{r}_\alpha\|_2}{\|\mathbf{d}_\alpha\|_2} \right) \tag{4.77}$$

where

$$\bar{\kappa}_\alpha = \frac{\|\mathbf{G}\|_2}{\alpha} \tag{4.78}$$

$$\mathbf{E} = \mathbf{G} - \bar{\mathbf{G}} \tag{4.79}$$

$$\mathbf{e} = \mathbf{d} - \bar{\mathbf{d}} \tag{4.80}$$

$$\epsilon = \frac{\|\mathbf{E}\|_2}{\|\mathbf{G}\|_2} \tag{4.81}$$

$$\mathbf{d}_\alpha = \mathbf{Gm}_\alpha \tag{4.82}$$

and

$$\mathbf{r}_\alpha = \mathbf{d} - \mathbf{d}_\alpha. \tag{4.83}$$

In the particular case when $\mathbf{G} = \bar{\mathbf{G}}$, and the only difference between the two problems is $\mathbf{e} = \mathbf{d} - \bar{\mathbf{d}}$, the inequality becomes even simpler

$$\frac{\|\mathbf{m}_\alpha - \bar{\mathbf{m}}_\alpha\|_2}{\|\mathbf{m}_\alpha\|_2} \leq \bar{\kappa}_\alpha \frac{\|\mathbf{e}\|_2}{\|\mathbf{d}_\alpha\|_2}. \tag{4.84}$$

The condition number $\bar{\kappa}_\alpha$ is inversely proportional to α. Thus increasing α will decrease the sensitivity of the solution to perturbations in the data. Of course, increasing α also increases the error in the solution due to regularization bias and decreases resolution.

The second question is whether we can establish any sort of bound on the norm of the difference between the regularized solution and the true model. This bound would incorporate both sensitivity to noise and the bias introduced by Tikhonov regularization. Such a bound must of course depend on the magnitude of the noise in the data. It must also depend on the particular regularization parameter chosen. Tikhonov developed a beautiful theorem that addresses this question in the context of inverse problems involving IFKs [161]. More recently, Neumaier has developed a version of Tikhonov's theorem that can be applied directly to discretized problems [116].

Recall that in a discrete ill-posed linear inverse problem, the matrix $\mathbf{G}$ commonly has a smoothing effect, in that when we multiply $\mathbf{Gm}$, the result is smoother than $\mathbf{m}$. Similarly, if we multiply $\mathbf{Gm}$ by $\mathbf{G}^T$, the result will be even smoother than $\mathbf{Gm}$. This smoothing in such problems is a consequence of the fact that the singular vectors corresponding to the larger singular values of $\mathbf{G}$ are smooth. Note, however, that this is not a universal property of all matrices; for example, if $\mathbf{G}$ is a matrix that approximates the differentiation operator, then $\mathbf{Gm}$ will be rougher than $\mathbf{m}$.

For discrete ill-posed problems, models in the range of $\mathbf{G}^T$ will form a relatively smooth subspace of all possible models in R^n. Models in this subspace can be written using the range of $\mathbf{G}^T$ as a basis as $\mathbf{m} = \mathbf{G}^T\mathbf{w}$, using the coefficients $\mathbf{w}$. Furthermore, models in the range of $\mathbf{G}^T\mathbf{G}$ form a subspace of $R(\mathbf{G}^T)$, since any model in $R(\mathbf{G}^T\mathbf{G})$ can be written as $\mathbf{m} = \mathbf{G}^T(\mathbf{Gw})$, which is a linear combination of columns of $\mathbf{G}^T$. Because of the smoothing effect of $\mathbf{G}$ and $\mathbf{G}^T$, we would expect these models to be even smoother than the models in $R(\mathbf{G}^T)$. We could construct smaller subspaces of R^n that contain even smoother models, but it turns out that with zeroth-order Tikhonov regularization these are the only subspaces of interest.

There is another way to see that models in $R(\mathbf{G}^T)$ will be relatively smooth. Recall that the vectors $\mathbf{V}_{\cdot,1}$, $\mathbf{V}_{\cdot,2}$, . . . , $\mathbf{V}_{\cdot,p}$ from the SVD of $\mathbf{G}$ form an orthonormal basis for $R(\mathbf{G}^T)$. For discrete ill-posed problems, we know from Chapter 3 that these basis vectors will be relatively smooth, so linear combinations of these vectors in $R(\mathbf{G}^T)$ should be smooth.

The following theorem gives a bound on the total error including bias due to regularization and error due to noise in the data for zeroth-order Tikhonov regularization [116].

Theorem 4.2 *Suppose that we use zeroth-order Tikhonov regularization with regularization parameter α to solve $\mathbf{Gm} = \mathbf{d}$, and that $\mathbf{m}_{\text{true}}$ can be expressed as one of the following distinct cases, for some $\mathbf{w}$, and as parameterized by p,*

$$\mathbf{m}_{\text{true}} = \begin{cases} \mathbf{G}^T\mathbf{w} & p = 1 \\ \mathbf{G}^T\mathbf{Gw} & p = 2 \end{cases} \tag{4.85}$$

and that

$$\|\mathbf{Gm}_{\text{true}} - \mathbf{d}\|_2 \leq \Delta\|\mathbf{w}\|_2 \tag{4.86}$$

for some $\Delta > 0$. Then

$$\|\mathbf{m}_{\text{true}} - \mathbf{G}^\sharp\mathbf{d}\|_2 \leq \left(\frac{\Delta}{2\alpha} + \gamma\alpha^p\right)\|\mathbf{w}\|_2, \tag{4.87}$$

where

$$\gamma = \begin{cases} 1/2 & p = 1 \\ 1 & p = 2 \end{cases}. \tag{4.88}$$

Furthermore, if we begin with the bound

$$\|\mathbf{G}\mathbf{m}_{\text{true}} - \mathbf{d}\|_2 \leq \delta, \tag{4.89}$$

we can let

$$\Delta = \frac{\delta}{\|\mathbf{w}\|_2}. \tag{4.90}$$

Under this condition the optimal value of α is

$$\hat{\alpha} = \left(\frac{\Delta}{2\gamma p}\right)^{\frac{1}{p+1}} = O(\Delta^{\frac{1}{p+1}}). \tag{4.91}$$

With this choice of α,

$$\Delta = 2\gamma p \hat{\alpha}^{p+1} \tag{4.92}$$

and the error bound simplifies to

$$\|\mathbf{m}_{\text{true}} - \mathbf{G}_{\hat{\alpha}}^{\sharp}\mathbf{d}\|_2 \leq \gamma(p+1)\hat{\alpha}^p \|\mathbf{w}\|_2 = O(\Delta^{\frac{p}{p+1}}). \tag{4.93}$$

Theorem 4.2 tells us that the error in the Tikhonov regularization solution depends on both the noise level δ and on the regularization parameter α. For larger values of α, the error due to regularization in recovering the true model will be dominant. For smaller values of α, the error due to noise in the data will be dominant. We seek the optimal value of α that best balances these effects. Using the optimal α, (4.93) indicates that we can obtain an error bound of $O(\Delta^{2/3})$ if $p = 2$, and an error bound of $O(\Delta^{1/2})$ if $p = 1$.

Of course, the above result can only be applied when our true model lives in the restricted subspace of $R(\mathbf{G}^T)$. In practice, even if the model does lie in $R(\mathbf{G}^T)$, the vector $\mathbf{w}$ could have a very large norm, making the bound uninformative.

Thus, applying this theorem in a quantitative fashion is typically impractical. However, the theorem does provide some useful insight into the ability of Tikhonov regularization to recover a true model. The first point is that the accuracy of the regularized solution depends very much on the smoothness of the true model. If $\mathbf{m}_{\text{true}}$ is not smooth, then Tikhonov regularization simply will not give an accurate solution. Furthermore, if the model $\mathbf{m}_{\text{true}}$ is smooth, then we can hope for an error in the Tikhonov

regularized solution which is $O(\delta^{1/2})$ or $O(\delta^{2/3})$. Another way of saying this is that we can hope *at best* for an answer with about two thirds as many correct significant digits as the data.

Example 4.8

Recall the Shaw problem previously considered in Examples 4.1 and 4.3. Because $\mathbf{G}^T$ is a nonsingular matrix, the spike model should lie in $R(\mathbf{G}^T)$. However, $\mathbf{G}^T$ is numerically singular due to the ill–posedness of the problem, and the spike model thus lies outside of the effective range of $\mathbf{G}^T$. Any unregularized attempt to find $\mathbf{w}$ produces a meaningless answer due to numerical instability. Because the spike model does not lie in $R(\mathbf{G}^T)$, Theorem 4.2 does not apply.

Figure 4.26 shows a smooth model that does lie in the range of $\mathbf{G}^T$. For this model we constructed a synthetic data set with noise as before at $\delta = 4.47 \times 10^{-6}$. Equation (4.93) suggests using $\alpha = 8.0 \times 10^{-4}$. The resulting error bound is 8.0×10^{-4}, while the actual norm of the model error is 6.6×10^{-4}. Here the data were accurate to roughly six digits, while the solution was accurate to roughly four digits. Figure 4.27 shows the reconstruction of the model with $N(0, (1.0 \times 10^{-6})^2)$ noise added to the data vector, showing that the solution is well recovered. This example once again demonstrates the importance of smoothness in the true model in determining how accurately it can be reconstructed with Tikhonov regularization.
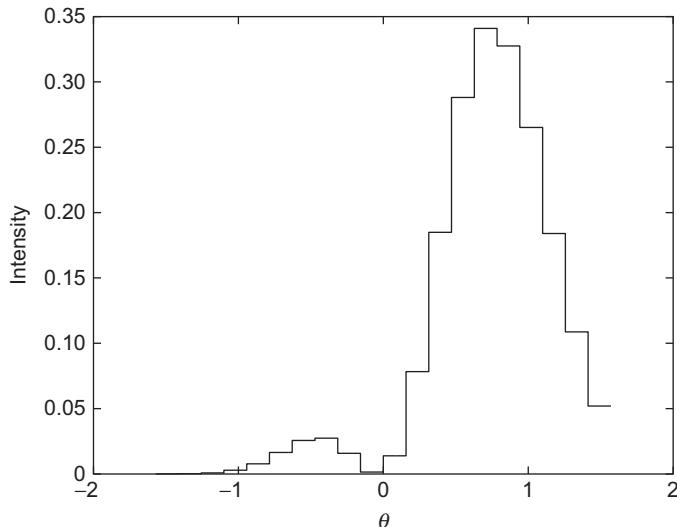
Figure 4.26 A smooth model in $R(\mathbf{G}^T)$.
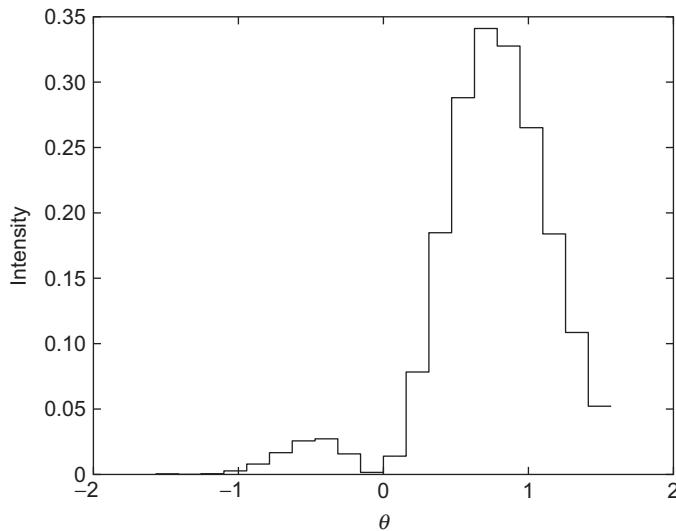
Figure 4.27 Reconstruction of the smooth model with $N(0, (1.0 \times 10^{-6})^2)$ noise.

4.9. EXERCISES

1. Use the method of Lagrange multipliers (Appendix C) to derive the damped least squares problem (4.4) from the discrepancy principle problem (4.2), and demonstrate that (4.4) can be written as (4.5).

2. Consider the integral equation and data set from Problem 3.5. You can find a copy of this data set in the file **ifk.mat**.

   a. Discretize the problem using simple collocation.

   b. Using the data supplied, and assuming that the numbers are accurate to four significant figures, determine a reasonable bound δ for the misfit.

   c. Use zeroth-order Tikhonov regularization to solve the problem. Use GCV, the discrepancy principle, and the L-curve criterion to pick the regularization parameter.

   d. Use first-order Tikhonov regularization to solve the problem. Use GCV, the discrepancy principle, and the L-curve criterion to pick the regularization parameter.

   e. Use second-order Tikhonov regularization to solve the problem. Use GCV, the discrepancy principle, and the L-curve criterion to pick the regularization parameter.

   f. Analyze the resolution of your solutions. Are the features you see in your inverse solutions unambiguously real? Interpret your results. Describe the size and location of any significant features in the solution.

3. Consider the following problem in **cross–well tomography**. Two vertical wells
   are located 1600 m apart. A seismic source is inserted in one well at depths of
   50, 150, . . . , 1550 m. A string of receivers is inserted in the other well at depths
   of 50 m, 150 m, . . . , 1550 m. See Figure 4.28. For each source–receiver pair, a
   travel time is recorded, with a measurement standard deviation of 0.5 ms. There are
   256 ray paths and 256 corresponding data points. We wish to determine the velocity
   structure in the two–dimensional plane between the two wells.

    Discretizing the problem into a 16 by 16 grid of 100 meter by 100 meter blocks
   gives 256 model parameters. The **G** matrix and noisy data, **d**, for this problem
   (assuming straight ray paths) are in the file **crosswell.mat**. The order of parameter
   indexing from the slowness grid to the model vector is row by row (e.g., Example
   1.12).

   a. Use the TSVD to solve this inverse problem using an L–curve. Plot the result.
   b. Use zeroth-order Tikhonov regularization to solve this problem and plot your
      solution. Explain why it is difficult to use the discrepancy principle to select the
      regularization parameter. Use the L–curve criterion to select your regularization
      parameter. Plot the L–curve as well as your solution.
   c. Use second-order Tikhonov regularization to solve this problem and plot your
      solution. Because this is a two–dimensional problem, you will need to implement
      a finite-difference approximation to the Laplacian (second derivative in the

Figure 4.28 Cross-well tomography problem, showing block discretization, block numbering conven-
tion, and one set of straight source-receiver ray paths.

horizontal direction plus the second derivative in the vertical direction) in the roughening matrix. The **L** matrix can be generated using the following MATLAB code:

```
L=zeros(14*14,256);
k=1;
for i=2:15,
  for j=2:15,
    M=zeros(16,16);
    M(i,j)=-4;
    M(i,j+1)=1;
    M(i,j-1)=1;
    M(i+1,j)=1;
    M(i-1,j)=1;
    L(k,:)=reshape(M,256,1)';
    k=k+1;
  end
end
```

What, if any, problems did you have in using the L-curve criterion on this problem? Plot the L-curve as well as your solution.

d. Discuss your results. If vertical bands appeared in some of your solutions, can you explain why?

4. Apply second-order Tikhonov regularization to solve the problem of finding a smooth curve that approximately fits a set of noisy data points. Write a MAT-LAB program to find a function specified at the 191 points $x = 1, 1.1, 1.2, \ldots, 20$ that approximately fits the 20 data points specified at $x = 1, 2, 3, \ldots, 20$ given in the file **interpdata.mat**. Construct an appropriate 20 by 191 **G** matrix, and use the library function **get_l_rough** to obtain the necessary second-order roughening matrix, **L**. Produce solutions for regularization parameter values of $\alpha = 0.2, 0.4, 0.6, \ldots, 10$ and show the tradeoff curve between 2-norm data misfit and model seminorm on a linear-linear plot. If the data noise is independent and normally distributed with a standard deviation of 0.3, use the discrepancy principle to find and plot an optimal interpolated curve along with the data points. What is the χ^2 value of this solution? Is it reasonable?

5. In some situations it is appropriate to bias the regularized solution toward a particular model $\mathbf{m}_0$. In this case, we would solve

$$\min \ \|\mathbf{Gm} - \mathbf{d}\|_2^2 + \alpha^2 \|\mathbf{L}(\mathbf{m} - \mathbf{m}_0)\|_2^2. \tag{4.94}$$

Write this as an ordinary linear least squares problem. What are the normal equations? Can you find a solution for this problem using the GSVD?

4.10. NOTES AND FURTHER READING

Hansen's book [65] is a very complete reference on the linear algebra of Tikhonov regularization. Arnold Neumaier's tutorial [116] is also a very useful reference. Two other surveys of Tikhonov regularization are [44, 45]. Vogel [169] includes an extensive discussion of methods for selecting the regularization parameter. Hansen's MATLAB Regularization Tools [64] is a collection of software for performing regularization within MATLAB.

The GSVD was first defined by Van Loan [101]. References on the GSVD and algorithms for computing the GSVD include [3, 53, 62, 65].

Selecting the regularization parameter is an important problem in both theory and practice. Much of the literature on functional analytic approaches assumes that the noise level is known. When the noise level is known, the discrepancy principle provides a scheme for selecting the regularization parameter for ill–posed problems that is convergent in the sense that, in the limit as the noise level goes to zero, the regularized solution goes to $\mathbf{m}_{\text{true}}$ [45].

In practice, the noise level is often unknown, so there has been a great deal of interest in schemes for selecting the regularization parameter without its prior knowledge. The two most popular approaches are the L–curve method and GCV. The L–curve method was introduced by Hansen [63, 65]. GCV was introduced by Craven and Wahba [36, 170]. The formula for GCV given here is very expensive to compute for large problems. A GCV algorithm for large-scale problems is given by Golub and von Matt [54]. Vogel has shown that the L–curve method can fail to converge [168]. It can be shown that no scheme that depends only on noisy data without knowledge of the noise level can be convergent in the limit as the noise level goes to zero [45].

Within statistics, poorly conditioned linear regression problems are said to suffer from "multicollinearity." A method called "ridge regression," which is identical to Tikhonov regularization, is often used to deal with such problems [40]. Statisticians also use a method called "principal components regression" (PCR), which is identical to the TSVD method [113].

This page intentionally left blank

Discretizing Problems Using Basis Functions

Synopsis

Techniques for discretizing continuous inverse problems characterized by Fredholm integral equations of the first kind using continuous basis functions are discussed, both for general basis functions and for representers. The Gram matrix is defined and used in analyzing these problems. The method of Backus and Gilbert is also introduced.

5.1. DISCRETIZATION BY EXPANSION OF THE MODEL

To this point, we have discretized continuous inverse problems using the collocation scheme discussed in Chapter 1. In this chapter, we discuss an alternative approach in which we expand the unknown model $m(x)$ in terms of a set of basis functions.

We begin with a Fredholm integral equation of the first kind,

$$d(x) = \int_a^b g(x, \xi) m(\xi) \ d\xi. \tag{5.1}$$

Note from Section A.11 that we can generalize linear algebra, including the concepts of linear independence, dot products, orthogonality, and the 2-norm, to spaces defined by functions. We will assume that our model $m(x)$ lies in the space $L_2(a, b)$ of functions that are square integrable on $[a, b]$.

In $L_2(a, b)$, the dot product of two functions $f(x)$ and $g(x)$ is defined as

$$\langle f(x), g(x) \rangle = \int_a^b f(x)g(x) \ dx, \tag{5.2}$$

the 2-norm of $f(x)$ is

$$\| f(x) \|_2 = \sqrt{\int_a^b f(x)^2 \ dx}, \tag{5.3}$$

and the distance between two functions under this norm is

$$\|f(x) - g(x)\|_2 = \sqrt{\int_a^b (f(x) - g(x))^2 \ dx}. \tag{5.4}$$

Ideally, we could try to solve the Tikhonov regularization problem

$$\min \ \|m(x)\|_2$$
$$\left\| \int_a^b g(x, \xi) m(\xi) \ d\xi - d(x) \right\|_2 \leq \Delta. \tag{5.5}$$

However, in practical cases, we do not fully know $d(x)$ in the sense that we normally have a finite number of data points $d_1 = d(x_1)$, $d_2 = d(x_2)$, . . . , $d_m = d(x_m)$ arranged in a vector $\mathbf{d}$. An appropriate approach is thus to minimize the model norm $\|m(x)\|_2$ subject to a constraint on the misfit at the data points:

$$\min \ \|m(x)\|_2$$
$$\|\mathbf{g}(m(x)) - \mathbf{d}\|_2 \leq \delta, \tag{5.6}$$

where

$$\mathbf{g}(m) = \begin{bmatrix} \int_a^b g(x_1, \xi) m(\xi) \ d\xi \\ \int_a^b g(x_2, \xi) m(\xi) \ d\xi \\ \vdots \\ \int_a^b g(x_m, \xi) m(\xi) \ d\xi \end{bmatrix}. \tag{5.7}$$

Consider a finite dimensional subspace H of $L_2(a, b)$, with the basis

$$h_1(x), \ h_2(x), \ . \ . \ . \ , \ h_n(x) \tag{5.8}$$

so that any model $m(x)$ in the subspace H can be written uniquely as

$$m(x) = \sum_{j=1}^n \beta_j h_j(x). \tag{5.9}$$

We will seek the model $m(x)$ in H that has minimum norm and adequately fits the data.

Substituting (5.9) into (5.1), and considering only those points x_i where we have data, we obtain

$$d(x_i) = \int_a^b g(x_i, \xi) \left(\sum_{j=1}^n \beta_j h_j(\xi) \right) d\xi \quad i = 1, 2, \ . \ . \ . \ , \ m. \tag{5.10}$$

To simplify the notation, let

$$g_i(\xi) = g(x_i, \xi) \quad i = 1, 2, \ldots, m \tag{5.11}$$

and

$$d_i = d(x_i) \quad i = 1, 2, \ldots, m. \tag{5.12}$$

We interchange the integral and sum to get

$$d_i = \sum_{j=1}^{n} \beta_j \int_a^b g_i(x) h_j(x) \ dx \quad i = 1, 2, \ldots, m. \tag{5.13}$$

If we let $\mathbf{G}$ be the $m \times n$ matrix with entries

$$G_{i,j} = \int_a^b g_i(x) h_j(x) \ dx, \tag{5.14}$$

then the discretized inverse problem can be written as

$$\mathbf{G}\boldsymbol{\beta} = \mathbf{d}, \tag{5.15}$$

which can be solved to find the functional expansion coefficients $\boldsymbol{\beta}$ in (5.9) to construct the desired model.

Because of noise in the data, we do not normally desire, nor will we be generally able, to solve (5.15) exactly. Furthermore, we typically need to regularize our solution to the inverse problem. We could try to solve

$$\begin{array}{c} \min \ \|\boldsymbol{\beta}\|_2 \\ \|\mathbf{G}\boldsymbol{\beta} - \mathbf{d}\|_2 \le \delta. \end{array} \tag{5.16}$$

However, $\|\boldsymbol{\beta}\|_2$ is generally not proportional to $\|m(x)\|_2$. Instead, note that

$$\begin{aligned} \|m(x)\|_2^2 &= \int_a^b m(x)^2 \ dx \\ &= \int_a^b \left(\sum_{i=1}^{n} \beta_i h_i(x) \right) \left(\sum_{j=1}^{n} \beta_j h_j(x) \right) \ dx \\ &= \sum_{i=1}^{n} \sum_{j=1}^{n} \beta_i \beta_j \int_a^b h_i(x) h_j(x) \ dx. \end{aligned} \tag{5.17}$$

If we let $\mathbf{H}$ be the n by n matrix with

$$H_{i,j} = \int_a^b h_i(x) h_j(x) \ dx, \tag{5.18}$$

then

$$\|m(x)\|_2^2 = \boldsymbol{\beta}^T \mathbf{H} \boldsymbol{\beta}. \tag{5.19}$$

The matrix $\mathbf{H}$ is called the **Gram matrix** of the functions $h_i(x)$. It is relatively easy to show that $\mathbf{H}$ is a symmetric and positive definite matrix, and thus has a Cholesky factorization

$$\mathbf{H} = \mathbf{R}^T \mathbf{R}, \tag{5.20}$$

so that

$$\begin{aligned} \|m(x)\|_2^2 &= \boldsymbol{\beta}^T \mathbf{R}^T \mathbf{R} \boldsymbol{\beta} \\ &= \|\mathbf{R}\boldsymbol{\beta}\|_2^2. \end{aligned} \tag{5.21}$$

We can write the Tikhonov regularization problem as

$$\begin{aligned} &\min \|\mathbf{R}\boldsymbol{\beta}\|_2 \\ &\|\mathbf{G}\boldsymbol{\beta} - \mathbf{d}\|_2 \leq \delta. \end{aligned} \tag{5.22}$$

We can use the Lagrange multiplier technique (Appendix C) to turn this into an unconstrained minimization problem

$$\min \|\mathbf{G}\boldsymbol{\beta} - \mathbf{d}\|_2^2 + \alpha^2 \|\mathbf{R}\boldsymbol{\beta}\|_2^2, \tag{5.23}$$

where the regularization parameter α is selected so that we obtain a solution that adequately fits the data. This problem can easily be solved using the Cholesky factorization of the normal equations, the generalized singular value decomposition, or iterative methods discussed in Chapter 6.

Note that if we choose an orthonormal basis of functions, then

$$H_{i,j} = \int_a^b h_i(x)h_j(x) \ dx = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases} \tag{5.24}$$

or $\mathbf{H} = \mathbf{R} = \mathbf{I}$, and the minimization problem simplifies to (5.16).

To minimize a model derivative norm (e.g., $\|m''(x)\|_2$ instead of $\|m(x)\|_2$), note that because

$$m''(x) = \sum_{j=1}^n \beta_j h_j''(x), \tag{5.25}$$

we can simply use the second derivative Gram matrix

$$H_{i,j} = \int_a^b h_i''(x)h_j''(x) \ dx \tag{5.26}$$

instead of (5.18).

The simple collocation scheme described in Chapter 1 can also be thought of as a special case of this approach, in which the $h_i(x)$ basis functions are "boxcar" functions centered at the measurement points x_i. A variety of other bases have been used in practice. For example, sine and cosine functions can be used to produce solutions that are Fourier series, and, for problems involving spherical symmetries, the spherical harmonic functions are often used. In recent years, many researchers have investigated the use of wavelet bases in the solution of inverse problems.

The choice of the finite dimensional subspace H and its basis is critical to the success of this method in solving practical problems. As we have seen, the method finds the model $m(x)$ in the subspace H that minimizes $\|m(x)\|_2$ subject to fitting the data. There is no guarantee that the model we obtain will actually have the minimum norm over the entire space $L_2(a, b)$. However, if the subspace includes a sufficiently large class of functions, the resulting solution may be an adequate approximation to the solution of (5.6).

5.2. USING REPRESENTERS AS BASIS FUNCTIONS

A particularly interesting choice of basis is to use the representers (1.33) themselves as the basis functions. This approach results in a solution that does effectively minimize $\|m(x)\|_2$ over the entire space $L_2(a, b)$.

Consider the space G spanned by the representers

$$g_j(\xi) = g(x_j, \xi) \quad j = 1, 2, \ldots, m. \tag{5.27}$$

Let $G^\perp$ be the perpendicular complement of G. $G^\perp$ consists of the functions $f(x)$ in $L_2(a, b)$ such that

$$\int_a^b f(x)g_j(x) \ dx = 0 \quad j = 1, 2, \ldots, m. \tag{5.28}$$

Thus every function $f(x)$ in $G^\perp$ is in the null space of the forward operator. Every function $m(x)$ in $L_2(a, b)$ can be written uniquely as

$$m(x) = m_1(x) + m_2(x) \tag{5.29}$$

where $m_1(x)$ is in G and $m_2(x)$ is in $G^\perp$. By the Pythagorean theorem,

$$\|m(x)\|_2^2 = \|m_1(x)\|_2^2 + \|m_2(x)\|_2^2. \tag{5.30}$$

Since any nonzero function $m_2(x)$ would have no effect on the fit to the data but would increase $\|m(x)\|_2$, the optimal norm-minimizing solution to (5.6) will always have $m_2(x) = 0$. This means that minimizing $\|m(x)\|_2$ over our subspace G will be equivalent to minimizing $\|m(x)\|_2$ over all of $L_2(a, b)$.

The Gram matrix of the representers, $\mathbf{\Gamma}$, has entries

$$\Gamma_{i,j} = \int\limits_a^b g_i(x)g_j(x) \ dx. \tag{5.31}$$

In terms of this matrix,

$$\|m(x)\|_2^2 = \boldsymbol{\beta}^T \mathbf{\Gamma} \boldsymbol{\beta}, \tag{5.32}$$

so that the discretized Tikhonov regularization problem (5.22) becomes

$$\begin{aligned} \min \ &\|\mathbf{R}\boldsymbol{\beta}\|_2 \\ &\|\mathbf{\Gamma}\boldsymbol{\beta} - \mathbf{d}\|_2 \le \delta, \end{aligned} \tag{5.33}$$

where $\mathbf{R}$ is the Cholesky factorization of $\mathbf{\Gamma}$.

Can we extend this approach to minimizing a derivative-based model roughness measure (e.g., $\|m''(x)\|$) as we did in (5.26)? It is still true that adding a nonzero function $m_2(x)$ from $G^\perp$ to $m(x)$ will have no effect on the fit to the data. Unfortunately, derivatives of models in G and $G^\perp$ will no longer generally satisfy the orthogonality relationship (5.30), that is,

$$\|m''(x)\|_2^2 \ne \|m_1''(x)\|_2^2 + \|m_2''(x)\|_2^2, \tag{5.34}$$

because orthogonality of two functions $f(x)$ and $g(x)$ does not imply orthogonality of $f''(x)$ and $g''(x)$. Because adding a nonzero function $m_2(x)$ from $G^\perp$ might actually decrease $\|m''(x)\|_2$, minimizing $\|m''(x)\|_2$ over our subspace G in this case will not be equivalent to minimizing over $L_2(a, b)$.

5.3. THE METHOD OF BACKUS AND GILBERT

The method of Backus and Gilbert [2, 126] is applicable to continuous linear inverse problems of the form

$$d(x) = \int\limits_a^b g(x, \xi)m(\xi) \ d\xi, \tag{5.35}$$

where we have observations at points $x_1, x_2, \ldots, x_n$. Let

$$d_j = d(x_j) \quad j = 1, 2, \ldots, m \tag{5.36}$$

and

$$g_j(\xi) = g(x_j, \xi) \quad j = 1, 2, \ldots, m. \tag{5.37}$$

Using (5.35), we can write the d_j in terms of the true model, m_{true}, as

$$d_j = \int_a^b g(x_j, \xi) m_{\text{true}}(\xi) \ d\xi$$

$$= \int_a^b g_j(\xi) m_{\text{true}}(\xi) \ d\xi.$$

(5.38)

The Backus and Gilbert method estimates $m_{\text{true}}(x)$ at some point $\hat{x}$, given the m data values d_j. The method produces model estimates that are linear combinations of the data elements

$$\hat{m} = \sum_{j=1}^m c_j d_j \approx m_{\text{true}}(\hat{x})$$

(5.39)

where the coefficients c_j are to be determined.

Combining (5.37) and (5.38) gives

$$\hat{m} = \sum_{j=1}^m c_j \int_a^b g_j(x) m_{\text{true}}(x) \ dx$$

$$= \int_a^b \left(\sum_{j=1}^m c_j g_j(x) \right) m_{\text{true}}(x) \ dx$$

(5.40)

$$= \int_a^b A(x) m_{\text{true}}(x) \ dx,$$

where

$$A(x) = \sum_{j=1}^m c_j g_j(x).$$

(5.41)

The function $A(x)$ is called an **averaging kernel**. Ideally, we would like the averaging kernel to closely approximate a delta function

$$A(x) = \delta(x - \hat{x})$$

(5.42)

because, assuming exact data, (5.40) would then produce exact agreement between the true and estimated model ($\hat{m} = m_{\text{true}}(\hat{x})$). Since this is not possible, we will instead select the coefficients so that the area under the averaging kernel is one, and so that the width of the averaging kernel around $\hat{x}$ is as small as possible.

In terms of the coefficients $\mathbf{c}$, the unit area constraint can be written as

$$\int_a^b A(x) \ dx = \int_a^b \sum_{j=1}^m c_j g_j(x) \ dx = \sum_{j=1}^m c_j \left(\int_a^b g_j(x) \ dx \right) = 1. \tag{5.43}$$

Letting

$$q_j = \int_a^b g_j(x) \ dx, \tag{5.44}$$

the unit area constraint (5.43) can be written as

$$\mathbf{q}^T \mathbf{c} = 1. \tag{5.45}$$

Averaging kernel widths can be usefully characterized in a variety of ways [126]. The most commonly used measure is the second moment of $A(x)$ about $\hat{x}$:

$$w = \int_a^b A(x)^2 (x - \hat{x})^2 \ dx. \tag{5.46}$$

In terms of the coefficients $\mathbf{c}$, this can be written using (5.43) as

$$w = \mathbf{c}^T \mathbf{H} \mathbf{c} \tag{5.47}$$

where

$$H_{j,k} = \int_a^b g_j(x) g_k(x) (x - \hat{x})^2 \ dx. \tag{5.48}$$

Now, the problem of finding the optimal coefficients can be written as

$$\begin{aligned} \min \ \mathbf{c}^T \mathbf{H} \mathbf{c} \\ \mathbf{c}^T \mathbf{q} = 1, \end{aligned} \tag{5.49}$$

which can be solved using the Lagrange multiplier technique.

In practice, the data may be noisy, and the solution may thus be unstable. For measurements with independent errors, the standard deviation of the estimate is given by

$$\mathrm{Var}(\hat{m}) = \sum_{j=1}^m c_j^2 \sigma_j^2, \tag{5.50}$$

where σ_j is the standard deviation of the jth observation.

The solution can be stabilized by adding a constraint on the variance to (5.49)

$$\min \mathbf{c}^T \mathbf{Hc}$$
$$\mathbf{q}^T \mathbf{c} = 1$$
$$\sum_{j=1}^{n} \hat{c}_j^2 \sigma_j^2 \leq \delta. \tag{5.51}$$

Again, this problem can be solved by the method of Lagrange multipliers. Smaller values of δ decrease the variance of the estimate, but restrict the choice of coefficients so that the width of the averaging kernel increases. There is thus a trade-off between stability of the solution and the width of the averaging kernel.

The method of Backus and Gilbert produces a model estimate at a particular point $\hat{x}$. It is possible to use the method to compute estimates on a grid of points $x_1, x_2, \ldots, x_n$. However, since the averaging kernels may not be well localized around their associated grid points, and may vary and overlap in complicated ways, this is not equivalent to the simple collocation method of model representation introduced in Chapter 1. Furthermore, such an approach requires the computationally intensive solution of (5.51) for each point. For these reasons, the method does not receive wide use.

Example 5.1

For a spherically symmetric Earth model, the mass M_e and moment of inertia I_e are determined by the radial density $\rho(r)$, where

$$M_e = \int_0^{R_e} \left(4\pi r^2\right) \rho(r) \, dr \tag{5.52}$$

and

$$I_e = \int_0^{R_e} \left(\frac{8}{3}\pi r^4\right) \rho(r) \, dr. \tag{5.53}$$

Using $R_e = 6.3708 \times 10^6$ m as the radius of a spherical Earth, and supposing that from astronomical measurements we can infer that $M_e = 5.973 \pm 0.0005 \times 10^{24}$ kg and $I_e = 8.02 \pm 0.005 \times 10^{37}$ kg$\cdot$m^2, we will estimate the density of the Earth in the lower mantle (e.g., at $r = 5000$ km), and core (e.g., at $r = 1000$ km).

Equations (5.52) and (5.53) include terms that span an enormous numerical range. Scaling so that

$$\hat{r} = r/R_e \quad \hat{\rho} = \rho/1000 \quad \hat{I}_e = I_e/10^{37} \quad \hat{M}_e = M_e/10^{24}$$

gives

$$\hat{M}_e = 0.2586 \int\limits_0^1 \left(4\pi\hat{r}^2\right) \hat{\rho}(\hat{r}) \; d\hat{r} \tag{5.54}$$

and

$$\hat{I}_e = 1.0492 \int\limits_0^1 \left(\frac{8}{3}\pi\hat{r}^4\right) \hat{\rho}(\hat{r}) \; d\hat{r}. \tag{5.55}$$

Applying (5.49) for $r = 5000$ km gives the coefficient values $\mathbf{c}^T = [1.1809, \; -0.1588]$ and a corresponding model density of 5.8 g/cm^3. This is a fairly accurate estimate of density for this radius, where standard earth models estimated using seismological methods [96] infer densities of approximately 5 g/cm^3. The associated standard deviation (5.50) is 0.001 g/cm^3, so there is very little sensitivity to data uncertainty.

At $r = 1000$ km, we obtain the coefficients $\mathbf{c}^T = [2.5537, \; -1.0047]$, and a corresponding density estimate of 7.2 g/cm^3. This is not an accurate estimate for the density of the inner core, where standard earth models have densities of around 13 g/cm^3. The corresponding standard deviation is just 0.005 g/cm^3, so this inaccuracy is not related to instability in the inverse problem.

Figure 5.1 shows the averaging kernels corresponding to these model element estimates, and explains both the successful mantle and failed core density estimates. In the mantle case, the averaging kernel has much of its area near the targeted radius of

Figure 5.1 Averaging kernels for target radii of 1000 and 5000 km.

5000 km. In the core case, however, the averaging kernel has most of its area at much greater r, and little area near the target radius of 1000 km. The fundamental reason for this situation is that both the mass and moment of inertia are relatively insensitive to the density of the innermost earth.

5.4. EXERCISES

1. Consider the earth density estimation problem of Example 5.1. Apply the generalized inverse to estimate the density as a function of radius, using the given values of mass and moment of inertia. Obtain a density model composed of 20 spherical shells of equal thickness, and compare your results to a standard model.

2. Use the Gram matrix technique to discretize the integral equation from Example 5.1.
   a. Solve the resulting linear system of equations, and plot the resulting model.
   b. What was the condition number of $\mathbf{\Gamma}$? What does this tell you about the accuracy of your solution?

3. Show that if the representers $g_i(t)$ are linearly independent, then the Gram matrix $\mathbf{\Gamma}$ is nonsingular.

4. Show that if the basis functions in (5.9) are orthonormal, then $\|m(x)\|_2 = \|\boldsymbol{\beta}\|_2$. Hint: Expand $\|m(x)\|_2^2$ using (5.3), and then simplify using the orthogonality of the basis functions.

5. Recall the polynomial regression problem from Exercise 2.5. Instead of using the polynomials $1, x, \ldots, x^{19}$, we will use the basis of Legendre polynomials, which are orthogonal on the interval $[-1, 1]$. These polynomials are generated by the recurrence relation

$$p_{n+1}(x) = \frac{(2n+1)xp_n(x) - np_{n-1}(x)}{n+1}, \tag{5.56}$$

starting with

$$p_0(x) = 1 \tag{5.57}$$

and

$$p_1(x) = x. \tag{5.58}$$

The next two Legendre polynomials are $p_2(x) = (3x^2 - 1)/2$ and $p_3(x) = (5x^3 - 3x)/2$. This recurrence relation can be used both to compute coefficients of the Legendre polynomials and to compute values of the polynomials for particular values of x.

Use the first 20 Legendre polynomials to fit a polynomial of degree 19 to the data from Exercise 2.5. Express your solution as a linear combination of the Legendre

polynomials and also as a regular polynomial. How well-conditioned was this system of equations? Plot your solution and compare it with your solution to Exercise 2.5.

6. Construct specific examples of functions $m_1(x)$ and $m_2(x)$ in $L_2(-1, 1)$, such that $m(x) = m_1(x) + m_2(x)$, $m_1(x) \perp m_2(x)$, but $\|m''(x)\|_2^2 \neq \|m_1''(x)\|_2^2 + \|m_2''(x)\|_2^2$.

5.5. NOTES AND FURTHER READING

Techniques for discretizing integral equations are discussed in [45, 126, 166, 172]. A variety of basis functions have been used to discretize integral equations including sines and cosines, spherical harmonics, B-splines, and wavelets. In selecting the basis functions, it is important to select a basis that can reasonably represent likely models. The basis functions must be linearly independent, so that a function can be written in terms of the basis functions in exactly one way, and (5.9) is thus unique. As we have seen, the use of an orthonormal basis has the further advantage that $\|\boldsymbol{\beta}\|_2 = \|m(x)\|_2$.

The selection of an appropriate basis for a particular problem is a fine art that requires detailed knowledge of the problem as well as of the behavior of the basis functions. Beware that a poorly selected basis may not adequately approximate the solution, resulting in an estimated model $m(x)$ that is very wrong. The choice of basis can also have a very large effect on the condition number of the discretized problem, potentially making it very ill-conditioned.

An important theoretical question is whether the solutions to discretized versions of a continuous inverse problem with noise-free data will converge to a solution of the continuous inverse problem. Engl, Hanke, and Neubauer provide an explicit example showing that nonconvergence of a discretization scheme is possible [45]. They also provide conditions under which convergence is guaranteed. For Fredholm integral equations of the first kind, the Gram matrix approach using representers can be shown to be convergent [45].

Iterative Methods

Synopsis

We present several methods for solving linear inverse problems that may be far too large for the methods previously discussed to be practical. These methods are iterative, in that a sequence of trial solutions is generated that converges to a final solution. Kaczmarz's algorithm and the related ART and SIRT methods form one class, while methods based on conjugate gradients form a second class. When the method of conjugate gradients is applied to the normal equations, the resulting conjugate gradient least squares (CGLS) method regularizes the solution to the inverse problem. Methods for estimating model resolution for very large model spaces are described. Illustrative examples involving tomography and image deblurring are given.

6.1. INTRODUCTION

Singular value decomposition (SVD)–based pseudoinverse and Tikhonov regularization solutions become impractical when we consider larger problems in which $\mathbf{G}$ has, for example, tens of thousands of rows and columns. Storing all of the elements in such a large $\mathbf{G}$ matrix can require a great deal of computer memory. If many of the elements in the $\mathbf{G}$ matrix are zero, as for example in ray-theory tomography, then $\mathbf{G}$ is a **sparse matrix**, and we can obviate this problem by storing only the nonzero elements of $\mathbf{G}$ and their locations. The **density** of $\mathbf{G}$ is the percentage of nonzero elements in the matrix. MATLAB can store sparse matrices and most built-in MATLAB functions operate efficiently on sparse matrices. **Dense matrices** contain enough nonzero elements that sparse storage schemes are not efficient.

Methods for the solution of linear systems of equations based on matrix factorizations such as the Cholesky factorization, QR factorization, or SVD do not tend to work well with sparse matrices. The problem is that the matrices that occur in the factorization of $\mathbf{G}$ are often more dense than $\mathbf{G}$ itself. In particular, the $\mathbf{U}$ and $\mathbf{V}$ matrices in the SVD and the $\mathbf{Q}$ matrix in the QR factorization are orthogonal matrices, which typically make these matrices dense.

The iterative methods discussed in this chapter do not require the storage of large dense matrices. Instead, they work by generating a sequence of models $\mathbf{m}$ that converge to an optimal solution. Steps in these algorithms typically involve multiplying $\mathbf{G}$ and $\mathbf{G}^T$ times vectors, which can be done without additional storage. Because iterative methods can also take advantage of the sparsity commonly found in the $\mathbf{G}$ matrix, they are often used for very large problems.

For example, consider a two-dimensional tomography problem where the model is of size 256 by 256 (65,536 model elements) and there are 100,000 ray paths. Each ray path will pass through less than 1% of the model cells, so the vast majority of the elements (e.g., more than 99%) in $\mathbf{G}$ will be zero. If we stored $\mathbf{G}$ as a full matrix, it would require about 50 gigabytes (GB) of storage. Furthermore, if we wished to apply the SVD, the $\mathbf{U}$ matrix would require 80 GB of storage, and the $\mathbf{V}$ matrix would require about 35 GB. Because of the extreme sparsity of $\mathbf{G}$, such a matrix can be stored in less than 1 GB using sparse storage.

At the time this book was written, computers with several gigabytes of main memory were common, while only computers classified as supercomputers would have hundreds or thousands of gigabytes of main memory. The point at which it becomes necessary to use sparse matrix storage depends on the computer that we are using. Although the memory capacity of computers has been increasing steadily for many years, it is safe to say that there will always be problems for which sparse matrix storage will be required. Furthermore, programs that use sparse matrix storage often run much faster than programs that do not take advantage of sparsity. Thus it is a good idea to consider using sparse storage even in cases where sufficient memory is available to store the full matrix.

6.2. ITERATIVE METHODS FOR TOMOGRAPHY PROBLEMS

We will concentrate in this section on Kaczmarz's algorithm and its ART and SIRT variants. These algorithms were originally developed for tomographic applications and are particularly effective for such problems.

Kaczmarz's algorithm is an easy-to-implement algorithm for solving a linear system of equations $\mathbf{Gm} = \mathbf{d}$. To understand the algorithm, note that each of the m equations $\mathbf{G}_{i,.}\mathbf{m} = d_i$ defines an n-dimensional hyperplane in R^m. Kaczmarz's algorithm starts with an initial solution $\mathbf{m}^{(0)}$, and then moves to a solution $\mathbf{m}^{(1)}$ by projecting the initial solution onto the hyperplane defined by the first row in $\mathbf{G}$. Next $\mathbf{m}^{(1)}$ is similarly projected onto the hyperplane defined by the second row in $\mathbf{G}$, and so forth. The process is repeated until the solution has been projected onto all m hyperplanes defined by the system of equations. At that point, a new cycle of projections begins. These cycles are repeated until the solution has converged sufficiently. Figure 6.1 shows an example
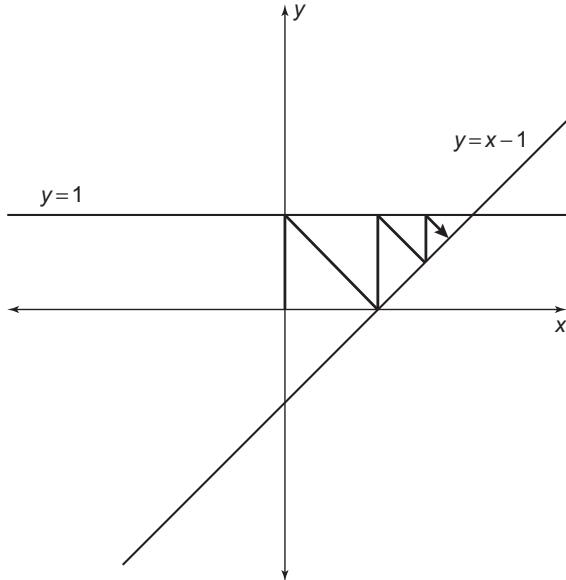
Figure 6.1 Kaczmarz's algorithm on a system of two equations.

in which Kaczmarz's algorithm is used to solve the system of equations

$$
\begin{aligned}
y &= 1 \\
-x + y &= -1.
\end{aligned}
$$
(6.1)

To implement the algorithm, we need a formula to compute the projection of a vector onto the hyperplane defined by equation i. Consider the hyperplane defined by $\mathbf{G}_{i+1,\cdot}\mathbf{m} = d_{i+1}$. Because the vector $\mathbf{G}_{i+1,\cdot}^{T}$ is perpendicular to this hyperplane, the update to $\mathbf{m}^{(i)}$ from the constraint due to row $i+1$ of $\mathbf{G}$ will be proportional to $\mathbf{G}_{i+1,\cdot}^{T}$.

$$
\mathbf{m}^{(i+1)} = \mathbf{m}^{(i)} + \beta\mathbf{G}_{i+1,\cdot}^{T}
$$
(6.2)

Using the fact that $\mathbf{G}_{i+1,\cdot}\mathbf{m}^{(i+1)} = d_{i+1}$ to solve for β, we obtain

$$
\mathbf{G}_{i+1,\cdot}\left(\mathbf{m}^{(i)} + \beta\mathbf{G}_{i+1,\cdot}^{T}\right) = d_{i+1}
$$
(6.3)

$$
\mathbf{G}_{i+1,\cdot}\mathbf{m}^{(i)} - d_{i+1} = -\beta\mathbf{G}_{i+1,\cdot}\mathbf{G}_{i+1,\cdot}^{T}
$$
(6.4)

$$
\beta = -\frac{\mathbf{G}_{i+1,\cdot}\mathbf{m}^{(i)} - d_{i+1}}{\mathbf{G}_{i+1,\cdot}\mathbf{G}_{i+1,\cdot}^{T}}.
$$
(6.5)

Thus the update formula is

$$\mathbf{m}^{(i+1)} = \mathbf{m}^{(i)} - \frac{\mathbf{G}_{i+1,\cdot}\mathbf{m}^{(i)} - d_{i+1}}{\mathbf{G}_{i+1,\cdot}\mathbf{G}_{i+1,\cdot}^{T}}\mathbf{G}_{i+1,\cdot}^{T}. \tag{6.6}$$

It can be shown that if the system of equations $\mathbf{Gm} = \mathbf{d}$ has a unique solution, then Kaczmarz's algorithm will converge to this solution. If the system of equations has many solutions, then the algorithm will converge to the solution that is closest to the point $\mathbf{m}^{(0)}$. In particular, if we start with $\mathbf{m}^{(0)} = \mathbf{0}$, we will obtain a minimum length solution. If there is no exact solution to the system of equations, then the algorithm will fail to converge, but will typically bounce around near an approximate solution.

A second important question is how quickly Kaczmarz's algorithm will converge to a solution. If the hyperplanes described by the system of equations are nearly orthogonal, then the algorithm will converge very quickly. However, if two or more hyperplanes are nearly parallel to each other, convergence can be extremely slow. Figure 6.2 shows a typical situation in which the algorithm zigzags back and forth without making much progress towards a solution. As the two lines become more nearly parallel, the problem becomes worse. This problem can be alleviated by picking an ordering of the equations such that adjacent equations describe hyperplanes that are nearly orthogonal to each other. In the context of tomography, this can be done by ordering the equations so that successive equations do not share common model cells.
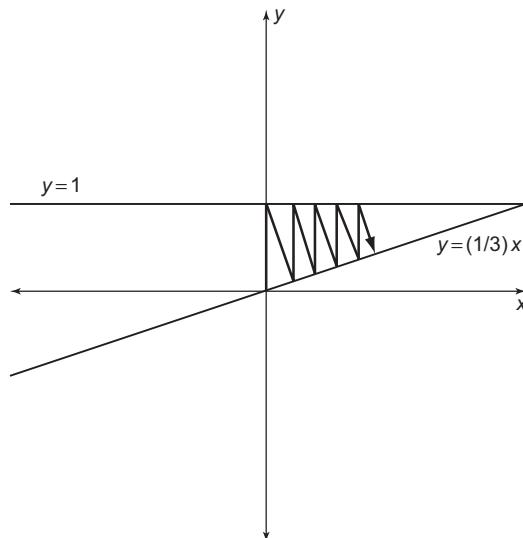
Figure 6.2 Slow convergence occurs when hyperplanes are nearly parallel.

Example 6.1

Consider a tomographic reconstruction problem with the same geometry used in Exercise 4.4, in which the slowness structure is parameterized in homogeneous blocks of size l by l. The true model is shown in Figure 6.3. Synthetic data were generated, with normally distributed random noise added. The random noise had a standard deviation of 0.01. Figure 6.4 shows the TSVD solution. The two anomalies are apparent, but it is not possible to distinguish the small hole within the larger of the two.

Figure 6.3 True model.
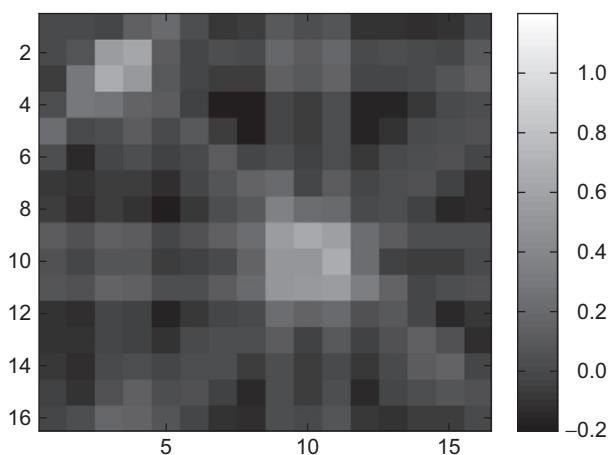
Figure 6.4 Truncated SVD solution.

Figure 6.5 Kaczmarz's algorithm solution.

Figure 6.5 shows the solution obtained after 200 iterations of Kaczmarz's algorithm. This solution is extremely similar to the TSVD solution, and both solutions are about the same distance from the true model.

The algebraic reconstruction technique (ART) is a version of Kaczmarz's algorithm that has been modified especially for the tomographic reconstruction problem. In (6.6), the updates to the solution always consist of adding a multiple of a row of $\mathbf{G}$ to the current solution. The numerator in the fraction is the difference between the right-hand side of equation $i+1$ and the value of the $i+1$ component of $\mathbf{Gm}$. The denominator in (6.6) is simply the square of the norm of $\mathbf{G}_{i+1,\cdot}$. Effectively, Kaczmarz's algorithm is determining the error in equation $i+1$, and then adjusting the solution by spreading the required correction over the elements of $\mathbf{m}$ that appear in equation $i+1$.

Algorithm 6.1 Kaczmarz's Algorithm

Given a system of equations $\mathbf{Gm} = \mathbf{d}$.
1. Let $\mathbf{m}^{(0)} = \mathbf{0}$.
2. For $i = 0, 1, \ldots, m-1$, let

$$\mathbf{m}^{(i+1)} = \mathbf{m}^{(i)} - \frac{\mathbf{G}_{i+1,\cdot}\mathbf{m}^{(i)} - d_{i+1}}{\mathbf{G}_{i+1,\cdot}\mathbf{G}_{i+1,\cdot}^{T}}\mathbf{G}_{i+1,\cdot}^{T}. \tag{6.7}$$

3. If the solution has not yet converged, return to step 2.

A crude approximation to the Kaczmarz update, used in ART, is to replace all of the nonzero elements in row $i+1$ of $\mathbf{G}$ with ones. We define

$$q_{i+1} = \sum_{\text{cell } j \text{ in ray path } i+1} \mathbf{m}_j l \tag{6.8}$$

as an approximation to the travel time along ray path $i+1$, where l is the cell dimension. The difference between q_{i+1} and d_{i+1} is roughly the residual for the predicted travel time of ray $i+1$.

Examining (6.6) for the ART-modified $\mathbf{G}$, we see that ART simply takes the total travel time misfit for ray $i+1$ and divides it by l times the number of cells in the ray path, N_{i+1}. This correction factor is then multiplied by a vector that has ones in cells along ray path $i+1$. This procedure has the effect of smearing the needed correction in travel time equally over all of the cells in the ray path.

The ART approximate update formula can thus be written as

$$m_j^{(i+1)} = \begin{cases} m_j^{(i)} - \frac{q_{i+1} - d_{i+1}}{l N_{i+1}} & \text{cell } j \text{ in ray path } i+1 \\ m_j^{(i)} & \text{cell } j \text{ not in ray path } i+1. \end{cases} \tag{6.9}$$

The approximation can be improved by taking into account that the ray path lengths actually will vary from cell to cell. If L_{i+1} is the length of ray path $i+1$, the corresponding improved update formula from (6.6) for the tomography problem is

$$m_j^{(i+1)} = \begin{cases} m_j^{(i)} + \frac{d_{i+1}}{L_{i+1}} - \frac{q_{i+1}}{l N_{i+1}} & \text{cell } j \text{ in ray path } i+1 \\ m_j^{(i)} & \text{cell } j \text{ not in ray path } i+1. \end{cases} \tag{6.10}$$

The main advantage of ART is that it saves storage. We need only store information about which rays pass through which cells, and we do not need to record the length of each ray in each cell. A second advantage of the method is that it reduces the number of floating-point multiplications compared to that required by Kaczmarz's algorithm. Although in current computers floating-point multiplications and additions require roughly the same amount of time, during the 1970s, when ART was first developed, multiplication was slower than addition.

Algorithm 6.2   ART

Given a system of equations $\mathbf{Gm} = \mathbf{d}$ arising from a tomography problem:
1. Let $\boldsymbol{m}^{(0)} = \mathbf{0}$.
2. For $i = 0, 1, \ldots, m$, let N_i be the number of cells traversed by ray path i.
3. For $i = 0, 1, \ldots, m$, let L_i be the length of ray path i.

Algorithm 6.2 (*Continued*)

---

4. For $i = 0, 1, \ldots, m-1, j = 1, 2, \ldots, n$, let

$$m_j^{(i+1)} = \begin{cases} m_j^{(i)} + \frac{d_{i+1}}{L_{i+1}} - \frac{q_{i+1}}{lN_{i+1}} & \text{cell } j \text{ in ray path } i+1 \\ m_j^{(i)} & \text{cell } j \text{ not in ray path } i+1. \end{cases} \tag{6.11}$$

5. If the solution has not yet converged, let $\boldsymbol{m}^{(0)} = \boldsymbol{m}^{(m)}$ and return to step 4. Otherwise, return the solution $\boldsymbol{m} = \boldsymbol{m}^{(m)}$.

One problem with ART is that the resulting tomographic images tend to be noisier than images produced by Kaczmarz's algorithm (6.7). The simultaneous iterative reconstruction technique (SIRT) is a variation on ART that gives slightly better images in practice, at the expense of a slightly slower algorithm. In the SIRT algorithm, all (up to m nonzero) updates using (6.10) are computed for each cell j of the model, for each ray that passes through cell j. The set of updates for cell j are then averaged before updating the corresponding model element m_j.

Algorithm 6.3 SIRT

Given a system of equations $\mathbf{Gm} = \mathbf{d}$ arising from a tomography problem:
1. Let $\mathbf{m}^{(0)} = \mathbf{0}$.
2. For $j = 0, 1, \ldots, n$, let K_j be the number of ray paths that pass through cell j.
3. For $i = 0, 1, \ldots, m$, let L_i be the length of ray path i.
4. For $i = 0, 1, \ldots, m$, let N_i be the number of cells traversed by ray path i.
5. Let $\Delta\boldsymbol{m}^{(i+1)} = \mathbf{0}$.
6. For $i = 0, 1, \ldots, m-1, j = 1, 2, \ldots, n$, let

$$\Delta m_j^{(i+1)} = \Delta m_j^{(i+1)} + \begin{cases} \frac{d_{i+1}}{L_{i+1}} - \frac{q_{i+1}}{lN_{i+1}} & \text{cell } j \text{ in ray path } i+1 \\ 0 & \text{cell } j \text{ not in ray path } i+1. \end{cases} \tag{6.12}$$

7. For $j = 1, 2, \ldots, n$, let

$$m_j^{(i+1)} = m_j^{(i+1)} + \frac{\Delta m_j^{(i+1)}}{K_j}. \tag{6.13}$$

8. If the solution has not yet converged, return to step 5. Otherwise, return the current solution.

Example 6.2

Returning to our earlier tomography example, Figure 6.6 shows the ART solution obtained after 200 iterations. Again, the solution is very similar to the TSVD solution.
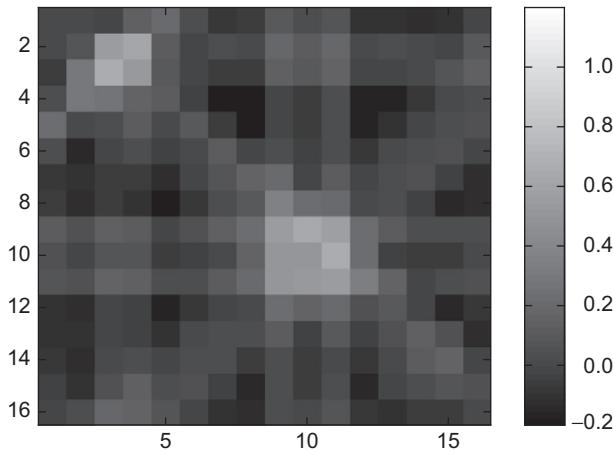
Figure 6.6 ART solution.

Figure 6.7 shows the SIRT solution for our example tomography problem after 200 iterations. This solution is similar to the TSVD, Kaczmarz's, and ART solutions.
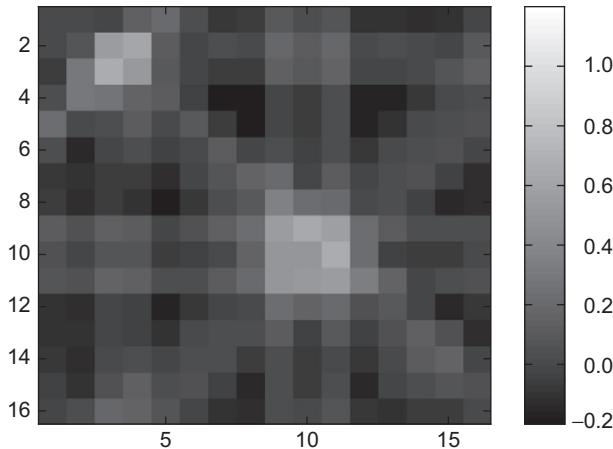
Figure 6.7 SIRT solution.

6.3. THE CONJUGATE GRADIENT METHOD

We next consider the **conjugate gradient** (CG) method for solving a symmetric and positive definite system of equations $\mathbf{Ax} = \mathbf{b}$. We will later apply the CG method to solving the normal equations for $\mathbf{Gm} = \mathbf{d}$. Consider the quadratic optimization problem,

$$\min \phi(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T\mathbf{Ax} - \mathbf{b}^T\mathbf{x}, \tag{6.14}$$

where $\mathbf{A}$ is an $n \times n$ symmetric and positive definite matrix. We require $\mathbf{A}$ be positive definite so that the function $\phi(\mathbf{x})$ will be convex and have a unique minimum. We can calculate $\nabla\phi(\mathbf{x}) = \mathbf{Ax} - \mathbf{b}$ and set it equal to zero to find the minimum. The minimum occurs at a point $\mathbf{x}$ that satisfies the equation

$$\mathbf{Ax} - \mathbf{b} = \mathbf{0} \tag{6.15}$$

or

$$\mathbf{Ax} = \mathbf{b}. \tag{6.16}$$

Thus solving the system of equations $\mathbf{Ax} = \mathbf{b}$ is equivalent to minimizing $\phi(\mathbf{x})$.

The conjugate gradient method approaches the problem of minimizing $\phi(\mathbf{x})$ by constructing a basis for R^n in which the minimization problem is extremely simple. The basis vectors $\mathbf{p}_0, \mathbf{p}_1, \ldots, \mathbf{p}_{n-1}$ are selected so that

$$\mathbf{p}_i^T\mathbf{Ap}_j = 0 \quad \text{when } i \neq j. \tag{6.17}$$

A collection of vectors with this property is said to be **mutually conjugate** with respect to $\mathbf{A}$. We express $\mathbf{x}$ in terms of these basis vectors as

$$\mathbf{x} = \sum_{i=0}^{n-1} \alpha_i\mathbf{p}_i \tag{6.18}$$

so that

$$\phi(\boldsymbol{\alpha}) = \frac{1}{2}\left(\sum_{i=0}^{n-1} \alpha_i\mathbf{p}_i\right)^T \mathbf{A}\left(\sum_{j=0}^{n-1} \alpha_j\mathbf{p}_j\right) - \mathbf{b}^T\left(\sum_{i=0}^{n-1} \alpha_i\mathbf{p}_i\right). \tag{6.19}$$

The product $\mathbf{x}^T\mathbf{Ax}$ can be written as a double sum, so the equation becomes

$$\phi(\boldsymbol{\alpha}) = \frac{1}{2}\sum_{i=0}^{n-1}\sum_{j=0}^{n-1} \alpha_i\alpha_j\mathbf{p}_i^T\mathbf{Ap}_j - \mathbf{b}^T\left(\sum_{i=0}^{n-1} \alpha_i\mathbf{p}_i\right). \tag{6.20}$$

Since the vectors are mutually conjugate with respect to $\mathbf{A}$, this simplifies to

$$\phi(\boldsymbol{\alpha}) = \frac{1}{2} \sum_{i=0}^{n-1} \alpha_i^2 \mathbf{p}_i^T \mathbf{A} \mathbf{p}_i - \mathbf{b}^T \left(\sum_{i=0}^{n-1} \alpha_i \mathbf{p}_i \right) \tag{6.21}$$

or

$$\phi(\boldsymbol{\alpha}) = \frac{1}{2} \sum_{i=0}^{n-1} (\alpha_i^2 \mathbf{p}_i^T \mathbf{A} \mathbf{p}_i - 2\alpha_i \mathbf{b}^T \mathbf{p}_i). \tag{6.22}$$

Equation (6.22) shows that $\phi(\boldsymbol{\alpha})$ consists of n terms, each of which is independent of the other terms. Thus we can minimize $\phi(\boldsymbol{\alpha})$ by selecting each α_i to minimize the *ith* term,

$$\alpha_i^2 \mathbf{p}_i^T \mathbf{A} \mathbf{p}_i - 2\alpha_i \mathbf{b}^T \mathbf{p}_i. \tag{6.23}$$

Differentiating with respect to α_i and setting the derivative equal to zero, we find that the optimal value for α_i is

$$\alpha_i = \frac{\mathbf{b}^T \mathbf{p}_i}{\mathbf{p}_i^T \mathbf{A} \mathbf{p}_i}. \tag{6.24}$$

This shows that if we have a basis of vectors that are mutually conjugate with respect to $\mathbf{A}$, then minimizing $\phi(\mathbf{x})$ is very easy. We have not yet shown how to construct the mutually conjugate basis vectors.

Our algorithm will actually construct a sequence of solution vectors $\mathbf{x}_i$, residual vectors $\mathbf{r}_i = \mathbf{b} - \mathbf{A}\mathbf{x}_i$, and basis vectors $\mathbf{p}_i$. The algorithm begins with $\mathbf{x}_0 = \mathbf{0}$, $\mathbf{r}_0 = \mathbf{b}$, $\mathbf{p}_0 = \mathbf{r}_0$, and $\alpha_0 = (\mathbf{r}_0^T \mathbf{r}_0)/(\mathbf{p}_0^T \mathbf{A}\mathbf{p}_0)$.

Suppose that at the start of iteration k of the algorithm, we have constructed $\mathbf{x}_0$, $\mathbf{x}_1, \ldots, \mathbf{x}_k, \mathbf{r}_0, \mathbf{r}_1, \ldots, \mathbf{r}_k, \mathbf{p}_0, \mathbf{p}_1, \ldots, \mathbf{p}_k$ and $\alpha_0, \alpha_1, \ldots, \alpha_k$. We assume that the first $k+1$ basis vectors $\mathbf{p}_i$ are mutually conjugate with respect to $\mathbf{A}$, the first $k+1$ residual vectors $\mathbf{r}_i$ are mutually orthogonal, and that $\mathbf{r}_i^T \mathbf{p}_j = 0$ when $i \neq j$.

We let

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k. \tag{6.25}$$

This effectively adds one more term of (6.18) into the solution. Next, we let

$$\mathbf{r}_{k+1} = \mathbf{r}_k - \alpha_k \mathbf{A}\mathbf{p}_k. \tag{6.26}$$

This correctly updates the residual, because

$$\mathbf{r}_{k+1} = \mathbf{b} - \mathbf{A}\mathbf{x}_{k+1} \tag{6.27}$$

$$= \mathbf{b} - \mathbf{A}(\mathbf{x}_k + \alpha_k \mathbf{p}_k) \tag{6.28}$$

$$= (\mathbf{b} - \mathbf{A}\mathbf{x}_k) - \alpha_k \mathbf{A}\mathbf{p}_k \tag{6.29}$$

$$= \mathbf{r}_k - \alpha_k \mathbf{A}\mathbf{p}_k. \tag{6.30}$$

We let

$$\beta_{k+1} = \frac{\|\mathbf{r}_{k+1}\|_2^2}{\|\mathbf{r}_k\|_2^2} = \frac{\mathbf{r}_{k+1}^T \mathbf{r}_{k+1}}{\mathbf{r}_k^T \mathbf{r}_k} \tag{6.31}$$

and

$$\mathbf{p}_{k+1} = \mathbf{r}_{k+1} + \beta_{k+1}\mathbf{p}_k. \tag{6.32}$$

In the following calculations, it will be useful to know that $\mathbf{b}^T\mathbf{p}_k = \mathbf{r}_k^T\mathbf{r}_k$. This is shown by

$$\mathbf{b}^T\mathbf{p}_k = (\mathbf{r}_k + \mathbf{A}\mathbf{x}_k)^T\mathbf{p}_k \tag{6.33}$$

$$= \mathbf{r}_k^T\mathbf{p}_k + \mathbf{x}_k^T\mathbf{A}\mathbf{p}_k \tag{6.34}$$

$$= \mathbf{r}_k^T\mathbf{p}_k + \mathbf{p}_k^T\mathbf{A}\mathbf{x}_k \tag{6.35}$$

$$= \mathbf{r}_k^T(\mathbf{r}_k + \beta_k\mathbf{p}_{k-1}) + \mathbf{p}_k^T\mathbf{A}\mathbf{x}_k \tag{6.36}$$

$$= \mathbf{r}_k^T\mathbf{r}_k + \beta_k\mathbf{r}_k^T\mathbf{p}_{k-1} + \mathbf{p}_k^T\mathbf{A}(\alpha_0\mathbf{p}_0 + \cdots + \alpha_{k-1}\mathbf{p}_{k-1}) \tag{6.37}$$

$$= \mathbf{r}_k^T\mathbf{r}_k + 0 + 0 \tag{6.38}$$

$$= \mathbf{r}_k^T\mathbf{r}_k. \tag{6.39}$$

We will now show that $\mathbf{r}_{k+1}$ is orthogonal to $\mathbf{r}_i$ for $i \leq k$. For every $i < k$,

$$\mathbf{r}_{k+1}^T\mathbf{r}_i = (\mathbf{r}_k - \alpha_k\mathbf{A}\mathbf{p}_k)^T\mathbf{r}_i \tag{6.40}$$

$$= \mathbf{r}_k^T\mathbf{r}_i - \alpha_k\mathbf{p}_k^T\mathbf{A}\mathbf{r}_i \tag{6.41}$$

$$= \mathbf{r}_k^T\mathbf{r}_i - \alpha_k\mathbf{r}_i^T\mathbf{A}\mathbf{p}_k. \tag{6.42}$$

Since $\mathbf{r}_k$ is orthogonal to all of the earlier $\mathbf{r}_i$ vectors,

$$\mathbf{r}_{k+1}^T\mathbf{r}_i = 0 - \alpha_k\mathbf{p}_k^T\mathbf{A}\mathbf{r}_k. \tag{6.43}$$

Also, since $\mathbf{p}_i = \mathbf{r}_i + \beta_i\mathbf{p}_{i-1}$,

$$\mathbf{r}_{k+1}^T\mathbf{r}_i = 0 - \alpha_k(\mathbf{p}_i - \beta_i\mathbf{p}_{i-1})^T\mathbf{A}\mathbf{p}_k. \tag{6.44}$$

Both $\mathbf{p}_i$ and $\mathbf{p}_{i-1}$ are conjugate with $\mathbf{p}_k$. Thus

$$\mathbf{r}_{k+1}^T \mathbf{r}_i = 0. \tag{6.45}$$

We also have to show that $\mathbf{r}_{k+1}^T \mathbf{r}_k = 0$:

$$\mathbf{r}_{k+1}^T \mathbf{r}_k = (\mathbf{r}_k - \alpha_k \mathbf{A}\mathbf{p}_k)^T \mathbf{r}_k \tag{6.46}$$

$$= \mathbf{r}_k^T \mathbf{r}_k - \alpha_k \mathbf{p}_k^T \mathbf{A}\mathbf{r}_k \tag{6.47}$$

$$= \mathbf{r}_k^T \mathbf{r}_k - \alpha_k \mathbf{p}_k^T \mathbf{A}(\mathbf{p}_k - \beta_k \mathbf{p}_{k-1}) \tag{6.48}$$

$$= \mathbf{r}_k^T \mathbf{r}_k - \alpha_k (\mathbf{p}_k - \beta_k \mathbf{p}_{k-1})^T \mathbf{A}\mathbf{p}_k \tag{6.49}$$

$$= \mathbf{r}_k^T \mathbf{r}_k - \alpha_k \mathbf{p}_k^T \mathbf{A}\mathbf{p}_k + \alpha_k \beta_k \mathbf{p}_{k-1}^T \mathbf{A}\mathbf{p}_k \tag{6.50}$$

$$= \mathbf{r}_k^T \mathbf{r}_k - \mathbf{r}_k^T \mathbf{r}_k + \alpha_k \beta_k 0 \tag{6.51}$$

$$= 0. \tag{6.52}$$

Next, we will show that $\mathbf{r}_{k+1}$ is orthogonal to $\mathbf{p}_i$ for $i \leq k$.

$$\mathbf{r}_{k+1}^T \mathbf{p}_i = \mathbf{r}_{k+1}^T \left(\mathbf{r}_i + \beta_i \mathbf{p}_{i-1}\right) \tag{6.53}$$

$$= \mathbf{r}_{k+1}^T \mathbf{r}_i + \beta_i \mathbf{r}_{k+1}^T \mathbf{p}_{i-1} \tag{6.54}$$

$$= 0 + \beta_i \mathbf{r}_{k+1}^T \mathbf{p}_{i-1} \tag{6.55}$$

$$= \beta_i \left(\mathbf{r}_k - \alpha_k \mathbf{A}\mathbf{p}_k\right)^T \mathbf{p}_{i-1} \tag{6.56}$$

$$= \beta_i (\mathbf{r}_k^T \mathbf{p}_{i-1} - \alpha_k \mathbf{p}_{i-1}^T \mathbf{A}\mathbf{p}_k) \tag{6.57}$$

$$= \beta_i (0 - 0) = 0. \tag{6.58}$$

Finally, we need to show that $\mathbf{p}_{k+1}^T \mathbf{A}\mathbf{p}_i = 0$ for $i \leq k$. For $i < k$,

$$\mathbf{p}_{k+1}^T \mathbf{A}\mathbf{p}_i = (\mathbf{r}_{k+1} + \beta_{k+1}\mathbf{p}_k)^T \mathbf{A}\mathbf{p}_i \tag{6.59}$$

$$= \mathbf{r}_{k+1}^T \mathbf{A}\mathbf{p}_i + \beta_{k+1}\mathbf{p}_k^T \mathbf{A}\mathbf{p}_i \tag{6.60}$$

$$= \mathbf{r}_{k+1}^T \mathbf{A}\mathbf{p}_i + 0 \tag{6.61}$$

$$= \mathbf{r}_{k+1}^T \left(\frac{1}{\alpha_i}(\mathbf{r}_i - \mathbf{r}_{i+1})\right) \tag{6.62}$$

$$= \frac{1}{\alpha_i}\left(\mathbf{r}_{k+1}^T \mathbf{r}_i - \mathbf{r}_{k+1}^T \mathbf{r}_{i+1}\right) \tag{6.63}$$

$$= 0. \tag{6.64}$$

For $i = k$,

$$\mathbf{p}_{k+1}^T \mathbf{A} \mathbf{p}_k = \left(\mathbf{r}_{k+1} + \beta_{k+1} \mathbf{p}_k \right)^T \left(\frac{1}{\alpha_k} \left(\mathbf{r}_k - \mathbf{r}_{k+1} \right) \right) \tag{6.65}$$

$$= \frac{1}{\alpha_k} \left(\beta_{k+1} \left(\mathbf{r}_k + \beta_k \mathbf{p}_{k-1} \right)^T \mathbf{r}_k - \mathbf{r}_{k+1}^T \mathbf{r}_{k+1} \right) \tag{6.66}$$

$$= \frac{1}{\alpha_k} \left(\beta_{k+1} \mathbf{r}_k^T \mathbf{r}_k + \beta_{k+1} \beta_k \mathbf{p}_{k-1}^T \mathbf{r}_k - \mathbf{r}_{k+1}^T \mathbf{r}_{k+1} \right) \tag{6.67}$$

$$= \frac{1}{\alpha_k} \left(\mathbf{r}_{k+1}^T \mathbf{r}_{k+1} + \beta_{k+1} \beta_k 0 - \mathbf{r}_{k+1}^T \mathbf{r}_{k+1} \right) \tag{6.68}$$

$$= 0. \tag{6.69}$$

We have now shown that the algorithm generates a sequence of mutually conjugate basis vectors. In theory, the algorithm will find an exact solution to the system of equations in n iterations. In practice, due to round-off errors in the computation, the exact solution may not be obtained in n iterations. In practical implementations of the algorithm, we iterate until the residual is smaller than some tolerance that we specify. The algorithm can be summarized as follows.

Algorithm 6.4   Conjugate Gradient Method

Given a positive definite and symmetric system of equations $\mathbf{A} \mathbf{x} = \mathbf{b}$, and an initial solution $\mathbf{x}_0$, let $\beta_{-1} = 0$, $\mathbf{p}_{-1} = \mathbf{0}$, $\mathbf{r}_0 = \mathbf{b} - \mathbf{A} \mathbf{x}_0$, and $k = 0$.

1. Let $\mathbf{p}_k = -\mathbf{r}_k + \beta_{k-1} \mathbf{p}_{k-1}$.
2. Let $\alpha_k = \frac{\|\mathbf{r}_k\|_2^2}{\mathbf{p}_k^T \mathbf{A} \mathbf{p}_k}$.
3. Let $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k$.
4. Let $\mathbf{r}_{k+1} = \mathbf{r}_k + \alpha_k \mathbf{A} \mathbf{p}_k$.
5. Let $\beta_k = \frac{\|\mathbf{r}_{k+1}\|_2^2}{\|\mathbf{r}_k\|_2^2}$.
6. Let $k = k + 1$.
7. Repeat the previous steps until convergence.

A major advantage of the CG method is that it requires storage only for the vectors $\mathbf{x}_k$, $\mathbf{p}_k$, $\mathbf{r}_k$, and the symmetric matrix $\mathbf{A}$. If $\mathbf{A}$ is large and sparse, then sparse matrix techniques can be used to store $\mathbf{A}$ more efficiently. Unlike factorization methods such as QR, SVD, or Cholesky factorization, there will be no fill-in of the zero elements in $\mathbf{A}$ at any stage in the solution process. Thus it is possible to solve extremely large systems using CG in cases where direct factorization would require far too much storage. In fact, the only way in which the algorithm uses $\mathbf{A}$ is in one multiplication of $\mathbf{A} \mathbf{p}_k$ for each

iteration. In some applications of the CG method, it is possible to perform these matrix vector multiplications without explicitly constructing $\mathbf{A}$.

6.4. THE CGLS METHOD

The CG method by itself can only be applied to positive definite systems of equations, and is thus not directly applicable to general least squares problems. In the **conjugate gradient least squares** (CGLS) method, we solve a least squares problem,

$$\min \|\mathbf{Gm} - \mathbf{d}\|_2, \tag{6.70}$$

by applying CG to the normal equations,

$$\mathbf{G}^T\mathbf{Gm} = \mathbf{G}^T\mathbf{d}. \tag{6.71}$$

In implementing this algorithm it is important to avoid round-off errors. One important source of error is the evaluation of the residual, $\mathbf{G}^T\mathbf{Gm} - \mathbf{G}^T\mathbf{d}$. It turns out that this calculation is more accurate when we factor out $\mathbf{G}^T$ and compute $\mathbf{G}^T(\mathbf{Gm} - \mathbf{d})$. We will use the notation $\mathbf{s}_k = \mathbf{Gm}_k - \mathbf{d}$, and $\mathbf{r}_k = \mathbf{G}^T\mathbf{s}_k$. Note that we can compute $\mathbf{s}_{k+1}$ recursively from $\mathbf{s}_k$ as follows

$$\mathbf{s}_{k+1} = \mathbf{Gm}_{k+1} - \mathbf{d} \tag{6.72}$$

$$= \mathbf{G}(\mathbf{m}_k + \alpha_k\mathbf{p}_k) - \mathbf{d} \tag{6.73}$$

$$= (\mathbf{Gm}_k - \mathbf{d}) + \alpha_k\mathbf{Gp}_k \tag{6.74}$$

$$= \mathbf{s}_k + \alpha_k\mathbf{Gp}_k. \tag{6.75}$$

With this trick, we can now state the CGLS algorithm.

Algorithm 6.5 CGLS

Given a least squares problem $\min \|\mathbf{Gm} - \mathbf{d}\|_2$, let $k = 0$, $\mathbf{m}_0 = \mathbf{0}$, $\mathbf{p}_{-1} = \mathbf{0}$, $\beta_{-1} = 0$, $\mathbf{s}_0 = -\mathbf{d}$, and $\mathbf{r}_0 = \mathbf{G}^T\mathbf{s}_0$.

1. Let $\mathbf{p}_k = -\mathbf{r}_k + \beta_{k-1}\mathbf{p}_{k-1}$.

2. Let $\alpha_k = \dfrac{\|\mathbf{r}_k\|_2^2}{(\mathbf{p}_k^T\mathbf{G}^T)(\mathbf{Gp}_k)}$.

3. Let $\mathbf{m}_{k+1} = \mathbf{m}_k + \alpha_k\mathbf{p}_k$.

4. Let $\mathbf{s}_{k+1} = \mathbf{s}_k + \alpha_k\mathbf{Gp}_k$.

5. Let $\mathbf{r}_{k+1} = \mathbf{G}^T\mathbf{s}_{k+1}$.

6. Let $\beta_k = \dfrac{\|\mathbf{r}_{k+1}\|_2^2}{\|\mathbf{r}_k\|_2^2}$.

7. Let $k = k + 1$.

8. Repeat the previous steps until convergence.

Note that this algorithm only requires one multiplication of $\mathbf{Gp}_k$ and one multiplication of $\mathbf{G}^T\mathbf{s}_{k+1}$ per iteration. We never explicitly compute $\mathbf{G}^T\mathbf{G}$, which might require considerable time, and which might have far more nonzero elements than $\mathbf{G}$ itself.

The CGLS algorithm has an important property that makes it particularly useful for ill-posed problems. It can be shown that, for exact arithmetic, $\|\mathbf{m}_k\|_2$ increases monotonically and $\|\mathbf{Gm}_k - \mathbf{d}\|_2$ decreases monotonically with increasing iterations [61, 65]. We can use the discrepancy principle together with this property to obtain a regularized solution. Simply stop the CGLS algorithm as soon as $\|\mathbf{Gm}_k - \mathbf{d}\|_2 < \delta$. In practice, this algorithm typically gives good solutions after a very small number of iterations.

An alternative way to use CGLS is to solve the Tikhonov regularization problem (4.4) by applying CGLS to

$$\min \left\| \begin{bmatrix} \mathbf{G} \\ \alpha\mathbf{L} \end{bmatrix} \mathbf{m} - \begin{bmatrix} \mathbf{d} \\ \mathbf{0} \end{bmatrix} \right\|_2^2. \tag{6.76}$$

For nonzero values of the regularization parameter α, this least squares problem should be reasonably well-conditioned. By solving this problem for several values of α, we can compute an L-curve. The disadvantage of this approach is that the number of CGLS iterations for each value of α may be large, and we need to solve the problem for several values of α. Thus the computational effort is far greater.

Example 6.3

A commonly used mathematical model of image blurring involves the two-dimensional convolution of the true image $I_{\text{true}}(x, y)$ with a **point spread function**, $\Psi(u, v)$ [14]:

$$I_{\text{blurred}}(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} I_{\text{true}}(x - u, y - v)\Psi(u, v) \, du \, dv. \tag{6.77}$$

Here the point spread function shows how a point in the true image is altered in the blurred image. A point spread function that is commonly used to represent the blurring that occurs because an image is out of focus is the **Gaussian point spread function**

$$\Psi(u, v) = e^{\frac{u^2 + v^2}{2\sigma^2}}. \tag{6.78}$$

Here the parameter σ controls the relative width of the point spread function. In practice, the blurred image and point spread function are discretized into pixels. In theory, Ψ is nonzero for all values of u and v. However, it becomes small quickly as u and v increase. If we set small values of Ψ to 0, then the $\mathbf{G}$ matrix in the discretized problem will be sparse.
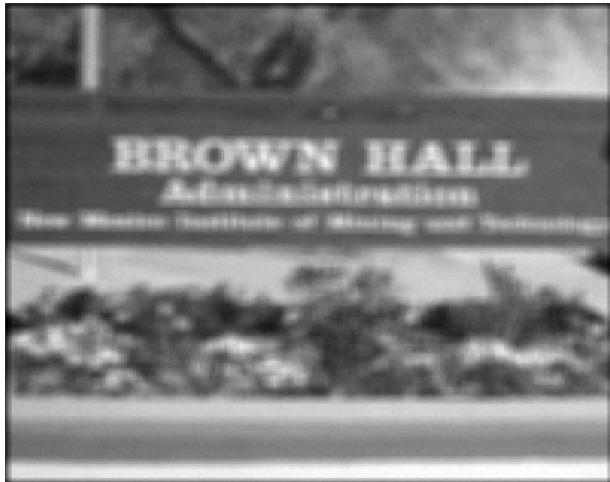
Figure 6.8 Blurred image.

Figure 6.8 shows an image that has been blurred and also has a small amount of added noise. This image is of size 200 pixels by 200 pixels, so the **G** matrix for the blurring operator is of size 40,000 by 40,000. Fortunately, the blurring matrix **G** is quite sparse, with less than 0.1% nonzero elements. The sparse matrix requires about 12 MB of storage. A dense matrix of this size would require about 13 GB of storage. Using the SVD approach to Tikhonov regularization would require far more storage than many current computers have. However, CGLS works quite well on this problem.

Figure 6.9 shows the L-curve for the solution of this problem by CGLS without explicit regularization and by CGLS with explicit regularization. The line with circles shows the solutions obtained by CGLS without explicit regularization. For the first 30 or so iterations of CGLS without explicit regularization, $\|\mathbf{Gm} - \mathbf{d}\|_2$ decreases quickly. After that point, the improvement in misfit slows down, while $\|\mathbf{m}\|_2$ increases rapidly.

Figure 6.10 shows the CGLS solution without explicit regularization after 30 iterations. The blurring has been greatly improved. Note that 30 iterations is far less than the size of the matrix ($n = 40,000$). Unfortunately, further CGLS iterations do not significantly improve the image. In fact, noise builds up rapidly, both because of the accumulation of round-off errors and because the algorithm is converging slowly towards an unregularized least squares solution. Figure 6.11 shows the CGLS solution after 100 iterations. In this image the noise has been greatly amplified, with little or no improvement in the clarity of the image.

We also computed CGLS solutions with explicit Tikhonov regularization for 22 values of α. For each value of α, 200 iterations of CGLS were performed. The resulting
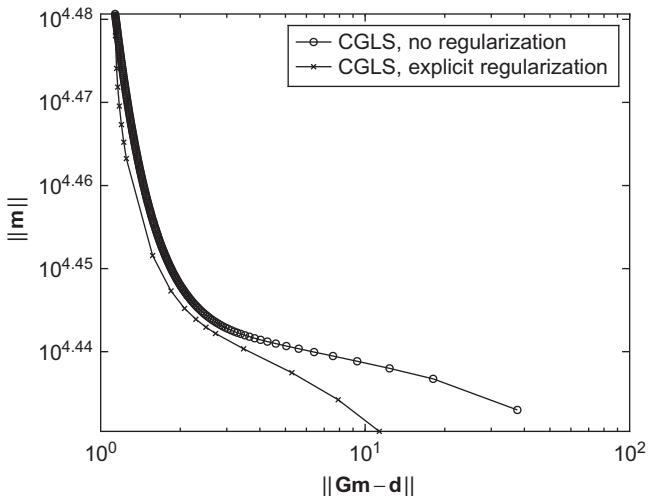
Figure 6.9 L-curves for CGLS deblurring.
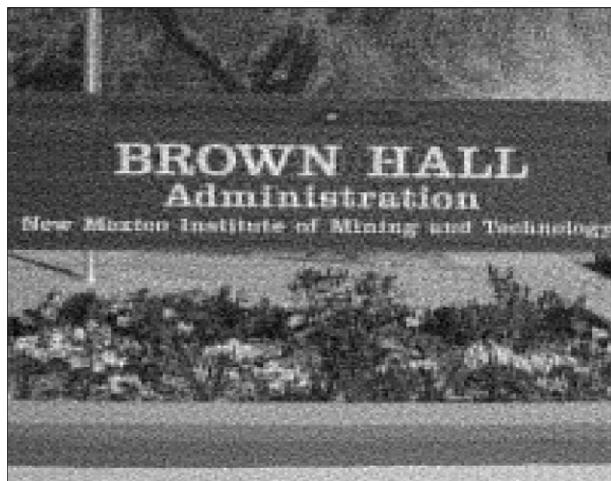
Figure 6.10 CGLS solution after 30 iterations, no explicit regularization.

L-curve is shown in Figure 6.9 with "x" markers for each regularized solution that was obtained. This L-curve is slightly better than the L-curve from the CGLS solution without explicit regularization in that the values of $\|\mathbf{m}\|_2$ and $\|\mathbf{Gm} - \mathbf{d}\|_2$ are smaller. However, it required 40 times as much computational effort. The corner solution for $\alpha = 7.0 \times 10^{-4}$ is shown in Figure 6.12. This solution is similar to the one shown in Figure 6.10.
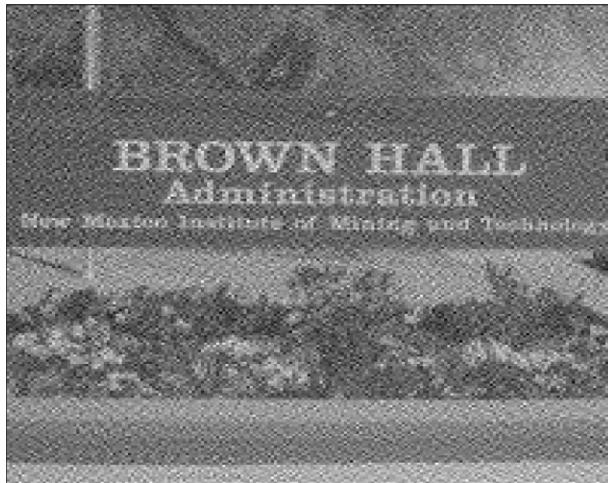
Figure 6.11 CGLS solution after 100 iterations, no explicit regularization.
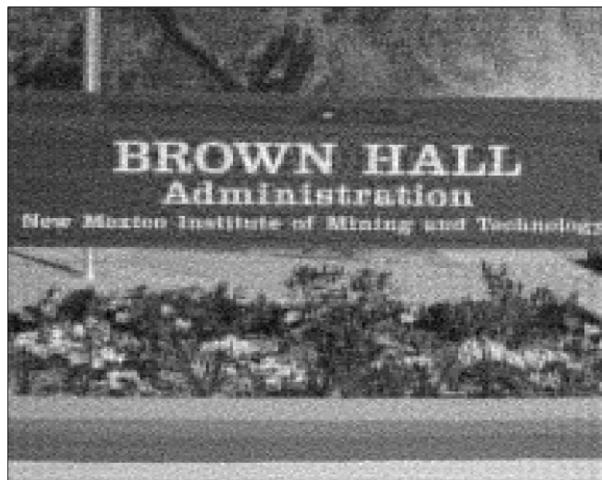
Figure 6.12 CGLS solution, explicit regularization, $\alpha = 7.0 \times 10^{-4}$.

We have focused on the CGLS method in this section because it is particularly easy to derive and because of the implicit regularization effect of the CGLS method. However, many other iterative methods have been developed for large-scale least squares problems [5, 135]. An analysis of the implicit regularizing effects of these methods can be found in Hanke [61]. The LSQR method of Paige and Saunders has been very widely used in many geophysical applications [123, 124]. MATLAB has an efficient and robust implementation of LSQR in its **lsqr** routine.

6.5. RESOLUTION ANALYSIS FOR ITERATIVE METHODS

Analyzing the resolution of a regularized solution computed by the iterative methods described in this chapter is a challenging problem. The formulas developed in Chapter 4 for the model resolution matrix under zeroth-order Tikhonov regularization were based on the SVD of the $\mathbf{G}$ matrix, but as we have seen it is often unfeasible to compute the SVD of a large matrix, and even when it is extremely sparse, the SVD is typically dense. For higher-order regularization, we have formulas for the resolution matrix based on the GSVD of the $\mathbf{G}$ matrix, such as (4.20), but it is also impractical to compute the GSVD of very large matrices.

   A brute force approach that offers some advantages over the computation of the SVD or GSVD is to use (4.52) with Cholesky factorization of $(\mathbf{G}^T\mathbf{G} + \alpha^2\mathbf{L}^T\mathbf{L})$ [19]. Cholesky factorization is considerably faster in practice than computing the SVD. However, since the Cholesky factor of this matrix is typically dense, this is still a very large-scale computation that would generally need to be performed in parallel on a computer with sufficient memory to store a dense n by n matrix.

   If we want to compute an individual column of the model resolution matrix, this can be accomplished by solving an additional least squares problem. Applying the expression for the least squares Tikhonov inverse operator $\mathbf{G}^\sharp$ (4.52), the *ith* column of $\mathbf{R}_m$ can be expressed as

$$\begin{aligned}
\mathbf{R}_{.,i} &= \mathbf{R}_m\mathbf{e}_i \\
&= \mathbf{G}^\sharp\mathbf{G}\mathbf{e}_i \\
&= \mathbf{G}^\sharp\mathbf{G}_{.,i} \\
&= \left(\mathbf{G}^T\mathbf{G} + \alpha^2\mathbf{L}^T\mathbf{L}\right)^{-1}\mathbf{G}^T\mathbf{G}_{.,i}.
\end{aligned} \tag{6.79}$$

This is (4.26), a regularized least squares problem

$$\min \left\| \begin{bmatrix} \mathbf{G} \\ \alpha\mathbf{L} \end{bmatrix} \mathbf{R}_{.,i} - \begin{bmatrix} \mathbf{G}_{.,i} \\ \mathbf{0} \end{bmatrix} \right\|_2^2 \tag{6.80}$$

that can be solved for $\mathbf{R}_{.,i}$ using CGLS, LSQR, or some other iterative method. We could conceivably solve (6.80) to retrieve every column of the model resolution matrix in this manner, but this would require the solution of n least squares problems.

   A number of authors have considered iterative techniques for computing low-rank approximations to the resolution matrix in combination with iterative least squares solution methods [12, 13, 174]. In such approaches, iterative methods are applied to compute the largest k singular values of a matrix and the associated singular vectors. For a sparse matrix, these methods are vastly more efficient than standard algorithms for computing the SVD.

Suppose that we obtain a low-rank (k singular values and vectors) approximation to the SVD

$$\mathbf{G} \approx \mathbf{U}_k \mathbf{L}_k \mathbf{V}_k^T \tag{6.81}$$

where $\mathbf{L}_k$ is a k by k nonsingular matrix, and $\mathbf{U}_k$ and $\mathbf{V}_k$ are matrices with k orthogonal columns. The pseudoinverse of $\mathbf{G}$ is then approximately

$$\mathbf{G}^\dagger \approx \mathbf{V}_k \mathbf{L}_k^{-1} \mathbf{U}_k^T. \tag{6.82}$$

Combining these approximations, we have

$$\mathbf{R}_m \approx \mathbf{V}_k \mathbf{V}_k^T. \tag{6.83}$$

Note that we would not want to explicitly multiply out $\mathbf{V}_k \mathbf{V}_k^T$, creating a huge and dense n by n resolution matrix. Rather, we could store the relatively small vector $\mathbf{V}_k$ and use it to compute desired entries of the $\mathbf{R}_m$ matrix as needed.

A more sophisticated version of this approach is to compute a low-rank approximation to the singular value decomposition of the augmented $\mathbf{G}$ matrix [173]. Consider an augmented system of equations

$$\mathbf{A} = \begin{bmatrix} \mathbf{G} \\ \alpha \mathbf{L} \end{bmatrix} \tag{6.84}$$

using truncated SVD approximations for $\mathbf{A}$ and its pseudoinverse, $\mathbf{A}^\dagger$,

$$\mathbf{A} \approx \mathbf{U}_k \mathbf{S}_k \mathbf{V}_k^T \tag{6.85}$$

and

$$\mathbf{A}^\dagger \approx \mathbf{V}_k \mathbf{S}_k^{-1} \mathbf{U}_k^T, \tag{6.86}$$

where $\mathbf{U}_k$ and $\mathbf{V}_k$ are composed of the first k columns of $\mathbf{U}$ and $\mathbf{V}$, and $\mathbf{S}_k$ is a square k by k matrix diagonal matrix of the largest k singular values.

It is convenient to write $\mathbf{U}_k$ as

$$\mathbf{U}_k = \begin{bmatrix} \mathbf{U}_G \\ \mathbf{U}_L \end{bmatrix}, \tag{6.87}$$

where $\mathbf{U}_G$ and $\mathbf{U}_L$ contain the rows of $\mathbf{U}_k$ corresponding to those of $\mathbf{G}$ and $\alpha \mathbf{L}$, respectively, in $\mathbf{A}$. We can thus partition (6.85) using

$$\mathbf{G} \approx \mathbf{U}_G \mathbf{S}_k \mathbf{V}_k^T. \tag{6.88}$$

Given (6.88) and (6.86), we can calculate an approximation of the model resolution matrix

$$\mathbf{R}_m = \mathbf{A}^\dagger \begin{bmatrix} \mathbf{G} \\ \mathbf{0} \end{bmatrix} \approx \mathbf{V}_k \mathbf{S}_k^{-1} \begin{bmatrix} \mathbf{U}_G^T \mathbf{U}_L^T \end{bmatrix} \begin{bmatrix} \mathbf{U}_G \mathbf{S}_k \mathbf{V}_k^T \\ \mathbf{0} \end{bmatrix}, \tag{6.89}$$

which simplifies to

$$\mathbf{R}_m \approx \mathbf{V}_k \mathbf{S}_k^{-1} \mathbf{U}_G^T \mathbf{U}_G \mathbf{S}_k \mathbf{V}_k^T. \tag{6.90}$$

Using the identity

$$\mathbf{U}_G^T \mathbf{U}_G = \mathbf{I} - \mathbf{U}_L^T \mathbf{U}_L \tag{6.91}$$

we can alternatively write (6.90) as

$$\mathbf{R}_m \approx \mathbf{V}_k \left(\mathbf{I} - \mathbf{S}_k^{-1} \mathbf{U}_L^T \mathbf{U}_L \mathbf{S}_k \right) \mathbf{V}_k^T. \tag{6.92}$$

Depending on the relative sizes of $\mathbf{U}_G$ and $\mathbf{U}_L$, one or the other of (6.90) or (6.92) might be more easily evaluated. Again, we would probably not want to, or be able to, explicitly multiply out the matrices to obtain the entire dense n by n model resolution matrix for a large problem, but could instead use (6.90) or (6.92) to compute specific desired elements of the resolution matrix.

In practice, it may not be feasible for very large problems to compute, or even store, a sufficient number of singular values and vectors to adequately approximate the model resolution matrix using low-rank, SVD-based approximations [37]. We will thus consider an alternative stochastic approach based on [9] that estimates solely the resolution matrix diagonal, but which is not limited by low-rank matrix approximations.

Consider a sequence of s random vectors from $\mathbf{R}^n$, $\mathbf{v}_1, \ldots, \mathbf{v}_s$, with independent elements drawn from a standard normal distribution. The sth corresponding estimate for the diagonal of an n by n square matrix $\mathbf{A}$ is given by

$$\mathbf{q}_s = \left[\sum_{k=1}^{s} \mathbf{v}_k \odot \mathbf{A}\mathbf{v}_k \right] \oslash \left[\sum_{k=1}^{s} \mathbf{v}_k \odot \mathbf{v}_k \right] \tag{6.93}$$

where $\odot$ denotes element-wise multiplication and $\oslash$ denotes element-wise division of vectors. Note that the only way in which $\mathbf{A}$ is used within this algorithm is in matrix-vector multiplications, so as long as we have a function that can implement the matrix vector multiplication, then we can implement the algorithm. As we have already seen, the multiplication of a large matrix with a vector can be effected by solving a least squares problem (6.80). Specifically, to solve for the kth product $\mathbf{y} = \mathbf{R}_m$, we solve [103]

$$\min \left\| \begin{bmatrix} \mathbf{G} \\ \alpha\mathbf{L} \end{bmatrix} \mathbf{y} - \begin{bmatrix} \mathbf{G}\mathbf{v}_k \\ \mathbf{0} \end{bmatrix} \right\|_2^2. \tag{6.94}$$

It is necessary to solve s least squares systems of equations (6.94) to calculate (6.93). In practice, relatively small values of s in (6.93) (e.g., 100 to 1000) are adequate even when there are tens or hundreds of thousands of model parameters [103].

The stochastic algorithm (6.93) can also be used to implement generalized cross-validation. In (4.74), the most difficult computation is evaluation of the trace of

$\mathbf{I} - \mathbf{GG}^\sharp$. Computing this matrix explicitly and then evaluating the trace would require excessive time and storage. Using (6.93) and evaluating the necessary $\mathbf{G}^\sharp$ matrix-vector multiplications by solving an associated least squares problem using an iterative method such as LSQR or CGLS makes it tractable to compute an approximation to the GCV function (4.74), where we solve s least squares problems for each value of α [103].

Example 6.4

We will examine the solution to a large-scale tomography problem involving 30,000 rays passing through a cubic volume discretized into a 30 by 30 by 30 model of 27,000 cells (Figure 6.13).
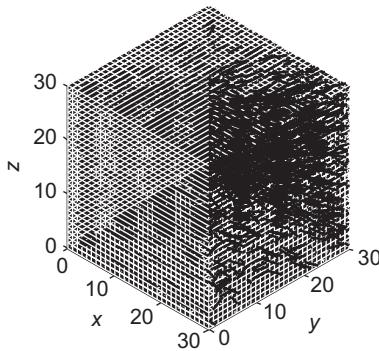
Figure 6.13 A 30 by 30 by 30 block tomography model interrogated by 30,000 ray paths.

Ray path start and end points are uniformly randomly distributed across the sides of the volume with 10,000 rays passing between each pair of parallel sides. The corresponding $\mathbf{G}$ matrix is 30,000 by 27,000, but only 0.2% of its entries are nonzero. Synthetic travel time data (with travel time perturbation values between approximately -20 and 30 time units) were generated from a zero-mean smooth model with $N(0, (0.5)^2)$ noise added. We implemented second-order Tikhonov regularization with a small amount of additional zeroth-order regularization to regularize the solution.

Figure 6.14 shows the L-curve for this problem. Figure 6.15 shows a plot of the GCV function (4.74) computed using stochastic estimates of the diagonal of $\mathbf{I} - \mathbf{GG}^\sharp$. The L-curve corner and GCV minimum both occur at about $\alpha = 0.5$. For the solution obtained using $\alpha = 0.5$, we computed a stochastic estimate of the diagonal of the resolution matrix using (6.93). For comparison, we also computed exact resolution matrix diagonal element values for 100 randomly selected indices using (6.79). Figure 6.16 shows a scatter plot of the stochastic estimates compared with corresponding exact determinations, and indicates that the stochastic estimates are accurate to a few percentage
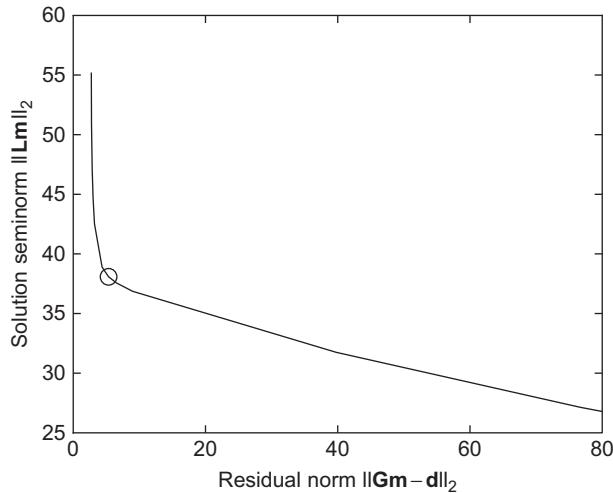
Figure 6.14 L-curve for the tomography example of Figure 6.13 and its corner near $\alpha = 0.5$.
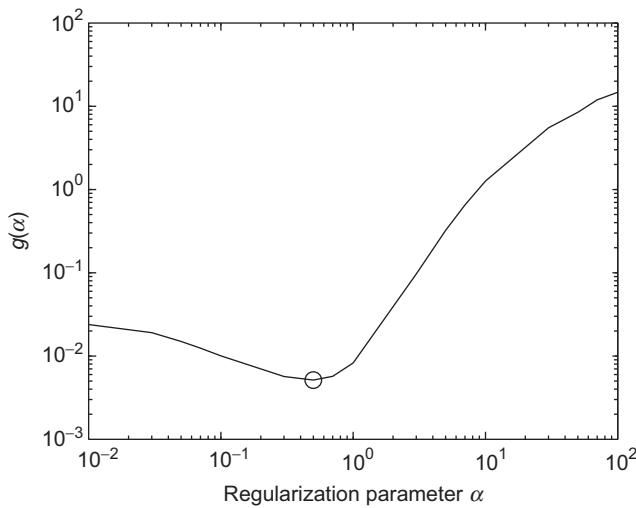
Figure 6.15 GCV plot for the tomography example of Figure 6.13 and its minimum near $\alpha = 0.5$.

points. It is common to use counts of the number of rays transecting each cell as a proxy for assessing the diagonal of the resolution matrix in tomographic problems. In general, the association between ray density and resolution in a tomography problem will depend on the specific ray path geometry. Figure 6.17 shows a scatter plot of the ray densities versus the accurately computed values of the diagonal elements of $\mathbf{R}_m$ for the 100 random indices plotted in Figure 6.16. The relationship shows a nonlinear trend between ray density and the resolution diagonal element values, and there are also anomalously
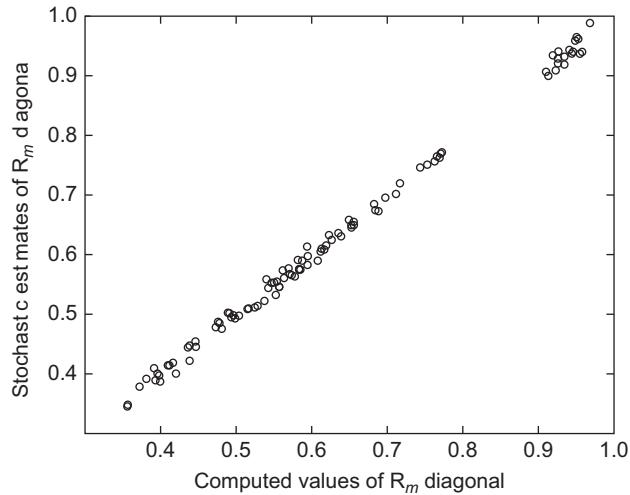
Figure 6.16 Stochastic estimates versus computed values of the diagonal of $\mathbf{R}_m$.
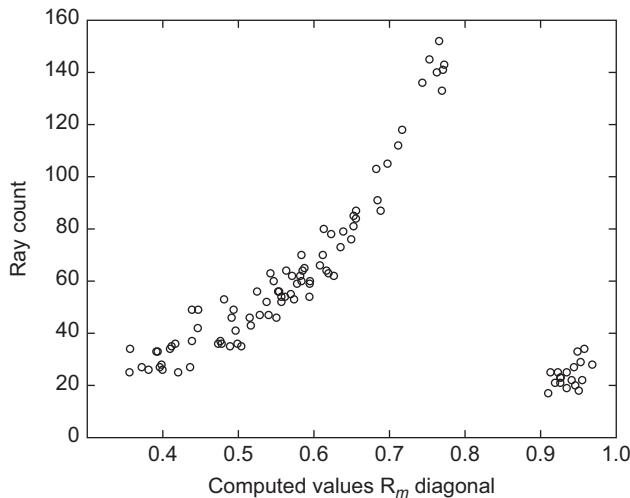
Figure 6.17 Comparison of ray densities and resolution.

well–resolved outliers in the plot with resolution diagonal element values greater than 0.9. Further examination shows that the anomalously well–resolved outliers correspond to cells that reside on the model exterior and are thus constrained by two–dimensional ray path geometries.

6.6. EXERCISES

1. Consider the cross-well tomography problem of Exercise 4.3.
   a. Apply Kaczmarz's algorithm to this problem.
   b. Apply ART to this problem.
   c. Apply SIRT to this problem.
   d. Comment on the solutions that you obtained.
2. A very simple iterative regularization method is the **Landweber iteration** [67]. The algorithm begins with $\mathbf{m}^{(0)} = \mathbf{0}$, and then follows the iteration,

$$\mathbf{m}^{(k+1)} = \mathbf{m}^{(k)} - \omega(\mathbf{G}\mathbf{m}^{(k)} - \mathbf{d}). \tag{6.95}$$

   To ensure convergence, the parameter ω must be selected so that $0 < \omega < 2/s_1^2$, where s_1 is the largest singular value of $\mathbf{G}$.

   In practice, the CGLS method generally works better than the Landweber iteration. However, it is easier to analyze the performance of the Landweber iteration. It can be shown that the kth iterate of the Landweber iteration is exactly the same as the SVD solution with filter factors of

$$f_i^{(k)} = 1 - (1 - \omega s_i^2)^k. \tag{6.96}$$

   a. Implement the Landweber iteration and apply it to the Shaw problem from Example 4.3.
   b. Verify that $\mathbf{m}^{(10)}$ from the Landweber iteration matches the SVD solution with filter factors given by (6.96).
   c. Derive (6.96).
3. The Landweber iteration described in the previous exercise converges under the condition that $0 < \omega < 2/s_1^2$, where s_1 is the largest singular value of $\mathbf{G}$ (or equivalently, $s_1 = \|\mathbf{G}\|_2$). As a practical matter we typically cannot compute the full SVD of $\mathbf{G}$. However, it is possible to rapidly estimate this quantity using an iterative method that we will derive in this exercise. Recall from Appendix A that $\|\mathbf{G}\|_2 = s_1$ is the square root of the largest eigenvalue of the matrix $\mathbf{G}^T\mathbf{G}$.
   a. Using (A.77), diagonalize the matrix $\mathbf{A} = \mathbf{G}^T\mathbf{G}$ and use the diagonalization to show that

$$\mathbf{A}^n = \mathbf{P}\mathbf{\Lambda}^n\mathbf{P}^{-1}. \tag{6.97}$$

   Assume that the eigenvalues of $\mathbf{A}$ are sorted so that $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n \geq 0$.
   b. Take an arbitrary vector $\mathbf{x}$ in $\mathbf{R}^n$, and write it in terms of the eigenvectors of $\mathbf{A}$ as

$$\mathbf{x} = \alpha_1\mathbf{v}_1 + \cdots + \alpha_n\mathbf{v}_n. \tag{6.98}$$

Then show that

$$\mathbf{A}^k x = \alpha_1 \lambda_1^k \mathbf{v}_1 + \cdots + \alpha_n \lambda_n^k \mathbf{v}_n. \tag{6.99}$$

c. Starting with a random vector $\mathbf{x}_0$, and assuming that $\alpha_1 \neq 0$ (a reasonable assumption for a random vector $\mathbf{x}_0$), show that

$$\lim_{k \to \infty} \frac{\|\mathbf{A}^k \mathbf{x}_0\|_2}{\|\mathbf{A}^{k-1} \mathbf{x}_0\|_2} = \lambda_1. \tag{6.100}$$

This leads to a simple iterative algorithm for estimating $s_1 = \sqrt{\lambda_1}$. Start with a random vector $\mathbf{x}_0$. In each iteration, let

$$\mathbf{x}_{k+1} = \frac{\mathbf{G}^T (\mathbf{G} \mathbf{x}_k)}{\|\mathbf{x}_k\|_2} \tag{6.101}$$

and let

$$\rho_k = \sqrt{\|\mathbf{G}^T (\mathbf{G} \mathbf{x}_k)\|_2}. \tag{6.102}$$

The sequence ρ_k converges to s_1. This function is implemented in MATLAB as the **normest** function. Write your own implementation of this function and compare the results you obtain with **normest**.

4. Use the Landweber iteration discussed in Exercise 6.2 to deblur the image from Example 6.3. In order to use the Landweber iteration, you will need to estimate $s_1 = \|\mathbf{G}\|_2$, which can be done using the **normest** command discussed in Exercise 6.3.

5. In Example 6.3 we have included a function **blur** that computes the system matrix for the problem of deblurring an image that has been blurred by a Gaussian point spread function. The file **blur.mat** contains a blur-inducing sparse $\mathbf{G}$ matrix and a data vector $\mathbf{d}$, where the 100 by 100 pixel gray-scale image is mapped into $\mathbf{d}$ using a column-by-column indexing convention.

   a. How large is the $\mathbf{G}$ matrix? How many nonzero elements does it have? How much storage would be required for the $\mathbf{G}$ matrix if all of its elements were nonzero? How much storage would the SVD of $\mathbf{G}$ require?

   b. Plot the raw image.

   c. Using the MATLAB **cgls** algorithm with 100 iterations, deblur the image by solving $\mathbf{Gm} = \mathbf{d}$ and interpret the evolution of your solutions with increasing iterations.

6. Show that if $\mathbf{p}_0, \mathbf{p}_1, \ldots, \mathbf{p}_{n-1}$ are nonzero and mutually conjugate with respect to an $n \times n$ symmetric and positive definite matrix $\mathbf{A}$, then the vectors are also linearly independent. Hint: Use the definition of linear independence.

7. Recall the iteratively reweighted least squares method introduced in Chapter 2. The system of equations (2.92) solved in each iteration of the IRLS algorithm might well

be dense, even if the matrix $\mathbf{G}$ is sparse. However, this system of equations can also be thought of as the normal equations for a particular least squares problem that will be sparse when $\mathbf{G}$ is sparse. Write down this least squares problem. Write a MATLAB function that uses the **lsqr** routine to solve these least squares problems in an IRLS scheme to minimize $\|\mathbf{Gm} - \mathbf{d}\|_1$. Hint: Use the fact that $\mathbf{R}$ is a diagonal matrix with positive elements; you can take its square root.

6.7. NOTES AND FURTHER READING

Iterative methods for the solution of linear systems of equations are an important topic in numerical analysis. Some basic references include [5, 87, 135, 142].

Iterative methods for tomography problems including Kaczmarz's algorithm, ART, and SIRT are discussed in [84, 115, 163], and parallel algorithms based on these methodologies are discussed in [30]. These methods are often referred to as **row action methods** because they access only one row of the matrix at a time, which makes them relatively easy to implement in parallel processing. In practice, the conjugate gradient and LSQR methods generally provide better performance than the row action methods. There are some interesting connections between SIRT and the conjugate gradient method discussed in [120, 146, 147].

Hestenes and Stiefel are generally credited with the invention of the conjugate gradient method [72]. However, credit is also due to Lanczos [92]. The history of the conjugate gradient method is discussed in [55, 71]. Shewchuk's technical report [142] provides an introduction to the conjugate gradient method with illustrations that help to make the geometry of the method very clear. Filter factors for the CGLS method similar to those in Exercise 2 can be determined. These are derived in [65]. The LSQR method of Paige and Saunders [65, 123, 124] is an alternative way to apply the CG method to the normal equations. The resolution of LSQR solutions is discussed in [12, 13]. Schemes have been developed for using CGLS with explicit regularization and dynamic adjustment of the regularization parameter α [85, 86, 106]. This can potentially remove the computational burden of solving the problem for many values of α. An alternative approach can be used to compute regularized solutions for several values of α at once [50]. The performance of the CG algorithm degrades dramatically on poorly conditioned systems of equations. In such situations a technique called **preconditioning** can be used to improve the performance of CG. Essentially, preconditioning involves a change of variables $\bar{x} = \mathbf{C}x$. The matrix $\mathbf{C}$ is selected so that the resulting system of equations will be better conditioned than the original system of equations [38, 53, 164]. The conjugate gradient method can also be generalized to nonlinear minimization problems [58, 142].

Inverse problems in image processing are a very active area of research. Some books on inverse problems in imaging include [14, 115].

Additional Regularization Techniques

Synopsis

Alternatives or adjuncts to Tikhonov regularization are introduced. Bounds constraints allow the use of prior knowledge on the permissible range of parameter values. Sparsity regularization selects solutions with the minimum number of nonzero model parameters. A modification of the iteratively reweighted least squares algorithm presented in Chapter 2 for 1-norm parameter estimation is introduced to solve these problems. In compressive sensing, sparsity regularization is applied in association with a change of basis, where the basis is chosen so that the desired model will be sparse (i.e., have only a few nonzero coefficients in the model expansion). Total variation regularization uses a regularization term based on the 1-norm of the model gradient that allows for discontinuous jumps in the model to bias the solution towards piecewise-constant functions.

7.1. USING BOUNDS AS CONSTRAINTS

In many physical situations, bounds exist, or can be usefully estimated, on the maximum and/or minimum values of model parameters. For example, the model parameters may represent a physical quantity such as density that is inherently non-negative, establishing a strict lower bound for model parameters of $\mathbf{0}$. The problem of solving for a least squares solution that includes this constraint, referred to as non-negative least squares (NNLS), can be expressed as

$$\begin{aligned} \min\ &\|\mathbf{Gm}-\mathbf{d}\|_2 \\ &\mathbf{m} \geq \mathbf{0}, \end{aligned} \tag{7.1}$$

where $\mathbf{m} \geq \mathbf{0}$ means that every element of the vector $\mathbf{m}$ must be non-negative. This problem can be solved by an algorithm called **non-negative least squares (NNLS)** that was originally developed by Lawson and Hanson [94]. MATLAB includes a command, **lsqnonneg**, that solves the NNLS problem.

We might also declare a strict upper bound, so that model parameters may not exceed some value, for example, a density $3500\,\mathrm{kg/m}^3$ for crustal rocks in a particular region.

Given the lower and upper bound vectors **l** and **u**, we can pose the **bounded variables least squares** (BVLS) problem:

$$\min \|\mathbf{Gm} - \mathbf{d}\|_2$$
$$\mathbf{m} \geq \mathbf{l} \tag{7.2}$$
$$\mathbf{m} \leq \mathbf{u}.$$

Given a BVLS algorithm for solving (7.2), we can also perform Tikhonov regularization with bounds by augmenting the system of equations (e.g., (4.5)) and then solving the augmented system under bounding constraints. Stark and Parker [150] developed an algorithm for solving the BVLS problem, which we employ here as a MATLAB function **bvls.m**. A similar algorithm is given in the 1995 edition of Lawson and Hanson's book [94].

A related optimization problem involves minimizing or maximizing a linear function of the model for a set of n coefficients c_i, subject to bounds constraints and a constraint on the misfit. This problem can be formulated as

$$\min \mathbf{c}^T\mathbf{m}$$
$$\|\mathbf{Gm} - \mathbf{d}\|_2 \leq \delta$$
$$\mathbf{m} \geq \mathbf{l} \tag{7.3}$$
$$\mathbf{m} \leq \mathbf{u}.$$

● ──

Example 7.1

Recall the source history reconstruction problem of Example 1.7, where data are taken in concentration units at spatial positions, x, at a particular time (assumed dimensionless here), T. Figure 7.1 shows the true (smooth) source model used to generate the data, and Figure 7.2 shows these data as a function of distance, x, at time $T = 300$, with $N(0,0.001^2)$ noise added.

Figure 7.3 shows the least squares solution, which has the extremely large amplitudes and oscillatory behavior characteristic of an unregularized solution to an ill-posed problem. This solution is, furthermore, physically unrealistic in having negative concentrations. Figure 7.4 shows the non-negative least squares solution, which, although certainly more realistic in having all of the concentration values nonnegative, consists of high-amplitude spikes that do not accurately reconstruct the (smooth) true source history. Suppose that the solubility limit of the contaminant in water is known to be 1.1 units, thus providing a natural upper bound on model parameters. Figure 7.5 shows the corresponding BVLS solution, which exhibits spiky features similar to those of Figure 7.4. Further regularization is indicated if we believe that the true model is smooth.

Figure 7.1 True source history.
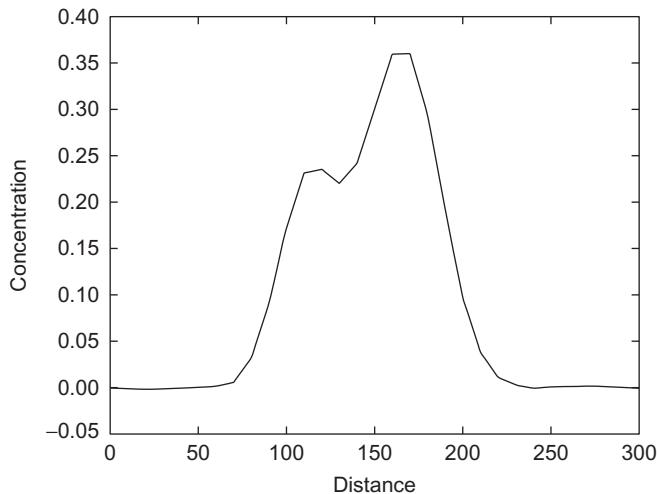
Figure 7.2 Concentration data as a function of position, x, taken at $T = 300$.

Figure 7.6 shows the L–curve for a second–order Tikhonov regularization solution applied with bounds of $0 \leq \mathbf{m} \leq 1.1$ on the model vector elements. Figure 7.7 shows the regularized solution for $\alpha = 0.01$. This solution correctly reveals the two major input concentration peaks. As is typical for cases of non–ideal model resolution, the solution peaks are somewhat lower and broader than those of the true model. This solution does not, however, resolve the smaller subsidiary peak near $t = 150$.
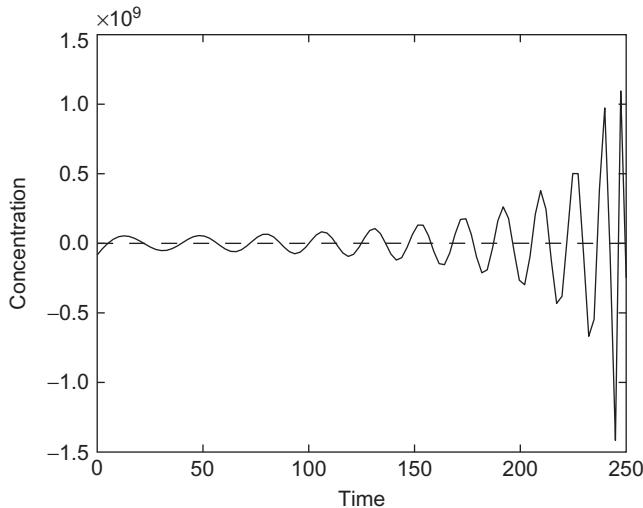
Figure 7.3 Least squares source history solution with true history (Figure 7.1) shown as a dashed curve. Because the least squares solution has extremely large amplitudes, the true model appears as a flat line at this scale.

Figure 7.4 NNLS source history solution with true history (Figure 7.1) shown as a dashed curve.

We can additionally use (7.3) to establish bounds on linear combinations of the model parameters. For example, we might want to establish bounds on the average concentration from $t = 125$ to $t = 150$. These concentrations appear in positions 51 through 60 of the model vector $\mathbf{m}$. We let c_i be zero in positions 1 through 50 and 61 through 100, and let c_i be 0.1 in positions 51 through 60 to form a 10-time-sample averaging function.

Figure 7.5 BVLS source history solution ($\mathbf{l} = \mathbf{0}, \mathbf{u} = \mathbf{1.1}$) with true history (Figure 7.1) shown as a dashed curve.
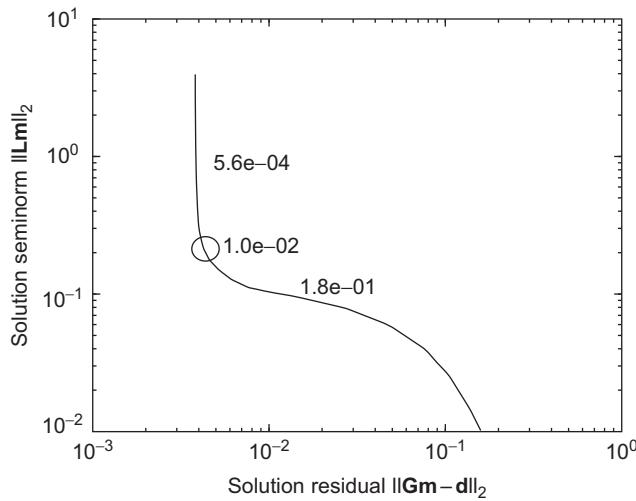
Figure 7.6 L-curve for the second-order Tikhonov solution with BVLS ($\mathbf{l} = \mathbf{0}, \mathbf{u} = \mathbf{1.1}$) implementation, corner at $\alpha = 0.01$.

The value of $\mathbf{c}^T \mathbf{m}$ is the average of model parameters 51 through 60. The value of the solution to (7.3) will then be a lower bound on the average concentration from $t = 125$ to $t = 150$. Similarly, by maximizing $\mathbf{c}^T \mathbf{m}$ or equivalently by minimizing $-\mathbf{c}^T \mathbf{m}$, we can obtain an upper bound on the average concentration from $t = 125$ to $t = 150$. Solving the minimization problems for $\mathbf{c}^T \mathbf{m}$ and $-\mathbf{c}^T \mathbf{m}$, we obtain, respectively, a lower
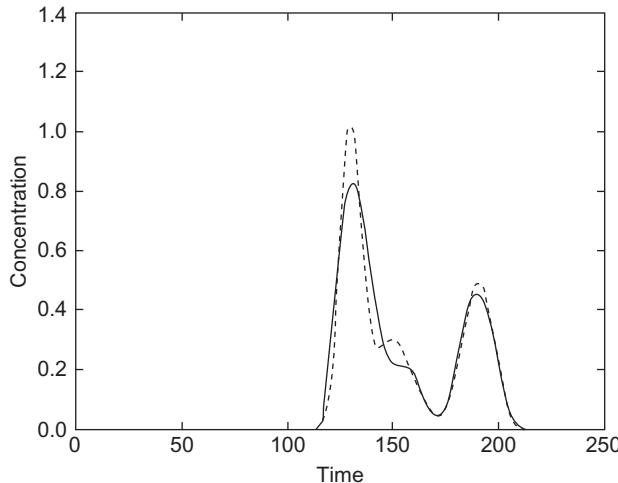
Figure 7.7 Second-order Tikhonov regularization source history solution determined from the L-curve of Figure 7.6, with true history (Figure 7.1) shown as a dashed curve.

bound of 0.36 and an upper bound of 0.73 for the average concentration during this time period. The true concentration average over this interval (Figure 7.1) is 0.57.

7.2. SPARSITY REGULARIZATION

In some situations there are reasons to expect that many of the unknown model parameters will be zero. Rather than using Tikhonov regularization to minimize $\|\mathbf{m}\|_2$, we may choose to minimize the number of nonzero entries in $\mathbf{m}$ to obtain a sparse model.

The notation $\|\mathbf{m}\|_0$ is commonly used to denote the number of nonzero entries in $\mathbf{m}$ (Note that this 0-norm definition is nonstandard because it is inconsistent with the definition of the p-norm in (A.85) and does not satisfy the requirements for a vector norm in Section A.7.) We can formulate a corresponding regularized inverse problem as

$$\begin{aligned} \min \ & \|\mathbf{m}\|_0 \\ & \|\mathbf{Gm} - \mathbf{d}\|_2 \le \delta. \end{aligned} \tag{7.4}$$

Unfortunately, these kinds of optimization problems can be extremely difficult to solve.

A surprisingly effective alternative to (7.4) is to instead find the least squares solution that minimizes $\|\mathbf{m}\|_1$. To see that this is a reasonable approach, consider the set of models with $\|\mathbf{m}\|_2 = 1$. Among the models with $\|\mathbf{m}\|_2 = 1$, it turns out that the models
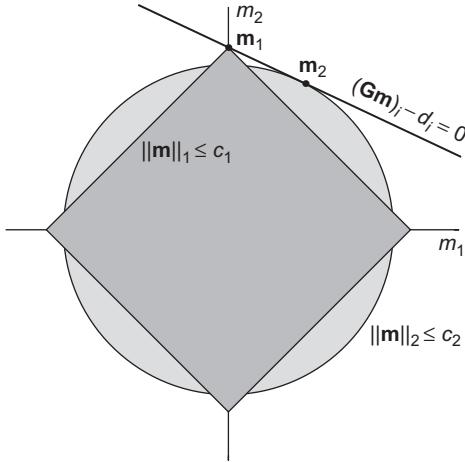
Figure 7.8 Two-dimensional demonstration of the use of model 1-norm minimization (7.5) to obtain sparsity regularization. The square shaded area shows the region $\|\mathbf{m}\|_1 \leq c_1$, while the circle shows the region with $\|\mathbf{m}\|_2 \leq c_2$. An arbitrary constraint equation in this 2-dimensional model space, $(\mathbf{Gm})_i - d_i = 0$, defines a line. The minimum 2-norm residual model satisfying the constraint, $\mathbf{m}_2$, will not generally be sparse. However, the minimum 1-norm model satisfying the constraint, $\mathbf{m}_1 = [0 \; c_1]^T$, will tend to be sparse due to the presence of corners in the 1-norm contour.

with precisely one nonzero entry of $+1$ or -1 have the smallest 1–norms (Figure 7.8). Thus, regularizing a least squares problem to minimize its model 1–norm will tend to produce sparse solutions. This tendency for 1–norm regularized models to be sparse becomes even more prominent in higher dimensions. The heuristic approach of minimizing $\|\mathbf{m}\|_1$ instead of $\|\mathbf{m}\|_0$ works very well in practice, and recent work [21] has demonstrated reasonable conditions under which the solution to the 1–norm regularized problem is identical to or at least close to the solution of the 0–norm regularized problem (7.4).

The L_1 regularized least squares problem can be written as

$$\min \; \|\mathbf{m}\|_1$$
$$\|\mathbf{Gm} - \mathbf{d}\|_2 \leq \delta. \tag{7.5}$$

Using the standard approach of moving the constraint into the objective function, we can select a positive regularization parameter, α, so that this is equivalent to

$$\min \; \|\mathbf{Gm} - \mathbf{d}\|_2^2 + \alpha \|\mathbf{m}\|_1. \tag{7.6}$$

This is a convex optimization problem that can be solved efficiently by many different algorithms. We next present an iterative least squares solution method.

7.3. USING IRLS TO SOLVE L_1 REGULARIZED PROBLEMS

To solve problems of the form of (7.6) we extend the IRLS algorithm introduced in Chapter 2 for L_1 regression. Let

$$f(\mathbf{m}) = \|\mathbf{Gm} - \mathbf{d}\|_2^2 + \alpha\|\mathbf{Lm}\|_1 \tag{7.7}$$

and let

$$\mathbf{y} = \mathbf{Lm}. \tag{7.8}$$

At points where any element of $\mathbf{Lm}$ is zero, $f(\mathbf{m})$ is not differentiable because of the nondifferentiability of the absolute values in the L_1 norm. However, at other points we can easily compute the gradient of f.

$$\nabla f(\mathbf{m}) = 2\mathbf{G}^T\mathbf{Gm} - 2\mathbf{G}^T\mathbf{d} + \alpha \sum_{i=1}^{m} \nabla|y_i|. \tag{7.9}$$

For y_i nonzero,

$$\frac{\partial|y_i|}{\partial m_k} = L_{i,k}\,\text{sgn}(y_i). \tag{7.10}$$

Writing $\text{sgn}(y_i)$ as $y_i/|y_i|$, and using $\|\mathbf{Lm}\|_1 = \sum_{i=1}^{m} |y_i|$, we obtain

$$\frac{\partial\|\mathbf{Lm}\|_1}{\partial m_k} = \sum_{i=1}^{m} L_{i,k}\frac{y_i}{|y_i|}. \tag{7.11}$$

Let $\mathbf{W}$ be the diagonal matrix with elements

$$W_{i,i} = \frac{1}{|y_i|}. \tag{7.12}$$

Then

$$\nabla\left(\|\mathbf{Lm}\|_1\right) = \mathbf{L}^T\mathbf{WLm} \tag{7.13}$$

and

$$\nabla f(\mathbf{m}) = 2\mathbf{G}^T\mathbf{Gm} - 2\mathbf{G}^T\mathbf{d} + \alpha\mathbf{L}^T\mathbf{WLm}. \tag{7.14}$$

Setting $\nabla f(\mathbf{m}) = \mathbf{0}$, we obtain

$$\left(2\mathbf{G}^T\mathbf{G} + \alpha\mathbf{L}^T\mathbf{WL}\right)\mathbf{m} = 2\mathbf{G}^T\mathbf{d}. \tag{7.15}$$

Because $\mathbf{W}$ depends on $\mathbf{Lm}$, this is a nonlinear system of equations. Furthermore, $\mathbf{W}$ is not defined at any point where $\mathbf{Lm}$ has a zero element.

To accommodate the nondifferentiability of the 1-norm at the origin, as in Chapter 2, we set a tolerance ϵ, and let

$$W_{i,i} = \begin{cases} 1/|\gamma_i| & |\gamma_i| \geq \epsilon \\ 1/\epsilon & |\gamma_i| < \epsilon. \end{cases} \tag{7.16}$$

In the IRLS procedure, we solve (7.15), update $\mathbf{W}$ according to (7.16), and iterate until the solution has converged.

The system of equations (7.15) can be seen as the normal equations for the least squares problem,

$$\min \left\| \begin{bmatrix} \mathbf{G} \\ \sqrt{\tfrac{\alpha}{2}}\sqrt{\mathbf{W}}\mathbf{L} \end{bmatrix} \mathbf{m} - \begin{bmatrix} \mathbf{d} \\ \mathbf{0} \end{bmatrix} \right\|_2. \tag{7.17}$$

When $\mathbf{G}$ is large and sparse it can be extremely advantageous to apply LSQR to solve the least squares problem (7.17) rather than solving the system of equations (7.15).

Example 7.2

Consider a deconvolution example in which a controlled surface source sends seismic waves down into the earth. Seismic reflections will occur from abrupt material discontinuities at various depths, and a seismogram of the reflected signals (which will generally include multiple reflections) is recorded at the surface. Here, $g(t)$ is the known source signal, $d(t)$ is the recorded seismogram, and $m(t)$ is the unknown reflection response of the earth. We'd like to recover the times of these reflections to study the responsible earth structure. This is a linear forward problem characterized by a convolution (e.g., (1.11))

$$d(t) = \int_{-\infty}^{\infty} g(t - \xi)\, m(\xi)\, d\xi. \tag{7.18}$$

In this highly simplified reflection seismology example, we will assume a one-dimensional structure with depth and consider upward- and downward-traveling plane compressional seismic waves detected by a vertically oriented seismometer. In this case, the presence of horizontally oriented seismic structural discontinuities in an otherwise smooth medium will ideally be manifested in $m(t)$ as a sequence of delta functions. As in Chapter 1, we will discretize $m(t)$ as a model vector $\mathbf{m}$. We seek models where $\mathbf{m}$ consists of an optimal sequence of spikes.

The source signal is a short pulse with a characteristic time of around $0.4\,\text{s}$:

$$g(t) = e^{-5t}\, \sin(10t). \tag{7.19}$$

We'll add independent normally distributed random noise to the corresponding recorded data (7.18) and attempt to recover $m(t)$ in regularized inversions. All time series were sampled uniformly at 100 samples/s to create model and data vectors of 1000 points spanning 10-s time intervals. The true reflectivity model (Figure 7.9) is the response of a single subsurface reflector with a coefficient of $r = 0.4$, located at a depth corresponding to a two-way seismic travel time from the surface of $\tau = 1.5$ s. This source signal $g(t)$ begins at 0.5 s, and the primary reflection and surface-reflector

Figure 7.9   The target reflectivity model $m(t)$.

Figure 7.10   Data (7.18) with $N(0, (0.002))^2$ noise added.

multiples appear in Figure 7.9 as alternating–sign spikes at intervals of τ with amplitudes that decrease by a factor of r upon each reflection. Figure 7.10 shows the corresponding data from (7.18) with $N(0, (0.002^2))$ noise added. The noise and smoothing from the convolution largely obscure the signal corresponding to the later impulses of Figure 7.9.

Figure 7.11 shows the L–curve for this problem with zeroth–order regularization, using the frequency–domain techniques described in Chapter 8, and Figure 7.12 shows

Figure 7.11   L-curve for zeroth-order Tikhonov solution with corner at $\alpha = 0.01$.

Figure 7.12   Zeroth-order Tikhonov solution.
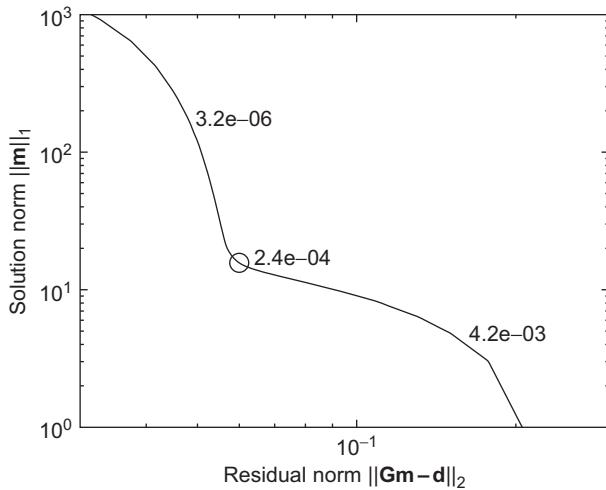
Figure 7.13 L-curve for the 1-norm regularized solution with corner at $\alpha = 2.4 \times 10^{-4}$.
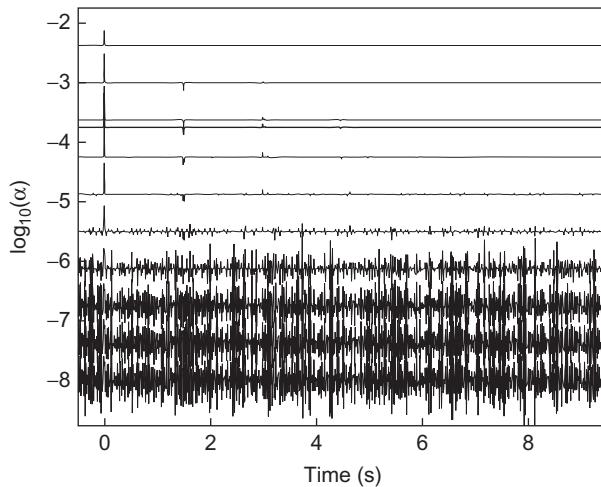
Figure 7.14 1-norm regularized solutions corresponding to the range of regularization parameters used to construct Figure 7.13. Model corresponding to the L-curve corner is shown in bold.

a corresponding solution. The impulses are broadened relative to the true model due to the associated regularization bias, and there are noise-generated secondary peaks. Figures 7.13, 7.14, and 7.15, however, show that using (7.5) produces excellent recovery of $m(t)$. Note that the spikes are correctly placed in time and that all are resolved save the tiny pulse at 7.5 s, which has an amplitude of only $r^5 \approx 0.01$, or just 1% of the amplitude

Figure 7.15 1-norm regularized solution for $\alpha = 2.4 \times 10^{-4}$ from Figure 7.14.

of the source spike. The amplitude of the spikes is reduced and they are slightly broader than they should be, but the model is vastly better than that obtained with Tikhonov regularization.

In many situations, there is no reason to bias models to be sparse. If the true model is not sparse, then sparsity regularization will of course produce poor model recovery. However, it is possible in some cases to implement a change of variables so that the coefficients of the model with respect to the new basis will be sparse, or can at least be well approximated by a sparse model.

Recall from Appendix A that if a set of vectors $\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_n$ form a basis for R^n, then any vector $\mathbf{m}$ in R^n can be written as a linear combination of the basis vectors with n coefficients, x_i:

$$\mathbf{m} = x_1 \mathbf{w}_1 + x_2 \mathbf{w}_2 + \cdots + x_n \mathbf{w}_n. \tag{7.20}$$

In vector–matrix form, this can be written as

$$\mathbf{m} = \mathbf{W}\mathbf{x} \tag{7.21}$$

where the basis vectors $\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_n$ are the columns of $\mathbf{W}$. We can also perform the change of variables in the reverse direction as

$$\mathbf{x} = \mathbf{W}^{-1}\mathbf{m}. \tag{7.22}$$

In practice, linear transformation algorithms, such as the fast Fourier transform (FFT; Chapter 8) or discrete cosine transform (DCT), are often used in place of the matrix-vector multiplications of (7.21) and (7.22) to effect the corresponding forward and inverse operations.

Example 7.3

Figure 7.16 shows the 40,000-pixel image from Example 6.3. If we were to zero out any significant fraction of these pixels, the effect on the image would be quite obvious. However, if instead we encode the image in terms of the discrete cosine transform (DCT) of the pixel values, a sparser representation of the image is possible that closely approximates the original. Calculating the discrete cosine transform of the two-dimensional image using MATLAB's **dct2** command produces a set of 40,000 coefficients. Of these 40,000 coefficients, the 16,000 coefficients (40% of the original coefficients) that were smallest in absolute value were set to zero.

After inverting the DCT, we produced the image shown in Figure 7.17. This second image appears to be very similar to the original image, yet 40% fewer coefficients were used to represent the image in terms of the DCT basis. This is because the
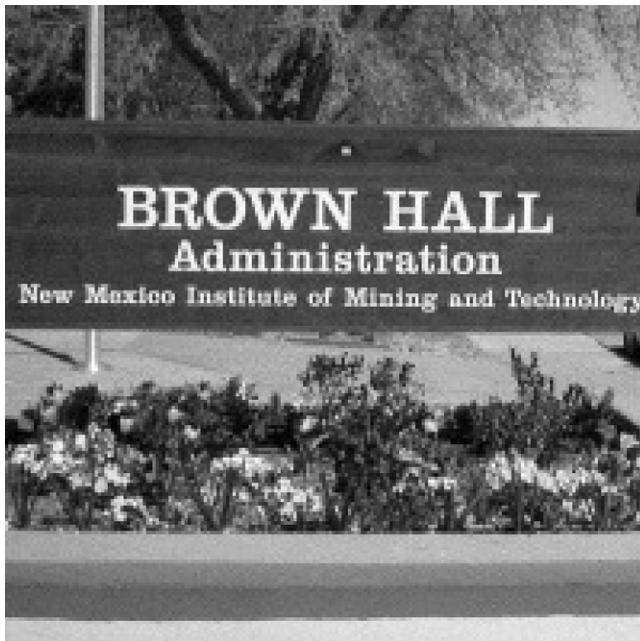
Figure 7.16　Image before discrete cosine transform threshholding.
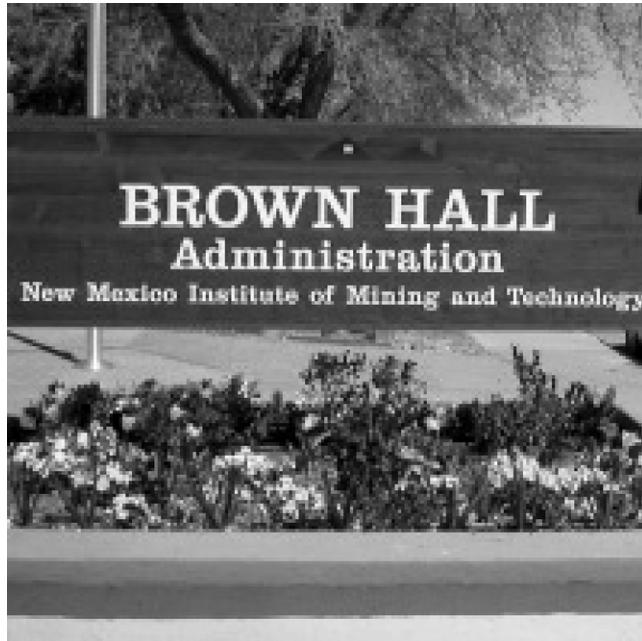
Figure 7.17 Image after removing smallest 40% of DCT coefficients.

coefficients associated with the 16,000 unused basis vectors in the original image were small.

The discrete cosine transform is the basis for the widely used JPEG standard for image compression. Wavelet bases are also widely used in image processing, and thresholding the wavelet transform produces similar results in compressing images. The newer JPEG 2000 standard for image compression uses wavelets rather than the discrete cosine transform.

In **compressive sensing**, we apply sparsity regularization using a basis chosen so that the model can be represented by a linear combination of the basis vectors with a sparse set of coefficients. The regularized problem can be written using (7.21) and (7.6) as

$$\min \ \|\mathbf{GWx} - \mathbf{d}\|_2^2 + \alpha \|\mathbf{x}\|_1, \tag{7.23}$$

where $\mathbf{G}$ is called the **measurement matrix**.

The recovery of a sparse solution using (7.23) depends on the details and appropriateness of the particular $\mathbf{G}$ and $\mathbf{W}$ used in (7.23). The analysis and implementation of compressive schemes become relatively simple if the measurement matrix is random [27].

Example 7.4

Consider the recovery of a signal, $\mathbf{m}$, shown in Figure 7.18. This 10-s long time series of $n = 1001$ time points, t_i, is sampled at 100 samples/s and consists of two sine waves at $f_1 = 25$ and $f_2 = 35$ Hz:

$$m_i = h_i \cdot \left(5 \, \cos(2\pi f_1 t_i) + 2 \, \cos(2\pi f_2 t_i) \right) \quad 1 \leq i \leq n, \qquad (7.24)$$

where the signal envelope has also been smoothed with term-by-term multiplication by a Hann taper function,

$$h_i = \frac{1}{2} \, (1 - \cos(2\pi (i-1)/n)) \quad 1 \leq i \leq n. \qquad (7.25)$$

In the standard basis, the signal (7.24), shown in Figure 7.18, is obviously not sparse. However, because it is constructed from two cosine components, this model has a very sparse representation in the DCT basis.

We use a random measurement matrix $\mathbf{G}$ that is $m = 100$ by $n = 1001$, with entries chosen independently from $N(0, 1)$ to generate a synthetic set of 100 data measurements, $\mathbf{d} = \mathbf{Gm}$, and then add independent $N(0, 25)$ noise. The matrix of basis vectors, $\mathbf{W}$, is constructed by applying the discrete cosine transform to standard basis vectors for the model space R^n.

Figures 7.19 and 7.20 show solutions obtained by solving $\mathbf{Gm} = \mathbf{d}$ using second-order Tikhonov regularization. The highlighted solution ($\alpha = 100$) fits the data with a
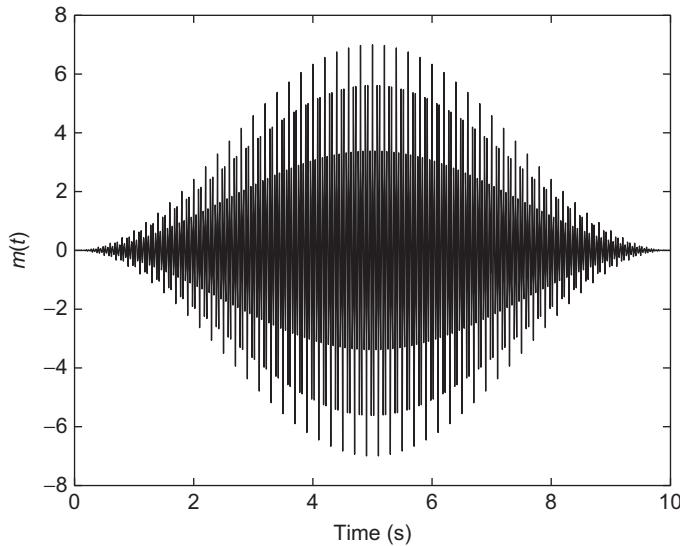
Figure 7.18 A 1001-point signal consisting of 25- and 35-Hz cosines (7.24), multiplied by a Hann taper.
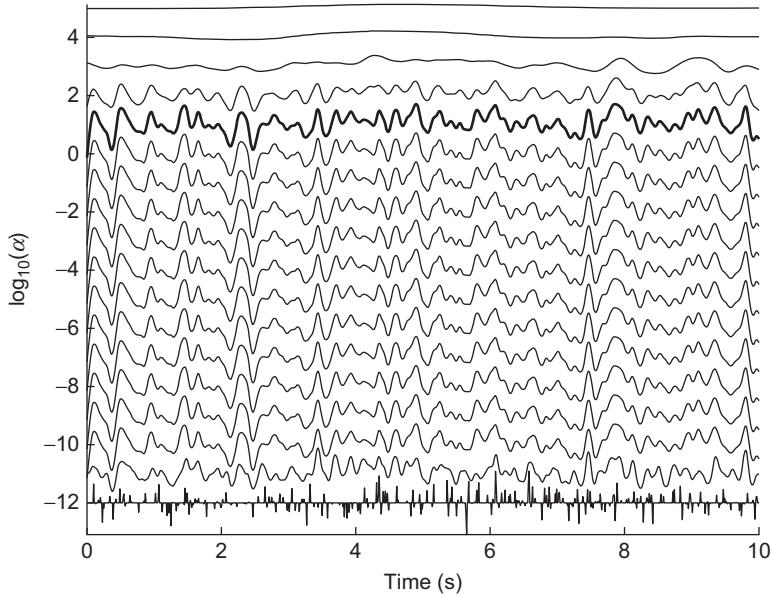
Figure 7.19 Signal recovery using second-order Tikhonov regularization. Solution amplitudes are normalized to improve legibility.
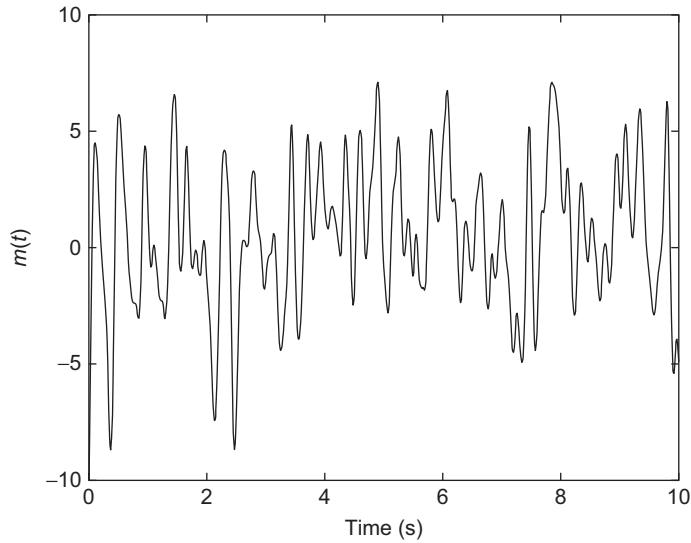
Figure 7.20 A representative solution using second-order Tikhonov regularization that approximately satisfies the discrepancy principle from Figure 7.19 ($\alpha = 10$).

residual norm that approximately satisfies the discrepancy principle, so that $\|\mathbf{Gm} - \mathbf{d}\|$ $\approx 5 \cdot \sqrt{m} = 50$. However, this solution, and others across a wide range of α values, bear essentially no resemblance to the true signal shown in Figure 7.18.

An L-curve obtained by solving (7.23) for n DCT coefficients $\mathbf{x}$, and subsequently constructing models $\mathbf{Wx}$ for various α values, shows a very abrupt corner near $\alpha = 10^{-11}$. For very low levels of regularization beyond this corner, solutions rapidly transition from excellent model recovery to essentially random noise. For α values greater than a few hundred, on the other hand, solutions rapidly approach the zero model. For a wide range of α values between these extreme values, the recovered signal $\mathbf{Wx}$ is stable and is quite close to the true signal (Figures 7.21 and 7.22).

It may seem paradoxical that we can recover a 1001-point signal so well from only 100 random linear combination measurements, when the Nyquist sampling theorem (8.40) suggests that at least $10\,\text{s} \times 35\,\text{Hz} \times 2 = 700$ equally spaced samples should be required to accurately represent it. This is possible in the current example because the measured signal is sparse with respect to the DCT basis, and just a few cosine basis functions are therefore necessary to represent it well. The corresponding sparse set of coefficients is easily recovered when sparsity regularization is employed.

7.4. TOTAL VARIATION

Total variation (TV) regularization is appropriate for problems where we expect there to be discontinuous jumps in the model. In the one-dimensional case, the TV regularization function is

$$\text{TV}(\mathbf{m}) = \sum_{i=1}^{n-1} |m_{i+1} - m_i| \tag{7.26}$$

$$= \|\mathbf{Lm}\|_1, \tag{7.27}$$

where, for one-dimensional models, $\mathbf{L} = \mathbf{L}_1$ (4.27). In higher-dimensional problems, $\mathbf{L}$ is a discretization of the gradient operator.

In first- and second-order Tikhonov regularization, discontinuities in the model are smoothed out and do not show up well in the inverse solution. This is because smooth changes are penalized less by the regularization term than sharp ones. The particular feature of (7.27) is that the regularization term imposes a constraint to keep the number of discontinuous transitions to a minimum (i.e., keeping $\mathbf{Lm}$ sparse). We can insert the TV regularization term (7.27) in place of $\|\mathbf{Lm}\|_2^2$ in the Tikhonov regularization optimization problem to obtain

$$\min \|\mathbf{Gm} - \mathbf{d}\|_2^2 + \alpha \|\mathbf{Lm}\|_1, \tag{7.28}$$

which is a convex optimization problem in the form of (7.6) that imposes a higher-order sparsity constraint.
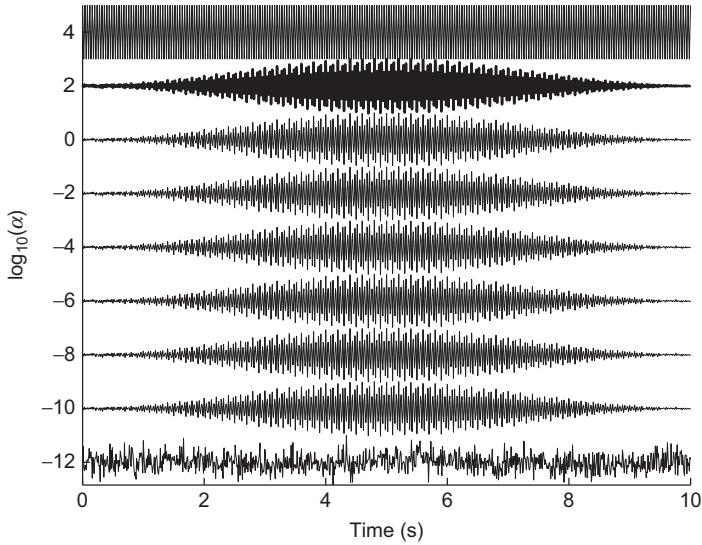
Figure 7.21 Signal recovery using compressive sensing with 100 signal measurements. Solution amplitudes are normalized to improve legibility.
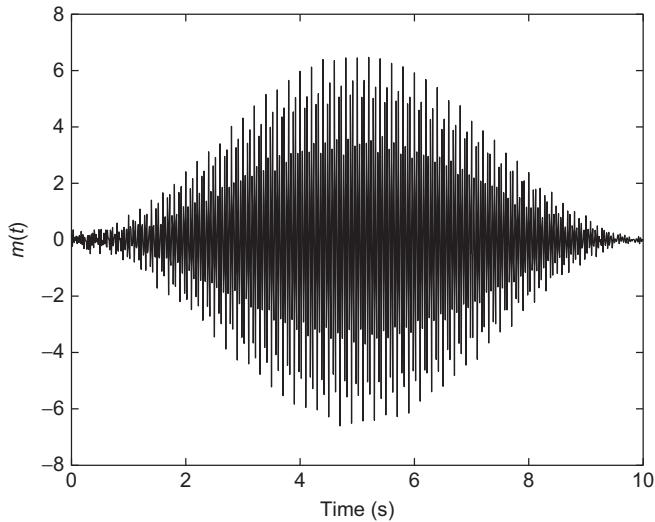
Figure 7.22 A representative solution obtained from Figure 7.21 using compressive sensing with $\alpha = 100$ that approximately satisfies the discrepancy principle.

This approach has seen wide use in the problem of "denoising" a model [122]. Denoising is a linear inverse problem in which $\mathbf{G} = \mathbf{I}$. In denoising, the general goal is to remove features considered to be spurious from a signal, while still retaining other key features, such as long-term trends and perhaps sharp discontinuities.

Example 7.5

Figure 7.23 shows a signal that is smooth with the exception of two discontinuous jumps at 2 and 7 s. Figure 7.24 shows the signal after $N(0, 4)$ independent noise has
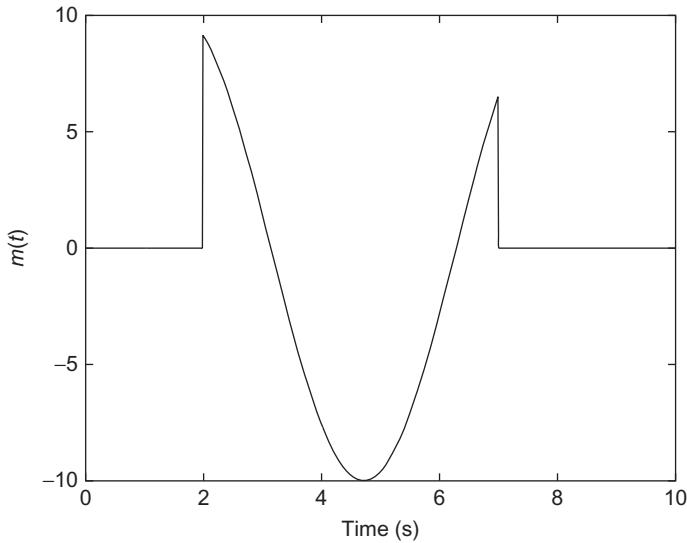
Figure 7.23 An otherwise smooth signal with two sharp discontinuities.
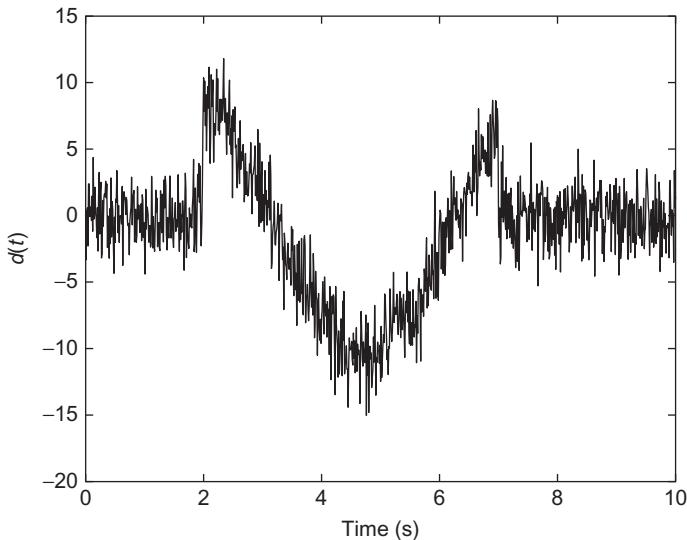
Figure 7.24 The signal with $N(0, 4)$ noise added.

been added. Our goal is to remove this noise from the signal while optimally recovering the discontinuities.

We first attempt to denoise the signal using second–order Tikhonov regularization with **L** given by (4.28). The L–curve is shown in Figure 7.25, and the resulting solution is shown in Figure 7.26. If the location of the large discontinuities is a feature that we wish
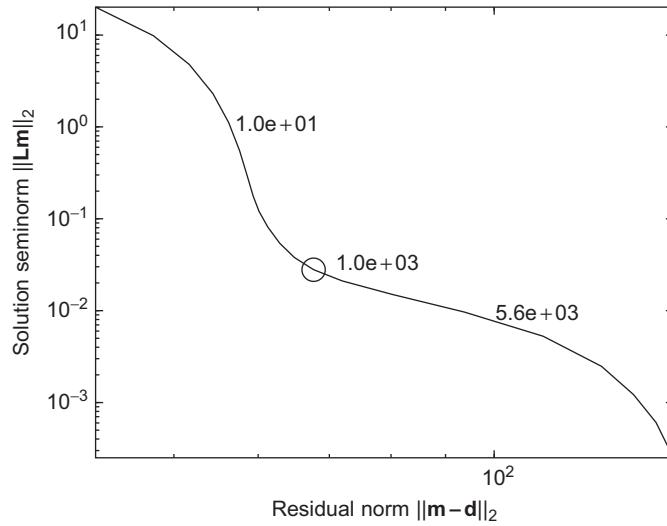
Figure 7.25 L-curve with second-order Tikhonov regularization with corner at $\alpha = 1000$.
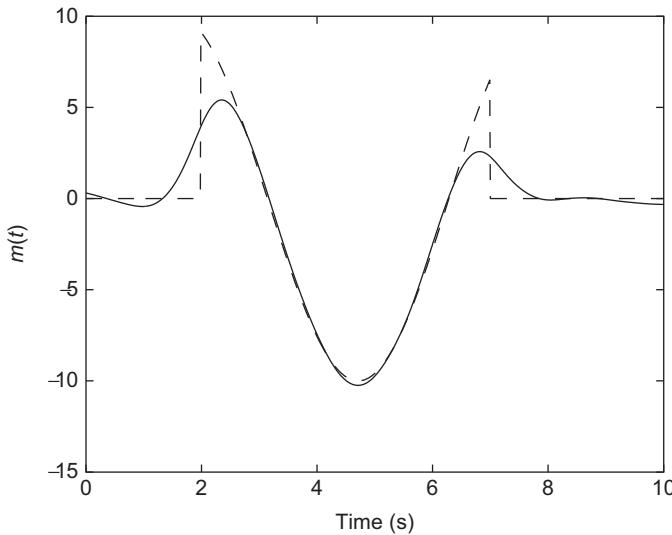
Figure 7.26 Second-order Tikhonov solution ($\alpha = 1000$) with the true model plotted as a dashed curve.

to recover, this solution obviously does not do a good job because the regularization scheme biases the model based on smoothness.

We next apply TV regularization to solve the problem. Figure 7.27 shows the L-curve trade-off between $\|\mathbf{m} - \mathbf{d}\|_2$ and $\|\mathbf{Lm}\|_1$, and the presence of a corner near
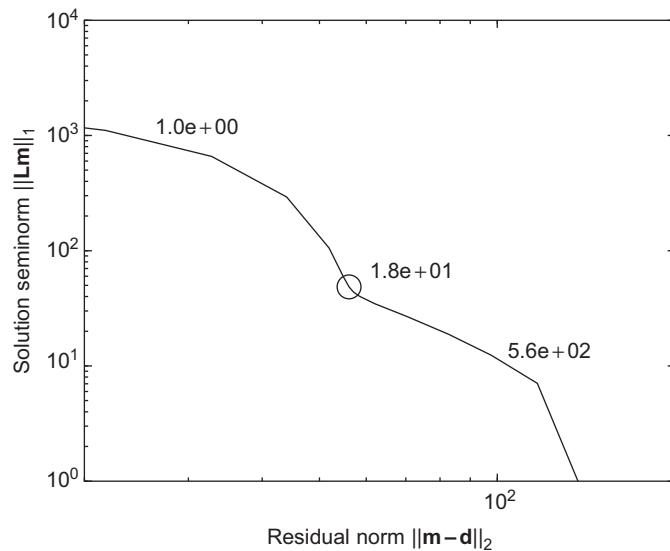
Figure 7.27   L-curve for TV regularization with corner at $\alpha = 18$.
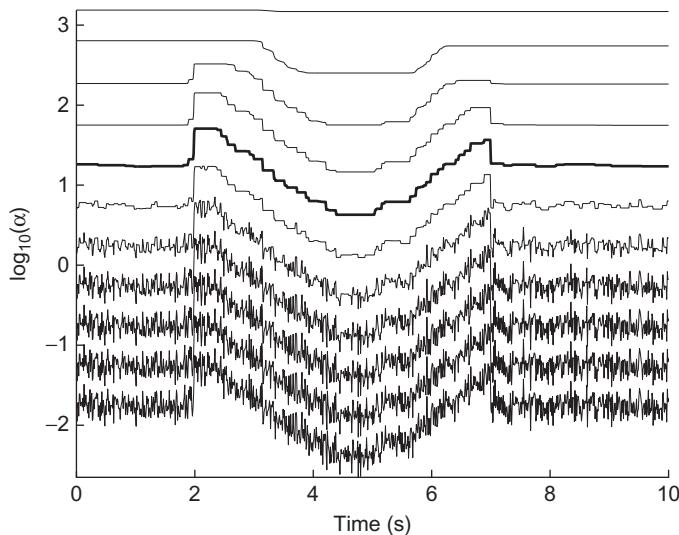
Figure 7.28   Signal recovered with TV regularization with model corresponding to the L-curve corner shown in bold.
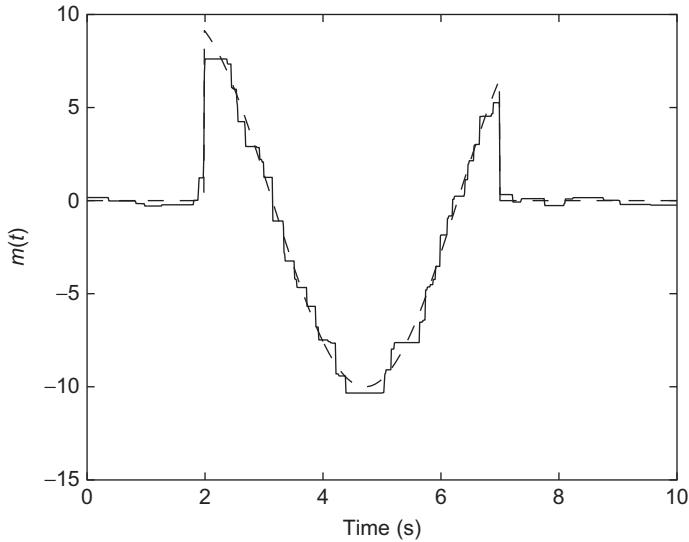
Figure 7.29 Signal recovered with TV regularization for $\alpha = 18$ from Figure 7.28 with the true model plotted as a dashed curve.

$\alpha \approx 18$. Figure 7.28 shows the corresponding models for various levels of regularization, with the optimal one selected from the L-curve shown in Figure 7.29. This piecewise-constant recovery of the original signal reveals the large discontinuities at 2 s and 7 s quite well.

7.5. EXERCISES

1. Using the method of Lagrange multipliers, develop a formula that can be used to solve

$$\min \mathbf{c}^T \mathbf{m}$$
$$\|\mathbf{Gm} - \mathbf{d}\|_2 \leq \delta. \tag{7.29}$$

2. Noisy seismic travel time data (units of s) from equally spaced (every 20 m) seismometers deployed in a 1000-m deep vertical seismic profile experiment (the same geometry as in Example 4.4) are to be inverted to reveal the slowness of the earth as a function of depth. Invert the data in **vsp.mat** using TV regularization with an L-curve analysis and assess the presence of any likely discontinuities. Compare this with the least squares solution and with a direct finite-difference solution suggested by (1.22).

3. In many cases, it is valuable to implement multiple regularization constraints, such as smoothness combined with bounds. Consider the gravity anomaly data in the vector **dn** (units of m/s^2) found in the file **gdata.mat**. The data are taken at the surface above a 1000 m section of a partially debris–blocked buried pipe located with its center at a depth of 25 m. Use the methodology of Example 1.4 to invert these data for subsurface density perturbations, assuming that the pipe has a cross-sectional area of 1 m^2. Anomalous density along the pipe relative to background density is parameterized as point masses located at 100 equally spaced points along the section, as specified by the vector **rhox**. The noisy surface gravity anomaly observations contained in the vector **dn** are made at 500 points along the same 1000 m span as the buried pipe section of interest at points specified in the vector **x**. Invert the data for density perturbations along the buried pipe by assuming that all mass in each segment can be concentrated at the points specified by **rhox**.
   a. Invert for density perturbations along the pipe transect in kg/m^3 using least squares.
   b. Invert for density perturbations along the pipe transect in kg/m^3 using second-order Tikhonov regularization and an L-curve analysis.
   c. Solve the problem using second-order Tikhonov regularization combined with BVLS and an L-curve analysis. Employ the knowledge that density variations for $x < 91$ and $x > 899$ m are negligible, and that the density variations overall are bounded by $-150 \text{ kg/m}^3 \leq \delta\rho(x) \leq 150 \text{ kg/m}^3$.
   d. Compare and discuss your three solutions. Where are the prominent maxima and minima located?
4. Returning to the problem in Exercise 3.1, solve for the density profile using total variation regularization. How does your solution compare to the solutions obtained in Exercise 3.1 and Exercise 4.4?

7.6. NOTES AND FURTHER READING

Methods for bounded variables least squares problems and minimizing a linear function subject to a bound on the misfit are given in [150]. Some applications of these techniques can be found in [73, 125, 128, 148, 149]. There is a long history of sparsity regularization in geophysical inverse problems. Claerbout and Muir were early advocates of this approach [31]; however, there was little theoretical justification for this prior to the work of Candes, Romberg, and Tao, which brought about a resurgence of interest in this methodology [24, 25, 26]. In recent years there has been an explosion of interest in compressive (or compressed) sensing, with applications in many areas of signal and image processing and inverse problems. There are numerous surveys and tutorials on compressive sensing, including [21, 27, 104]. Methods for total variation regularization are discussed in [122, 169].

Fourier Techniques

Synopsis

The formulation of a general linear forward problem as a convolution is derived. The Fourier transform, Fourier basis functions, and the convolution theorem are introduced for continuous- and discrete-time systems. The inverse problem of deconvolution is explored in the context of the convolution theorem. Water level and Tikhonov regularization in the frequency domain are employed to solve discrete deconvolution problems.

8.1. LINEAR SYSTEMS IN THE TIME AND FREQUENCY DOMAINS

A remarkable feature of linear time-invariant systems is that the forward problem can generally be described by a **convolution** (1.11),

$$d(t) = \int_{-\infty}^{\infty} m(\tau)g(t-\tau) \ d\tau. \tag{8.1}$$

Inverse problems involving such systems can be solved by **deconvolution**. Here, the independent variable t is time, and the data d, model m, and system kernel g are all time functions. However, the results here are equally applicable to spatial problems (e.g., Example 8.1) and to higher dimensions. We will first overview the essentials of Fourier theory in the context of performing convolutions and deconvolutions.

Consider a linear time-invariant forward operator, G, that maps an unknown model function, $m(t)$, into an observable data function, $d(t)$,

$$d(t) = G[m(t)]. \tag{8.2}$$

Because it is linear and time-invariant, (8.2) obeys the principles of superposition (1.5) and scaling (1.6). It is possible to show that any such system can be cast in the form of (8.1), by utilizing the **sifting property** of the **impulse** or **delta function**, $\delta(t)$. The delta function can be conceptualized as the limiting case of a pulse as its width goes to zero, its height goes to infinity, and its area stays constant and equal to one, for example,

as the limiting case of a unit area Gaussian pulse centered on the origin,

$$\delta(t) = \lim_{\sigma \to 0} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}t^2/\sigma^2}. \tag{8.3}$$

The sifting property of the delta function extracts the value of a function at a particular point from within a definite integral

$$\int_a^b f(t)\delta(t - t_0)\ dt = \begin{cases} f(t_0) & a \le t_0 \le b \\ 0 & \text{elsewhere,} \end{cases} \tag{8.4}$$

for any $f(t)$ that is continuous at finite $t = t_0$. The **impulse response**, or **Green's function**, $g(t)$, for a system where the model and data are related by an operator G, as in (8.2), is defined as the system output (data) for a delta function input (model),

$$g(t) = G[\delta(t)]. \tag{8.5}$$

The concept of the impulse response can be used to demonstrate our initial assertion in this chapter that linear time-invariant forward problems can generally be expressed as convolutions.

Theorem 8.1 *We begin with a restatement of (8.2);*

$$d(t) = G[m(t)]. \tag{8.6}$$

m(t) can be written as a summation of impulse functions by invoking (8.4),

$$m(t) = \int_{-\infty}^{\infty} m(\tau)\delta(t - \tau)\ d\tau, \tag{8.7}$$

and we can thus write

$$d(t) = G\left[\int_{-\infty}^{\infty} m(\tau)\delta(t - \tau)\ d\tau\right]. \tag{8.8}$$

Applying the definition of the integral as a limit of a sum of Δt-width rectangular areas as Δt goes to zero, we have

$$d(t) = G\left[\lim_{\Delta\tau \to 0} \sum_{n=-\infty}^{\infty} m(\tau_n)\delta(t - \tau_n)\Delta\tau\right]. \tag{8.9}$$

Because G is a linear and time-invariant operator, superposition allows us to move the operator inside of the summation in (8.9). Furthermore, using the scaling relation, we can factor out the

coefficients $m(\tau_n)$ from inside of the operator to obtain

$$d(t) = \lim_{\Delta\tau \to 0} \sum_{n=-\infty}^{\infty} m(\tau_n) G[\delta(t - \tau_n)] \Delta\tau. \tag{8.10}$$

In the limit as $\Delta t \to 0$ and substituting (8.5), (8.10) thus defines the integral

$$d(t) = \int_{-\infty}^{\infty} m(\tau) g(t - \tau) \, d\tau, \tag{8.11}$$

which is identical to (8.1), the convolution of $m(t)$ and $g(t)$.

Note that it is common to denote the convolution of two functions $h_1(t)$ and $h_2(t)$ simply as $h_1(t) * h_2(t) = h_2(t) * h_1(t)$. The indicated reciprocity is easily demonstrated from (8.11) with a change of variables.

Convolution can thus be used to describe the mapping of models to data for any linear time-invariant system. Important examples include the mapping between a physical process and a set of observables that occurs in a forward problem, or the output of a linear instrumentation system. For example, a perfect instrument that recorded some model $m(t)$ with perfect fidelity, but imposed a time delay t_0, would have a time-delayed delta function impulse response:

$$d(t) = m(t) * \delta(t - t_0) \tag{8.12}$$

$$= \int_{-\infty}^{\infty} m(\tau)\delta(t - t_0 - \tau) \, d\tau \tag{8.13}$$

$$= m(t - t_0). \tag{8.14}$$

We define the **Fourier transform** of a function, $h(t)$, as

$$\mathcal{H}(f) = \mathcal{F}[h(t)] \tag{8.15}$$

$$= \int_{-\infty}^{\infty} h(t)e^{-\imath 2\pi ft} \, dt, \tag{8.16}$$

and the corresponding **inverse Fourier transform** as

$$h(t) = \mathcal{F}^{-1}[\mathcal{H}(f)] \tag{8.17}$$

$$= \int_{-\infty}^{\infty} \mathcal{H}(f)e^{\imath 2\pi ft} \, df, \tag{8.18}$$

where f has units of reciprocal time (frequency) and $\imath$ is the square root of -1.

The impulse response of a linear system, $g(t) = G[\delta(t)]$, is commonly called the **time domain response** of the system when the independent variable is time. Note, however, that the impulse response concept and Fourier theory are trivially generalizable to functions of space or other variables. The Fourier transform or **spectrum** of the impulse response, $\mathcal{F}(g(t))$, is variously referred to as the **frequency response**, or **transfer function** of the system.

The Fourier transform (8.16) provides a formula for evaluating the spectrum. The inverse Fourier transform (8.18) states that the time domain function $g(t)$ can be exactly reconstructed via a weighted integration of functions of the form $e^{\iota 2\pi ft}$, where the weighting is provided by the spectrum $\mathcal{G}(f)$. The essence of Fourier analysis expressed in the transforms (8.16) and (8.18) is that general functions can be expressed and analyzed as a continuous weighted superposition of **Fourier basis functions** of the form $e^{\iota 2\pi ft}$. Note that this representation differs from the finite basis function sets considered in Chapter 5 in that the sets of Fourier basis functions in (8.16) and (8.18) are infinite.

Because the Fourier basis functions, $e^{\iota 2\pi ft} = \cos(2\pi ft) + \iota \sin(2\pi ft)$, are complex, a general real-valued function $h(t)$ will have a corresponding spectrum, $\mathcal{H}(f)$, that can be complex valued. $|\mathcal{H}(f)|$ is called the **spectral amplitude**. It is commonly very convenient to express spectra in polar form,

$$\mathcal{H}(f) = |\mathcal{H}(f)|e^{\iota \theta(f)}, \tag{8.19}$$

where the angle that $\mathcal{H}(f)$ makes in the complex plane,

$$\theta(f) = \tan^{-1}\left(\frac{\text{imag}(\mathcal{G}(f))}{\text{real}(\mathcal{G}(f))}\right), \tag{8.20}$$

is called the **spectral phase**.

A useful feature of the Fourier transform is that it is **length preserving** for the 2-norm measure, in the sense that the 2-norm of a function and its Fourier transform are identical. Consider a time domain norm of the form of (5.3) for a general complex function $h(t)$

$$\|h(t)\|_2^2 = \int\limits_{-\infty}^{\infty} h(t)h^*(t) \; dt. \tag{8.21}$$

Expressing $h(t)$ using the inverse Fourier transform (8.18) and applying complex conjugation (denoted by an asterisk) gives

$$\|h(t)\|_2^2 = \int\limits_{-\infty}^{\infty} h(t)\left(\int\limits_{-\infty}^{\infty} \mathcal{H}^*(f)e^{-\iota 2\pi ft} \; df\right) dt. \tag{8.22}$$

Interchanging the order of integration and utilizing (8.16), we have

$$\|h(t)\|_2^2 = \int_{-\infty}^{\infty} \mathcal{H}^*(f) \left(\int_{-\infty}^{\infty} h(t)e^{-\iota 2\pi ft} \, dt \right) df$$

$$= \int_{-\infty}^{\infty} \mathcal{H}^*(f)\mathcal{H}(f) \, df \qquad (8.23)$$

$$= \|\mathcal{H}(f)\|_2^2,$$

which is the 2-norm of the Fourier transform of $h(t)$. Equation (8.23) is most commonly referred to as **Parseval's theorem**.

We have shown in (8.11) that the data produced by a linear time–invariant system for a model, $m(t)$, is the convolution of $m(t)$ with the system impulse response. Evaluating the Fourier transform of this convolution produces an especially important result.

Theorem 8.2

$$\mathcal{F}[m(t) * g(t)] = \int_{-\infty}^{\infty} \left(\int_{-\infty}^{\infty} m(\tau)g(t-\tau) \, d\tau \right) e^{-\iota 2\pi ft} \, dt. \qquad (8.24)$$

Reversing the order of integration gives

$$\mathcal{F}[m(t) * g(t)] = \int_{-\infty}^{\infty} m(\tau) \left(\int_{-\infty}^{\infty} g(t-\tau)e^{-\iota 2\pi ft} \, dt \right) d\tau. \qquad (8.25)$$

Introducing a change of variables, $\xi = t - \tau$, we obtain

$$\mathcal{F}[m(t) * g(t)] = \int_{-\infty}^{\infty} m(\tau) \left(\int_{-\infty}^{\infty} g(\xi)e^{-\iota 2\pi f(\xi+\tau)} \, d\xi \right) d\tau$$

$$= \left(\int_{-\infty}^{\infty} m(\tau)e^{-\iota 2\pi f\tau} \, d\tau \right) \left(\int_{-\infty}^{\infty} g(\xi)e^{-\iota 2\pi f\xi} \, d\xi \right) \qquad (8.26)$$

$$= \mathcal{M}(f)\mathcal{G}(f).$$

Equation (8.26), which applies to any pair of functions, is called the **convolution theorem**. The theorem states that convolution of two functions in the time domain corresponds to the multiplication of their Fourier transforms in the frequency domain. When the two functions are a model, $m(t)$, and an impulse response, $g(t)$, in a forward problem, (8.26) indicates that the corresponding data spectrum is simply the product of the model and impulse response spectra.

To examine the implications of the convolution theorem more explicitly, consider the data produced by a linear time-invariant system, characterized by the spectrum of the impulse response, $\mathcal{G}(f)$, for a model, $m_0(t)$, that is a Fourier basis function of frequency f_0,

$$m_0(t) = e^{\iota 2\pi f_0 t}. \tag{8.27}$$

The spectrum of (8.27) is a delta function located at $f = f_0$, $\delta(f - f_0)$. This can be seen by constructing an inverse Fourier transform (8.18) that invokes the sifting property of the delta function

$$e^{\iota 2\pi f_0} = \int_{-\infty}^{\infty} \delta(f - f_0)e^{\iota 2\pi ft} \, df. \tag{8.28}$$

The spectrum of the corresponding data is thus, by (8.26), a delta function at $f = f_0$, scaled by the spectrum of $G(t)$:

$$\mathcal{F}\left[G\left[e^{\iota 2\pi f_0 t}\right]\right] = \mathcal{F}\left[\mathcal{G}(f)e^{\iota 2\pi f_0 t}\right] = \delta(f - f_0)\mathcal{G}(f_0). \tag{8.29}$$

The corresponding time domain response is, by (8.18),

$$\int_{-\infty}^{\infty} \mathcal{G}(f_0)\delta(f - f_0)e^{\iota 2\pi ft} \, df = \mathcal{G}(f_0)e^{\iota 2\pi f_0 t}$$
$$= |\mathcal{G}(f_0)|e^{\iota \theta(f_0)}e^{\iota 2\pi f_0 t} \tag{8.30}$$
$$= |\mathcal{G}(f_0)|e^{\iota \theta(f_0)}m_0(t).$$

Linear time-invariant systems thus transform the Fourier basis functions (8.27) to identical functions, altering only their amplitude and phase by frequency-dependent factors $|\mathcal{G}(f_0)|e^{\iota \theta(f_0)}$. Because the system is linear, the transformation for a general input function is just the superposition of all such components at all frequencies, where the appropriate functional weighting is $\mathcal{G}(f)$. Of particular relevance here when considering inverse methods is the result that model basis function amplitudes at frequencies that are weakly mapped to the data (frequencies where $|\mathcal{G}(f)|$ is small) in a forward problem, and/or are obscured by noise, may be difficult or impossible to recover in an inverse problem.

The spectrum of the impulse response (transfer function) can be evaluated in a particularly useful analytical manner when we can express the forward problem $d(t) = G[m(t)]$ as a linear differential equation,

$$a_r m^{(r)} + a_{r-1} m^{(r-1)} + \cdots + a_1 m^{(1)} + a_0 m$$
$$= b_q d^{(q)} + b_{q-1} d^{(q-1)} + \cdots + b_1 d^{(1)} + b_0 d, \quad (8.31)$$

where the superscripts denote the order of time differentiation of m and d, and the a_i and b_i are constant coefficients. Because each term in (8.31) is linear (there are no powers or

other nonlinear functions of m, d, or their derivatives), and because differentiation is itself a linear operation, (8.31) expresses a linear time-invariant system obeying superposition and scaling.

If (8.31) operates on a model of the form $m(t) = e^{\iota 2\pi f t}$, (8.30) indicates that the corresponding output will be $d(t) = \mathcal{G}(f)e^{\iota 2\pi f t}$. Substituting this form of $m(t)$ and differentiating each term, each time derivative, $m^{(k)}$ and $d^{(k)}$, will generate corresponding multipliers $(2\pi \iota f)^k$. Finally, dividing the resulting equation on both sides by $e^{\iota 2\pi f t}$, and solving for $\mathcal{G}(f)$ gives

$$\mathcal{G}(f) = \frac{\mathcal{D}(f)}{\mathcal{M}(f)} \tag{8.32}$$

$$= \frac{\sum_{j=0}^{q} b_j(2\pi \iota f)^j}{\sum_{k=0}^{r} a_k(2\pi \iota f)^k}. \tag{8.33}$$

The transfer function can thus be expressed as a ratio of two complex polynomials in f for any system expressible in the form of (8.31). The $q+1$, generally complex, frequencies f_z, for which the numerator of (8.33) is zero, are referred to as **zeros** of the transfer function (8.33). The predicted data for the forward problem will thus be zero for inputs of the form $e^{\iota 2\pi f_z t}$, regardless of their amplitude. Any real-valued frequency, $f_z = f_0$, corresponding to the Fourier model basis function $e^{\iota 2\pi f_0 t}$ will thus lie in the model null space and be unrecoverable by any inverse methodology. The $r+1$, generally complex, frequencies f_p for which the denominator of (8.33) is zero, are called **poles** of (8.33). The system will be unstable when excited by model basis functions $e^{\iota 2\pi f_p t}$. Along with a scalar gain factor, the transfer function (and hence the response) of a general linear system that is expressible in the form of (8.31) can be completely characterized by tabulating the poles and zeros, along with a scalar gain factor. This characterization is commonly employed in specifying instrument responses.

8.2. LINEAR SYSTEMS IN DISCRETE TIME

To implement Fourier methods numerically, a discrete theory is required with properties that are analogous to the continuous transforms (8.16) and (8.18). This is achieved by the **discrete Fourier transform**, or **DFT**. In its most basic formulation, the DFT operates on a uniformly sampled (e.g., space or time) sequence with a specified number of terms n. The frequency, f_s, at which sampling occurs is called the **sampling rate**. The forward discrete Fourier transform of an n-point sequence, m_j, $j = 0, 1, \ldots, n-1$, is

$$\mathcal{M}_k = (\text{DFT}[\mathbf{m}])_k \tag{8.34}$$

$$= \sum_{j=0}^{n-1} m_j e^{-\iota 2\pi jk/n}, \tag{8.35}$$

and its inverse is

$$m_j = \left(\text{DFT}^{-1}[\mathcal{M}]\right)_j \tag{8.36}$$

$$= \frac{1}{n} \sum_{k=0}^{n-1} \mathcal{M}_k e^{\imath 2\pi jk/n}. \tag{8.37}$$

Equation (8.37) states that the sequence m_j can be expressed as a linear combination of the n basis functions $e^{\imath 2\pi jk/n}$, where the complex weights in the linear combination are the discrete spectral elements $\mathcal{M}_k$ (with an additional scaling factor of $1/n$ in (8.37)). The DFT basis functions are orthonormal on an n-point interval in that, for any integers l and k:

$$\frac{1}{n} \sum_{j=0}^{n-1} e^{\imath 2\pi jl} e^{-\imath 2\pi jk/n} = \begin{cases} 1 & l = k \\ 0 & l \neq k. \end{cases} \tag{8.38}$$

The DFT operations (8.35) and (8.37) are also widely referred to as the **FFT** and **IFFT** because a particularly efficient algorithm, the **fast Fourier transform**, is nearly ubiquitously used for their evaluation. The transforms can be calculated in MATLAB using the **fft** and **ifft** commands (note that FFT vectors and arrays in MATLAB, as is standard, are indexed beginning with one instead of zero). The DFT expression of Parseval's theorem (8.23) is

$$\sum_{j=0}^{n-1} |m_j|^2 = \frac{1}{n} \sum_{k=0}^{n-1} |\mathcal{M}_k|^2. \tag{8.39}$$

DFT spectra, $\mathcal{M}_k$, are generally complex, discrete, and periodic, where the period is n. There is an implicit assumption in DFT theory that the associated time domain sequence, m_j, is also periodic with period n. Because of these periodicities, DFT results can be stored in complex vectors of length n without loss of information, although (8.35) and (8.37) are valid for any integer index k. The DFT of a real–valued sequence has **Hermitian symmetry** about $k = 0$ and $k = n/2$, as defined by $\mathcal{M}_k = \mathcal{M}_{n-k}^*$ (see Exercise 8.2).

The mapping of the discrete spectrum index to specific frequencies is proportional to the sampling rate f_s. For n even, the positive frequencies, lf_s/n, where $l = 1, \ldots, n/2 - 1$, correspond to indices $k = 1, \ldots, n/2 - 1$, and the negative frequencies, $-lf_s/n$, correspond to indices $k = n/2 + 1, \ldots, n - 1$. The frequencies $\pm f_s/2$ have identical DFT values and correspond to index $k = n/2$. For n odd, there is no integer k corresponding to exactly half of the sampling rate. In this case, positive frequencies correspond to indices 1 through $(n-1)/2$ and negative frequencies correspond to indices $(n+1)/2$ through $n - 1$. Figure 8.1 displays the discrete spectrum index-frequency mapping with respect to k for an $n = 16$-length DFT.
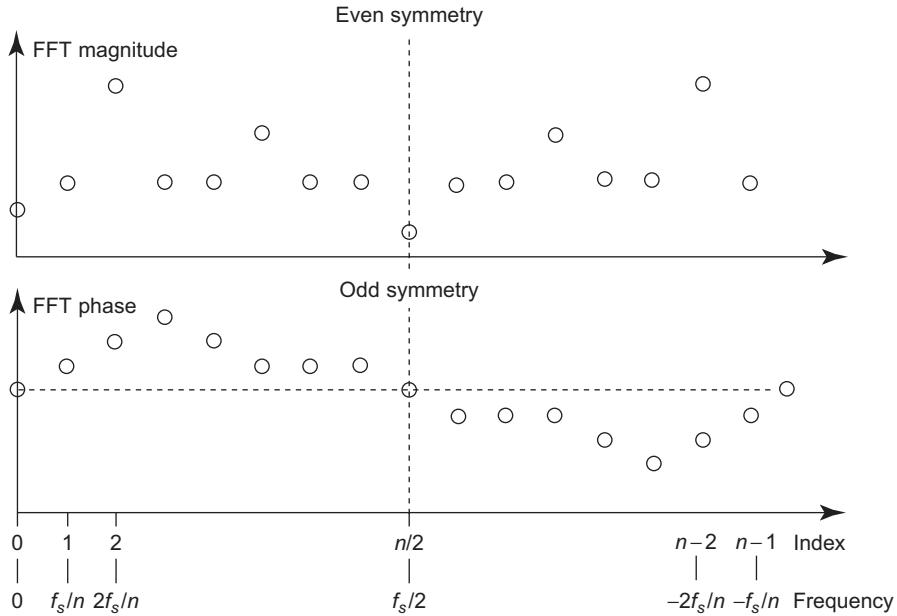
Figure 8.1 Frequency and index mapping for the DFT of a real-valued sequence ($n = 16$) sampled at f_s. For DFT theory to accurately represent Fourier operations on a continuous time signal, f_s must be greater than or equal to the Nyquist frequency (8.40).

The Hermitian symmetry of the DFT implies that, for a real–valued sequence, the spectral amplitude, $|\mathcal{M}|$, is symmetric and the spectral phase is antisymmetric with respect to $k = 0$ and $k = n/2$. See Figure 8.1. For this reason it is customary to just plot the spectral amplitude and phase for positive frequencies in depicting the spectrum of a real signal.

For a uniformly sampled sequence to accurately represent a continuous function that contains nonnegligible spectral energy up to some maximum frequency $f_{\max}$, the sampling rate, f_s, must be at least as large as the **Nyquist frequency**, f_N, so that

$$f_s \geq f_N = 2f_{\max}. \tag{8.40}$$

Should (8.40) not be met, a nonlinear and generally irreversible distortion called **aliasing** occurs. Generally speaking, aliasing causes spectral energy at frequencies $f > f_s/2$ to be "folded" and superimposed onto the DFT spectrum within the frequency range $-f_s/2 \leq f \leq f_s/2$.

Consider a model sequence m_j of length n and an impulse response sequence g_j of length p, where both sequences are synchronously and uniformly sampled at $f_s = 1/\Delta t$. The discrete convolution of the two sequences can be performed in two ways.

The first method is a **serial** convolution, which produces a sequence of length $n + p - 1$:

$$d_j = \sum_{k=0}^{n-1} m_k g_{j-k} \, \Delta t \quad j = 0, 1, \ldots, n+p-2. \tag{8.41}$$

Serial convolution implements the discrete approximation to the convolution integral, and is implemented in MATLAB by the **conv** command.

The second type of discrete convolution is a **circular convolution**. Circular convolution is applicable to two sequences of equal length. If the lengths initially differ, they may be equalized by padding the shorter of the two sequences with zeros. The result of a circular convolution is as if each sequence is first expanded to be periodic (with period n), and then serially convolved solely on the index interval $j = 0, 1, \ldots, n-1$. A circular convolution can be implemented using the discrete counterpart of the convolution theorem,

$$\begin{aligned} \mathbf{d} &= \mathrm{DFT}^{-1}[\mathrm{DFT}[\mathbf{m}] \odot \mathrm{DFT}[\mathbf{g}]]\Delta t \\ &= \mathrm{DFT}^{-1}[\mathcal{M} \odot \mathcal{G}]\Delta t, \end{aligned} \tag{8.42}$$

where $\mathcal{M} \odot \mathcal{G}$ indicates element-by-element multiplication of the vectors $\mathcal{M}$ and $\mathcal{G}$.

A very important practical consideration is that circular convolution implemented using the discrete convolution theorem (8.42) is much more efficient than serial convolution invoked with simple summation as in (8.41). To avoid **wrap–around** effects arising due to the implied n-length periodicity of $\mathbf{m}$ and $\mathbf{g}$ in the circular convolution, and thus obtain a result that is indistinguishable from the serial convolution (8.41), it may be necessary to pad one or both series with up to n zeros and apply (8.42) on the extended sequences. Because of the factoring strategy used in the FFT algorithm, it is also desirable for computational efficiency to pad $\mathbf{m}$ and $\mathbf{g}$ to lengths that are powers of 2, or are at least have many small prime factors.

Consider the case where we have a theoretically known, or accurately estimated, system impulse response, $g(t)$, convolved with an unknown model, $m(t)$. The continuous forward problem is

$$d(t) = \int_a^b g(t - \tau)m(\tau) \, d\tau. \tag{8.43}$$

Uniformly discretizing this expression using simple collocation with a sampling rate, $f_s = 1/\Delta t$, that is rapid enough to satisfy (8.40) and thus avoid aliasing, gives

$$\mathbf{d} = \mathbf{Gm}, \tag{8.44}$$

where $\mathbf{d}$ and $\mathbf{m}$ are m and n length sequences, respectively, and $\mathbf{G}$ is a matrix with m rows of length n. Each row of $\mathbf{G}$ is a time-reversed and time–shifted representation of the impulse response, scaled by Δt to approximate the convolution integral,

$$G_{j,k} = g_{j-k}\Delta t. \tag{8.45}$$

This time domain representation of a forward problem convolution was previously examined in Example 3.2.

An inverse solution using Fourier methods can be obtained by first padding $\mathbf{d}$ and $\mathbf{g}$ appropriately with zeros so that they are of some equal and sufficient length, n, to render moot any wrap-around artifacts associated with circular convolution. $\mathbf{G}$ then becomes an n by n matrix. Applying the DFT and (8.26) allows us to cast the forward problem as a complex-valued linear system,

$$\mathcal{D} = \mathbf{G}\mathcal{M}. \tag{8.46}$$

$\mathbf{G}$ in (8.46) is a complex-valued diagonal matrix with

$$G_{k,k} = \mathcal{G}_k, \tag{8.47}$$

where $\mathcal{G}$ is the discrete Fourier transform of the sampled impulse response $\mathbf{g}$, $\mathcal{D}$ is the discrete Fourier transform of the data vector $\mathbf{d}$, and $\mathcal{M}$ is the discrete Fourier transform of the model vector $\mathbf{m}$. We can write (8.46) more simply using the element-by-element multiplication operator as

$$\mathcal{D} = \mathcal{G} \odot \mathcal{M}. \tag{8.48}$$

Equation (8.46) suggests a solution by **spectral division**, where we first solve for the Fourier transform of the model using the element-by-element division operation (or equivalently, inverting the diagonal matrix $\mathbf{G}$ in (8.46)),

$$\mathcal{M} = \mathcal{D} \oslash \mathcal{G}, \tag{8.49}$$

and then obtaining the model $\mathbf{m} = \mathrm{DFT}^{-1}[\mathcal{M}]$.

Equation (8.49) is appealing in its simplicity and efficiency. The application of (8.26), combined with the efficient FFT implementation of the DFT, reduces the necessary computational effort from solving a potentially very large linear system of time domain equations (8.44) to just three n-length DFT operations (taking the DFT of the data and impulse response vectors, and then the inverse DFT of the element–by-element quotient). If $\mathbf{d}$ and $\mathbf{g}$ are real, packing/unpacking algorithms also exist that allow DFT operations to be further reduced to complex vectors of length $n/2$.

However, (8.49) does not avoid the instability that is potentially associated with deconvolution if the reciprocal of any element in $\mathcal{G}$ is large. Equation (8.49) typically requires regularization to be useful.

8.3. WATER LEVEL REGULARIZATION

A straightforward and widely applied method of regularizing spectral division is **water level regularization**. The water level strategy employs a modified impulse response spectrum, $\mathcal{G}_w$, in (8.49), where

$$(\mathcal{G}_w)_i = \begin{cases} \mathcal{G}_i & (|\mathcal{G}_i| > w) \\ w(\mathcal{G}_i/|\mathcal{G}_i|) & (0 < |\mathcal{G}_i| \leq w) \\ w & (\mathcal{G}_i = 0). \end{cases} \tag{8.50}$$

The water level regularized model estimate is then

$$\mathbf{m}_w = \mathrm{DFT}^{-1}\left[\mathcal{D} \oslash \mathcal{G}_w\right]\Delta t. \tag{8.51}$$

The colorful name for this technique arises from the construction of $\mathcal{G}_w$, applying the analogy of pouring water into the low-amplitude "holes" of $\mathcal{G}$ until the spectral amplitude levels there reach w. The effect in (8.51) is to prevent undesirable noise amplification from occurring at frequencies where $\|\mathcal{G}\|$ is small.

An optimal water level value, w, will reduce the sensitivity to noise in the inverse solution while still recovering important model features. As is typical of the regularization process, it is possible to choose a "best" solution by assessing the trade-off between the norm of the residuals (fitting the data) and the model norm (smoothness of the model) as the regularization parameter w is varied. In calculating a trade-off curve, Parseval's theorem (8.39) usefully facilitates calculations of the model and residual norms from spectra without calculating inverse Fourier transforms. Note that the 2-norm of the water level–regularized solution, $\mathbf{m}_w$, will be nonincreasing with increasing w because $|(\mathcal{G}_w)_i| \geq |\mathcal{G}_i|$.

Example 8.1

In Example 3.2, we investigated time domain deconvolution for uniformly sampled data with a sampling rate of $f_s = 2$ Hz using the truncated single value decomposition. Here, we solve this problem using frequency-domain deconvolution regularized via the water level technique. The impulse response, true model, and noisy data for this example are plotted in Figures 3.9, 3.11, and 3.12, respectively. We first pad the 210-point data and impulse response vectors with 210 additional zeros to eliminate wrap-around artifacts, and apply the fast Fourier transform to both vectors to obtain corresponding discrete spectra. The spectral amplitudes of the impulse response, data, and noise are critical in assessing the stability of the spectral division solution. See Figure 8.2. The frequencies range from 0 to $f_s/2 = 1$ Hz. Because spectral amplitudes for real-valued sequences are symmetric about $k = 0$ and $k = n/2$ (Figure 8.1), only positive frequencies are shown.

Figure 8.2 Amplitude spectra for the system impulse response (solid, smooth curve), noise-free data (dashed curve), and noisy data (solid, rough curve).

Examining the impulse response spectral amplitude, $|\mathcal{G}_k|$, in Figure 8.2, we note that it decreases by approximately three orders of magnitude between very low frequencies and half of the sampling frequency ($f_s/2 = 1$ Hz). The convolution theorem (8.26) shows that the forward problem convolution multiplies the spectrum of the model by $\mathcal{G}(f)$ in mapping it to the data. Thus, the convolution of a general signal with broad frequency content with this impulse response will strongly attenuate higher frequencies. Figure 8.2 also shows that the spectral amplitudes of the noise-free data fall off more quickly than the impulse response. This indicates that spectral division will be a stable process for noise-free data in this problem. Figure 8.2 also shows that the spectral amplitudes of the noisy data dominate the signal at frequencies higher than $f \approx 0.1$ Hz. Because of the small values of $\mathcal{G}_k$ at these frequencies, the spectral division solution using the noisy data will be dominated by noise (as was the case in the time domain solution of Example 3.2; see Figure 3.14). Figure 8.3 shows the amplitude spectrum of the noisy data (Figure 3.14) divided by the spectrum of the impulse response. The resulting model spectrum is dominated by noise at frequencies above about 0.1 Hz.

To regularize the spectral division solution, an optimal water level, w, is sought, where w should be large enough to avoid the undesirable amplification of noise. Figures 8.2 and 8.3 suggest that the optimal value of w is near the crossover spectral amplitude where the data spectrum is surpassed by noise, or near w somewhat greater than 1. However, such a determination might be more difficult for data with a more
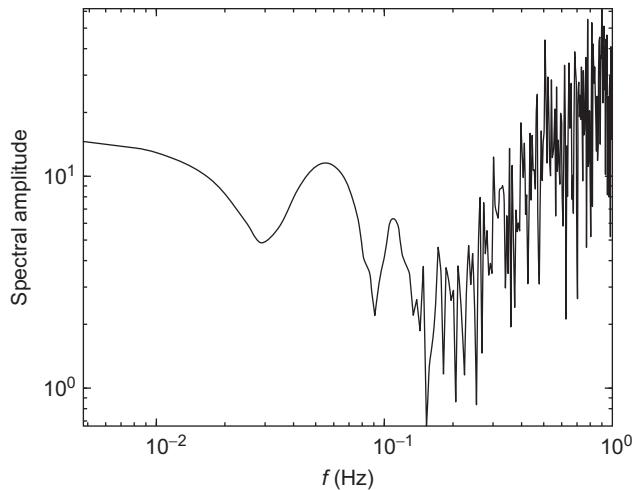
Figure 8.3 Spectral amplitudes resulting from the Fourier transform of the noisy data divided by the Fourier transform of the impulse response (the transfer function).
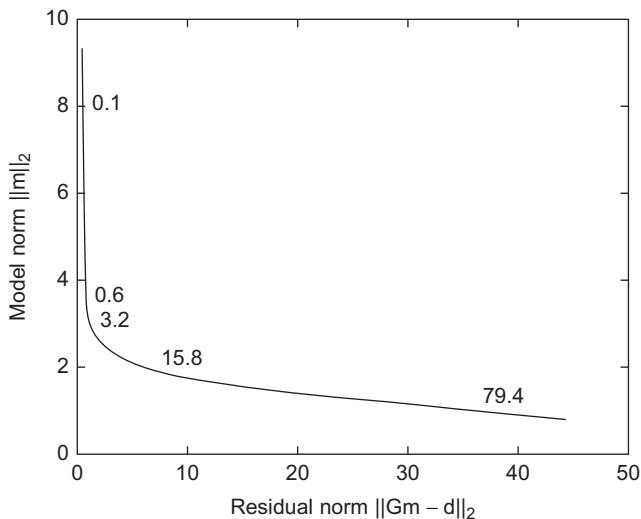
Figure 8.4 L-curve for a logarithmically distributed range of water level values as indicated.

complex spectrum, and/or where the distinction between signal and noise spectra is generally less clear. Figure 8.4 shows an L-curve for this example, which suggests an optimal w close to 3. Figure 8.5 shows a corresponding range of solutions, and Figure 8.6 shows the solution for $w = 3.16$.

Figure 8.5 Models corresponding to the range of water level values used to construct Figure 8.4. Dashed curves show the true model, and bold trace shows the L-curve determined model with $w = 3.16$ (Figure 8.4).

Figure 8.6 Water level–regularized model corresponding to $w = 3.16$. Dashed curve shows the true model.

The solution shown in Figure 8.6, chosen from the corner of the trade-off curve of Figure 8.4, shows features of limited resolution that are typical of regularized solutions. In this case, imperfect resolution induced by regularization is manifested by reduced amplitude, oscillatory side lobes, and model broadening relative to the true model.

8.4. TIKHONOV REGULARIZATION IN THE FREQUENCY DOMAIN

An alternative regularized approach to solving the frequency-domain forward problem
(8.46) is to consider the system as a least squares minimization problem, as we did in
Chapter 4 (e.g., (4.4)):

$$\min \|\mathbf{G}\mathcal{M} - \mathcal{D}\|_2^2 + \alpha^2 \|\mathcal{M}\|_2^2. \tag{8.52}$$

Expressing 8.52 as an augmented forward problem gives

$$\min \left\| \begin{bmatrix} \mathbf{G} \\ \alpha\mathbf{I} \end{bmatrix} \mathcal{M} - \begin{bmatrix} \mathcal{D} \\ \mathbf{0} \end{bmatrix} \right\|_2^2. \tag{8.53}$$

We solve (8.53) by applying the normal equations using the conjugate transpose of the
diagonal matrix $\mathbf{G}$, $\mathbf{G}^H$, which gives

$$\begin{bmatrix} \mathbf{G}^H & \alpha\mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{G} \\ \alpha\mathbf{I} \end{bmatrix} \mathcal{M} = \begin{bmatrix} \mathbf{G}^H & \alpha\mathbf{I} \end{bmatrix} \begin{bmatrix} \mathcal{D} \\ \mathbf{0} \end{bmatrix}. \tag{8.54}$$

Equation (8.47) corresponds to the zeroth-order Tikhonov-regularized formulation

$$(\mathbf{G}^H\mathbf{G} + \alpha^2\mathbf{I})\mathcal{M} = \mathbf{G}^H\mathcal{D}. \tag{8.55}$$

with the frequency-domain solution,

$$\mathcal{M}_\alpha = (\mathbf{G}^H\mathbf{G} + \alpha^2\mathbf{I})^{-1}\mathbf{G}^H\mathcal{D}. \tag{8.56}$$

Because $\mathbf{G}^H\mathbf{G}$ is diagonal, (8.56) can be expressed solely using n-length vector element-
by-element operations as

$$\mathcal{M}_\alpha = (\mathcal{G}^H \odot \mathcal{D}) \oslash (|\mathcal{G}|_2^2 + \alpha^2\mathbf{e}), \tag{8.57}$$

where $\mathbf{e}$ is an n-length vector of ones. Applying the IDFT to $\mathcal{M}_\alpha$ then produces the
corresponding regularized time domain solution, $\mathbf{m}$.

Although we have implemented $\|\mathcal{M}\|_2 = \|\mathrm{DFT}[\mathbf{m}]\|_2$ rather than $\|\mathbf{m}\|_2$ in (8.53) to
regularize the problem, by Parseval's theorem (8.39), the norm of the model spectrum
is proportional to the norm of the time domain model. Thus, (8.57) is equivalent to
that obtained if we were to regularize using the norm of the model in the time domain
model (although the specific values of α will be different).

An important consideration is that the inverse operation (8.57) is applied by element-
by-element spectral division of n-length vectors, so the solution can be calculated

extremely efficiently relative to inverting a nondiagonal n by n matrix. Even with the additional calculations associated with the FFT, this methodology is typically much more efficient than a time domain approach, such as is described in Example 3.2.

To implement higher-order Tikhonov regularization, we note that

$$
\begin{aligned}
\frac{d}{dt}m(t) &= \frac{d}{dt}\int_{-\infty}^{\infty}\mathcal{M}(f)e^{i2\pi ft}\,df \\
&= \int_{-\infty}^{\infty}\frac{\partial}{\partial t}[\mathcal{M}(f)e^{i2\pi ft}]\,df \\
&= \int_{-\infty}^{\infty}2\pi if\mathcal{M}(f)e^{i2\pi ft}\,df \\
&= \mathcal{F}^{-1}[2\pi if\mathcal{M}(f)].
\end{aligned}
\tag{8.58}
$$

Taking the Fourier transform of both sides gives

$$
\mathcal{F}\left[\frac{d}{dt}m(t)\right] = 2\pi if\mathcal{M}(f).
\tag{8.59}
$$

Equation (8.59) enables us to effect a first derivative seminorm by multiplying each element of $\mathcal{M}$ by $2\pi f_j$, choosing the f_j to be proportional to the spectral frequency of the jth-element (Figure 8.1). Proportionality is sufficient, because the exact constant frequency-scaling factor for converting the DFT index to frequency can be absorbed into the range of the regularization parameter α. Thus, we can effect pth-order Tikhonov regularization by solving

$$
\min\left\|\begin{bmatrix}\mathbf{G}\\\alpha\mathbf{K}^p\end{bmatrix}\mathcal{M}-\begin{bmatrix}\mathcal{D}\\\mathbf{0}\end{bmatrix}\right\|_2^2,
\tag{8.60}
$$

where $\mathbf{K}$ is an n by n diagonal matrix with diagonal elements (e.g., for n even)

$$
K_{j,j}=\begin{cases}(j-1)/n & j=1,\,2,\,\ldots,\,n/2\\(j-1)/n-1 & j=n/2+1,\,n/2+2,\,\ldots,\,n\end{cases}
\tag{8.61}
$$

that are proportional to the frequency represented by the jth element of $\mathcal{M}$. The least squares solution, obtained utilizing the normal equations, is thus

$$
\mathcal{M}_\alpha = (\mathbf{G}^H\mathbf{G}+\alpha^2\mathbf{K}^{2p})^{-1}\mathbf{G}^H\mathcal{D}
\tag{8.62}
$$

or

$$\mathcal{M}_\alpha = (\mathcal{G}^H \odot \mathcal{D}) \oslash (|\mathcal{G}|_2^2 + \alpha^2 \mathbf{k}^{2p}), \tag{8.63}$$

where $\mathbf{k}^{2p}$ is the vector composed of the diagonal elements of $\mathbf{K}^{2p}$.

As with the zeroth-order solution (8.57), (8.63) can be evaluated very efficiently with element–by–element n-length vector operations. Note that the implementation of the (nonidentity) matrix $\mathbf{K}$ in (8.62) weights the higher frequency components of the model more in the regularization, and thus tends to make them small. It is thus straightforward to consider and implement more general regularization matrices that will penalize particular Fourier components (e.g., within a particular frequency range).

Example 8.2

Let us reconsider Example 8.1 in a Tikhonov regularization framework.

We first implement zeroth-order regularization using (8.56) and examine the trade-off curve (Figure 8.7). The suite of solutions is shown in Figure 8.8, and a solution selected from the trade-off curve is shown in Figure 8.9 compared with the true model. Note that, compared to the water level solution shown in Figure 8.6, the Tikhonov regularized solution has better amplitude recovery and is somewhat smoother. On the
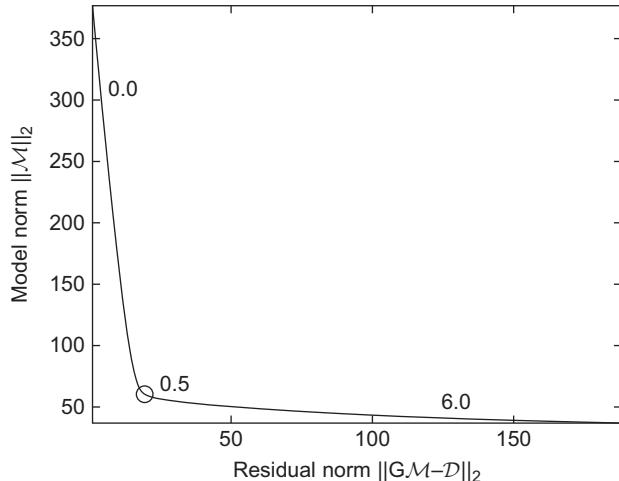
Figure 8.7 L-curve for a logarithmically distributed range of regularization-parameter values, zeroth-order Tikhonov regularization.
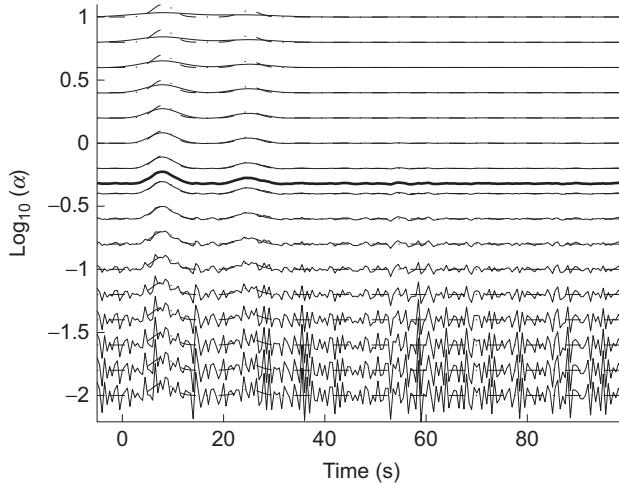
Figure 8.8 Zeroth-order Tikhonov regularized models corresponding to the range of regularization parameters used to construct Figure 8.7. Dashed curves show the true model, and bold trace shows the L-curve determined model with $\alpha = 0.48$.
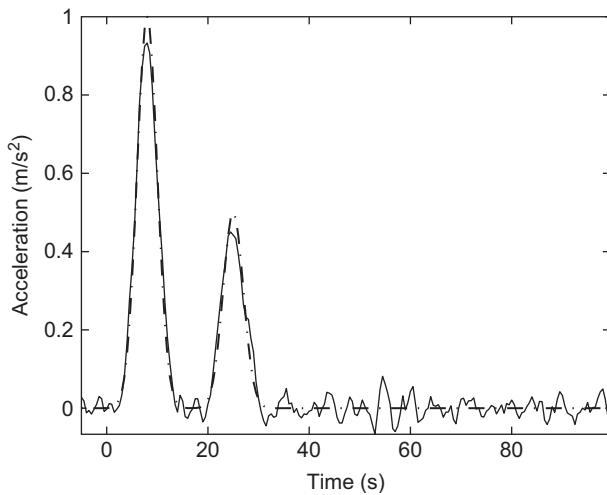
Figure 8.9 Zeroth-order Tikhonov regularized model corresponding to $\alpha = 0.48$. Dashed curve shows the true model.

other hand, the recovered zeroth–order Tikhonov regularized model shows higher amplitude structure later in the time series. Applying second–order Tikhonov regularization, we obtain a corresponding trade–off curve (Figure 8.10) and recover a still
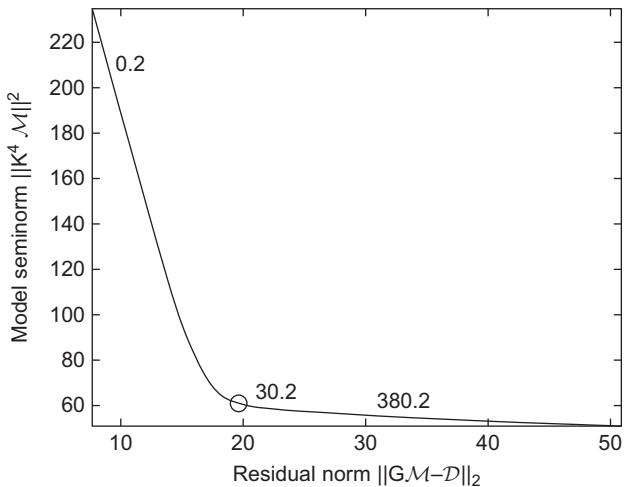
Figure 8.10 L-curve for a logarithmically distributed range of regularization-parameter values, second-order Tikhonov regularization.
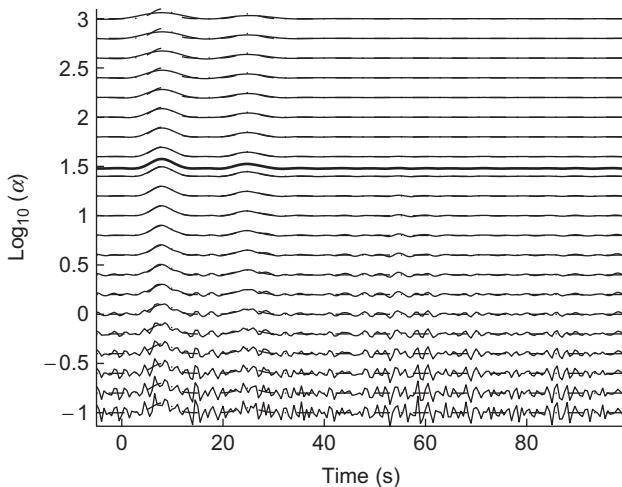
Figure 8.11 Second-order Tikhonov regularized models corresponding to the range of regularization parameters used to construct Figure 8.7. Dashed curves show the true model, and bold trace shows the L-curve determined model with $\alpha = 30.2$ (Figure 8.10).

smoother model that is generally closer to the true model than either water level or zeroth–order Tikhonov regularization (Figures 8.11 and 8.12). The final accuracy of any regularized solution will, of course, depend on properties of the true solution. In this

Figure 8.12 Second-order Tikhonov regularized model corresponding to $\alpha = 30.2$. Dashed curve shows the true model.

particular case the true model is smooth, and the second-order regularization solution is closest.

A significant new idea introduced by the Fourier methodology is that it provides a set of orthonormal model and data basis functions of the form of (8.27), the complex exponentials, that have the property of passing through a linear system altered in phase and amplitude, but not in frequency or functional character (8.30). This remarkable fact is the essence of the convolution theorem (8.26), which leads to frequency-domain inverse methodologies that are very efficient when coupled with the FFT algorithm. This efficiency can become critically important when larger and/or higher-dimensional models are of interest, a large number of deconvolutions must be performed, or computational speed is critical, such as in real-time applications. The spectrum of the impulse response (such as in Figures 8.2 and 8.3) can be used to understand what frequency components may exhibit instability in an inverse solution. The information contained in the spectrum of Figure 8.2 is thus analogous to that obtained with a Picard plot in the context of the SVD (Chapters 3 and 4). The Fourier perspective also provides a link between linear inverse theory and the (vast) field of linear filtering. The deconvolution problem in this context is identical to finding an optimal inverse filter to recover the model while suppressing the influence of noise, and Tikhonov regularization in the frequency domain applies a preferential filtering to the solution that reduces amplitudes of high-frequency Fourier components.

8.5. EXERCISES

1. Given that the Fourier transform of a real-valued linear system, $g(t)$,

$$\mathcal{F}[g(t)] = \mathcal{G}(f) = \text{real}(\mathcal{G}(f)) + \text{imag}(\mathcal{G}(f)) = \alpha(f) + \iota\beta(f), \qquad (8.64)$$

   is Hermitian,

$$\mathcal{G}(f) = \mathcal{G}^*(-f), \qquad (8.65)$$

   show that convolving $g(t)$ with $\sin(2\pi f_0 t)$ and $\cos(2\pi f_0 t)$ produces the scaled and phase-shifted sinusoids,

$$g(t) * \sin(2\pi f_0 t) = |\mathcal{G}(f_0)| \, \sin(2\pi f_0 t + \theta(f_0)) \qquad (8.66)$$

$$g(t) * \cos(2\pi f_0 t) = |\mathcal{G}(f_0)| \, \cos(2\pi f_0 t + \theta(f_0)), \qquad (8.67)$$

   where the scale factor is the spectral amplitude,

$$|\mathcal{G}(f_0)| = (\alpha^2(f_0) + \beta^2(f_0))^{\frac{1}{2}}, \qquad (8.68)$$

   and the phase-shift factor is the spectral phase

$$\theta(f_0) = \tan^{-1}\left(\frac{\beta(f_0)}{\alpha(f_0)}\right). \qquad (8.69)$$

2. **(a)** Demonstrate using (8.35) that the DFT of an n-point, real-valued sequence, **x**, is Hermitian, i.e.,

$$\mathcal{X}_{n-k} = \mathcal{X}_k^* \qquad (8.70)$$

   (b) Demonstrate that the Hermitian symmetry shown in part (a) implies that the N independent elements in a time series **x** produce $N/2 + 1$ independent elements (N even) or $(N-1)/2 + 1$ independent elements (N odd) in the DFT $\mathcal{X}$. As the DFT has an inverse (8.37) that reproduces **x** from $\mathcal{X}$, clearly information has not been lost in taking the DFT, yet the number of independent elements in **x** and $\mathcal{X}$ differ. Explain.

3. A linear damped vertical harmonic oscillator consisting of a mass suspended on a lossy spring is affixed to the surface of a terrestrial planet to function as a seismometer, where the recorded displacement, $z(t)$, of the mass relative to its equilibrium position will depend on ground motion (note that when the surface of the planet moves upward, the inertia of the mass will tend to make it remain at rest, and the corresponding motion of the mass relative to its suspension system will be downward). For an upward ground displacement, $u(t)$, the system can be mathematically modeled as the linear differential equation (8.31) as

$$\frac{d^2z}{dt^2} + \frac{D}{M}\frac{dz}{dt} + \frac{K}{M}z = \frac{d^2u}{dt^2}, \qquad (8.71)$$

where the physical properties of the oscillator are defined by the mass M, the displacement-proportional spring force constant K, and the velocity-proportional damping force constant D.

(a) By taking the Fourier transform of (8.71), obtain the transfer function $\mathcal{G}(f) = \mathcal{Z}(f)/\mathcal{U}(f)$, where $\mathcal{Z}(f)$ and $\mathcal{U}(f)$ are the Fourier transforms of $z(t)$ and $u(t)$, respectively.

In terms of M, K, and D, and $f_s = \sqrt{K/M}$:

(b) For what general frequency range of ground motion will the response of this instrument be difficult to remove via a deconvolution?

(c) For what general frequency range of ground motion will the output of this instrument be nearly identical to the true ground motion?

4. A displacement seismogram is observed from a large earthquake at a far-field seismic station, from which the source region can be approximated as a point. A much smaller aftershock from the main shock region is used as an empirical Green's function for this event. It is supposed that the observed signal from the large event should be approximately equal to the convolution of the main shock's rupture history with this empirical Green's function. The 256-point seismogram is in the file **seis.mat**. The impulse response of the seismometer is in the file **impresp.mat**.

(a) Deconvolve the impulse response from the observed main shock seismogram using frequency-domain Tikhonov zeroth-order deconvolution to solve for the source time function of the large earthquake. Note that the source time function is expected to consist of a nonnegative pulse or set of pulses. Estimate the source duration in samples and assess any evidence for subevents and their relative durations and amplitudes. Approximately what water level do you believe is best for this data set? Why?

(b) Perform second-order ($p = 2$), frequency-domain Tikhonov deconvolution to solve this problem. See Exercise 8.4(b).

(c) Recast the problem as a discrete linear inverse problem, as described in the example for Chapter 3, and solve the system using second-order Tikhonov regularization.

(d) Are the results in (c) better or worse than in (a) or (b)? How and why? Compare the amount of time necessary to find the solution in each case on your computing platform.

8.6. NOTES AND FURTHER READING

Although we examine one-dimensional convolution and deconvolution problems here for conceptual simplicity, note that these results are readily generalizable to higher dimensions, and higher-dimensional formulations of the DFT are widely utilized.

MATLAB has a general n-dimensional set of FFT algorithms, **fftn** and **ifftn**. Many regularization matrices (e.g., the **L** roughening matrices introduced for one-dimensional Tikhonov regularization in this text (4.27) and (4.28)) are easily recognized as convolutions, where the rows are time-reversed sampled representations of the impulse response of the convolution effected by **Lm**. Fourier analysis of the frequency response of regularization matrices thus facilitates more sophisticated regularization matrix design.

In some physics and geophysics treatments, the sign convention chosen for the complex exponentials in the Fourier transform and its inverse may be reversed, so that the forward transform (8.16) has a plus sign in the exponent and the inverse transform (8.18) has a minus sign in the exponent. This alternative sign convention merely induces a complex conjugation in the spectrum that is reversed when the corresponding inverse transform is applied. An additional convention issue arises as to whether to express frequency in Hz (f) or radians per second ($\omega = 2\pi f$). Alternative Fourier transform formulations using ω differ from (8.16) and (8.18) by a simple change of variables, and introduce scaling factors of 2π in the forward, reverse, or both transforms.

Gubbins [60] also explores connections between Fourier and inverse theory in a geophysical context. Kak and Slaney [84] give an extensive treatment of Fourier-based methods for tomographic imaging. Vogel [169] discusses Fourier methods for image deblurring. Because of the tremendous utility of Fourier techniques, there are numerous resources on their use in the physical sciences, engineering, and pure mathematics. A basic treatment covering theory and some applications at the approximate level of this text is [20], and a recommended advanced text on the topic is [131].

Nonlinear Regression

Synopsis

Common approaches to solving nonlinear regression problems are introduced, extending the development of linear regression in Chapter 2. We begin with a discussion of Newton's method, which provides a general framework for solving nonlinear systems of equations and nonlinear optimization problems. Then we discuss the Gauss-Newton (GN) and Levenberg-Marquardt (LM) methods, which are versions of Newton's method specialized for nonlinear regression problems. The distinction between LM and Tikhonov regularization is also made. Statistical aspects and implementation issues are addressed, and examples of nonlinear regression are presented.

9.1. INTRODUCTION TO NONLINEAR REGRESSION

In previous chapters we have concentrated on linear forward and inverse problems, and have seen that such problems are uniformly approachable using a variety of solution methods. We will next consider problems that are nonlinear, i.e., forward and inverse problems that do not obey the rules of superposition (1.5) and scaling (1.6). There is no general theory for the solution of nonlinear parameter estimation and inverse problems. However, we will see that iterative strategies incorporating linear concepts can be applied to solve them in many circumstances.

9.2. NEWTON'S METHOD FOR SOLVING NONLINEAR EQUATIONS

Consider a nonlinear system of m equations in m unknowns

$$\mathbf{F}(\mathbf{x}) = \mathbf{0}. \tag{9.1}$$

We will construct a sequence of vectors, $\mathbf{x}^0$, $\mathbf{x}^1$, . . . , that will converge to a solution $\mathbf{x}^*$. If the nonlinear vector function $\mathbf{F}$ is continuously differentiable, we can construct a Taylor series approximation about some starting solution estimate, $\mathbf{x}^0$,

$$\mathbf{F}\big(\mathbf{x}^0 + \Delta\mathbf{x}\big) \approx \mathbf{F}\big(\mathbf{x}^0\big) + \mathbf{J}\big(\mathbf{x}^0\big)\Delta\mathbf{x}, \tag{9.2}$$

where $\mathbf{J}(\mathbf{x}^0)$ is the Jacobian,

$$\mathbf{J}(\mathbf{x}^0) = \begin{bmatrix} \frac{\partial F_1(\mathbf{x})}{\partial x_1} & \cdots & \frac{\partial F_1(\mathbf{x})}{\partial x_m} \\ \vdots & \ddots & \vdots \\ \frac{\partial F_m(\mathbf{x})}{\partial x_1} & \cdots & \frac{\partial F_m(\mathbf{x})}{\partial x_m} \end{bmatrix}\Bigg|_{\mathbf{x}=\mathbf{x}^0} . \tag{9.3}$$

Using (9.2), and expressing the difference between the desired solution, $\mathbf{x}^*$, and $\mathbf{x}^0$ as

$$\Delta\mathbf{x} = \mathbf{x}^* - \mathbf{x}^0 \tag{9.4}$$

gives

$$\mathbf{F}(\mathbf{x}^*) = \mathbf{0} \approx \mathbf{F}(\mathbf{x}^0) + \mathbf{J}(\mathbf{x}^0)\Delta\mathbf{x}, \tag{9.5}$$

which produces a linear system of equations,

$$\mathbf{J}(\mathbf{x}^0)\Delta\mathbf{x} \approx -\mathbf{F}(\mathbf{x}^0), \tag{9.6}$$

that can be solved for $\Delta\mathbf{x}$ to improve the solution estimate.

Algorithm 9.1   Newton's Method

Given a system of equations $\mathbf{F}(\mathbf{x}) = \mathbf{0}$ and an initial solution $\mathbf{x}^0$, repeat the following steps to compute a sequence of solutions $\mathbf{x}^k$. Stop if and when the sequence adequately converges to a solution with $\mathbf{F}(\mathbf{x}) = \mathbf{0}$.
1. Calculate the Jacobian $\mathbf{J}(\mathbf{x}^k)$ and $\mathbf{F}(\mathbf{x}^k)$.
2. Solve $\mathbf{J}(\mathbf{x}^k)\Delta\mathbf{x} = -\mathbf{F}(\mathbf{x}^k)$.
3. Let $\mathbf{x}^{k+1} = \mathbf{x}^k + \Delta\mathbf{x}$.
4. Let $k = k+1$.

The theoretical properties of Newton's method are summarized in the following theorem. For a proof, see [39].

Theorem 9.1 *If $\mathbf{x}^0$ is close enough to $\mathbf{x}^*$, $\mathbf{F}(\mathbf{x})$ is continuously differentiable in a neighborhood of $\mathbf{x}^*$, and $\mathbf{J}(\mathbf{x}^*)$ is nonsingular, then Newton's method will converge to $\mathbf{x}^*$. The convergence rate is quadratic in the sense that there is a constant c, such that for large k,*

$$\left\| \mathbf{x}^{k+1} - \mathbf{x}^* \right\|_2 \le c \left\| \mathbf{x}^k - \mathbf{x}^* \right\|_2^2 . \tag{9.7}$$

In practical terms, quadratic convergence means that as we approach $\mathbf{x}^*$, the number of accurate digits in the solution doubles at each iteration. Unfortunately, if the hypotheses in the above theorem are not satisfied, then Newton's method can converge very slowly or even fail altogether.

A simple modification to Newton's method often helps with convergence problems. In the **damped Newton's method**, we use the Newton's method equations at each iteration to compute an iterative correction to the working solution. However, instead of taking the full step $\mathbf{x}^{k+1} = \mathbf{x}^k + \Delta\mathbf{x}$, we perform a **line search** across solutions lying along the line $\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha\,\Delta\mathbf{x}$ for a range of positive α values and take the one with minimum $\|\mathbf{F}(\mathbf{x}^{k+1})\|_2$.

Now suppose that we wish to minimize a scalar-valued function $f(\mathbf{x})$. If we assume that $f(\mathbf{x})$ is twice continuously differentiable, we can construct a Taylor series approximation,

$$f(\mathbf{x}^0 + \Delta\mathbf{x}) \approx f(\mathbf{x}^0) + \nabla f(\mathbf{x}^0)^T \Delta\mathbf{x} + \frac{1}{2}\Delta\mathbf{x}^T \mathbf{H}(f(\mathbf{x}^0))\Delta\mathbf{x}, \tag{9.8}$$

where $\nabla f(\mathbf{x}^0)$ is the gradient

$$\nabla f(\mathbf{x}^0) = \begin{bmatrix} \frac{\partial f(\mathbf{x})}{\partial x_1} \\ \vdots \\ \frac{\partial f(\mathbf{x})}{\partial x_m} \end{bmatrix}\Bigg|_{\mathbf{x}=\mathbf{x}^0} \tag{9.9}$$

and $\mathbf{H}(f(\mathbf{x}^0))$ is the Hessian

$$\mathbf{H}(f(\mathbf{x}^0)) = \begin{bmatrix} \frac{\partial^2 f(\mathbf{x})}{\partial x_1^2} & \cdots & \frac{\partial^2 f(\mathbf{x})}{\partial x_1 \partial x_m} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f(\mathbf{x})}{\partial x_m \partial x_1} & \cdots & \frac{\partial^2 f(\mathbf{x})}{\partial x_m^2} \end{bmatrix}\Bigg|_{\mathbf{x}=\mathbf{x}^0}. \tag{9.10}$$

A necessary condition for $\mathbf{x}^*$ to be a minimum of $f(\mathbf{x})$ is that $\nabla f(\mathbf{x}^*) = \mathbf{0}$. We can approximate the gradient in the vicinity of x^0 by

$$\nabla f(\mathbf{x}^0 + \Delta\mathbf{x}) \approx \nabla f(\mathbf{x}^0) + \mathbf{H}(f(\mathbf{x}^0))\Delta\mathbf{x}. \tag{9.11}$$

Setting the approximate gradient (9.11) equal to zero gives

$$\mathbf{H}(f(\mathbf{x}^0))\Delta\mathbf{x} = -\nabla f(\mathbf{x}^0). \tag{9.12}$$

Solving the linear system of equations (9.12) for successive solution steps leads to **Newton's method for minimizing $f(\mathbf{x})$**.

Since Newton's method for minimizing $f(\mathbf{x})$ is exactly Newton's method for solving a nonlinear system of equations applied to $\mathbf{F} = \nabla f(\mathbf{x}) = \mathbf{0}$, the convergence proof follows immediately from the proof of Theorem 9.1.

Theorem 9.2 *If $f(\mathbf{x})$ is twice continuously differentiable in a neighborhood of a local minimizer $\mathbf{x}^*$, there is a constant λ such that $\|\mathbf{H}(f(\mathbf{x})) - \mathbf{H}(f(\mathbf{y}))\|_2 \le \lambda\|\mathbf{x} - \mathbf{y}\|_2$ for every*

vector **y** *in the neighborhood,* $\mathbf{H}(f(\mathbf{x}^*))$ *is positive definite, and* $\mathbf{x}^0$ *is close enough to* $\mathbf{x}^*$, *then Newton's method will converge quadratically to* $\mathbf{x}^*$.

Algorithm 9.2 Newton's Method for Minimizing $f(\mathbf{x})$

Given a twice continuously differentiable function $f(\mathbf{x})$, and an initial solution $\mathbf{x}^0$, repeat the following steps to compute a sequence of solutions $\mathbf{x}^1$, $\mathbf{x}^2$, ... Stop if and when the sequence adequately converges to a solution with $\nabla f(\mathbf{x}^*) = \mathbf{0}$.
1. Calculate the gradient $\nabla f(\mathbf{x}^k)$ and Hessian $\mathbf{H}(f(\mathbf{x}^k))$.
2. Solve $\mathbf{H}(f(\mathbf{x}^k))\Delta\mathbf{x} = -\nabla f(\mathbf{x}^k)$.
3. Let $\mathbf{x}^{k+1} = \mathbf{x}^k + \Delta\mathbf{x}$.
4. Let $k = k + 1$.

Newton's method for minimizing $f(\mathbf{x})$ can be very efficient, but the method can also fail to converge. As with Newton's method for solving a system of equations, the convergence properties of the algorithm can be improved in practice by modifying the model update step with a line search.

9.3. THE GAUSS-NEWTON AND LEVENBERG-MARQUARDT METHODS FOR SOLVING NONLINEAR LEAST SQUARES PROBLEMS

Newton's method is not directly applicable to most nonlinear regression and inverse problems. We may not have equal numbers of data points and model parameters, there may not be an exact solution to $\mathbf{G}(\mathbf{m}) = \mathbf{d}$, or $\mathbf{G}(\mathbf{m}) = \mathbf{d}$ may have multiple solutions. Here, we will use a specialized version of Newton's method to minimize a nonlinear least squares problem.

Given a nonlinear system of equations $\mathbf{G}(\mathbf{m}) = \mathbf{d}$, consider the problem of finding an n-length parameter vector, $\mathbf{m}$, constrained by an n-length data vector, $\mathbf{d}$, with associated specified data standard deviations. Our goal is to find a set of parameters that best fits the data in the sense of minimizing the 2-norm of the residuals.

As with linear regression, if we assume that the measurement errors are normally distributed, then the maximum likelihood principle leads us to minimizing the sum of squared residuals normalized by their respective standard deviations (2.12). We seek to minimize the weighted residual norm

$$f(\mathbf{m}) = \sum_{i=1}^{m} \left(\frac{G(\mathbf{m})_i - d_i}{\sigma_i} \right)^2. \tag{9.13}$$

We define the scalar-valued functions

$$f_i(\mathbf{m}) = \frac{G(\mathbf{m})_i - d_i}{\sigma_i} \quad i = 1, 2, \ldots, m \tag{9.14}$$

and the vector–valued function

$$\mathbf{F}(\mathbf{m}) = \begin{bmatrix} f_1(\mathbf{m}) \\ \vdots \\ f_m(\mathbf{m}) \end{bmatrix}. \tag{9.15}$$

Thus

$$f(\mathbf{m}) = \sum_{i=1}^{m} f_i(\mathbf{m})^2 = \|\mathbf{F}(\mathbf{m})\|_2^2. \tag{9.16}$$

The gradient of $f(\mathbf{m})$ can be written as the sum of the gradients of the individual terms,

$$\nabla f(\mathbf{m}) = \sum_{i=1}^{m} \nabla\big(f_i(\mathbf{m})^2\big). \tag{9.17}$$

The elements of the gradient of $f(\mathbf{m})$ are

$$(\nabla f(\mathbf{m}))_j = \sum_{i=1}^{m} 2f_i(\mathbf{m})(\nabla f_i(\mathbf{m}))_j, \tag{9.18}$$

and the gradient can thus be written in matrix notation as

$$\nabla f(\mathbf{m}) = 2\mathbf{J}(\mathbf{m})^T \mathbf{F}(\mathbf{m}), \tag{9.19}$$

where $\mathbf{J}(\mathbf{m})$ is the Jacobian

$$\mathbf{J}(\mathbf{m}) = \begin{bmatrix} \frac{\partial f_1(\mathbf{m})}{\partial m_1} & \cdots & \frac{\partial f_1(\mathbf{m})}{\partial m_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m(\mathbf{m})}{\partial m_1} & \cdots & \frac{\partial f_m(\mathbf{m})}{\partial m_n} \end{bmatrix}. \tag{9.20}$$

Similarly, we can express the Hessian of $f(\mathbf{m})$ using the $f_i(\mathbf{m})$ terms to obtain

$$\mathbf{H}(f(\mathbf{m})) = \sum_{i=1}^{m} \mathbf{H}(f_i(\mathbf{m})^2) \tag{9.21}$$

$$= \sum_{i=1}^{m} \mathbf{H}^i(\mathbf{m}), \tag{9.22}$$

where $\mathbf{H}^i(\mathbf{m})$ is the Hessian of $f_i(\mathbf{m})^2$.

The $j,\ k$ element of $\mathbf{H}^i(\mathbf{m})$ is

$$H^i_{j,k}(\mathbf{m}) = \frac{\partial^2(f_i(\mathbf{m})^2)}{\partial m_j \partial m_k} \tag{9.23}$$

$$= \frac{\partial}{\partial m_j}\left(2f_i(\mathbf{m})\frac{\partial f_i(\mathbf{m})}{\partial m_k}\right) \tag{9.24}$$

$$= 2\left(\frac{\partial f_i(\mathbf{m})}{\partial m_j}\frac{\partial f_i(\mathbf{m})}{\partial m_k} + f_i(\mathbf{m})\frac{\partial^2 f_i(\mathbf{m})}{\partial m_j \partial m_k}\right). \tag{9.25}$$

Thus

$$\mathbf{H}(f(\mathbf{m})) = 2\mathbf{J}(\mathbf{m})^T\mathbf{J}(\mathbf{m}) + \mathbf{Q}(\mathbf{m}), \tag{9.26}$$

where

$$\mathbf{Q}(\mathbf{m}) = 2\sum_{i=1}^{m} f_i(\mathbf{m})\mathbf{H}(f_i(\mathbf{m})). \tag{9.27}$$

In the **Gauss–Newton (GN) method**, we ignore the $\mathbf{Q}(\mathbf{m})$ term in (9.26) and approximate the Hessian as

$$\mathbf{H}(f(\mathbf{m})) \approx 2\mathbf{J}(\mathbf{m})^T\mathbf{J}(\mathbf{m}). \tag{9.28}$$

In the context of nonlinear regression, we expect that the $f_i(\mathbf{m})$ terms will be small as we approach the optimal parameters $\mathbf{m}^*$, so that this should be a reasonable approximation in the vicinity of the solution. Conversely, this is not a reasonable approximation for nonlinear least squares problems in which the values of $f_i(\mathbf{m})$ can be large.

Implementing Newton's method for minimizing $f(\mathbf{m})$ (9.12) using the gradient (9.19) and the approximate Hessian (9.28), and dividing both sides by 2, we obtain

$$\mathbf{J}(\mathbf{m}^k)^T\mathbf{J}(\mathbf{m}^k)\Delta\mathbf{m} = -\mathbf{J}(\mathbf{m}^k)^T\mathbf{F}(\mathbf{m}^k), \tag{9.29}$$

which provides a formula for solving for successive update steps $\Delta\mathbf{m}$. The n by n matrix $\mathbf{J}(\mathbf{m}^k)^T\mathbf{J}(\mathbf{m}^k)$ is symmetric and positive semidefinite. If the matrix is actually positive definite then we can use the Cholesky factorization or another full–rank method to solve the system of equations for $\Delta\mathbf{m}$. However, if the matrix is singular then such straight-forward approaches will fail. Although the GN method often works well in practice, it is based on Newton's method, and can thus fail by converging to a local maximum or saddle point where $\nabla f(\mathbf{m}) \approx 0$, or by not converging at all. An additional point to consider in using GN and other such iterative methods is that the algorithm may converge to a local minimum rather than to a global minimum (see Section 9.5).

In the **Levenberg–Marquardt (LM) method**, the GN method model update equations (9.29) are modified to

$$(\mathbf{J}(\mathbf{m}^k)^T\mathbf{J}(\mathbf{m}^k) + \lambda\mathbf{I})\Delta\mathbf{m} = -\mathbf{J}(\mathbf{m}^k)^T\mathbf{F}(\mathbf{m}^k), \tag{9.30}$$

and the positive parameter λ is adjusted during the course of iterations to ensure convergence. One important reason for this modification is to ensure that the matrix on the left-hand side of (9.30) is nonsingular. Since the matrix in this system of equations is symmetric and positive definite, we can use the Cholesky factorization to efficiently solve the system for the model update steps $\delta\mathbf{m}$.

For very large values of λ,

$$\mathbf{J}(\mathbf{m}^k)^T \mathbf{J}(\mathbf{m}^k) + \lambda \mathbf{I} \approx \lambda \mathbf{I}, \tag{9.31}$$

and the solution to (9.30) is

$$\Delta\mathbf{m} \approx -\frac{1}{\lambda}\nabla f(\mathbf{m}). \tag{9.32}$$

This is called a **steepest-descent** step, meaning that the algorithm simply moves down-gradient to most rapidly reduce $f(\mathbf{m})$. The steepest-descent approach provides very slow but certain convergence to a local minimum. Conversely, for very small values of λ, the LM method reverts to the GN method (9.29), which gives potentially fast but uncertain convergence.

A challenge associated with implementing the LM method is determining the optimal value of λ. The general strategy is to use small values of λ in situations where the GN method is working well, but to increase λ when the GN method fails to make progress in reducing the residual norm. A simple approach is to start with a small value of λ, and then adjust it in every iteration. If the LM method leads to a residual norm reduction, then update $\mathbf{m}$ and decrease λ by a constant multiplicative factor (e.g., a factor of 2) before the next iteration. Conversely, if the LM method does not improve the solution, we then increase λ by a constant factor and try again, repeating this process until a model update is found that decreases the residual norm. Robust implementations of the LM method use more sophisticated strategies for adjusting λ, but even this simple strategy works surprisingly well.

In practice, a careful LM implementation offers the good performance of the GN method as well as robust convergence properties, and LM is usually the method of choice for small- to medium-sized nonlinear least squares problems.

Note that although the LM stabilization term $\lambda\mathbf{I}$ in (9.30) resembles expressions used elsewhere (e.g., Chapter 4) for Tikhonov regularization purposes, it does not alter the ultimate model achieved at convergence. The $\lambda\mathbf{I}$ term is used to stabilize the solution of the linear system of equations that determines the search direction to be used. Because the $\lambda\mathbf{I}$ term is only used as a way to improve the convergence of the algorithm, and does *not* enter into the residual norm objective function that is being minimized, it does not regularize the nonlinear least squares problem. We discuss the regularization of nonlinear problems in Chapter 10.

9.4. STATISTICAL ASPECTS OF NONLINEAR LEAST SQUARES

Recall from Appendix B that if a vector $\mathbf{d}$ has a multivariate normal distribution, and $\mathbf{A}$ is an appropriately sized matrix, then $\mathbf{Ad}$ also has a multivariate normal distribution with an associated covariance matrix

$$\text{Cov}(\mathbf{Ad}) = \mathbf{A}\text{Cov}(\mathbf{d})\mathbf{A}^T. \tag{9.33}$$

In Chapter 2, we applied this formula to the linear least squares problem for $\mathbf{Gm} = \mathbf{d}$, which we solved by the normal equations. The resulting formula for $\text{Cov}(\mathbf{m})$ was

$$\text{Cov}(\mathbf{m}_{L_2}) = (\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T\text{Cov}(\mathbf{d})\mathbf{G}(\mathbf{G}^T\mathbf{G})^{-1}. \tag{9.34}$$

In the simplest case, where $\text{Cov}(\mathbf{d}) = \sigma^2\mathbf{I}$, (9.34) simplified to

$$\text{Cov}(\mathbf{m}_{L_2}) = \sigma^2(\mathbf{G}^T\mathbf{G})^{-1}. \tag{9.35}$$

For the nonlinear regression problem, we no longer have a linear relationship between the data and the estimated model parameters, so we cannot assume that the estimated model parameters have a multivariate normal distribution, and cannot use the above formulas. However, we can obtain useful corresponding formulas by linearization if the data errors are not too large.

If small data perturbations result in small model perturbations in a nonlinear system, we can consider a linearization of the misfit function (9.15) about a solution, $\mathbf{m}^*$:

$$\mathbf{F}(\mathbf{m}^* + \Delta\mathbf{m}) \approx \mathbf{F}(\mathbf{m}^*) + \mathbf{J}(\mathbf{m}^*)\Delta\mathbf{m}. \tag{9.36}$$

Under this approximation, there is a linear relationship between changes in $\mathbf{F}$ and changes in the parameters $\mathbf{m}$:

$$\mathbf{F}(\mathbf{m}^* + \Delta\mathbf{m}) - \mathbf{F}(\mathbf{m}^*) = \Delta\mathbf{F} \approx \mathbf{J}(\mathbf{m}^*)\Delta\mathbf{m}. \tag{9.37}$$

To the extent that the residual misfit terms (the elements of $\mathbf{F}(\mathbf{m}^*)$) are small, the Hessian can be approximated by (9.28). In this case $\mathbf{J}(\mathbf{m}^*)$ in nonlinear regression can take the place of $\mathbf{G}$ in linear problems (e.g., (9.35)) to estimate the covariance of the model parameters. Equation (9.13) incorporates the respective data element standard deviations σ_i into the formula for the residual norm $f(\mathbf{m})$, and explicitly weights the nonlinear constraint equations. $\text{Cov}(\mathbf{d})$, in the case of independent data errors, is thus the identity matrix in such a weighted system. In this case we have

$$\text{Cov}(\mathbf{m}^*) \approx (\mathbf{J}(\mathbf{m}^*)^T\mathbf{J}(\mathbf{m}^*))^{-1}. \tag{9.38}$$

As in Chapter 2, we can construct a 95% confidence ellipsoid for the fitted parameters in terms of this covariance matrix,

$$(\mathbf{m} - \mathbf{m}^*)^T \mathbf{J}(\mathbf{m}^*)^T \mathbf{J}(\mathbf{m}^*)(\mathbf{m} - \mathbf{m}^*) \leq \Delta^2, \tag{9.39}$$

where Δ^2 is the ninety-fifth percentile of a χ^2 distribution with n degrees of freedom. It is important to reiterate that this confidence ellipsoid only defines an approximate confidence region because we linearized $\mathbf{F}(\mathbf{m})$ in (9.37).

As in Chapter 2, there is an alternative derivation of the 95% confidence region for the fitted parameters in terms of $\chi^2(\mathbf{m}) - \chi^2(\mathbf{m}^*)$. If we let

$$\chi^2(\mathbf{m}) = \|\mathbf{F}(\mathbf{m})\|_2^2 = \mathbf{F}(\mathbf{m})^T \mathbf{F}(\mathbf{m}), \tag{9.40}$$

then the inequality

$$\chi^2(\mathbf{m}) - \chi^2(\mathbf{m}^*) \leq \Delta^2 \tag{9.41}$$

defines an approximate 95% confidence region for $\mathbf{m}^*$. Because of the nonlinearity in the $\chi^2(\mathbf{m})$ function, this 95% confidence region is typically not an ellipsoid. Unfortunately, this confidence region is not exact, because this development is still based on the assumption of multivariate normality, and $\mathbf{m}^*$, because it is not a linear combination of the data vector elements, will not in general have a multivariate normal distribution in nonlinear problems.

Should (9.39) or (9.41) be used in practice? If $G(\mathbf{m})$ is not too strongly nonlinear, then there should be very little difference between the two confidence regions. However, if $G(\mathbf{m})$ is more strongly nonlinear, then the difference between the two confidence regions may be large and neither confidence region can really be trusted. In such situations, Monte Carlo methods are a more useful and appropriate way to estimate parameter uncertainty. This is discussed further in Chapter 11.

As with linear regression, it is possible to apply nonlinear regression when the measurement errors are independent and normally distributed and the standard deviations are unknown but assumed to be equal (Section 3.3). We set the σ_i to 1 and minimize the sum of squared errors, and define a residual vector,

$$r_i = G(\mathbf{m}^*)_i - d_i \quad i = 1, 2, \ldots, m. \tag{9.42}$$

Our estimate of the measurement standard deviation is then

$$s = \sqrt{\frac{\|\mathbf{r}\|_2^2}{m - n}}, \tag{9.43}$$

and the corresponding approximate covariance matrix for the estimated model parameters is

$$\text{Cov}(\mathbf{m}^*) = s^2 (\mathbf{J}(\mathbf{m}^*)^T \mathbf{J}(\mathbf{m}^*))^{-1}. \tag{9.44}$$

Given $\mathbf{m}^*$ and $\text{Cov}(\mathbf{m}^*)$, we can establish confidence intervals for the model parameters using the methods described in Chapter 2. As with any parameter estimation problem, it is also important to examine the residuals for systematic patterns or deviations from normality. If we have not estimated the measurement standard deviation s from the fitted residuals, then it is also important to test the χ^2 value for goodness of fit and perform an associated **p-value** test (2.21) if we have normally distributed data errors. The appropriateness of this test will depend on how well the nonlinear model is approximated by the Jacobian linearization for points near the optimal parameter values.

● ──

Example 9.1

A classic method in hydrology for determining the transmissivity and storage coefficient of an aquifer is called the "slug test" [49].

A known volume Q of water (the slug) is injected into a well, and the resulting effects on the head, h, at an observation well a distance d away from the injection well are observed at various times t. The change in the head measured at the observation well typically increases rapidly and then decreases more slowly. We wish to determine the dimensionless storage coefficient, S, and the transmissivity T.

The mathematical model for the slug test is

$$h = \frac{Q}{4\pi\,Tt} e^{-d^2 S/(4Tt)}. \tag{9.45}$$

We know the parameters $Q = 50$ m^3 and $d = 60$ m, and the times t at which the head h is measured. Our data are given in Table 9.1, where head measurements are roughly accurate to 0.01 m ($\sigma_i = 0.01$ m).

The optimal parameter values are $S = 0.00207$ and $T = 0.585$ m^2/hr. The observed χ^2 value is 2.04, with a corresponding p-value of 0.73. Thus this fit passes the χ^2 test. The data points and fitted curve are shown in Figure 9.1.

Using the Jacobian evaluated at the optimal parameter values, we computed an approximate covariance matrix for the fitted parameters. The resulting 95% confidence intervals for S and T are

$$S = 0.00207 \pm 0.00012 \tag{9.46}$$

$$T = 0.585 \pm 0.029 \text{ m}^2/\text{hr}. \tag{9.47}$$

Table 9.1 Slug Test Data

t (hr)	5	10	20	30	40	50
h (m)	0.72	0.49	0.30	0.20	0.16	0.12
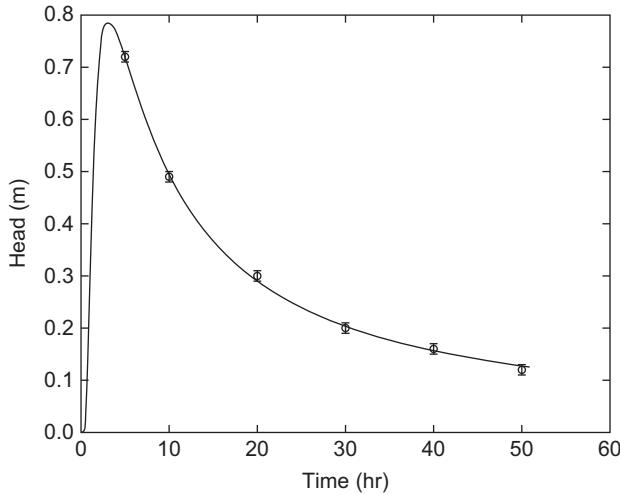
Figure 9.1 Data, with one standard deviation error bars, and fitted model for the slug test.
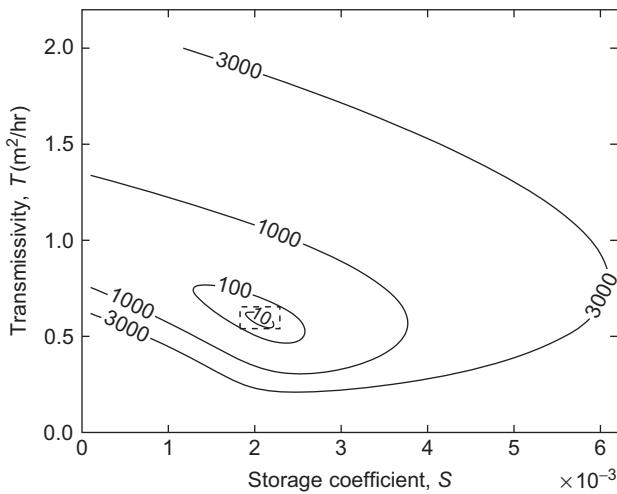
Figure 9.2 χ^2 contour plot for the slug test for a wide range of parameters. The small box shows the close-up axis range displayed in Figure 9.3.

A contour plot of the χ^2 surface obtained by varying S and T is shown in Figure 9.2. Note that, unlike our earlier linear regression problems, the contours are not even approximately elliptical because of the nonlinearity of the problem. However, if we zoom into the immediate vicinity of the optimal parameters (Figure 9.3), we find that the χ^2 contours are approximately elliptical at this scale. The approximate ellipticity
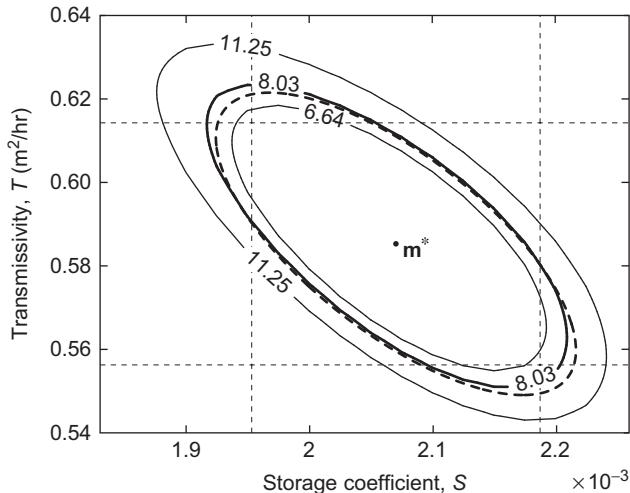
Figure 9.3 Close-up χ^2 contour plot for the immediate vicinity of the optimal parameters $\mathbf{m}^*$ for the slug test example, showing nonlinear 90%, 95% (bold), and 99% confidence contours. The nearly ellip-tical contours are indicative of a approximately linear behavior at this scale. Dashed ellipse shows the linearized approximation to the 95% confidence contour calculated using (9 38). Ninety-five percent confidence intervals for the parameters calculated from (9.38) are shown as dashed lines.

of the contours indicates that the linear approximation of $\mathbf{G}(\mathbf{m})$ around the optimal parameter values is a reasonable approximation for model perturbations in this range.

9.5. IMPLEMENTATION ISSUES

Iterative methods for solving nonlinear problems require the computation of the func-tions $f_i(\mathbf{m})$ and their partial derivatives with respect to the model parameters m_j. These partial derivatives in turn depend on the derivatives of the nonlinear $\mathbf{G}$:

$$\frac{\partial f_i(\mathbf{m})}{\partial m_j} = \frac{1}{\sigma_i}\frac{\partial \mathbf{G}(\mathbf{m})_i}{\partial m_j}. \tag{9.48}$$

In some cases, we have explicit formulas for $\mathbf{G}(\mathbf{m})$ and its derivatives. In other cases, $\mathbf{G}(\mathbf{m})$ exists only as a **black box** subroutine that can be called as required to evaluate the function.

When an explicit formula for $\mathbf{G}(\mathbf{m})$ is available, and the number of parameters is relatively small, we can differentiate analytically. There also exist **automatic differen-tiation** software packages that can translate the source code of a program that computes $\mathbf{G}(\mathbf{m})$ into a program that computes the derivatives of $\mathbf{G}(\mathbf{m})$.

Another approach is to use **finite differences** to approximate the derivatives of $G(\mathbf{m})_i$. A simple first-order scheme is to employ the linearization

$$\frac{\partial G(\mathbf{m})_i}{\partial m_j} \approx \frac{G(\mathbf{m} + h\mathbf{e}_j)_i - G(\mathbf{m})_i}{h}, \tag{9.49}$$

where $\mathbf{e}_j$ is the *jth* standard basis vector, and the scalar h is chosen to be suitably small. However, if h becomes very small, significant round–off error in the numerator of (9.49) may occur. A good rule of thumb is to set $h = \sqrt{\epsilon}$, where ϵ is the relative accuracy of the evaluations of $G(\mathbf{m})_i$. For example, if the function evaluations are accurate to 0.0001, then an appropriate choice of h would be about 0.01. Determining the actual accuracy of function evaluations can be difficult, especially when G is a black box routine. One useful assessment technique is to plot function values as a parameter of interest is varied over a small range. These plots should be smooth at the scale of h. When G is available only as a black box subroutine that can be called with particular values of $\mathbf{m}$, and the source code for the subroutine is not available, then the only available approach is to use finite differences.

In practice, many difficulties in solving nonlinear regression problems can be traced back to incorrect derivative computations. It is thus a good idea to cross–check any available analytical formulas for the derivative with finite-difference approximations. Many software packages for nonlinear regression include options for checking the accuracy of derivative formulas.

A second important issue in the implementation of the GN and LM methods is deciding when to terminate the iterations. We would like to stop when the gradient $\nabla f(\mathbf{m})$ is approximately $\mathbf{0}$ and $\mathbf{m}$ has stopped changing substantially from one iteration to the next. Because of scaling issues, it is not possible to set an absolute tolerance on $\|\nabla f(\mathbf{m})\|_2$ that would be appropriate for all problems. Similarly, it is difficult to pick a single absolute tolerance on $\|\mathbf{m}^{k+1} - \mathbf{m}^k\|_2$ or $|f(\mathbf{m}^{k+1}) - f(\mathbf{m}^k)|$.

The following convergence tests have been normalized so that they will work well on a wide variety of problems. We assume that values of $G(\mathbf{m})$ can be calculated with a relative accuracy of ϵ. To ensure that the gradient of $f(\mathbf{m})$ is approximately $\mathbf{0}$, we require that

$$\|\nabla f(\mathbf{m}^k)\|_2 < \sqrt{\epsilon}(1 + |f(\mathbf{m}^k)|). \tag{9.50}$$

To ensure that successive values of $\mathbf{m}$ are close, we require

$$\|\mathbf{m}^k - \mathbf{m}^{k-1}\|_2 < \sqrt{\epsilon}(1 + \|\mathbf{m}^k\|_2). \tag{9.51}$$

Finally, to make sure that the values of $f(\mathbf{m})$ have stopped changing, we require that

$$|f(\mathbf{m}^k) - f(\mathbf{m}^{k-1})| < \epsilon(1 + |f(\mathbf{m}^k)|). \tag{9.52}$$

There are a number of additional problems that can arise during the solution of a nonlinear regression problem by the GN or LM methods related to the functional behavior of $f(\mathbf{m})$.

The first issue is that our methods, because they involve calculation of the Jacobian and the Hessian, assume that $f(\mathbf{m})$ is a smooth function. This means not only that $f(\mathbf{m})$ must be continuous, but also that its first and second partial derivatives with respect to the parameters must be continuous. Figure 9.4 shows a function that is itself continuous, but has discontinuities in the first derivative at $m = 0.2$ and the second derivative at $m = 0.5$. When $\mathbf{G}(\mathbf{m})$ is given by an explicit formula, it is usually easy to verify that $f(\mathbf{m})$ is smooth, but when $\mathbf{G}(\mathbf{m})$ is implemented as a black box routine it may be difficult.

A second issue is that the function $f(\mathbf{m})$ may have a "flat bottom." See Figure 9.5. In such cases, there are many values of $\mathbf{m}$ that come close to fitting the data, and it is difficult to determine the optimal $\mathbf{m}^*$. In practice, this condition is seen to occur when $\mathbf{J}(\mathbf{m}^*)^T\mathbf{J}(\mathbf{m}^*)$ is nearly singular. Because of this ill–conditioning, computing accurate confidence intervals for the model parameters can be effectively impossible. We will address this difficulty in Chapter 10 by applying regularization.

The final problem that we will consider is that $f(\mathbf{m})$ may be nonconvex and therefore have multiple local minimum points. See Figure 9.6. The GN and LM methods are designed to converge to a local minimum, but depending on where we begin the search, there is no way to be certain that such a solution will be a global minimum. Depending on the particular problem, the optimization algorithm might well converge to a locally optimal solution.

Figure 9.4 Example of a nonsmooth function.

Figure 9.5 Example of a function with a flat bottom.

Figure 9.6 An example of a function with multiple local minima.

Global optimization methods have been developed to deal with this issue [17, 75, 76, 140]. Deterministic global optimization procedures can be used on problems with a very small number of variables, while stochastic search procedures can be applied to large-scale problems. Stochastic search procedures can be quite effective in practice, even though they do not find a global optimum with certainty.

However, even a deterministic global optimization procedure is not a panacea. In the context of nonlinear regression, if the nonlinear least squares problem has multiple locally

optimal solutions with similar objective function values, then each of these solutions will correspond to a statistically likely solution. We cannot simply report one globally optimal solution as our best estimate and construct confidence intervals using (9.38), because this would mean ignoring other likely solutions. However, if we could show that there is one globally optimal solution and other locally optimal solutions have very small p-values, then it would be appropriate to report the globally optimal solution and corresponding confidence intervals.

Although a thorough discussion of global optimization is beyond the scope of this book, we will discuss one simple global optimization procedure called the **multistart method** in the next example. In the multistart method, we randomly generate a large number of initial solutions, and then apply the LM method starting with each of these initial solutions. We then examine the local minimum solutions found by the procedure, and examine ones with acceptable values of $f(\mathbf{m})$. The multistart approach has two important practical advantages. First, by potentially identifying multiple locally optimal solutions, we can determine whether there is more than one statistically likely solution. Second, we can make effective use of the fast convergence of the LM method to find the locally optimal solutions.

Example 9.2

Consider the problem of fitting a model of two superimposed exponential decay functions characterized by four parameters,

$$y_i = m_1 e^{m_2 x_i} + m_3 x_i e^{m_4 x_i}, \tag{9.53}$$

to a set of observations. The true model parameters are $m_1 = 1.0$, $m_2 = -0.5$, $m_3 = 1.0$, and $m_4 = -0.75$, and the x_i values are 25 evenly spaced points between 1 and 7. We compute corresponding y_i values and add independent normally distributed noise with a standard deviation of 0.01 to obtain a synthetic data set.

We next apply the LM method to solve the problem 20 times, using random initial solutions, with each initial parameter uniformly distributed between -1 and 1. This produces a total of three different locally optimal solutions (Table 9.2).

Solution number 1, with a χ^2 value of approximately 17, has an acceptable p-value of about 0.69 for a regression with 21 degrees of freedom. The other two solutions have

Table 9.2 Locally Optimal Solutions for the Sample Problem

Solution Number	m_1	m_2	m_3	m_4	χ^2	p-value
1	0.9874	−0.5689	1.0477	−0.7181	17.3871	0.687
2	1.4368	0.1249	−0.5398	−0.0167	40.0649	0.007
3	1.5529	−0.1924	−0.1974	−0.1924	94.7845	2×10^{-11}

unreasonably large χ^2 values, and hence much lower p-values. We will thus analyze only the first solution. Figure 9.7 shows the data points with 1σ error bars and the fitted curve for solution number 1, and Figure 9.8 shows the corresponding residuals normalized by the data standard deviations. Note that the majority of the residuals are within 0.5 standard deviations, with a few residuals as large as 1.9 standard deviations. There is no obvious residual trend as x ranges from 1 to 7.

Figure 9.7 Data points and fitted curve.
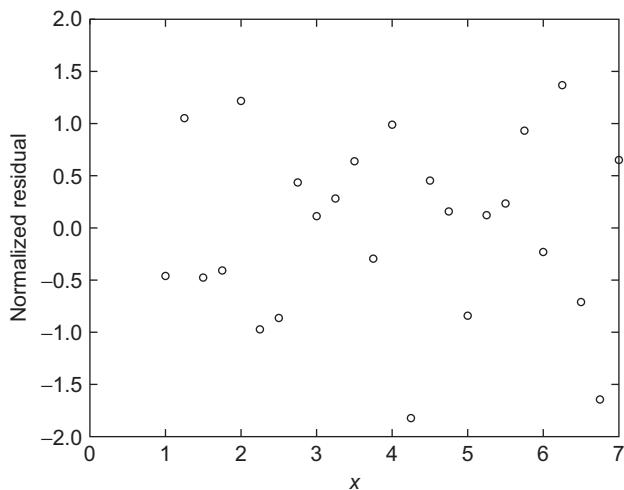
Figure 9.8 Normalized residuals corresponding to Figure 9.7.

Next, we compute the approximate covariance matrix for the model parameters using (9.38). The square roots of the diagonal elements of the covariance matrix are standard deviations for the individual model parameters. These are then used to compute 95% confidence intervals for model parameters. The solution parameters with 95% confidence intervals are

$$
\begin{aligned}
m_1 &= 0.98 \pm 0.22 \\
m_2 &= -0.57 \pm 0.77 \\
m_3 &= 1.05 \pm 0.50 \\
m_4 &= -0.72 \pm 0.20.
\end{aligned}
\tag{9.54}
$$

The true parameters $(1, -0.5, 1, \text{ and } -0.75)$ are all covered by these confidence intervals. However, there is a large degree of uncertainty. This is an example of a poorly conditioned nonlinear regression problem in which the data do not strongly constrain the parameter values.

The correlation matrix provides some insight into the nature of the ill-conditioning in this example by quantifying strong parameter trade-offs. For our preferred solution, the correlation matrix calculated using (9.38) is

$$
\rho = \begin{bmatrix}
1.00 & -0.84 & 0.68 & 0.89 \\
-0.84 & 1.00 & -0.96 & -0.99 \\
0.68 & -0.96 & 1.00 & 0.93 \\
0.89 & -0.99 & 0.93 & 1.00
\end{bmatrix}.
\tag{9.55}
$$

Note the strong positive and negative correlation between pairs of parameters. The high negative correlation between m_1 and m_2 tells us that by increasing m_1 and simultaneously decreasing m_2 we can obtain a solution that is very nearly as good as our optimal solution. There are also strong negative correlations between m_2 and m_3 and between m_2 and m_4.

9.6. EXERCISES

1. Show that (9.29) is equivalent to the normal equations (2.3) when $G(\mathbf{m}) = \mathbf{Gm}$, where $\mathbf{G}$ is a matrix of constant coefficients.

2. A recording instrument sampling at 50 Hz records a noisy sinusoidal voltage signal in a 40-s–long record. The data are to be modeled using

$$
y(t) = A \sin(2\pi f_0 t + \phi) + c + \sigma \eta(t)
\tag{9.56}
$$

where $\eta(t)$ is believed to be unit standard deviation, independent, and normally distributed noise, and σ is an unknown standard deviation. Using the data in the MATLAB data file **instdata.mat**, solve for the parameter (A, f_0, ϕ, c), using the

LM method. Show that it is critical to choose a good initial solution (suitable initial parameters can be found by examining a plot of the time series by eye). Once you are satisfied that you have found a good solution, use it to estimate the noise amplitude σ. Use your solution and estimate of σ to find corresponding covariance and correlation matrices and 95% parameter confidence intervals. Which pair of parameters is most strongly correlated? Are there multiple equally good solutions for this problem?

3. In hydrology, the van Genuchten model is often used to relate the volumetric water content in an unsaturated soil to the head [52]. The model is

$$\theta(h) = \theta_r + \frac{\theta_s - \theta_r}{(1 + (-\alpha h)^n)^{(1-1/n)}} \tag{9.57}$$

where θ_s is the volumetric water content at saturation, θ_r is the residual volumetric water content at a very large negative head, and α and n are two parameters which can be fit to laboratory measurements.

The file **vgdata.mat** contains measurements for a loam soil at the Bosque del Apache National Wildlife Refuge in New Mexico [68]. Fit the van Genuchten model to the data. The volumetric water content at saturation is $\theta_s = 0.44$, and the residual water content is $\theta_r = 0.09$. You may assume that the measurements of $\theta(h)$ are accurate to about 2% of the measured values.

You will need to determine appropriate values for σ_i, write functions to compute $\theta(h)$ and its derivatives, and then use the LM method to estimate the parameters. In doing so, you should consider whether or not this problem might have local minima. It will be helpful to know that typical values of α range from about 0.001 to 0.02, and typical values of n run from 1 to 10.

4. An alternative version of the LM method stabilizes the GN method by multiplicative damping. Instead of adding $\lambda \mathbf{I}$ to the diagonal of $\mathbf{J}(\mathbf{m}^k)^T \mathbf{J}(\mathbf{m}^k)$, this method multiplies the diagonal of $\mathbf{J}(\mathbf{m}^k)^T \mathbf{J}(\mathbf{m}^k)$ by a factor of $(1 + \lambda)$. Show that this method can fail by producing an example in which the modified $\mathbf{J}(\mathbf{m}^k)^T \mathbf{J}(\mathbf{m}^k)$ matrix is singular, no matter how large λ becomes.

5. A cluster of 10 small earthquakes occurs in a shallow geothermal reservoir. The field is instrumented with nine seismometers, eight of which are at the surface and one of which is 300 m down a borehole. The P-wave velocity of the fractured granite medium is thought to be an approximately uniform 2 km/s. The station locations (in meters relative to a central origin) are given in Table 9.3 and in the MATLAB data file **stmat.mat**.

The arrival times of P-waves from the earthquakes are carefully measured at the stations, with an estimated error of approximately 1 ms. The arrival time estimates for each earthquake, e_i, at each station (in seconds relative to an arbitrary reference), are given in Table 9.4. These data can also be found in the MATLAB data file **eqdata.mat**.

Table 9.3 Station Locations for the
Earthquake Location Problem

Station	x(m)	y(m)	z(m)
1	500	−500	0
2	−500	−500	0
3	100	100	0
4	−100	0	0
5	0	100	0
6	0	−100	0
7	0	−50	0
8	0	200	0
9	10	50	−300

Table 9.4 Data for the Earthquake Location Problem.
Times are in Seconds

Station	e_1	e_2	e_3	e_4	e_5
1	0.8423	1.2729	0.8164	1.1745	1.1954
2	0.8680	1.2970	0.8429	1.2009	1.2238
3	0.5826	1.0095	0.5524	0.9177	0.9326
4	0.5975	1.0274	0.5677	0.9312	0.9496
5	0.5802	1.0093	0.5484	0.9145	0.9313
6	0.5988	1.0263	0.5693	0.9316	0.9480
7	0.5857	1.0141	0.5563	0.9195	0.9351
8	0.6017	1.0319	0.5748	0.9362	0.9555
9	0.5266	0.9553	0.5118	0.8533	0.8870

Station	e_6	e_7	e_8	e_9	e_{10}
1	0.5361	0.7633	0.8865	1.0838	0.9413
2	0.5640	0.7878	0.9120	1.1114	0.9654
3	0.2812	0.5078	0.6154	0.8164	0.6835
4	0.2953	0.5213	0.6360	0.8339	0.6982
5	0.2795	0.5045	0.6138	0.8144	0.6833
6	0.2967	0.5205	0.6347	0.8336	0.6958
7	0.2841	0.5095	0.6215	0.8211	0.6857
8	0.3025	0.5275	0.6394	0.8400	0.7020
9	0.2115	0.4448	0.5837	0.7792	0.6157

a. Apply the LM method to this data set to estimate least squares locations of the earthquakes.

b. Estimate the uncertainties in x, y, z (in meters) and origin time (in seconds) for each earthquake using the diagonal elements of the appropriate covariance matrix. Do the earthquake locations follow any discernible trend?

Table 9.5 Data for the Lightning Mapping Array Problem

Station	t(s)	x(km)	y(km)	z(km)
1	0.0922360280	−24.3471411	2.14673146	1.18923667
2	0.0921837940	−12.8746056	14.5005985	1.10808551
3	0.0922165500	16.0647214	−4.41975194	1.12675062
4	0.0921199690	0.450543748	30.0267473	1.06693166
6	0.0923199800	−17.3754105	−27.1991732	1.18526730
7	0.0922839580	−44.0424408	−4.95601205	1.13775547
8	0.0922030460	−34.6170855	17.4012873	1.14296361
9	0.0922797660	17.6625731	−24.1712580	1.09097830
10	0.0922497250	0.837203704	−10.7394229	1.18219520
11	0.0921672710	4.88218031	10.5960946	1.12031719
12	0.0921702350	16.9664920	9.64835135	1.09399160
13	0.0922357370	32.6468622	−13.2199767	1.01175261

6. The Lightning Mapping Array is a portable system that locates the sources of light-ning radio-frequency radiation in three spatial dimensions and time [132]. The system measures the arrival times of impulsive radiation events with an uncertainty specified by a standard deviation of 7×10^{-2} µs. Measurements are made at multiple locations, typically in a region 40 to 60 km in diameter. Each station records the peak radiation event in successive 100-µs time intervals; from such data, several hundred to over a thousand distinct radiation sources may typically be located per lightning discharge. Example data from the LMA are shown in Table 9.5 and are found in the MATLAB data file **lightningdata.mat**.

  a. Use the arrival times at stations 1, 2, 4, 6, 7, 8, 10, and 13 to find the time and location of the associated source. Assume that radio wave signals travel along straight paths at the speed of light in a vacuum (2.997×10^8 m/s).

  b. A challenge in dealing with the large number of detections from the LMA is to disentangle overlapping signals from multiple events. Locate using subsets of the data set to find the largest subset of the data for nine or more stations that gives a good solution, and compare it to the station subset from part (a).

9.7. NOTES AND FURTHER READING

Newton's method is central to the field of optimization [39, 58, 87, 88, 119], and is, because of its speed, the basis for most methods of nonlinear optimization. A number of modifications to the method are used to ensure convergence to a local minimum of $f(\mathbf{x})$ [58, 119]. One important difficulty in Newton's method is that, for very large problems, it may be impractical to store the Hessian matrix. Specialized methods have been developed for the solution of such large-scale optimization problems [58, 119].

The GN and LM methods are discussed in more detail in [16, 58, 119]. Statistical aspects of nonlinear regression are discussed in [7, 40, 113]. A more detailed discussion of the termination criteria for the LM method described in Section 9.5 can be found in [58]. There are a number of freely available and commercial software packages for nonlinear regression, including GaussFit [82], MINPACK [111], and ODRPACK [18]. Automatic differentiation has applications in many areas of numerical computing, including optimization and numerical solution of ordinary and partial differential equations. Two books that survey this topic are [34, 57]. Global optimization is a large field of research. Some basic references include [17, 75, 76]. For a survey of global optimization methods in geophysical inversion, see [140].

Nonlinear Inverse Problems

Synopsis

The nonlinear regression approaches of Chapter 9 are generalized to problems requiring regularization. The Tikhonov regularization and Occam's inversion approaches are introduced. Seismic tomography and electrical conductivity inversion examples are used to illustrate the application of these methods. Resolution analysis for nonlinear problems is addressed.

10.1. REGULARIZING NONLINEAR LEAST SQUARES PROBLEMS

As with linear problems, the nonlinear least squares approaches can run into difficulty with ill-conditioned problems. This typically happens as the number of model parameters grows. Here, we will discuss regularization of nonlinear inverse problems and algorithms for computing a regularized solution to a nonlinear inverse problem.

The basic ideas of Tikhonov regularization can be extended to nonlinear problems. Suppose that we are given a nonlinear discrete inverse problem where an n-element model $\mathbf{m}$ and an m-element data vector $\mathbf{d}$ are related by a nonlinear system of equations $\mathbf{G}(\mathbf{m}) = \mathbf{d}$. For convenience, we will assume that the m nonlinear constraint equations have been scaled to incorporate the measurement standard deviations σ_i. We seek the solution with the smallest seminorm $\|\mathbf{Lm}\|_2$ that comes sufficiently close to fitting the data vector, where $\mathbf{L}$ is an appropriate roughening matrix (e.g., (4.28)).

We can formulate this problem as

$$\begin{aligned} \min \ &\|\mathbf{Lm}\|_2 \\ &\|\mathbf{G}(\mathbf{m}) - \mathbf{d}\|_2 \leq \delta. \end{aligned} \tag{10.1}$$

Note that the form of the problem is virtually identical to that which was considered in the linear case (e.g., (4.25)), with the only difference being that we now have a general function $\mathbf{G}(\mathbf{m})$ instead of a matrix-vector multiplication $\mathbf{Gm}$. As in the linear case, we can reformulate this problem in terms of minimizing the misfit subject to a constraint on $\|\mathbf{Lm}\|_2$,

$$\begin{aligned} \min \ &\|\mathbf{G}(\mathbf{m}) - \mathbf{d}\|_2 \\ &\|\mathbf{Lm}\|_2 \leq \epsilon, \end{aligned} \tag{10.2}$$

or as a regularized (often referred to as "damped") least squares problem,

$$\min \|\mathbf{G}(\mathbf{m}) - \mathbf{d}\|_2^2 + \alpha^2 \|\mathbf{Lm}\|_2^2. \tag{10.3}$$

All three versions of the regularized least squares problem can be solved by applying standard nonlinear optimization software. In particular, (10.3) is a nonlinear least squares problem, so we can apply the LM or GN methods to it. Of course, any such approach will still have to deal with the possibility of local minima that are not global minimum points. In some cases, it is possible to show that nonlinear least squares problems are convex, and thus possess only global minima. In other cases we will have to employ multistart or some other global optimization strategy to determine whether there are multiple minima.

To apply the GN method to (10.3), we rewrite it as

$$\min \left\| \begin{matrix} \mathbf{G}(\mathbf{m}) - \mathbf{d} \\ \alpha \mathbf{Lm} \end{matrix} \right\|_2^2. \tag{10.4}$$

The Jacobian of (10.4) for the kth iteration is

$$\mathbf{K}(\mathbf{m}^k) = \begin{bmatrix} \mathbf{J}(\mathbf{m}^k) \\ \alpha \mathbf{L} \end{bmatrix}, \tag{10.5}$$

where $\mathbf{J}(\mathbf{m}^k)$ is the Jacobian of $\mathbf{G}(\mathbf{m}^k)$. A GN model step is obtained by applying (9.29) and solving

$$\mathbf{K}(\mathbf{m}^k)^T \mathbf{K}(\mathbf{m}^k) \Delta \mathbf{m} = -\mathbf{K}(\mathbf{m}^k)^T \begin{bmatrix} \mathbf{G}(\mathbf{m}^k) - \mathbf{d} \\ \alpha \mathbf{Lm}^k \end{bmatrix}, \tag{10.6}$$

or, combining (10.5) and (10.6), by solving

$$\left(\mathbf{J}(\mathbf{m}^k)^T \mathbf{J}(\mathbf{m}^k) + \alpha^2 \mathbf{L}^T \mathbf{L} \right) \Delta \mathbf{m} = -\mathbf{J}(\mathbf{m}^k)^T (\mathbf{G}(\mathbf{m}^k) - \mathbf{d}) - \alpha^2 \mathbf{L}^T \mathbf{Lm}^k. \tag{10.7}$$

Equation (10.7) resembles the LM method (9.30). Note, however, that α in (10.7) now appears in the objective function being minimized (10.4) and thus introduces regularization. To further stabilize the iterations, as in the LM method, a variable $\lambda \mathbf{I}$ term could be added to the matrix term of the left-hand side of (10.7) to steer iterative updates toward the direction of steepest descent. This will not be necessary if the explicit regularization of (10.7) sufficiently stabilizes the system of equations.

●———————————————————————————————————

Example 10.1

Consider a modified version of the cross-well tomography example from Exercise 4.3. We introduce nonlinearity by employing a more realistic forward model that incorporates ray-path bending due to seismic velocity changes. The two-dimensional velocity

structure is parameterized using a 64-element matrix of uniformly spaced slowness nodes on an 8 by 8 grid spanning a 1600 m by 1600 m region.

We apply an approximate ray-bending technique to estimate refracted ray paths within the slowness model and to estimate travel times and their partial derivatives with respect to the model parameters [167]. Figure 10.1 shows the true velocity model and a corresponding set of 64 ray paths. The true model consists of a background velocity of 2.9 km/s with large embedded fast (+10%) and slow (−15%) Gaussian-shaped anomalies. The data set consists of the 64 travel times between each pair of opposing sources and receivers with $N(0, (0.001 \text{ s})^2)$ noise added.

Note that refracted ray paths tend to be "repelled" from low-velocity regions (dark shading) and are, conversely, "attracted" to high-velocity regions (light shading) in accordance with Fermat's least-time principle. In practice this effect makes low-velocity regions more difficult to resolve in such studies because they will be less well sampled by ray paths.

A discrete approximation of the two-dimensional Laplacian operator is used to regularize this problem. Iterative GN (10.7) solutions were obtained for a range of 16 values of α ranging logarithmically from approximately 4.9 to 367. Figure 10.2 shows the suite of solutions after five iterations. An L-curve plot of seminorm versus data misfit is plotted in Figure 10.3, along with the discrepancy principle value $\delta = 0.001 \cdot \sqrt{64} = 0.008$. Note that the two most lightly regularized solutions are out of their expected monotonically decreasing residual norm positions on the L-curve. This is because the linearization-based GN method is unable to accurately solve the corresponding poorly conditioned least squares systems (10.7) in these cases. Such solutions

Figure 10.1 True velocity model and the corresponding ray paths for the bent-ray cross-well tomography example.
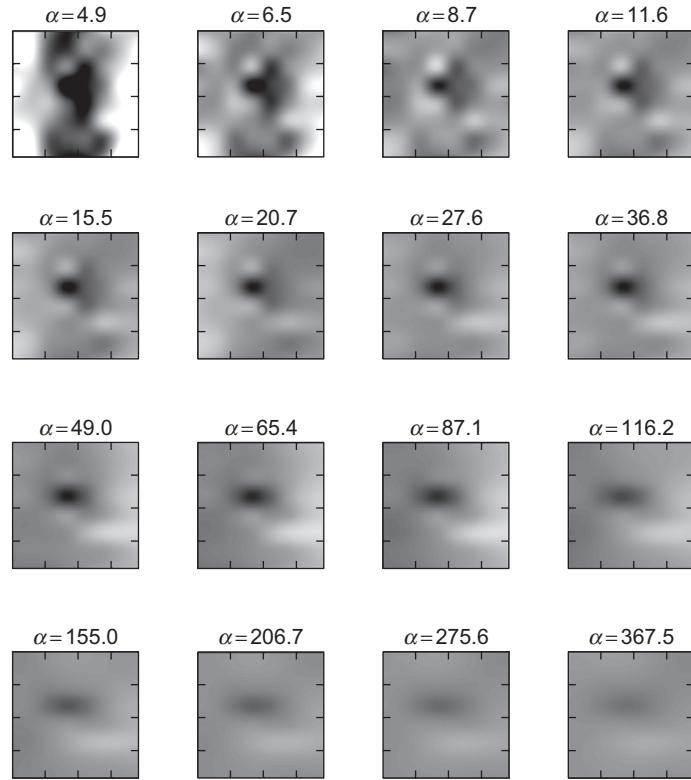
Figure 10.2 Suite of GN (10.7), second-order regularized solutions, ranging from least (upper left) to most (lower right) regularized, and associated α values. The physical dimensions and gray scale are identical to those of Figures 10.1 and 10.4.
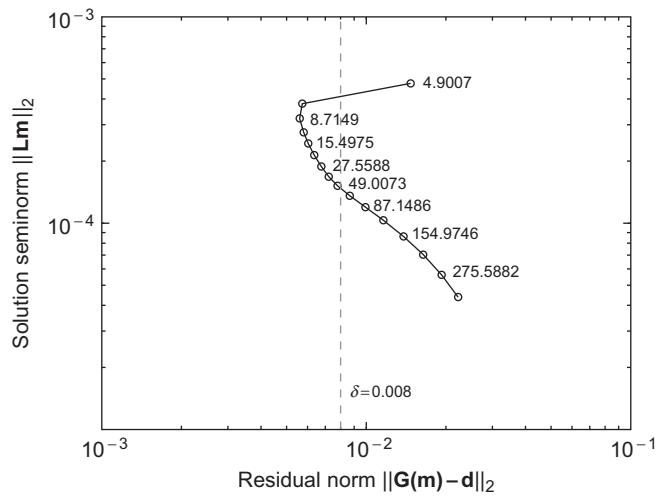
Figure 10.3 L-curve and corresponding α values for the solutions of Figure 10.2.

could be improved by modifying the system of equations with an LM stabilizing term as described above. The solution best satisfying the discrepancy principle corresponds to $\alpha \approx 37$ (Figure 10.4).

Because we know the true model in this example, it is instructive to examine how well the regularized solutions of Figure 10.2 compare to it. Figure 10.5 shows the 2-norm model misfit as a function of α, and demonstrates that the discrepancy principle solution for this problem, and for this particular noise realization, is indeed close to

Figure 10.4 Best-solution velocity structure (m/s), α selected using the discrepancy principle, $\alpha \approx 37$.
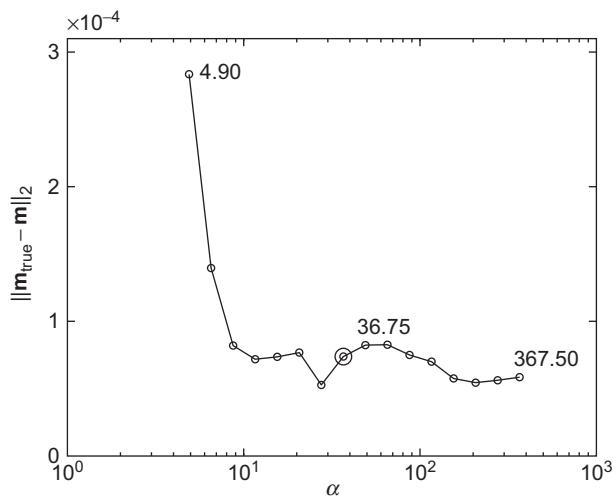
Figure 10.5 Model misfit 2-norm as a function of regularization parameter α, with preferred model highlighted.

the minimum in $\|\mathbf{m} - \mathbf{m}_{\text{true}}\|_2$. Note that the solution shown in Figure 10.4 exhibits resolution artifacts that are common in regularized solutions, such as streaking, side lobes, and amplitude under-recovery (see Example 10.3).

10.2. OCCAM'S INVERSION

Occam's inversion is a popular algorithm for nonlinear inversion introduced by Constable, Parker, and Constable [33]. The name refers to the 14th-century philosopher William of Ockham, who argued that simpler explanations should always be preferred to more complicated explanations. A similar statement occurs as rule 1 in Newton's "Rules for the Study of Natural Philosophy" [118]. This principle has become known as "Occam's razor."

Occam's inversion uses the discrepancy principle, and searches for the solution that minimizes $\|\mathbf{Lm}\|_2$ subject to the constraint $\|\mathbf{G}(\mathbf{m}) - \mathbf{d}\|_2 \leq \delta$. The algorithm is straightforward to implement, requires only the nonlinear forward model $\mathbf{G}(\mathbf{m})$ and its Jacobian, and works well in practice.

We assume that our nonlinear inverse problem has been cast in the form of (10.1). The roughening matrix $\mathbf{L}$ can be $\mathbf{I}$ to implement zeroth-order Tikhonov regularization, or it can be a finite difference approximation of a first (4.27) or second (4.28) derivative for higher-order regularization. In practice, Occam's inversion is often used on two- or three-dimensional problems where $\mathbf{L}$ is a discrete approximation of the Laplacian operator.

As usual, we will assume that the measurement errors in $\mathbf{d}$ are independent and normally distributed. For convenience, we will also assume that the system of equations $\mathbf{G}(\mathbf{m}) = \mathbf{d}$ has been scaled so that the standard deviations σ_i are equal.

The basic idea behind Occam's inversion is an iteratively applied local linearization. Given a trial model $\mathbf{m}^k$, Taylor's theorem is applied to obtain the local approximation,

$$\mathbf{G}(\mathbf{m}^k + \Delta\mathbf{m}) \approx \mathbf{G}(\mathbf{m}^k) + \mathbf{J}(\mathbf{m}^k)\Delta\mathbf{m}, \tag{10.8}$$

where $\mathbf{J}(\mathbf{m}^k)$ is the Jacobian,

$$\mathbf{J}(\mathbf{m}^k) = \begin{bmatrix} \frac{\partial G_1(\mathbf{m}^k)}{\partial m_1} & \cdots & \frac{\partial G_1(\mathbf{m}^k)}{m_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial G_m(\mathbf{m}^k)}{\partial m_1} & \cdots & \frac{\partial G_m(\mathbf{m}^k)}{\partial m_n} \end{bmatrix}. \tag{10.9}$$

Using (10.8), the regularized least squares problem (10.3) becomes

$$\min \|\mathbf{G}(\mathbf{m}^k) + \mathbf{J}(\mathbf{m}^k)\Delta\mathbf{m} - \mathbf{d}\|_2^2 + \alpha^2 \|\mathbf{L}(\mathbf{m}^k + \Delta\mathbf{m})\|_2^2, \tag{10.10}$$

where the variable is $\Delta \mathbf{m}$ and $\mathbf{m}^k$ is constant. Reformulating this as a problem in which the variable is $\mathbf{m}^{k+1} = \mathbf{m}^k + \Delta \mathbf{m}$, and letting

$$\hat{\mathbf{d}}(\mathbf{m}^k) = \mathbf{d} - \mathbf{G}(\mathbf{m}^k) + \mathbf{J}(\mathbf{m}^k)\mathbf{m}^k, \tag{10.11}$$

gives

$$\min \|\mathbf{J}(\mathbf{m}^k)(\mathbf{m}^k + \Delta \mathbf{m}) - (\mathbf{d} - \mathbf{G}(\mathbf{m}^k) + \mathbf{J}(\mathbf{m}^k)\mathbf{m}^k)\|_2^2 + \alpha^2 \|\mathbf{L}(\mathbf{m}^k + \Delta \mathbf{m})\|_2^2 \tag{10.12}$$

or

$$\min \|\mathbf{J}(\mathbf{m}^k)\mathbf{m}^{k+1} - \hat{\mathbf{d}}(\mathbf{m}^k)\|_2^2 + \alpha^2 \|\mathbf{L}(\mathbf{m}^{k+1})\|_2^2. \tag{10.13}$$

Because $\mathbf{J}(\mathbf{m}^k)$ and $\hat{\mathbf{d}}(\mathbf{m}^k)$ are constant within a given iteration, (10.13) is in the form of a regularized linear least squares problem. If the system is of full rank, the solution is given by

$$\mathbf{m}^{k+1} = \mathbf{m}^k + \Delta \mathbf{m} = \left(\mathbf{J}(\mathbf{m}^k)^T \mathbf{J}(\mathbf{m}^k) + \alpha^2 \mathbf{L}^T \mathbf{L}\right)^{-1} \mathbf{J}(\mathbf{m}^k)^T \hat{\mathbf{d}}(\mathbf{m}^k). \tag{10.14}$$

Note that this method is similar to the GN method applied to the regularized least squares problem (10.3) (see Exercise 10.1). The difference is that in Occam's inversion the parameter α is dynamically adjusted so that the solution will not exceed the allowable misfit. At each iteration we pick the largest value of α that keeps the χ^2 value of the solution from exceeding the bound on δ specified in (10.1). If this is impossible, we instead pick the value of α that minimizes the χ^2 value. At the end of the procedure, we should have a solution with $\chi^2 = \delta^2$. We can now state the algorithm.

Algorithm 10.1 Occam's Inversion

Beginning with an initial solution, $\mathbf{m}^0$, repeat the following steps to compute a sequence of solutions $\mathbf{m}^k$. Stop if and when the sequence converges to a solution with $\chi^2 = \delta^2$.
1. Calculate the Jacobian $\mathbf{J}(\mathbf{m}^k)$ and the vector $\hat{\mathbf{d}}(\mathbf{m}^k)$.
2. Calculate updated models corresponding to a range of regularization parameter values, α,

$$\mathbf{m}^{k+1} = \left(\mathbf{J}(\mathbf{m}^k)^T \mathbf{J}(\mathbf{m}^k) + \alpha^2 \mathbf{L}^T \mathbf{L}\right)^{-1} \mathbf{J}(\mathbf{m}^k)^T \hat{\mathbf{d}}(\mathbf{m}^k). \tag{10.15}$$

3. Choose the particular $\mathbf{m}^{k+1}$ with the largest value of α such that $\chi^2(\mathbf{m}^{k+1}) \leq \delta^2$. If no such value exists, then use the value of α that minimizes $\chi^2(\mathbf{m}^{k+1})$.
4. Let $k = k + 1$.

Example 10.2

We will consider the problem of estimating subsurface electrical conductivities from above-ground electromagnetic induction measurements. The instrument used in this example is the Geonics EM–38 ground conductivity meter. A description of the instrument and the mathematical model of its response can be found in [69]. The mathematical model is complicated, but we will treat it as a black box, and concentrate on the inverse problem.

Measurements are taken at heights of 0, 10, 20, 30, 40, 50, 75, 100, and 150 cm above the surface, with the coils oriented in both the vertical and horizontal orientations. There are a total of 18 observations (Table 10.1). We will assume measurement standard deviations of 0.1 mS/m.

We discretize the subsurface electrical conductivity profile into 10- and 20-cm thick layers and a semi–infinite layer below 2 m, giving us 11 parameters to estimate. The forward problem function $\mathbf{G}(\mathbf{m})$ is available to us as a subroutine. Since we do not have simple formulas for $\mathbf{G}(\mathbf{m})$, we cannot write down analytic expressions for the elements of the Jacobian. However, we can use finite difference approximations to estimate the necessary partial derivatives.

We first apply the LM method to estimate unregularized model parameters. After 50 iterations, the LM method produced the model shown in Figure 10.6. The χ^2 value for this model is 9.62 and there are $18 - 11 = 7$ degrees of freedom, so the model actually fits the data reasonably well. However, the least squares problem is very badly conditioned, with a condition number for $\mathbf{J}^T\mathbf{J}$ of approximately 2×10^{17}. Furthermore, the resulting model is unrealistic because it includes negative electrical conductivities and exhibits the high amplitudes and high-frequency oscillations that are characteristic of under-regularized solutions to inverse problems.

Table 10.1 Data for the EM-38 Example

Height (cm)	EMV (mS/m)	EMH (mS/m)
0	134.5	117.4
10	129.0	97.7
20	120.5	81.7
30	110.5	69.2
40	100.5	59.6
50	90.8	51.8
75	70.9	38.2
100	56.8	29.8
150	38.5	19.9
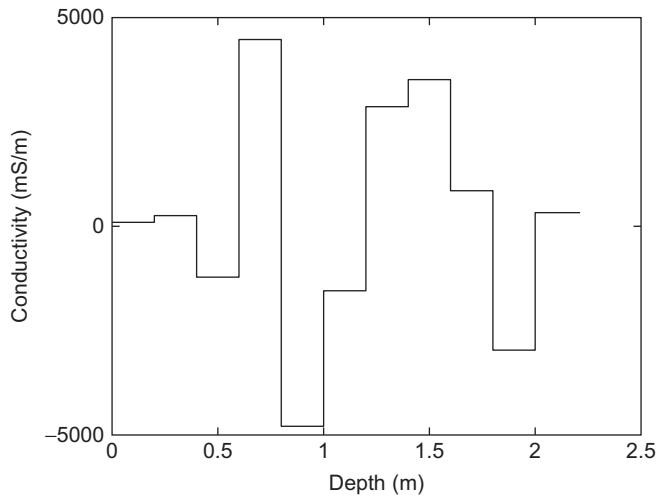
Figure 10.6 LM solution.
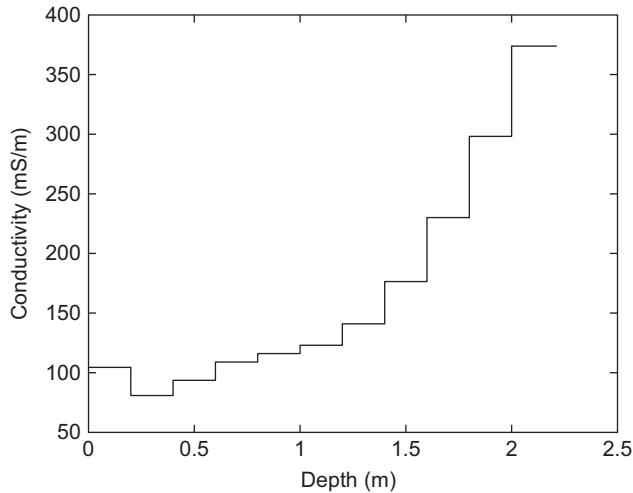
Figure 10.7 Occam's inversion solution.

We next apply Occam's inversion with second-order regularization and a discrepancy principle value of $\delta = 0.1 \cdot \sqrt{18} \approx 0.424$. The resulting model is shown in Figure 10.7, and Figure 10.8 shows the true model. The Occam's inversion method provides a fairly good reproduction of the true model.
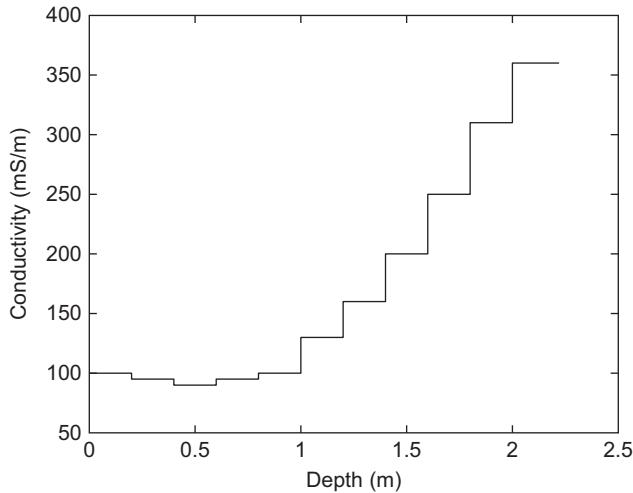
Figure 10.8 True model.

10.3. MODEL RESOLUTION IN NONLINEAR INVERSE PROBLEMS

We introduced the concept of model resolution in Chapter 3 (see (3.63)) by expressing the generalized inverse solution for a linear problem as

$$\mathbf{m} = \mathbf{R}_m \mathbf{m}_{\text{true}} = \mathbf{G}^\dagger \mathbf{G} \mathbf{m}_{\text{true}} \tag{10.16}$$

where $\mathbf{G}$ is a (forward problem) matrix that produces data from a model, $\mathbf{G}^\dagger$ is the generalized inverse matrix used to recover a model from data, and $\mathbf{m}$ and $\mathbf{m}_{\text{true}}$ are the recovered and true models, respectively. In a linear problem, the action of mapping a model to data and then back to a recovered model can be completely characterized by the model resolution matrix $\mathbf{R}_m = \mathbf{G}^\dagger \mathbf{G}$ in (10.16). In Chapter 4 we saw that this concept could easily be extended to Tikhonov regularization (4.20), by substituting the corresponding Tikhonov inverse matrix, $\mathbf{G}^\sharp$, for $\mathbf{G}^\dagger$ in (10.16).

We can recast (10.16) for nonlinear problems as

$$\mathbf{m} = G^{-1}(G(\mathbf{m}_{\text{true}})) \tag{10.17}$$

where G^{-1} and G are inverse and forward operators, respectively. However, the combined action of the forward and inverse operations is not representable as a matrix, such as $\mathbf{R}_m$ in (10.17), because the forward operator is a nonlinear function or algorithm, and the inverse operator is typically realized with an iterative method based on stepwise

linearization, such as GN or Occam's inversion. Furthermore, model resolution for a nonlinear inversion will not only depend on the physics, model parameterization, and data collection specifics, as was the case for linear problems, but may furthermore depend on the choice of starting model used in the solution algorithm, chosen convergence criteria, and possibly on the existence of multiple equally good solutions. Finally, as with linear methods, nonlinear model resolution will depend on the level and nature of the imposed regularization.

Because of these complexities, nonlinear resolution is typically analyzed using resolution tests. In seismic tomography, for example, it is common to evaluate the effects of nonideal resolution by generating noise-free synthetic data from a spike, checkerboard, or other test model using an identical source and receiver geometry as for the actual data in the problem of interest. A model is then recovered using the identical inverse methodology as was used for the actual data, and is compared to the test model to evaluate inversion artifacts. If there are specific features of the true model that will affect the resolution, such as a known strong velocity gradient with depth in seismic problems that significantly affects the curvature of the ray paths, those features should also be incorporated into the resolution test model. Because the problem is nonlinear, resolution analysis results will also potentially be dependent on test model amplitudes. A second resolution analysis strategy, sometimes referred to as a "squeeze" test, restricts the freedom of the model in regions that are suspected to be unnecessary or marginal for achieving an acceptable fit to the data. One example would be to modify the regularization constraints to strongly penalize model variations in the deeper part of a tomographic model to assess whether an acceptable data fit can still be achieved when structure is predominantly restricted to shallower depths.

●──

Example 10.3

Revisiting Example 10.1, we calculate noise-free synthetic data for a checkerboard velocity structure, using an identical starting model and ray-path end points. The checkerboard model (Figure 10.9) consists of an 8 by 8 node 2.9 km/s background model with alternating 10% variations for the 36 interior blocks. Inverting these data using the identical methodology as in Example 10.1 for the same regularization constraints and range of regularization parameters, we obtain the suite of models shown in Figure 10.10. For lower levels of regularization, the checkerboard is more apparent, while for higher levels, the horizontally varying velocity structure is substantially smoothed out. For the level of regularization chosen from the discrepancy principle for Example 10.1, $\alpha \approx 37$, we note substantial smearing that makes it difficult to discern the full checkerboard pattern, indicating that structural variations of this character and spatial scale will be difficult to recover in some parts of the model without additional

Figure 10.9 Checkerboard test model and ray paths for the cross-well tomography problem of Example 10.1.

$\alpha{=}4.9$ $\alpha{=}6.5$ $\alpha{=}8.7$ $\alpha{=}11.6$

$\alpha{=}15.5$ $\alpha{=}20.7$ $\alpha{=}27.6$ $\alpha{=}36.8$

$\alpha{=}49.08$ $\alpha{=}65.4$ $\alpha{=}87.1$ $\alpha{=}116.2$

$\alpha{=}155.0$ $\alpha{=}206.7$ $\alpha{=}275.6$ $\alpha{=}367.5$
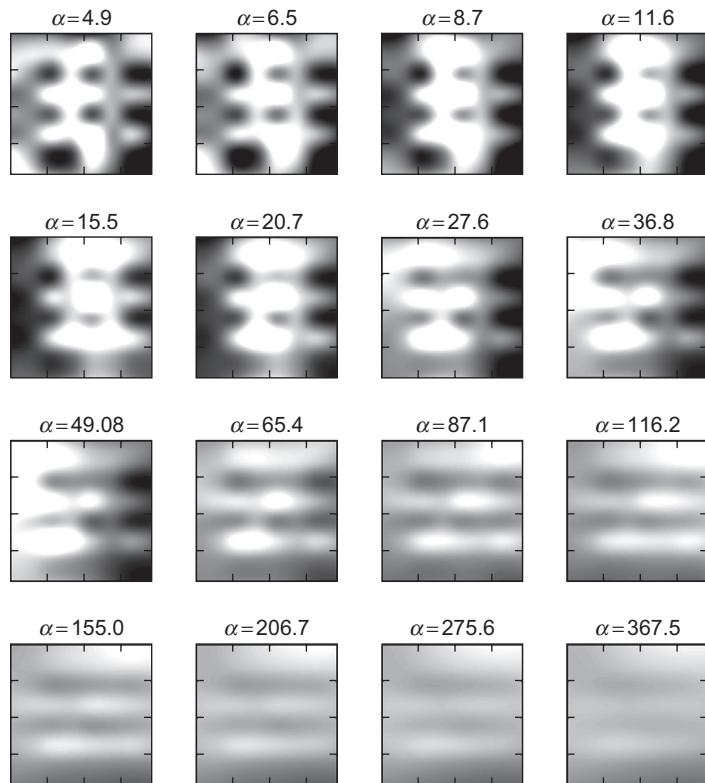
Figure 10.10 Suite of recovered models for a range of regularization parameter values (same as in Figure 10.2) for the checkerboard test model of Figure 10.9.

data. If we did not know the true model in Example 10.1, this test would helpfully show that we would expect significant horizontal smearing for the inversion result shown in Figure 10.4, and that the true model anomalies are thus not necessarily horizontally elongated.

10.4. EXERCISES

1. Show that, for a given value of α, the GN (10.7) and Occam's inversion model (10.14) update steps are mathematically equivalent.

2. Recall Example 1.5, in which we had gravity anomaly observations above a density perturbation of variable depth $m(x)$, and fixed density $\Delta\rho$. Use Occam's inversion to solve an instance of this inverse problem. Consider a gravity anomaly along a 1-km section, with observations taken every 50 m, and density perturbation of $\Delta\rho = 200$ kg/m^3 (0.2 g/cm^3). The perturbation is expected to be at a depth of roughly 200 m.

The MATLAB data file **gravprob.mat** contains a vector **x** of observation locations. Use the same coordinates for your discretization of the model. The vector **obs** contains the actual observations. Assume that the observations are accurate to about 1.0×10^{-12}.

a. Derive a formula for the elements of the Jacobian.

b. Write MATLAB routines to compute the model predictions and the Jacobian for this problem.

c. Use the supplied implementation of Occam's inversion from Example 10.2 to solve the inverse problem with second-order regularization.

d. Discuss your results. What features in the inverse solution appear to be real? What is the resolution of your solution? Were there any difficulties with local minimum points?

e. What would happen if the true density perturbation was instead at a depth of about 1000 m?

3. Apply the GN method with explicit regularization to the EM inversion problem by modifying the MATLAB code from Example 10.2. Compare your solution with the solution obtained by Occam's inversion. Which method required more computational effort?

4. Apply Occam's inversion to a cross-well, bent-ray tomography problem with identical geometry to Example 10.1. Use the example subroutine **getj.m** to forward model the travel times, calculate the Jacobian, and implement second-order regularization. Travel-time data and subroutine control parameters are contained in the MATLAB data file **benddata.mat**. Start with the uniform 2900 m/s velocity 8 by 8 node initial

velocity model in **benddata.mat**, and assume independent and normally distributed data errors with $\sigma = 0.001$ ms.

Hint: A search range of α^2 between 10 and 10^5 is appropriate for this problem. MATLAB code for generating a second–order roughening matrix that approximates a two–dimensional Laplacian operator can be found in **makerough.m**.

10.5. NOTES AND FURTHER READING

In inverse problems with a large number of parameters, the most difficult computational problem is often computing derivatives of $\mathbf{G(m)}$ with respect to the model parameters, often referred to as **Frećhet derivatives** [107]. Computation of analytic formulas is commonly impractical. Finite difference estimates require computational effort, which increases with the number of parameters and may become impractical for large problems. A promising technique for calculating Frećhet derivatives is the adjoint approach [46, 154, 165], which is likely to see increasing use in large problems as forward modeling capabilities advance. An alternative approach involves using the discretized differential equation as a set of constraints to be added to the nonlinear least squares problem [15].

For large-scale problems, it may be impractical to use direct factorization to solve the systems of equations (10.7) or (10.14) involved in computing the GN or Occam step. One approach in this case is to use an iterative method such as conjugate gradients to solve the linear systems of equations [58]. The conjugate gradient method can also be extended to minimize the nonlinear objective function directly [58, 142].

Bayesian Methods

Synopsis

Following a review of the classical least squares approach to solving inverse problems, we introduce the Bayesian approach, which treats the model as a random variable with a probability distribution that we seek to estimate. A prior distribution for the model parameters is combined with the data to produce a posterior distribution for the model parameters. In special cases, the Bayesian approach produces solutions that are equivalent to the least squares, maximum likelihood, and Tikhonov regularization solutions. Several examples of the Bayesian approach are presented. Markov Chain Monte Carlo methods for sampling from the posterior distribution are presented and demonstrated.

11.1. REVIEW OF THE CLASSICAL APPROACH

In the classical approach to parameter estimation and inverse problems with discrete data and models, we begin with a mathematical model of the form $\mathbf{Gm} = \mathbf{d}$ in the linear case or $G(\mathbf{m}) = \mathbf{d}$ in the nonlinear case. We assume that there exists a true model, $\mathbf{m}_{\text{true}}$, and a true data set, $\mathbf{d}_{\text{true}}$, such that $\mathbf{Gm}_{\text{true}} = \mathbf{d}_{\text{true}}$. We acquire an actual data set, $\mathbf{d}$, which is generally the sum of $\mathbf{d}_{\text{true}}$ and measurement noise. Our goal is to recover $\mathbf{m}_{\text{true}}$ from the noisy data.

For well-conditioned linear problems, under the assumption of independent and normally distributed data errors, the theory is well developed. In Chapter 2 it was shown that the maximum likelihood principle leads to the least squares solution, which is found by minimizing the 2-norm of the residual, $\|\mathbf{Gm} - \mathbf{d}\|_2$. Since there is noise in the data, we should expect some misfit between the data predictions of the forward model and the data, so that observed values of the square of the 2-norm of the inverse standard deviation–weighted residual, χ^2_{obs}, will not typically be zero. We saw that the χ^2 distribution can be used to test the goodness-of-fit of a least squares solution. We showed that the least squares solution, $\mathbf{m}_{L_2}$, is an unbiased estimate of $\mathbf{m}_{\text{true}}$. We were also able to compute a covariance matrix for the estimated parameters

$$\text{Cov}(\mathbf{m}_{L_2}) = (\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T\text{Cov}(\mathbf{d})\mathbf{G}(\mathbf{G}^T\mathbf{G})^{-1} \tag{11.1}$$

to compute confidence intervals for and correlations between the estimated parameters.

This approach works very well for linear regression problems where the least squares problem is well-conditioned. We found, however, that in many cases the least squares problem is not well-conditioned. In such situations, the set of solutions that adequately fits the data is large and diverse, and commonly contains physically unreasonable models.

In Chapters 4 through 8, we discussed a number of approaches to regularizing the least squares problem. These approaches pick a single "best" solution out of those that adequately fit the data, based on a preference for what sort of model features constitute a good solution. Zeroth-order Tikhonov regularization selects the model that minimizes the model 2-norm $\|\mathbf{m}\|_2$ subject to the residual norm constraint, $\|\mathbf{Gm} - \mathbf{d}\|_2 < \delta$, while higher-order Tikhonov regularization selects the model that minimizes a model seminorm $\|\mathbf{Lm}\|_2$ subject to $\|\mathbf{Gm} - \mathbf{d}\|_2 < \delta$. We introduced L_1 regularization and the related technique of total variation regularization which are now widely used in compressive sensing.

For relatively small linear problems, straightforward, insightful, and robust computation of regularized solutions can be performed with the help of singular value decomposition (SVD). For large sparse linear problems, iterative methods such as conjugate gradient least squares (CGLS) or LSQR are widely used.

For nonlinear problems, as discussed in Chapters 9 and 10, the Gauss–Newton, Levenberg-Marquardt, or Occam's inversion methods can be used to find local minima of the nonlinear least squares problem. We showed how approximate confidence intervals for the fitted parameters can be obtained by linearizing the nonlinear model around the best fit parameters. As in linear inverse problems, the nonlinear least squares problem can be badly conditioned in which case regularization may be needed to obtain a stable solution.

Unfortunately nonlinear problems may have a large number of local minimum solutions, and finding the global minimum can be difficult. Furthermore, if there are several local minimum solutions with acceptable data fits, then it may be difficult to select a single "best" solution.

How can we more generally justify selecting one solution from the set of models that adequately fit the data? One justification is Occam's razor, which is the philosophy that when we have several different hypotheses to consider, we should select the simplest. Solutions selected by regularization are in some sense the simplest models that fit the data. However, this approach is not by itself entirely satisfactory because different choices of the regularization term used in obtaining regularized solutions can result in very different models, and the specific choice of regularization may be subjective.

Recall from Chapter 4 (e.g., Example 4.3) that once we have regularized a least squares problem, we lose the ability to obtain statistically useful confidence intervals for the parameters because regularization introduces bias. In particular, the expected value of the regularized solution is not generally equal to the true model. In practice

this regularization bias is often much more significant than the effect of noise in the data. Bounds on the error in Tikhonov regularized solution error were discussed in Section 4.8, but these estimates require knowledge of the true model that is typically not available in practice.

11.2. THE BAYESIAN APPROACH

The Bayesian approach is named after Thomas Bayes, an 18th-century pioneer in prob-ability theory. The methodology is based on philosophically different ideas than we have considered so far. However, as we will see, it frequently results in similar solutions.

The most fundamental difference between the classical and Bayesian approaches is in the conceptualization of the solution. In the classical approach, there is a specific but unknown model $\mathbf{m}_{\text{true}}$ that we would like to uncover. In the Bayesian approach the model is not deterministic, but is rather a random variable, and the solution takes the form of a probability distribution for the model parameters called the **posterior distri-bution**. Once we have this probability distribution, we can use it to answer probabilistic questions about the model, such as "What is the probability that m_5 is less than 1?" In the classical approach such questions do not make sense, since the true model that we seek is not a random variable.

A second very important difference between the classical and Bayesian approaches is that the Bayesian approach naturally incorporates prior information about the solution, ranging from hard additional constraints to experience-based intuition. This information is expressed mathematically as a **prior distribution** for the model. Once data have been collected, they are combined with the prior distribution using Bayes' theorem (B.54) to produce the desired posterior distribution for the model parameters.

If no other information is available, then under the **principle of indifference**, we may alternatively pick a prior distribution where all model parameter values have equal likelihood. Such a prior distribution is said to be **uninformative**.

It should be pointed out that, in the common case where the model parameters are contained in the range $(-\infty, \infty)$, the uninformative prior is not a proper probability distribution. This is because a probability density function $f(x)$ does not exist which satisfies (B.4) so that

$$\int_{-\infty}^{\infty} f(x) \ dx = 1 \tag{11.2}$$

and $f(x)$ is constant. In practice, the use of this improper prior distribution in Bayesian methods can nevertheless be justified because the resulting posterior distribution for the model is a proper distribution.

One of the main objections to the Bayesian approach is that the method is "unscientific" because it allows the analyst to incorporate subjective judgments into the model that are not solely based on the data. Proponents of the approach rejoin that there are also subjective aspects to the classical approach embodied in the choice of regularization biases, and furthermore, that one is free to choose an uninformative prior distribution.

Here, we denote the prior distribution by $p(\mathbf{m})$, and assume that we can compute the conditional probability distribution, $f(\mathbf{d}|\mathbf{m})$, that, given a particular model, corresponding data, $\mathbf{d}$, will be observed. Given a prior distribution, we then seek the conditional (posterior) distribution of the model parameter(s) given the data. We will denote this posterior probability distribution for the model parameters by $q(\mathbf{m}|\mathbf{d})$. Bayes' theorem relates the prior and posterior distributions in a way that makes the computation of $q(\mathbf{m}|\mathbf{d})$ possible, and can be stated as follows.

Theorem 11.1

$$q(\mathbf{m}|\mathbf{d}) = \frac{f(\mathbf{d}|\mathbf{m})p(\mathbf{m})}{c} \tag{11.3}$$

where

$$c = \int_{\text{all models}} f(\mathbf{d}|\mathbf{m})p(\mathbf{m})\,d\mathbf{m}. \tag{11.4}$$

Note that the constant c in (11.3) simply normalizes the conditional distribution $q(\mathbf{m}|\mathbf{d})$ so that its integral in model space is one.

For some purposes, knowing the normalization constant, c, is not necessary. For example, we can compare two models $\hat{\mathbf{m}}$ and $\bar{\mathbf{m}}$ by computing the likelihood ratio:

$$\frac{q(\hat{\mathbf{m}}|\mathbf{d})}{q(\bar{\mathbf{m}}|\mathbf{d})} = \frac{f(\mathbf{d}|\hat{\mathbf{m}})p(\hat{\mathbf{m}})}{f(\mathbf{d}|\bar{\mathbf{m}})p(\bar{\mathbf{m}})}. \tag{11.5}$$

A very small likelihood ratio would indicate that the model $\bar{\mathbf{m}}$ is far more likely than the model $\hat{\mathbf{m}}$. Because c is not always needed, (11.3) is often written as a statement of proportionality,

$$q(\mathbf{m}|\mathbf{d}) \propto f(\mathbf{d}|\mathbf{m})p(\mathbf{m}). \tag{11.6}$$

However, there are many other situations in which knowing c in (11.3) is required. In particular, c is required to compute any posterior probabilities and to compute the expected value and variance of the posterior distribution.

It is important to re-emphasize that the probability distribution $q(\mathbf{m}|\mathbf{d})$ does not provide a single model that we can consider to be the "answer." However, in cases where we want to single out a representative model, it may be appropriate to identify

the one corresponding to the largest value of $q(\mathbf{m}|\mathbf{d})$. This model is referred to as the **maximum a posteriori (MAP)** model. Another possibility is to select the mean of the posterior distribution. In situations where the posterior distribution is normal, the MAP and posterior mean models will be identical.

In general, the computation of a posterior distribution using (11.3) can be difficult. The chief difficulty lies in evaluating the integral in (11.4). This integral often has very high dimensionality, and numerical integration techniques may thus be computationally daunting.

Fortunately, there are a number of useful special cases in which the computation of the posterior distribution is greatly simplified. One simplification occurs when the prior distribution $p(\mathbf{m})$ is uninformative, in which case (11.6) simplifies to

$$q(\mathbf{m}|\mathbf{d}) \propto f(\mathbf{d}|\mathbf{m}), \tag{11.7}$$

and the posterior distribution is precisely the likelihood function, $L(\mathbf{m}|\mathbf{d})$. Under the maximum likelihood principle, we would select the model $\mathbf{m}_{\mathrm{ML}}$ that maximizes $L(\mathbf{m}|\mathbf{d})$, which is the MAP model.

A further simplification occurs when data noise elements are independent and normally distributed with standard deviation σ. Because the data errors are independent, we can write the likelihood function as

$$L(\mathbf{m}|\mathbf{d}) = f(\mathbf{d}|\mathbf{m}) = f(d_1|\mathbf{m}) \cdot f(d_2|\mathbf{m}) \cdots f(d_n|\mathbf{m}). \tag{11.8}$$

If the data points d_i are normally distributed with expected values given by the (linear or nonlinear) operation $(G(\mathbf{m}))_i$, and each has standard deviation σ, we can write

$$f(d_i|\mathbf{m}) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{((G(\mathbf{m}))_i - d_i)^2}{2\sigma^2}} \tag{11.9}$$

and

$$L(\mathbf{m}|\mathbf{d}) = \left(\frac{1}{\sigma\sqrt{2\pi}}\right)^m e^{-\sum_{i=1}^{m} \frac{((G(\mathbf{m}))_i - d_i)^2}{2\sigma^2}}. \tag{11.10}$$

We can maximize (11.10) by maximizing the exponent or equivalently minimizing the negative of the exponent.

$$\min \sum_{i=1}^{m} \frac{((G(\mathbf{m}))_i - d_i)^2}{2\sigma^2}. \tag{11.11}$$

This is a weighted least squares problem. Thus we have shown that when we have independent and normally distributed measurement errors and we use an uninformative prior, the MAP solution is the least squares solution.

Example 11.1

Consider a very simple parameter estimation problem where we perform repeated weighings of a microscopic object to determine its mass in micrograms. The measurement errors are normally distributed with zero mean and standard deviation $\sigma = 1$ µg. Our goal is to estimate the mass of the object.

For the specified normally distributed and zero mean measurement error, the probability density function for a measurement d given m is

$$f(d|m) = \frac{1}{\sqrt{2\pi}} e^{-(m-d)^2/2}. \tag{11.12}$$

Suppose that we weigh the mass once and obtain a measurement of $d_1 = 10.3$ µg. What do we now know about m? For an uninformative prior, (11.7) gives

$$q(m|d_1 = 10.3) \propto f(10.3|m) = \frac{1}{\sqrt{2\pi}} e^{-(m-10.3)^2/2}. \tag{11.13}$$

Because (11.13) is itself a normal probability distribution, the constant of proportionality in (11.3) is 1, and the posterior distribution for the mass in micrograms (Figure 11.1) is therefore

$$q(m|d_1 = 10.3) = \frac{1}{\sqrt{2\pi}} e^{-(m-10.3)^2/2}. \tag{11.14}$$

Next, suppose that we obtain a second statistically independent measurement of $d_2 = 10.1$ µg. We can then use the distribution (11.14) estimated from the first measurement
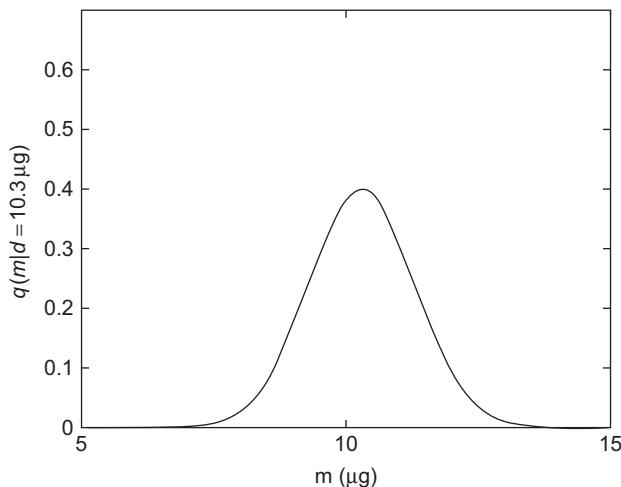
Figure 11.1 Posterior distribution $q(m|d_1 = 10.3 \, \mu g)$, uninformative prior.

as an informative prior distribution to compute a revised posterior distribution:

$$q(m|d_1 = 10.3, \ d_2 = 10.1) \propto f(d_2 = 10.1|m)q(m|d_1 = 10.3)$$

$$= \frac{1}{\sqrt{2\pi}}e^{-(m-10.1)^2/2}\frac{1}{\sqrt{2\pi}}e^{-(m-10.3)^2/2}. \tag{11.15}$$

Combining the exponents and absorbing the $1/\sqrt{2\pi}$ factors into the constant of proportionality gives

$$q(m|d_1 = 10.3, \ d_2 = 10.1) \propto e^{-((m-10.3)^2 + (m-10.1)^2)/2}. \tag{11.16}$$

Finally, we can simplify the exponent by combining terms and completing the square to obtain

$$(m-10.3)^2 + (m-10.1)^2 = 2(m-10.2)^2 + 0.02. \tag{11.17}$$

Thus,

$$q(m|d_1 = 10.3 \ \mu g, \ d_2 = 10.1) \propto e^{-(2(m-10.2)^2 + 0.02)/2}. \tag{11.18}$$

The constant $e^{-0.02/2}$ can be absorbed into the constant of proportionality, giving

$$q(m|d_1 = 10.3, \ d_2 = 10.1) \propto e^{-(10.2-m)^2}. \tag{11.19}$$

Normalizing (11.19) gives a normal posterior distribution

$$q(m|d_1 = 10.3, \ d_2 = 10.1) = \frac{1}{(1/\sqrt{2})\sqrt{2\pi}}e^{-\frac{(10.2-m)^2}{2(1/\sqrt{2})^2}}, \tag{11.20}$$

with mean 10.2 μg and $\sigma = 1/\sqrt{2}$ μg (Figure 11.2). Since we used an uninformative prior and the measurement errors were independent and normally distributed, the MAP solution is precisely the least squares solution for this problem.

It is notable in the second part of this example that we started with a normal prior distribution, accommodated normally distributed data, and obtained a normal posterior distribution (see (11.20)). In general we should not expect that the prior and posterior distributions will both be well-known distributions whose properties are well known. A prior distribution associated with a simple posterior distribution in this way is called a **conjugate prior**. There are other families of conjugate distributions for various parameter estimation problems, but in general this is unusual [51].
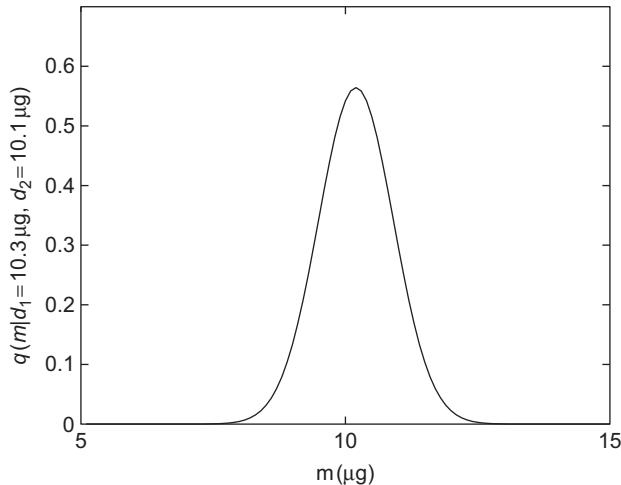
Figure 11.2 Posterior distribution $q(m|d_1 = 10.3\,\mu\text{g},\ d_2 = 10.1\,\mu\text{g})$.

11.3. THE MULTIVARIATE NORMAL CASE

The result shown in Example 11.1 in which a normal prior distribution and normally distributed data lead to a normal posterior distribution can be readily extended to problems with many model parameters. We next examine the problem of determining the posterior distribution for a linear model, multivariate normal (MVN) distributed data errors, and an MVN prior distribution.

Let $\mathbf{d}_{\text{obs}}$ be the observed data, $\mathbf{C}_D$ be the corresponding data covariance matrix, $\mathbf{m}_{\text{prior}}$ be the mean of the prior distribution, and $\mathbf{C}_M$ be the covariance matrix for the prior distribution. The prior distribution is thus, by (B.62),

$$p(\mathbf{m}) \propto e^{-\frac{1}{2}(\mathbf{m}-\mathbf{m}_{\text{prior}})^T \mathbf{C}_M^{-1}(\mathbf{m}-\mathbf{m}_{\text{prior}})}, \tag{11.21}$$

and the conditional distribution of the data, given $\mathbf{m}$, is

$$f(\mathbf{d}|\mathbf{m}) \propto e^{-\frac{1}{2}(\mathbf{Gm}-\mathbf{d})^T \mathbf{C}_D^{-1}(\mathbf{Gm}-\mathbf{d})}. \tag{11.22}$$

Thus, (11.6) gives

$$q(\mathbf{m}|\mathbf{d}) \propto e^{-\frac{1}{2}((\mathbf{Gm}-\mathbf{d})^T \mathbf{C}_D^{-1}(\mathbf{Gm}-\mathbf{d})+(\mathbf{m}-\mathbf{m}_{\text{prior}})^T \mathbf{C}_M^{-1}(\mathbf{m}-\mathbf{m}_{\text{prior}}))}. \tag{11.23}$$

Tarantola [155] showed that this can be simplified to

$$q(\mathbf{m}|\mathbf{d}) \propto e^{-\frac{1}{2}(\mathbf{m}-\mathbf{m}_{\text{MAP}})^T \mathbf{C}_{M'}^{-1}(\mathbf{m}-\mathbf{m}_{\text{MAP}})}, \tag{11.24}$$

where $\mathbf{m}_{\text{MAP}}$ is the MAP solution, and

$$\mathbf{C}_{M'} = (\mathbf{G}^T \mathbf{C}_D^{-1} \mathbf{G} + \mathbf{C}_M^{-1})^{-1}. \tag{11.25}$$

The MAP solution can be found by maximizing the exponent in (11.23), or equivalently by minimizing its negative:

$$\min \ (\mathbf{Gm} - \mathbf{d})^T \mathbf{C}_D^{-1} (\mathbf{Gm} - \mathbf{d}) + (\mathbf{m} - \mathbf{m}_{\text{prior}})^T \mathbf{C}_M^{-1} (\mathbf{m} - \mathbf{m}_{\text{prior}}). \tag{11.26}$$

The key to minimizing (11.26) is to rewrite it in terms of the matrix square roots of $\mathbf{C}_M^{-1}$ and $\mathbf{C}_D^{-1}$. Note that every covariance matrix is positive definite and has a unique positive definite matrix square root, which may be calculated in MATLAB using the **sqrtm** routine. This minimization problem can then be reformulated as

$$\min \ (\mathbf{C}_D^{-1/2}(\mathbf{Gm} - \mathbf{d}))^T (\mathbf{C}_D^{-1/2}(\mathbf{Gm} - \mathbf{d})) + \\ (\mathbf{C}_M^{-1/2}(\mathbf{m} - \mathbf{m}_{\text{prior}}))^T (\mathbf{C}_M^{-1/2}(\mathbf{m} - \mathbf{m}_{\text{prior}})), \tag{11.27}$$

or as the standard least squares problem,

$$\min \ \left\| \begin{bmatrix} \mathbf{C}_D^{-1/2}\mathbf{G} \\ \mathbf{C}_M^{-1/2} \end{bmatrix} \mathbf{m} - \begin{bmatrix} \mathbf{C}_D^{-1/2}\mathbf{d} \\ \mathbf{C}_M^{-1/2}\mathbf{m}_{\text{prior}} \end{bmatrix} \right\|_2^2. \tag{11.28}$$

Examining the right–hand terms in (11.28), note that

$$\text{Cov}(\mathbf{C}_D^{-1/2}\mathbf{d}) = \mathbf{C}_D^{-1/2}\mathbf{C}_D(\mathbf{C}_D^{-1/2})^T = \mathbf{I}. \tag{11.29}$$

The multiplication of $\mathbf{C}_D^{-1/2}$ times $\mathbf{d}$ in (11.28) can thus be conceptualized as a data transformation that both makes the resulting elements independent and normalizes the standard deviations. In the model space, multiplication by $\mathbf{C}_M^{-1/2}$ has the same effect.

An interesting limiting case is where the prior distribution provides essentially no information. Consider an MVN prior distribution with a covariance matrix $\mathbf{C}_M = \alpha^2 \mathbf{I}$, in the limit where α is extremely large. In this case, the diagonal elements of $\mathbf{C}_M^{-1}$ will be extremely small, and the posterior covariance matrix (11.25) will be well–approximated by

$$\mathbf{C}_{M'} \approx (\mathbf{G}^T \ \text{Cov}(\mathbf{d})^{-1} \ \mathbf{G})^{-1}. \tag{11.30}$$

If the data covariance matrix is $\mathbf{C}_D = \sigma^2 \mathbf{I}$, then

$$\mathbf{C}_{M'} \approx \sigma^2 (\mathbf{G}^T \mathbf{G})^{-1}, \tag{11.31}$$

which is precisely the covariance matrix for the least squares model parameters in (11.1). Furthermore, when we solve (11.28) to obtain the MAP solution, we find that it simplifies to the least squares problem of minimizing $\|\mathbf{Gm} - \mathbf{d}\|_2^2$. Thus, under the common assumption of normally distributed and independent data errors with constant variance,

a very broad prior distribution leads to a MAP solution that is the unregularized least squares solution.

It is also worthwhile to consider what happens in the special case where $\mathbf{C}_D = \sigma^2 \mathbf{I}$ and $\mathbf{C}_M = \alpha^2 \mathbf{I}$. In this case the corresponding matrix square roots are also proportional to identity matrices, and (11.28) simplifies to

$$\min \ (1/\sigma)^2 \|(\mathbf{Gm} - \mathbf{d})\|_2^2 + (1/\alpha)^2 \|\mathbf{m} - \mathbf{m}_{\text{prior}}\|_2^2, \tag{11.32}$$

which is a modified optimization problem for zeroth-order Tikhonov regularization (4.4), where the 2-norm regularization term is evaluated relative to $\mathbf{m}_{\text{prior}}$, and the equivalent Tikhonov regularization parameter is σ/α. Thus, the MAP solution obtained by using a prior with independent and normally distributed model parameters is precisely the zeroth-order Tikhonov regularized solution obtained by solving (11.32). However, this does not mean that the Bayesian approach is entirely equivalent to Tikhonov regularization, because the Bayesian solution is a probability distribution, whereas the Tikhonov solution is a single model from that distribution.

Once we have obtained the posterior distribution, it is straightforward to generate corresponding model realizations. We may wish to do this to assess likely or unlikely model features. Following the procedure outlined in Example B.10, we compute the Cholesky factorization of the posterior distribution covariance matrix,

$$\mathbf{C}_{M'} = \mathbf{R}^T \mathbf{R}, \tag{11.33}$$

and generate a random solution

$$\mathbf{m} = \mathbf{R}^T \mathbf{s} + \mathbf{m}_{\text{MAP}}, \tag{11.34}$$

where the vector $\mathbf{s}$ consists of independent and normally distributed random numbers with a zero mean and a unit standard deviation.

Example 11.2

We consider Bayesian solutions to the Shaw problem that was previously considered in Examples 3.3, 4.1, 4.2, 4.3, and 4.8.

We first use a relatively uninformative MVN prior distribution with mean 0.5, standard deviation 0.5, and zero covariances, so that $\mathbf{C}_M = 0.25 \ \mathbf{I}$. As in the previous examples, the measurement noise has standard deviation 1.0×10^{-6}, so that $\mathbf{C}_D = 1.0 \times 10^{-12} \ \mathbf{I}$. Solving (11.28) produces the $\mathbf{m}_{\text{MAP}}$ solution shown in Figure 11.3. Figure 11.4 shows this same solution with error bars. These error bars are not classical 95% confidence intervals (e.g., Figure 4.9). Rather, they are 95% probability intervals calculated from the MVN posterior distribution, so that there is 95% probability that each model parameter lies within the corresponding symmetric interval around $\mathbf{m}_{\text{MAP}}$.
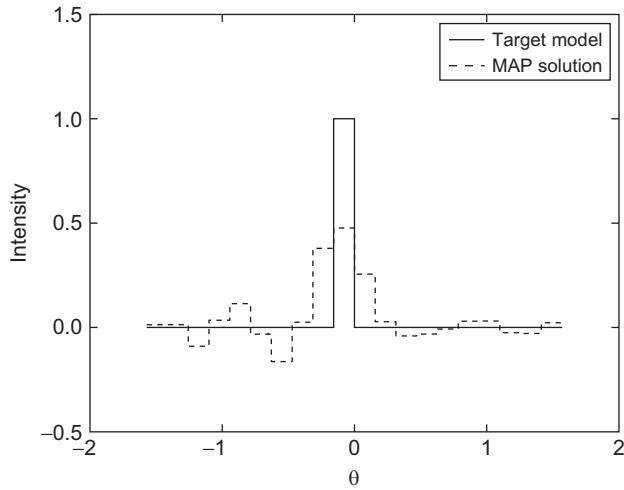
Figure 11.3 The MAP solution and the true model for the Shaw example using an MVN prior distribution with mean 0 5, standard deviation 0 5, and zero covariance.
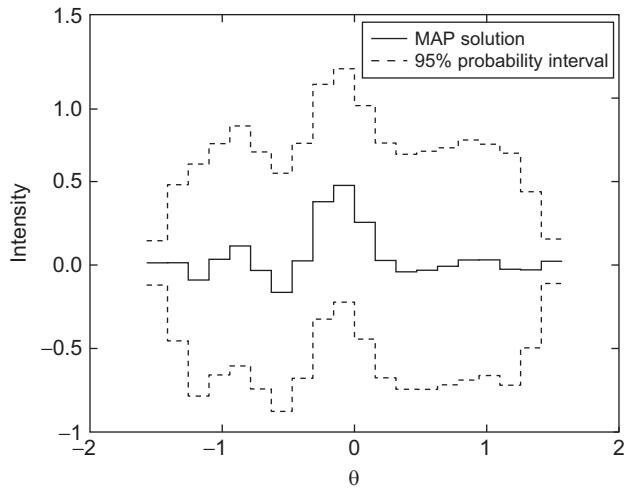
Figure 11.4 The MAP solution of Figure 11.3, with 95% probability intervals.

Figure 11.5 shows a random solution generated from the posterior distribution. This solution varies considerably from the true model, and demonstrates great uncertainty in the inverse solution, consistent with the large 95% probability intervals in Figure 11.3. The roughness of this solution realization is a consequence of the fact that the prior distribution $\mathbf{C}_M$ had zero covariances, so model realizations from the posterior distribution have no preference for smoothness.
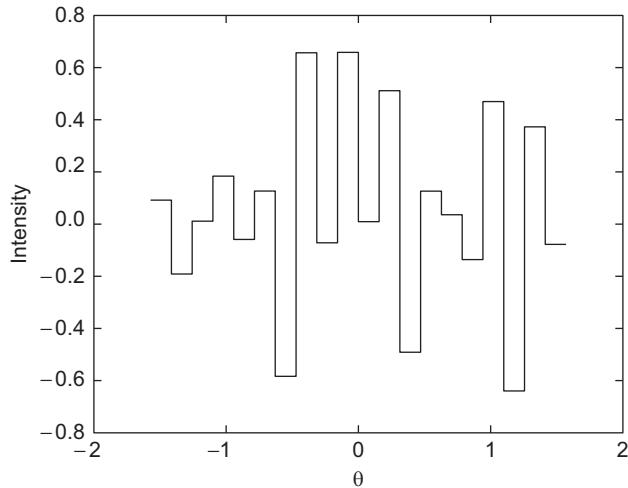
Figure 11.5 A model realization for the Shaw example using an MVN prior distribution with mean 0.5, standard deviation 0 5, and zero covariance.
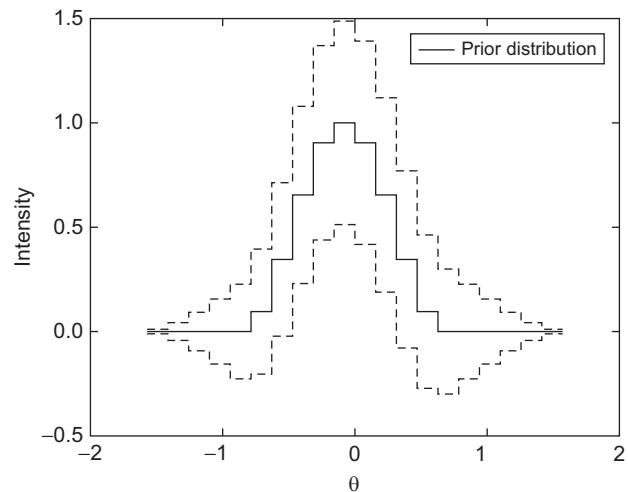
Figure 11.6 A more restrictive zero covariance prior distribution for the Shaw problem with 95% probability intervals.

Next, consider a more restrictive prior distribution. Suppose we have reason to believe that the largest amplitudes in the solution should be near the center of the model. We thus choose the bell–shaped zero-covariance prior distribution depicted in Figure 11.6. Figures 11.7 and 11.8 show the resulting MAP model and its probability
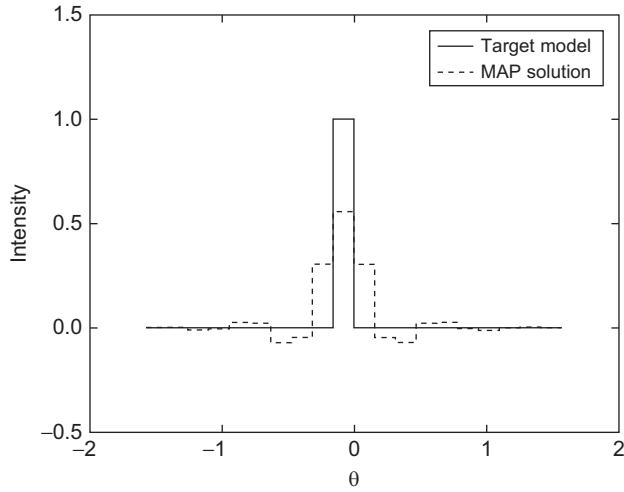
Figure 11.7 The MAP solution for the Shaw example using the prior distribution shown in Figure 11.6.
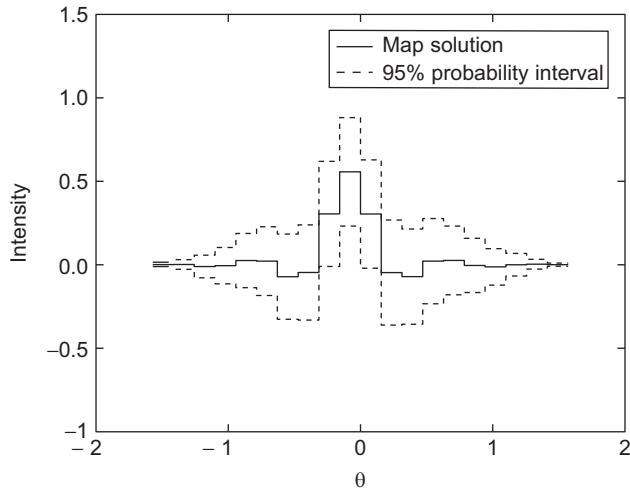
Figure 11.8 The MAP solution of Figure 11.7 with probability intervals.

intervals for this case. The solution recovery is, not surprisingly, improved by our more restrictive prior model given that the true model is highly restricted and is thus consistent with the prior distribution (Figure 11.3).

These results illustrate a principal issue with applying the Bayesian approach to poorly conditioned problems. To obtain a tight posterior distribution in such cases, we will have to make strong prior assumptions. Conversely, if such assumptions are not made, then

we cannot recover the true model features well. This situation is analogous to that of Tikhonov regularization, which must also impose strong and consistent model bias to produce "good" solutions and must also make strong model assumptions to estimate solution error bounds (Section 4.8).

In the previous example, we applied a prior that preferentially concentrated the model structure in a particular region (the center) by imposing a zero prior with small standard deviations near the model edges. Because this prior distribution had zero covariances, resulting model realizations were rough. Prior distributions can be readily designed to enforce smoothness constraints on realizations from the posterior distribution by specifying a nondiagonal prior covariance matrix. A straightforward way to accomplish this for a 1-dimensional model is to construct a correlation matrix with columns

$$\mathbf{R}_{i,.} = \text{shift}(a_j, i),\tag{11.35}$$

where a_j is the desired sequence of parameter correlations, with a zero lag correlation of 1. The shift operator shifts the sequence (truncating as necessary) so that the zero-lag (unit) maximum of the correlation sequence is centered on element i, and hence on the diagonal of $\mathbf{R}$. Suitable sequences a_i that ensure the positive definiteness of $\mathbf{R}$ can be constructed using autocorrelation (e.g., using the MATLAB **xcorr** function). Here, we use the autocorrelation of a triangle function, which produces a cubic approximation to a Gaussian function. Given the correlation matrix, $\mathbf{R}$, a corresponding prior distribution covariance matrix with uniform parameter variances, σ_M, can then be constructed as

$$\mathbf{C}_M = \sigma_M^2 \mathbf{R}.\tag{11.36}$$

Example 11.3

Consider the vertical seismic profile (VSP) problem first introduced in Example 1.3, which was solved and analyzed using Tikhonov regularization in Examples 4.4, 4.5, 4.6, and 4.7. We revisit this problem as a Bayesian problem, implementing an MVN prior with a covariance matrix of the form of (11.36). The model consists of 50 equally spaced slowness intervals along a 1000-m vertical borehole with an interval length of 20 m. Seismic travel time data are collected at 50 equally spaced depths with independent zero-mean normal errors with standard deviation of $\sigma_D = 2 \times 10^{-4}$ s, producing a data covariance matrix $\mathbf{C}_D = \sigma_D^2 \, \mathbf{I}$. We apply a prior distribution that is consistent with a seismic slowness decrease (velocity increase) with depth and has a constant gradient between
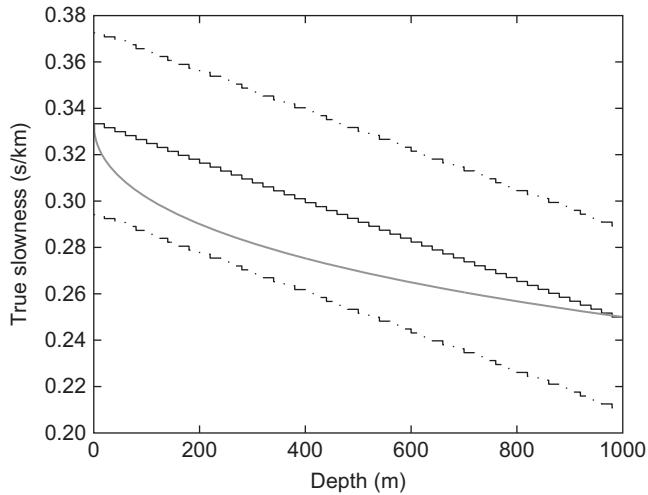
Figure 11.9 A constant-slowness gradient prior distribution and its 95% distribution intervals for the VSP problem. The true model is shown as the gray smooth curve.
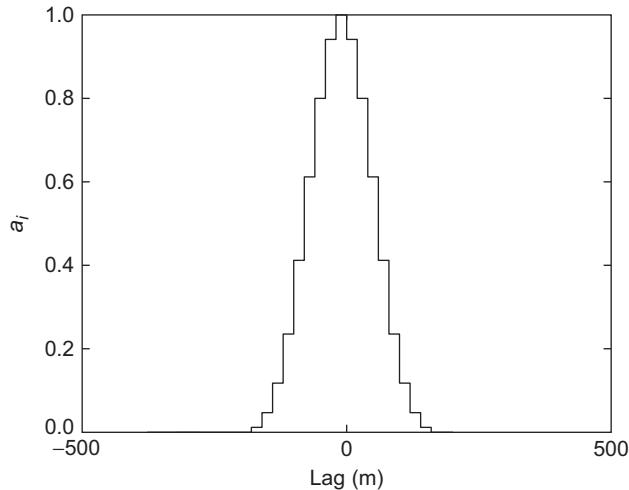
Figure 11.10 A correlation function for the prior distribution of Figure 11.9 with a $1/e$ correlation length of approximately five parameters (100 m).

known seismic slownesses from the top and the bottom of the borehole. We first impose a prior distribution standard deviation of $\sigma_M = 2 \times 10^{-5}$ s/m on all parameters to represent an estimated variability in model slownesses, and utilize a prior correlation function that falls off with a scale length of five model intervals (i.e., a correlation of $1/e$ at a model parameter lag of approximately five model parameters or 100 m). The prior
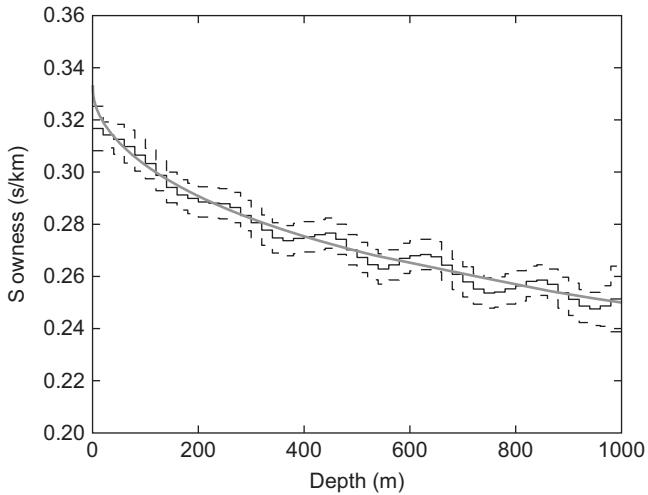
Figure 11.11 MAP model obtained from the posterior distribution, and its 95% distribution intervals, using the prior distribution described in Figures 11.9 and 11.10. The true model is shown as the gray smooth curve.
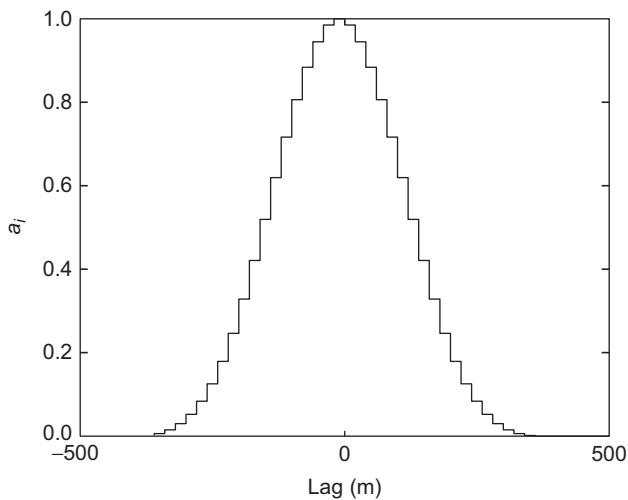
Figure 11.12 A correlation function for the prior distribution of Figure 11.9 with a $1/e$ correlation length of approximately 10 parameters (200 m).

distribution and its parameter standard deviations are shown in Figure 11.9, and the corresponding parameter correlation function is shown in Figure 11.10. The resulting posterior distribution and its standard deviations are shown in Figure 11.11. We next apply a prior with twice the correlation length (Figure 11.12). The resulting posterior distribution is shown in Figure 11.13.
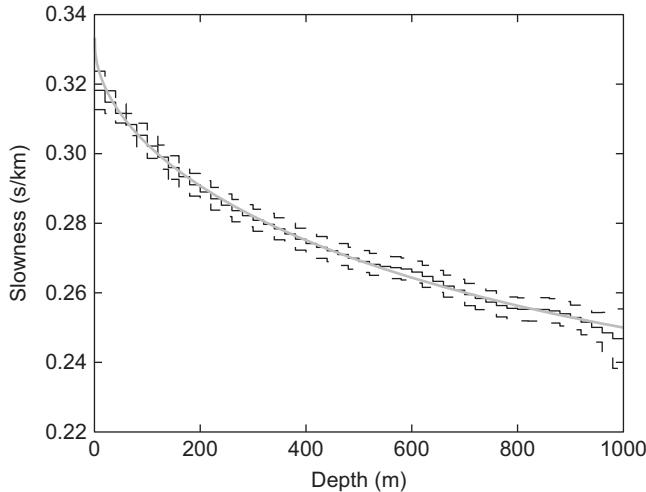
Figure 11.13 MAP model obtained from the posterior distribution, and its 95% distribution intervals, using the prior distribution depicted in Figures 11.9 and 11.12. The true model is shown as the gray smooth curve.

The approach described in this section can be extended to nonlinear problems. To find the MAP solution, solve the nonlinear least squares problem,

$$\min \quad \mathbf{G}(\mathbf{m}) - \mathbf{d})^T \mathbf{C}_D^{-1} (\mathbf{G}(\mathbf{m}) - \mathbf{d}) + (\mathbf{m} - \mathbf{m}_{\text{prior}})^T \mathbf{C}_M^{-1} (\mathbf{m} - \mathbf{m}_{\text{prior}}). \quad (11.37)$$

One can then linearize around the MAP solution to obtain the approximate posterior covariance,

$$\mathbf{C}_{M'} = (\mathbf{J}(\mathbf{m}_{\text{MAP}})^T \mathbf{C}_D^{-1} \mathbf{J}(\mathbf{m}_{\text{MAP}}) + \mathbf{C}_M^{-1})^{-1}, \quad (11.38)$$

where $\mathbf{J}(\mathbf{m})$ is the Jacobian. As with other nonlinear optimization problems, we must consider the possibility of multiple local optima. If (11.37) has multiple solutions with comparable likelihoods, then a single MAP solution and associated $\mathbf{C}_{M'}$ from (11.38) will not accurately characterize the posterior distribution.

11.4. THE MARKOV CHAIN MONTE CARLO METHOD

We next introduce the use of Markov Chain Monte Carlo (MCMC) methods to sample from a posterior distribution. Given a sufficient population of such samples, we can use them to characterize the solution of a Bayesian inverse problem. Because MCMC methods depend only on the forward model and associated likelihood calculations, they are easily applied to both linear and nonlinear problems.

A Markov chain is a sequence of random variables,

$$X^0, \ X^1, \ X^2, \ \ldots, \tag{11.39}$$

where the probability distribution of X^{n+1} depends solely on the previous value, X^n, and not on earlier values of random variables in the sequence. That is,

$$P(X^{n+1}|X^0, \ X^1, \ \ldots, \ X^n) = P(X^{n+1}|X^n). \tag{11.40}$$

The particular Markov chains considered here will be time-invariant, so that the Markov chain **transition kernel**

$$P(X^n, \ X^{n+1}) = P(X^{n+1}|X^n) \tag{11.41}$$

is independent of n.

For example, consider a one-dimensional random process in which we start at $X^0 = 0$, and then iteratively compute X^{n+1} from X^n by adding an $N(0, \ \sigma^2)$ realization to X^n. Clearly X^{n+1} depends directly on X^n, and this dependence is time-invariant. The transition kernel for this Markov chain is

$$k(x_1, \ x_2) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}(x_1 - x_2)^2/\sigma^2}. \tag{11.42}$$

In the MCMC method, we are interested in Markov chains that have limiting distributions, $q(\mathbf{x})$, such that

$$\lim_{n\to\infty} P(X^n|X^0) = q(\mathbf{x}). \tag{11.43}$$

Not all Markov chains have limiting distributions. For example, if X^n is the sum of n independent $N(0, \ \sigma^2)$ random variables, X^n has an $N(0, \ n\sigma^2)$ distribution. This does not approach a limiting probability distribution as $n \to \infty$.

For a general multivariate model, it can be shown that if $q(\mathbf{x})$ and $k(\mathbf{x}, \mathbf{y})$ satisfy the **local balance** equation

$$q(\mathbf{x})k(\mathbf{x}, \ \mathbf{y}) = q(\mathbf{y})k(\mathbf{y}, \ \mathbf{x}) \tag{11.44}$$

for all models $\mathbf{x}$ and $\mathbf{y}$, then $q(\mathbf{x})$ is the limiting distribution of the Markov chain. Equation (11.44) states that the rate of model transitions from $\mathbf{x}$ to $\mathbf{y}$ equals the rate from $\mathbf{y}$ to $\mathbf{x}$.

The **Metropolis–Hastings sampler** is an algorithm that simulates a Markov chain with a specified limiting distribution. We will apply it to produce samples from a posterior distribution $q(\mathbf{m}|\mathbf{d})$ that will tend to densely sample its higher likelihood regions. With enough such samples, we can usefully characterize the posterior distribution of a Bayesian parameter estimation or inverse problem. Since the data vector $\mathbf{d}$ is given, we will simply write $q(\mathbf{m}|\mathbf{d})$ as $q(\mathbf{m})$ throughout the following development.

We begin a Metropolis–Hastings sampler implementation by picking a **proposal distribution** $r(\mathbf{x}, \mathbf{y})$ that facilitates random steps in the posterior model space. These randomly perturbed samples will subsequently be subjected to a likelihood–based test in the Metropolis–Hastings sampler. A common choice for the proposal distribution is an MVN distribution with zero covariances and variances σ_i^2, so that

$$r(x_i, y_i) \propto e^{-\frac{1}{2}(x_i - y_i)^2 / \sigma_i^2}. \tag{11.45}$$

Although $r(\mathbf{x}, \mathbf{y})$ cannot be implemented as a transition kernel directly, because it does not satisfy (11.44), this can be remedied by introducing a factor called the **acceptance ratio**,

$$\alpha(\mathbf{x}, \mathbf{y}) = \min(1, s), \tag{11.46}$$

where

$$s = \frac{q(\mathbf{y})r(\mathbf{y}, \mathbf{x})}{q(\mathbf{x})r(\mathbf{x}, \mathbf{y})}. \tag{11.47}$$

Note that $0 \le \alpha(\mathbf{x}, \mathbf{y}) \le 1$. Also note that

$$\alpha(\mathbf{y}, \mathbf{x}) = \min\left(1, s^{-1}\right). \tag{11.48}$$

At least one of $\alpha(\mathbf{x}, \mathbf{y})$ and $\alpha(\mathbf{y}, \mathbf{x})$ will thus be equal to 1, depending on whether s is greater than or less than 1.

Now, let

$$\tilde{k}(\mathbf{x}, \mathbf{y}) = \alpha(\mathbf{x}, \mathbf{y})r(\mathbf{x}, \mathbf{y}). \tag{11.49}$$

$\alpha(\mathbf{x}, \mathbf{y}) \le 1$, so

$$\beta(x) = \int_{\text{all models}} \tilde{k}(\mathbf{x}, \mathbf{y}) \, d\mathbf{y} = \int_{\text{all models}} \alpha(\mathbf{x}, \mathbf{y})r(\mathbf{x}, \mathbf{y}) \, d\mathbf{y} \le 1. \tag{11.50}$$

Thus $\tilde{k}(\mathbf{x}, \mathbf{y})$ is not a properly normalized transition kernel. We can, however, obtain a normalized kernel by defining

$$k(\mathbf{x}, \mathbf{y}) = \tilde{k}(\mathbf{x}, \mathbf{y}) + (1 - \beta)\delta(\mathbf{x} - \mathbf{y}) \tag{11.51}$$

so that

$$\int_{\text{all models}} k(\mathbf{x}, \mathbf{y}) \, dy = \int_{\text{all models}} \alpha(\mathbf{x}, \mathbf{y})r(\mathbf{x}, \mathbf{y}) \, d\mathbf{y} + \int_{\text{all models}} (1 - \beta)\delta(\mathbf{x} - \mathbf{y}) \, d\mathbf{y}$$

$$= \beta + (1 - \beta) \tag{11.52}$$

$$= 1.$$

A simple algorithm can now be used to generate a random value $\mathbf{y}$ from $\mathbf{x}$ that satisfies (11.44).

Algorithm 11.1 Transition Kernel Evaluation

---

1. Generate a candidate $\mathbf{y}$ from $\mathbf{x}$ according to the proposal distribution $r(\mathbf{x}, \mathbf{y})$.
2. Compute $\alpha(\mathbf{x}, \mathbf{y})$.
3. With probability α, return the candidate $\mathbf{y}$.
4. With probability $1 - \alpha$, return the previous value $\mathbf{x}$.

Now, we need to show that $q(\mathbf{x})$ and $k(\mathbf{x}, \mathbf{y})$ as defined above satisfy the local balance equation (11.44). If $\mathbf{y} = \mathbf{x}$, this is obviously true. For $\mathbf{y} \neq \mathbf{x}$, we need to consider the two cases of $\alpha(\mathbf{x}, \mathbf{y}) = 1$ or $\alpha(\mathbf{y}, \mathbf{x}) = 1$. If $\alpha(\mathbf{x}, \mathbf{y}) = 1$, then

$$\alpha(\mathbf{y}, \mathbf{x}) = s^{-1} = \frac{q(\mathbf{x})r(\mathbf{x}, \mathbf{y})}{q(\mathbf{y})r(\mathbf{y}, \mathbf{x})} \tag{11.53}$$

and

$$q(\mathbf{x})k(\mathbf{x}, \mathbf{y}) = q(\mathbf{x})\alpha(\mathbf{x}, \mathbf{y})r(\mathbf{x}, \mathbf{y}) = q(\mathbf{x})r(\mathbf{x}, \mathbf{y}). \tag{11.54}$$

Also,

$$q(\mathbf{y})k(\mathbf{y}, \mathbf{x}) = q(\mathbf{y})\alpha(\mathbf{y}, \mathbf{x})r(\mathbf{y}, \mathbf{x}) = q(\mathbf{y})\frac{q(\mathbf{x})r(\mathbf{x}, \mathbf{y})}{q(\mathbf{y})r(\mathbf{y}, \mathbf{x})}r(\mathbf{y}, \mathbf{x}) = q(\mathbf{x})r(\mathbf{x}, \mathbf{y}). \tag{11.55}$$

However, $q(\mathbf{x})r(\mathbf{x}, \mathbf{y}) = q(\mathbf{x})k(\mathbf{x}, \mathbf{y})$, so

$$q(\mathbf{y})k(\mathbf{y}, \mathbf{x}) = q(\mathbf{x})k(\mathbf{x}, \mathbf{y}), \tag{11.56}$$

thus satisfying (11.44). A similar argument shows that (11.44) is satisfied for the case where $\alpha(\mathbf{y}, \mathbf{x}) = 1$.

There are several important tactics that help to simplify the method further. Because the product of q and r appears in both the numerator and denominator of s, we need only know these factors to constants of proportionality, and thus do not need to normalize q and r in individual calculations. Also note that the posterior distribution, $q(\mathbf{m})$, is proportional to the product of the prior, $p(\mathbf{m})$, and the likelihood, $f(\mathbf{d}|\mathbf{m})$ (11.3). We can thus write (11.46) as

$$\alpha(\mathbf{x}, \mathbf{y}) = \min\left(1, \frac{p(\mathbf{y})f(\mathbf{d}|\mathbf{y})r(\mathbf{y}, \mathbf{x})}{p(\mathbf{x})f(\mathbf{d}|\mathbf{x})r(\mathbf{x}, \mathbf{y})}\right). \tag{11.57}$$

If $r(\mathbf{x}, \mathbf{y})$ is a symmetric distribution, such as (11.45), then $r(\mathbf{x}, \mathbf{y}) = r(\mathbf{y}, \mathbf{x})$, and we can simplify (11.57) to

$$\alpha(\mathbf{x}, \mathbf{y}) = \min\left(1, \frac{q(\mathbf{y})}{q(\mathbf{x})}\right). \tag{11.58}$$

In computational practice, numbers in the numerator of (11.58) may be extremely small, and can thus generate floating point underflow problems. This is easily avoided by evaluating $\log \alpha(\mathbf{x}, \mathbf{y})$ instead of $\alpha(\mathbf{x}, \mathbf{y})$. We now have all the components to describe the Metropolis–Hastings sampler.

Algorithm 11.2 The Metropolis-Hastings Sampler

Given a starting model, $\mathbf{m}^0$, repeat the following steps for $j = 1, 2, \ldots$ until the posterior distribution is sufficiently sampled by the set of models $\mathbf{m}^j$.

1. Generate a candidate model $\mathbf{c}$ from the previous model, $\mathbf{m}^j$, using the proposal distribution $r(\mathbf{m}^j, \mathbf{c})$.
2. Compute $\log \alpha(\mathbf{m}^j, \mathbf{c})$.
3. Let t be a realization of a uniformly distributed random variable on $[0, 1]$.
4. If $\log t < \log \alpha(\mathbf{m}^j, \mathbf{c})$, then accept the candidate model and let $\mathbf{m}^{j+1} = \mathbf{c}$; otherwise reject the candidate model and let $\mathbf{m}^{j+1} = \mathbf{m}^j$.

Note that if $\log t$ is quite small, we will occasionally accept a new model that has a small acceptance ratio and thus move towards a model with reduced likelihood. This property of the algorithm helps overcome the problem of having the sampling become trapped near a localized likelihood maximum. The behavior of the algorithm and its ability to sample the posterior distribution fruitfully and efficiently will depend on the size of the steps taken in generating candidate models. In the case of the normal formulation for the proposal distribution (11.45), this will be controlled by the size of the σ_i. Smaller steps will result in higher acceptance rates, but the algorithm may be unacceptably slow at exploring the posterior distribution. Conversely, larger steps will result in lower acceptance ratios and will thus be less frequently accepted. Either situation may cause the algorithm to become stuck in a particular region of the posterior distribution. Often, the step size parameters are explored adaptively in multiple runs of the algorithm (which may be run as independent parallel processes on a multi–CPU computer system). Some studies, e.g., [51], suggest that the algorithm is optimally tuned when the new model acceptance rate is between approximately 20% and 50%.

11.5. ANALYZING MCMC OUTPUT

Although the limiting distribution of the Markov chain sampled by the Metropolis–Hastings algorithm is the desired posterior distribution, there are significant practical challenges in analyzing the output of an MCMC simulation.

First, successive models m^k, m^{k+1}, m^{k+2}, $\ldots$, produced by the simulation are typically strongly correlated with each other, but most statistical techniques require independent samples. In practice, this complication can be avoided by analyzing a

subset of samples that are far apart in the sample sequence. For example, if we examine $\mathbf{m}^k$, $\mathbf{m}^{k+1000}$, $\mathbf{m}^{k+2000}$, ..., it is likely that samples taken 1000 steps apart will not be highly correlated. We can verify this by plotting the successive model autocorrelations for the history of sampling over some moving window length.

Second, early Metropolis-Hastings algorithm samples will be biased by the initial model $\mathbf{m}^0$, which may not lie in a high likelihood region of the posterior distribution. This issue is commonly addressed by skipping over early samples in the chain to give time for the algorithm to "burn in," and/or by running the process with a variety of starting models, either sequentially or in parallel on multiple CPUs. For example, if it is determined that samples spaced 1000 steps apart are effectively uncorrelated, then it might be reasonable to let the Metropolis-Hastings sampler establish itself for 10,000 steps before beginning to collect samples.

Once we are confident that the procedure has produced a large enough collection of effectively independent samples, we can use the results to characterize the posterior distribution. For a suitably large sample, the MAP solution can be estimated as the retrieved posterior distribution sample that has the greatest likelihood. The posterior distribution may be approximately multivariate normal, which can be established by examining model histograms and quantile-quantile (Q-Q) plots. In this case we can readily construct probability intervals for describing the posterior distribution from the sample mean and covariance using normal assumptions. However, if the distribution is distinctly non-normal, it will be more difficult to produce a simple summary of its character, particularly for very high dimensional models. A common approach is to produce and evaluate scatter plots and/or histograms that display key features of the posterior distribution and to use counting statistics to establish probability intervals.

Example 11.4

Reconsidering the ill-posed nonlinear parameter estimation problem of Example 9.2, we apply the Metropolis-Hastings algorithm to the problem of fitting four parameters, m_i, to the nonlinear function

$$d_i = (G(\mathbf{m}))_i = m_1 e^{m_2 x_i} + m_3 x_i e^{m_4 x_i} \tag{11.59}$$

given a set of observations, d_i, with specified independent normally distributed data noise, specified by corresponding standard deviations σ_i. As in Example 9.2, the true model parameters are $m_1 = 1.0$, $m_2 = -0.5$, $m_3 = 1.0$, and $m_4 = -0.75$; data are produced at 25 equally spaced points, x_i, on the interval $[1, 7]$; and $N(0, 0.01^2)$ independent noise is added to each data element. The likelihood function is specified by

$$f(\mathbf{d}|\mathbf{m}) \propto \prod_{i=1}^{m} e^{-\frac{1}{2}(d_i - G(\mathbf{m})_i)^2/\sigma_i^2}, \tag{11.60}$$

which, after taking the natural logarithm, is

$$\log(f(\mathbf{d}|\mathbf{m})) = -\frac{1}{2}\sum_{i=1}^{m}(d_i - G(\mathbf{m})_i)^2/\sigma_i^2 + C, \qquad (11.61)$$

where C is the logarithm of the constant of proportionality in (11.60). We sample the posterior distribution using 410,000 steps, and produce a low-correlation set of 400 samples by allowing a 10,000 sample burn in and selecting subsequent samples spaced 1000 steps apart by down–sampling the full 410,000 length sequence. The procedure was initiated using a random model selected from a 4-dimensional uniform distribution
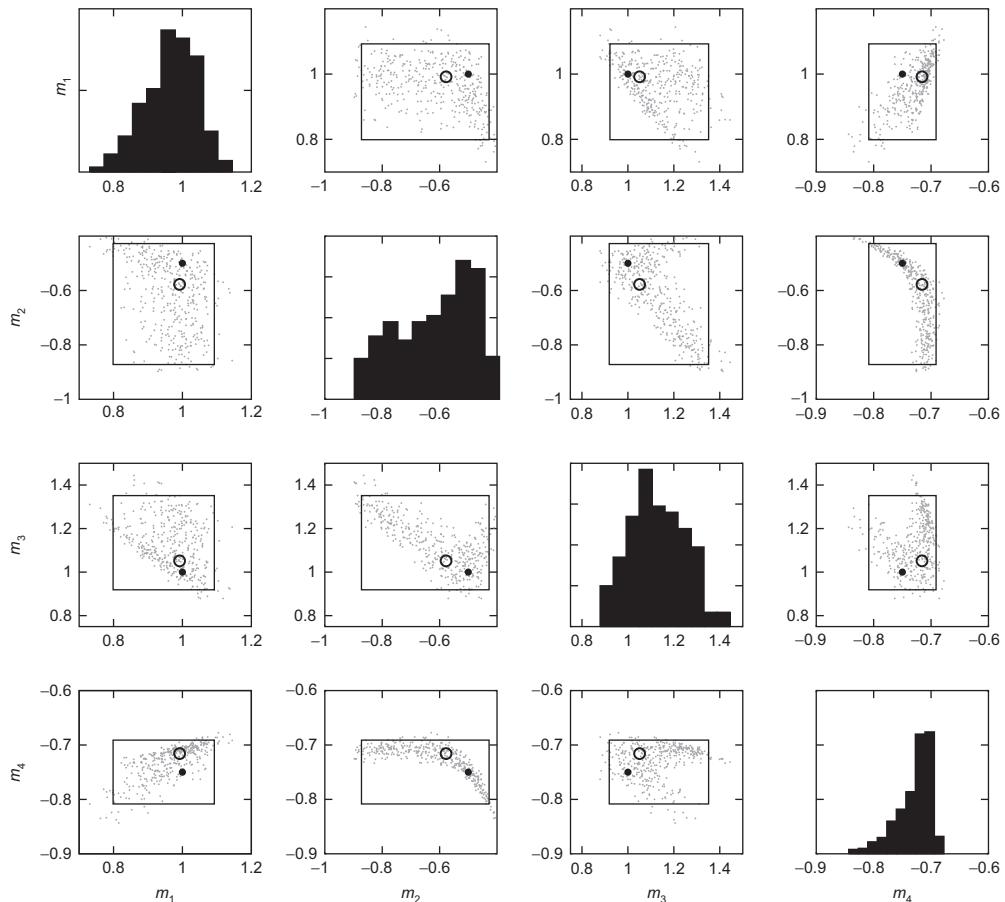
Figure 11.14 Sampled posterior distribution for Example 11.3. The true model is shown as the large black dot, and the MAP model estimated from the maximum likelihood posterior distribution sample is indicated by the open circle. 400 retrieved samples of the posterior distribution (every 1000th calcu-lated sample from the MCMC output) are shown as gray dots. Ninety-five percent probability intervals estimated from the MCMC posterior distribution samples are shown by boxes.

bounded by $[-1, 1]$ in each parameter direction and applying a uniform prior for the region $m_1 = [0, 2]$, $m_2 = [-1, 0]$, $m_3 = [0, 2]$, and $m_4 = [-1, 0]$. In each of the MCMC steps, we apply independent normally distributed random perturbations with standard deviations of 0.005 in each of the four model parameter directions, accepting or rejecting the corresponding candidate models according to the Metropolis–Hastings algorithm. The procedure produced an acceptance rate of approximately 39.5%, which is in the nominally acceptable range of not being too large or too small [51].

Figure 11.14 shows scatter plots and histograms of the resulting sampled posterior distribution, along with the corresponding MAP solution and 95% probability intervals for each parameter direction, which well enclose the true solution. Figure 11.15 shows
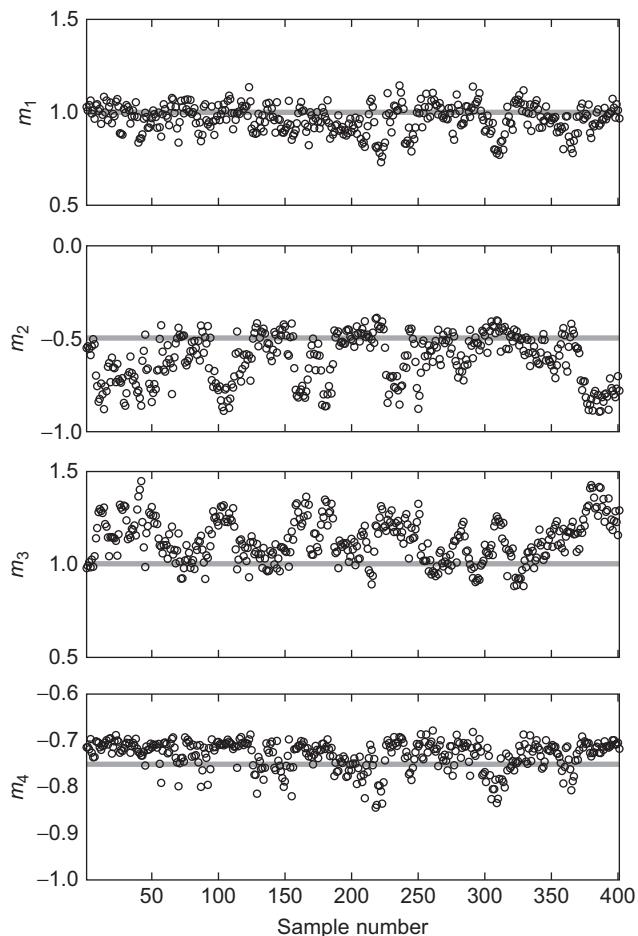
Figure 11.15 Time history of the posterior samples plotted in Figure 11.14. True parameters are shown as gray lines.
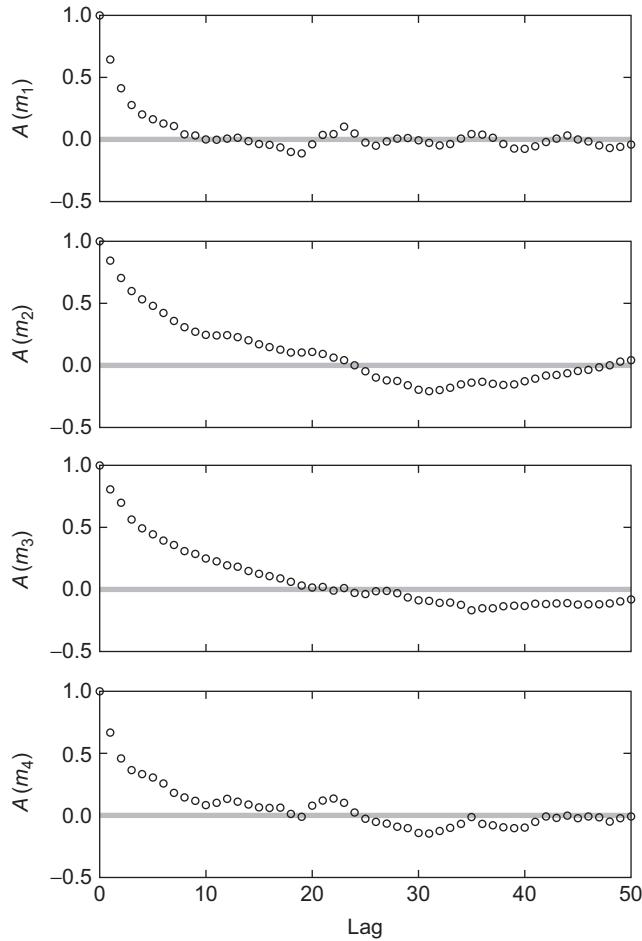
Figure 11.16 Twenty-point autocorrelations for posterior distribution parameters shown in Figure 11.15.

the time history of the corresponding 400 samples. The prominent quasilinear scattering trends in the output between m_2 and m_3 are indicative of a high degree of anticorrelation between some parameter pairs. This anticorrelation is also apparent in the sequence of parameter samples in Figure 11.15, for example in the anticorrelated trends of m_2 and m_3. An examination of the parameter autocorrelations for the samples shown in Figure 11.14 (Figure 11.16) shows that individual parameter estimates are effectively decorrelated after 20 or fewer steps. Figure 11.17 shows a histogram of the natural log of likelihood values for the posterior samples. The multimodal nature of this histogram is reflective of the "bumpy" likelihood function surface for this ill–posed problem with random data noise, a situation that also resulted in multiple solutions when we applied
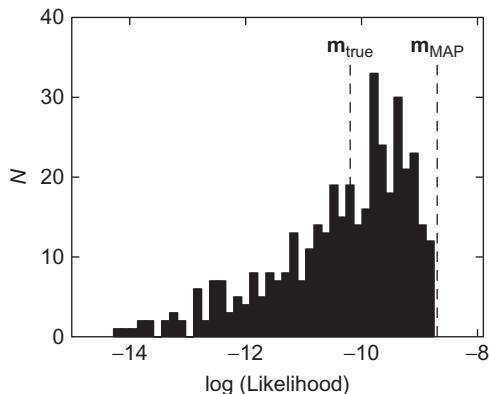
Figure 11.17 Natural logarithm of the likelihood function for the 400 posterior distribution samples shown in Figures 11.14 and 11.15. The value for the true model is approximately $-10\,2$, and the value for the MAP model is approximately -8.7.

classical nonlinear solution methods in Example 9.2. However, the MCMC method here provided a sufficiently thorough search of the model space to allow us to discriminate the MAP solution from solutions corresponding to subsidiary likelihood maxima.

11.6. EXERCISES

1. Reanalyze the data in Example 11.1 using a prior distribution that is uniform on the interval $[9, 11]$. Compute the posterior distribution after the first measurement of $10.3\ \mu g$ and after the second measurement of $10.1\ \mu g$. What is the posterior mean?
2. Consider the estimation problem $\mathbf{d} = \mathbf{m}$ (i.e., where $\mathbf{G} = \mathbf{I}$) in two dimensions. The data, $\mathbf{d} = [5\ 15]^T$, have identical and independent normal errors with standard deviations of $\sqrt{2}$. Apply a zero-mean MVN prior characterized by a covariance matrix where m_1 and m_2 have a correlation coefficient of 0.9, and equal standard deviations of $\sqrt{5}$. Calculate the MAP model, and compute and plot the 50%, 90%, and 95% contours of the MVN distributions $\mathbf{d}$, $\mathbf{m}_{\text{prior}}$, and the posterior model.
3. In writing (11.28) we made use of the matrix square root.
   a. Suppose that $\mathbf{A}$ is a symmetric and positive definite matrix. Using the SVD, find an explicit formula for the matrix square root. Your square root should itself be a symmetric and positive definite matrix.
   b. Show that instead of using the matrix square roots of $\mathbf{C}_D^{-1}$ and $\mathbf{C}_M^{-1}$, we could have used the Cholesky factorizations of $\mathbf{C}_D^{-1}$ and $\mathbf{C}_M^{-1}$ in formulating the least squares problem.
4. Consider the following coin tossing experiment. We repeatedly toss a coin, and each time record whether it comes up heads (0) or tails (1). The bias b of the coin is the

probability that it comes up heads. We have reason to believe that this is not a fair coin, so we will not assume that $b = 1/2$. Instead, we will begin with a uniform prior distribution $p(b) = 1$, for $0 \le b \le 1$.

a. What is $f(d|b)$? Note that the only possible data are 0 and 1, so this distribution will involve delta functions at $d = 0$, and $d = 1$.

b. Suppose that on our first flip, the coin comes up heads. Compute the posterior distribution $q(b|d_1 = 0)$.

c. The second, third, fourth, and fifth flips are 1, 1, 1, and 1. Find the posterior distribution $q(b|d_1 = 0, d_2 = 1, d_3 = 1, d_4 = 1, d_5 = 1)$. Plot the posterior distribution.

d. What is your MAP estimate of the bias?

e. Now, suppose that you initially felt that the coin was at least close to fair, with

$$p(b) \propto e^{-10(b-0.5)^2} \quad 0 \le b \le 1. \tag{11.62}$$

Repeat the analysis of the five coin flips described above.

5. Apply the Bayesian method to Exercise 2 in Chapter 4. Select what you consider to be a reasonable prior. How sensitive is your solution to the prior mean and covariance?

6. Apply the Bayesian method to Exercise 9.3. Assume an MVN prior distribution for α and n, with α and n independent. The prior for α should have a mean of 0.01 and a standard deviation of 0.005, and the prior for n should have a mean of 5 and a standard deviation of 3. Compare your solution with the solution that you obtained to Exercise 9.3. How sensitive is your solution to the choice of the prior?

7. Repeat Exercise 11.6, using MCMC to estimate the posterior distribution. Compare your solution to the solution that you obtained for Exercise 11.6.

8. Apply the Metropolis-Hastings sampler to produce a sampled posterior distribution for the nonlinear parameter estimation problem of Example 9.1. Use a prior distribution that is uniform on $S = [0, 0.01]$ and $T = [0, 2]$, a zero covariance MVN proposal distribution, and a starting model of $(S, T) = (5 \times 10^{-3}, 1.0)$. Generate 200,000 samples using a 10,000-sample burn-in and explore independent step sizes for the two parameters in your proposal distribution to obtain a Metropolis-Hastings sampler acceptance rate between 10% and 50%. Extract every 1000th sample for analysis and establish that these 191 samples are not highly dependent by examining sample autocorrelation functions. Examine the sampled distribution to obtain the MAP model and empirical 95% probability intervals on S and T. Apply a Q-Q plot and assess the multivariate normality of the sampled posterior distribution and compare normal assumption and empirical estimates of the 95% probability intervals.

9. Apply the Metropolis-Hastings sampler to produce a sampled posterior distribution for the nonlinear inverse problem for gravity observations above a buried density

perturbation with an unknown variable depth $m(x)$ and a fixed density perturbation $\Delta\rho$, as described in Exercise 10.2. Your prior should be selected to favor smooth (specified by nonzero parameter correlations) models.

11.7. NOTES AND FURTHER READING

The arguments for and against the use of Bayesian methods in statistics and inverse problems have raged for decades. Some classical references that provide context for these arguments include [35, 42, 80, 81, 136]. Sivia and Skilling's book [143] is a good general introduction to Bayesian ideas for scientists and engineers. The book by Calvetti and Somersalo introduces Bayesian methods for inverse problems including Markov Chain Monte Carlo sampling and includes MATLAB examples [22]. The book by Kaipio and Somersalo provides a more detailed theoretical treatment of Bayesian methods for inverse problems and includes some interesting case studies [83]. An early paper by Tarantola and Valette on the application of the Bayesian approach was quite influential [156], and Tarantola's book is the standard reference work on Bayesian methods for inverse problems [155]. The book by Rodgers [133] focuses on application of the Bayesian approach to problems in atmospheric sounding. The paper by Gouveia and Scales [56] discusses the relative advantages and disadvantages of Bayesian and classical methods for inverse problems. The draft textbook by Scales and Smith [137] takes a Bayesian approach to inverse problems.

In many cases the solution to an inverse problem will be used in making a decision, with measurable consequences for making the "wrong" decision. Statistical decision theory can be helpful in determining the optimal decision. The paper by Evans and Stark provides a good introduction to the application of statistical decision theory to inverse problems [48].

Epilogue

The theme of this book has been obtaining and analyzing solutions to discretized parameter estimation problems using classical and Bayesian approaches. We have discussed computational procedures for both linear and nonlinear problems. Classical procedures produce estimates of the parameters and their associated uncertainties. In Bayesian methods, the model is a random variable, and the solution is its probability distribution.

However, there are critical issues that arise in solving these problems. When we discretize a continuous problem, the choice of the discretization scheme, basis functions, and grid spacing can have large effects on the behavior of the discretized problem and its solutions, and these effects will not be reflected in the statistical analysis of the solution of the discretized problem. The discretization errors in the solution could potentially be far larger than any explicitly computed statistical uncertainty. Thus it is important to ensure that the discretization provides an adequate approximation to the continuous problem. If no formal analysis is performed, it is at least desirable to see whether varying the discretization has a significant effect on the solutions obtained.

For well-conditioned problems with normally distributed measurement errors, we can use the classical least squares approach. This results in unbiased parameter estimates and associated confidence intervals. For ill-conditioned problems, and for problems where we have good reason to prefer a specific bias in the character of the solution, Tikhonov regularization can be applied to obtain a solution. However, regularization introduces bias into the solution, and it is impossible to even bound this bias without making additional assumptions about the model.

Although Bayesian approaches are also applicable to well-conditioned situations, they are particularly interesting in the context of ill-conditioned problems. By selecting a prior distribution, we make explicit assumptions about the model. The resulting posterior distribution is not affected by regularization bias. In the multivariate normal case for linear problems the Bayesian approach is no more difficult computationally than the least squares approach.

Various efforts have been made to avoid the use of subjective priors in the Bayesian approach. Principles such as maximum entropy can be used to derive prior distributions that have been claimed to be, in some sense, "objective." However, we do not find these arguments completely convincing, and in many cases the choice of prior has a

similar effect on the choice of the regularization operations in Tikhonov and other classical approaches. Markov Chain Monte Carlo methods present a computationally intensive methodology to sample and characterize the Bayesian posterior distribution using forward model–based likelihood calculations.

Both the classical and Bayesian approaches can be extended to nonlinear inverse problems. The computation of the estimated model parameters becomes substantially more complex in that we must solve nonlinear optimization problems which may have multiple locally optimal solutions. In both standard approaches, the statistical analysis is typically performed approximately by analyzing a linearization of the nonlinear model around the estimated parameters. However, the validity of this approach will depend on the data uncertainties and the nonlinearity of the problem. The Bayesian approach can in theory be applied when measurement errors are not normally distributed. In practice, however, the associated analytical computations can be difficult. Markov Chain Monte Carlo methods provide a very general approach for characterizing the posterior distribution that avoids the difficulties of computing it in analytical form.

Review of Linear Algebra

Synopsis

A summary of essential concepts, definitions, and theorems in linear algebra used throughout this book.

A.1. SYSTEMS OF LINEAR EQUATIONS

Recall that a system of linear equations can be solved by the process of **Gaussian elimination**.

Example A.1

Consider the following system of equations:

$$\begin{aligned}
x_1 + 2x_2 + 3x_3 &= 14 \\
x_1 + 2x_2 + 2x_3 &= 11 \\
x_1 + 3x_2 + 4x_3 &= 19.
\end{aligned} \tag{A.1}$$

We eliminate x_1 from the second and third equations by subtracting the first equation from the second and third equations to obtain

$$\begin{aligned}
x_1 + 2x_2 + 3x_3 &= 14 \\
-x_3 &= -3 \\
x_2 + x_3 &= 5.
\end{aligned} \tag{A.2}$$

We would like x_2 to appear in the second equation, so we interchange the second and third equations:

$$\begin{aligned}
x_1 + 2x_2 + 3x_3 &= 14 \\
x_2 + x_3 &= 5 \\
-x_3 &= -3.
\end{aligned} \tag{A.3}$$

Next, we eliminate x_2 from the first equation by subtracting two times the second equation from the first equation:

$$
\begin{aligned}
x_1 + x_3 &= \quad 4 \\
x_2 + x_3 &= \quad 5 \\
-x_3 &= -3.
\end{aligned}
\tag{A.4}
$$

We then multiply the third equation by -1 to get an equation for x_3:

$$
\begin{aligned}
x_1 + x_3 &= 4 \\
x_2 + x_3 &= 5 \\
x_3 &= 3.
\end{aligned}
\tag{A.5}
$$

Finally, we eliminate x_3 from the first two equations to obtain a solution to the original system of equations:

$$
\begin{aligned}
x_1 &= 1 \\
x_2 &= 2 \\
x_3 &= 3.
\end{aligned}
\tag{A.6}
$$

Geometrically the constraints specified by the three equations of (A.1) describe three planes that, in this case, intersect in a single point, $x_1 = 1$, $x_2 = 2$, $x_3 = 3$.

In solving (A.1), we used three **elementary row operations**: adding a multiple of one equation to another equation, multiplying an equation by a nonzero constant, and swapping two equations. This process can be extended to solve systems of equations with an arbitrary number of variables.

In performing the elimination process, the actual names of the variables are insignificant. We could have renamed the variables in the above example to a, b, and c without changing the solution in any significant way. Because the actual names of the variables are insignificant, we can save space by writing down the significant coefficients from the system of equations in matrix form as an **augmented matrix**. The augmented matrix form is also useful in solving a system of equations in computer algorithms, where the elements of the augmented matrix are stored in an array.

In augmented matrix form (A.1) becomes

$$
\begin{bmatrix}
1 & 2 & 3 & \big| & 14 \\
1 & 2 & 2 & \big| & 11 \\
1 & 3 & 4 & \big| & 19
\end{bmatrix}.
\tag{A.7}
$$

In augmented notation, the elementary row operations become adding a multiple of one row to another row, multiplying a row by a nonzero constant, and interchanging

two rows. The Gaussian elimination process is essentially identical to the process used in Example A.1, with the final version of the augmented matrix given by

$$
\left[\begin{array}{ccc|c}
1 & 0 & 0 & 1 \\
0 & 1 & 0 & 2 \\
0 & 0 & 1 & 3
\end{array}\right].
\tag{A.8}
$$

Definition A.1 A matrix is said to be in **reduced row echelon form (RREF)** if it has the following properties:

1. The first nonzero element in each row is a 1. The first nonzero row elements of the matrix are called **pivot elements**. A column in which a pivot element appears is called a **pivot column**.
2. Except for the pivot element, all elements in pivot columns are 0s.
3. Each pivot element is to the right of the pivot elements in previous rows.
4. Any rows consisting entirely of 0s are at the bottom of the matrix.

In solving a system of equations in augmented matrix form, we apply elementary row operations to reduce the augmented matrix to RREF and then convert back to conventional notation to read off the solutions. The process of transforming a matrix into RREF can easily be automated. In MATLAB, this is done by the **rref** command.

It can be shown that any linear system of equations has either no solutions, exactly one solution, or infinitely many solutions [95]. In a system of two dimensions, for example, lines represented by the equations can fail to intersect (no solution), intersect at a point (one solution), or intersect in a line (many solutions). The following example shows how to determine the number of solutions from the RREF of the augmented matrix.

Example A.2

Consider a system of two equations in three variables that has many solutions:

$$
\begin{aligned}
x_1 + x_2 + x_3 &= 0 \\
x_1 + 2x_2 + 2x_3 &= 0.
\end{aligned}
\tag{A.9}
$$

We put this system of equations into augmented matrix form and then find the RREF, which is

$$
\left[\begin{array}{ccc|c}
1 & 0 & 0 & 0 \\
0 & 1 & 1 & 0
\end{array}\right].
\tag{A.10}
$$

We can translate this back into equation form as

$$
\begin{aligned}
x_1 &= 0 \\
x_1 + x_3 &= 0.
\end{aligned}
\tag{A.11}
$$

Clearly, x_1 must be 0 in any solution to the system of equations. However, x_2 and x_3 are not fixed. We could treat x_3 as a **free variable** and allow it to take on any value. Whatever value x_3 takes on, x_2 must be equal to $-x_3$. Geometrically, this system of equations describes the intersection of two planes, where the intersection consists of points on the line $x_2 = -x_3$ in the $x_1 = 0$ plane.

A linear system of equations may have more equation constraints than variables, in which case the system of equations is **over-determined**. Although over-determined systems often have no solutions, it is possible for an over-determined system of equations to have either many solutions or exactly one solution.

Conversely, a system of equations with fewer equations than variables is **under-determined**. Although in many cases under-determined systems of equations have infinitely many solutions, it is also possible for such systems to have no solutions.

A system of equations with all 0s on the right-hand side is **homogeneous**. Every homogeneous system of equations has at least one solution, the trivial solution in which all of the variables are 0s. A system of equations with a nonzero right-hand side is **nonhomogeneous**.

A.2. MATRIX AND VECTOR ALGEBRA

As we have seen in the previous section, a matrix is a table of numbers laid out in rows and columns. A **vector** is a matrix consisting of a single column or row of numbers (vectors in this text are typically expressed as column vectors). In general, matrices and vectors may contain complex numbers as well as real numbers. With the exception of Chapter 8, all of the vectors and matrices in this book are real.

There are several important notational conventions used here for matrices and vectors. Boldface capital letters such as $\mathbf{A}$, $\mathbf{B}$, . . . are used to denote matrices. Boldface lower-case letters such as $\mathbf{x}$, $\mathbf{y}$, . . . are used to denote vectors. Lower-case letters or Greek letters such as m, n, α, β, . . . will be used to denote scalars.

At times we will need to refer to specific parts of a matrix. The notation $A_{i,j}$ denotes the element of the matrix $\mathbf{A}$ in row i and column j. We denote the *j*th element of the vector $\mathbf{x}$ by x_j. The notation $\mathbf{A}_{.,j}$ is used to refer to column j of the matrix $\mathbf{A}$, while $\mathbf{A}_{i,.}$ refers to row i of $\mathbf{A}$.

We can also construct larger matrices from smaller matrices. The notation $\mathbf{A} = [\mathbf{B} \ \mathbf{C}]$ means that the matrix $\mathbf{A}$ is composed of the matrices $\mathbf{B}$ and $\mathbf{C}$, with matrix $\mathbf{C}$ to the right of $\mathbf{B}$.

If $\mathbf{A}$ and $\mathbf{B}$ are two matrices of the *same size*, we can add them by simply adding corresponding elements. Similarly, we can subtract $\mathbf{B}$ from $\mathbf{A}$ by subtracting the corresponding elements of $\mathbf{B}$ from those of $\mathbf{A}$. We can multiply a scalar times a matrix by

multiplying the scalar times each matrix element. Because vectors are just n by 1 matrices, we can perform the same arithmetic operations on vectors. A **zero matrix**, **0**, is a matrix composed of all zero elements. A zero matrix plays the same role in matrix algebra as the scalar 0, with

$$\mathbf{A} + \mathbf{0} = \mathbf{A} \tag{A.12}$$

$$= \mathbf{0} + \mathbf{A}. \tag{A.13}$$

Using vector notation, we can write a linear system of equations in **vector form**.

Example A.3

Recall the system of equations (A.9),

$$\begin{aligned} x_1 + x_2 + x_3 &= 0 \\ x_1 + 2x_2 + 2x_3 &= 0, \end{aligned} \tag{A.14}$$

from Example A.2. We can write this in vector form as

$$x_1 \begin{bmatrix} 1 \\ 1 \end{bmatrix} + x_2 \begin{bmatrix} 1 \\ 2 \end{bmatrix} + x_3 \begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \tag{A.15}$$

The expression on the left-hand side of (A.15) where vectors are multiplied by scalars and the results are summed together is called a **linear combination**.

If **A** is an m by n matrix, and **x** is an n element vector, we can multiply **A** times **x**, where the product is defined by

$$\mathbf{A}\mathbf{x} = x_1 \mathbf{A}_{\cdot,1} + x_2 \mathbf{A}_{\cdot,2} + \cdots + x_n \mathbf{A}_{\cdot,n}. \tag{A.16}$$

Example A.4

Given

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \tag{A.17}$$

and

$$\mathbf{x} = \begin{bmatrix} 1 \\ 0 \\ 2 \end{bmatrix}, \tag{A.18}$$

then

$$\mathbf{Ax} = 1 \begin{bmatrix} 1 \\ 4 \end{bmatrix} + 0 \begin{bmatrix} 2 \\ 5 \end{bmatrix} + 2 \begin{bmatrix} 3 \\ 6 \end{bmatrix} = \begin{bmatrix} 7 \\ 16 \end{bmatrix}. \tag{A.19}$$

The formula (A.16) for $\mathbf{Ax}$ is a linear combination much like the one that occurred in the vector form of a system of equations. It is possible to write any linear system of equations in the form $\mathbf{Ax} = \mathbf{b}$, where $\mathbf{A}$ is a matrix containing the coefficients of the variables in the equations, $\mathbf{b}$ is a vector containing the coefficients on the right-hand sides of the equations, and $\mathbf{x}$ is a vector containing the variables.

Definition A.2 If $\mathbf{A}$ is a matrix of size m by n, and $\mathbf{B}$ is a matrix of size n by r, then the product $\mathbf{C} = \mathbf{AB}$ is obtained by multiplying $\mathbf{A}$ times each of the columns of $\mathbf{B}$ and assembling the matrix vector products in $\mathbf{C}$:

$$\mathbf{C} = \begin{bmatrix} \mathbf{AB}_{\cdot,1} & \mathbf{AB}_{\cdot,2} & \dots & \mathbf{AB}_{\cdot,r} \end{bmatrix}. \tag{A.20}$$

This approach given in (A.20) for calculating a matrix–matrix product will be referred to as the **matrix–vector method**.

Note that the product (A.20) is only possible if the two matrices are of compatible sizes. If $\mathbf{A}$ has m rows and n columns, and $\mathbf{B}$ has n rows and r columns, then the product $\mathbf{AB}$ exists and is of size m by r. In some cases, it is thus possible to multiply $\mathbf{AB}$ but not $\mathbf{BA}$. It is important to note that when both $\mathbf{AB}$ and $\mathbf{BA}$ exist, $\mathbf{AB}$ is not generally equal to $\mathbf{BA}$!

An alternate way to compute the product of two matrices is the **row–column expansion method**, where the product element $C_{i,j}$ is calculated as the matrix product of row i of $\mathbf{A}$ and column j of $\mathbf{B}$.

Example A.5

Let

$$\mathbf{A} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} \tag{A.21}$$

and

$$\mathbf{B} = \begin{bmatrix} 5 & 2 \\ 3 & 7 \end{bmatrix}. \tag{A.22}$$

The product matrix $\mathbf{C} = \mathbf{AB}$ will be of size 3 by 2. We compute the product using both methods. First, using the matrix–vector approach (A.20), we have

$$\mathbf{C} = \begin{bmatrix} \mathbf{AB}_{\cdot,1} & \mathbf{AB}_{\cdot,2} \end{bmatrix} \tag{A.23}$$

$$= \begin{bmatrix} 5 \begin{bmatrix} 1 \\ 3 \\ 5 \end{bmatrix} + 3 \begin{bmatrix} 2 \\ 4 \\ 6 \end{bmatrix} & 2 \begin{bmatrix} 1 \\ 3 \\ 5 \end{bmatrix} + 7 \begin{bmatrix} 2 \\ 4 \\ 6 \end{bmatrix} \end{bmatrix} \tag{A.24}$$

$$= \begin{bmatrix} 11 & 16 \\ 27 & 34 \\ 43 & 52 \end{bmatrix}. \tag{A.25}$$

Next, we use the row–column approach:

$$\mathbf{C} = \begin{bmatrix} 1 \cdot 5 + 2 \cdot 3 & 1 \cdot 2 + 2 \cdot 7 \\ 3 \cdot 5 + 4 \cdot 3 & 3 \cdot 2 + 4 \cdot 7 \\ 5 \cdot 5 + 6 \cdot 3 & 5 \cdot 2 + 6 \cdot 7 \end{bmatrix} \tag{A.26}$$

$$= \begin{bmatrix} 11 & 16 \\ 27 & 34 \\ 43 & 52 \end{bmatrix}. \tag{A.27}$$

Definition A.3 The n by n **identity matrix** $\mathbf{I}_n$ is composed of 1s in the diagonal and 0s in the off-diagonal elements.

For example, the 3 by 3 identity matrix is

$$\mathbf{I}_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{A.28}$$

We often write $\mathbf{I}$ without specifying the size of the matrix in situations where the size of matrix is obvious from the context. It is easily shown that if $\mathbf{A}$ is an m by n matrix, then

$$\mathbf{AI}_n = \mathbf{A} \tag{A.29}$$

$$= \mathbf{I}_m \mathbf{A}. \tag{A.30}$$

Thus, multiplying by $\mathbf{I}$ in matrix algebra is similar to multiplying by 1 in conventional scalar algebra.

We have not defined matrix division, but it is possible at this point to define the matrix algebra equivalent of the reciprocal.

Definition A.4 If $\mathbf{A}$ is an n by n matrix, and there is a matrix $\mathbf{B}$ such that

$$\mathbf{AB} = \mathbf{BA} = \mathbf{I}, \tag{A.31}$$

then $\mathbf{B}$ is the **inverse** of $\mathbf{A}$. We write $\mathbf{B} = \mathbf{A}^{-1}$.

How do we compute the inverse of a matrix? If $\mathbf{AB} = \mathbf{I}$, then

$$\begin{bmatrix} \mathbf{AB}_{\cdot,1} & \mathbf{AB}_{\cdot,2} & \dots & \mathbf{AB}_{\cdot,n} \end{bmatrix} = \mathbf{I}. \tag{A.32}$$

Since the columns of the identity matrix and $\mathbf{A}$ are known, we can solve

$$\mathbf{AB}_{\cdot,1} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{A.33}$$

to obtain $\mathbf{B}_1$. We can find the remaining columns of the inverse in the same way. If any of these systems of equations are inconsistent, then $\mathbf{A}^{-1}$ does not exist.

The inverse matrix can be used to solve a system of linear equations with n equations and n variables. Given the system of equations $\mathbf{Ax} = \mathbf{b}$, and $\mathbf{A}^{-1}$, we can multiply $\mathbf{Ax} = \mathbf{b}$ on both sides by the inverse to obtain

$$\mathbf{A}^{-1}\mathbf{Ax} = \mathbf{A}^{-1}\mathbf{b}. \tag{A.34}$$

Because

$$\mathbf{A}^{-1}\mathbf{Ax} = \mathbf{Ix} \tag{A.35}$$

$$= \mathbf{x}, \tag{A.36}$$

this gives the solution

$$\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}. \tag{A.37}$$

This argument shows that if $\mathbf{A}^{-1}$ exists, then for any right-hand side $\mathbf{b}$, a system of equations has a *unique* solution. If $\mathbf{A}^{-1}$ does not exist, then the system of equations may have either many solutions or no solution.

Definition A.5 When $\mathbf{A}$ is an n by n matrix, $\mathbf{A}^k$ is the product of k copies of $\mathbf{A}$. By convention, we define $\mathbf{A}^0 = \mathbf{I}$.

Definition A.6 The **transpose** of a matrix $\mathbf{A}$, denoted $\mathbf{A}^T$, is obtained by taking the columns of $\mathbf{A}$ and writing them as the rows of the transpose. We will also use the notation $\mathbf{A}^{-T}$ for $(\mathbf{A}^{-1})^T$.

Example A.6

Let

$$\mathbf{A} = \begin{bmatrix} 2 & 1 \\ 5 & 2 \end{bmatrix}. \tag{A.38}$$

Then

$$\mathbf{A}^T = \begin{bmatrix} 2 & 5 \\ 1 & 2 \end{bmatrix}. \tag{A.39}$$

Definition A.7 A matrix is **symmetric** if $\mathbf{A} = \mathbf{A}^T$.

Although many elementary textbooks on linear algebra consider only square diagonal matrices, we will have occasion to refer to m by n matrices that have nonzero elements only on the diagonal.

Definition A.8 An m by n matrix $\mathbf{A}$ is **diagonal** if $A_{i,j} = 0$ whenever $i \neq j$.

Definition A.9 An m by n matrix $\mathbf{R}$ is **upper-triangular** if $R_{i,j} = 0$ whenever $i > j$. A matrix $\mathbf{L}$ is **lower-triangular** if $\mathbf{L}^T$ is upper-triangular.

Example A.7

The matrix

$$\mathbf{S} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 \end{bmatrix} \tag{A.40}$$

is diagonal, and the matrix

$$\mathbf{R} = \begin{bmatrix} 1 & 2 & 3 \\ 0 & 2 & 4 \\ 0 & 0 & 5 \\ 0 & 0 & 0 \end{bmatrix} \tag{A.41}$$

is upper-triangular.

Theorem A.1 *The following statements are true for any scalars s and t and matrices $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$. It is assumed that the matrices are of the appropriate size for the operations involved and that whenever an inverse occurs, the matrix is invertible.*

   1. $\mathbf{A} + 0 = 0 + \mathbf{A} = \mathbf{A}$.
   2. $\mathbf{A} + \mathbf{B} = \mathbf{B} + \mathbf{A}$.
   3. $(\mathbf{A} + \mathbf{B}) + \mathbf{C} = \mathbf{A} + (\mathbf{B} + \mathbf{C})$.
   4. $\mathbf{A}(\mathbf{BC}) = (\mathbf{AB})\mathbf{C}$.
   5. $\mathbf{A}(\mathbf{B} + \mathbf{C}) = \mathbf{AB} + \mathbf{AC}$.
   6. $(\mathbf{A} + \mathbf{B})\mathbf{C} = \mathbf{AC} + \mathbf{BC}$.
   7. $(st)\mathbf{A} = s(t\mathbf{A})$.
   8. $s(\mathbf{AB}) = (s\mathbf{A})\mathbf{B} = \mathbf{A}(s\mathbf{B})$.
   9. $(s + t)\mathbf{A} = s\mathbf{A} + t\mathbf{A}$.
   10. $s(\mathbf{A} + \mathbf{B}) = s\mathbf{A} + s\mathbf{B}$.
   11. $(\mathbf{A}^T)^T = \mathbf{A}$.
   12. $(s\mathbf{A})^T = s(\mathbf{A}^T)$.
   13. $(\mathbf{A} + \mathbf{B})^T = \mathbf{A}^T + \mathbf{B}^T$.
   14. $(\mathbf{AB})^T = \mathbf{B}^T\mathbf{A}^T$.
   15. $(\mathbf{AB})^{-1} = \mathbf{B}^{-1}\mathbf{A}^{-1}$.
   16. $(\mathbf{A}^{-1})^{-1} = \mathbf{A}$.
   17. $(\mathbf{A}^T)^{-1} = (\mathbf{A}^{-1})^T$.
   18. If $\mathbf{A}$ and $\mathbf{B}$ are n by n matrices, and $\mathbf{AB} = \mathbf{I}$, then $\mathbf{A}^{-1} = \mathbf{B}$ and $\mathbf{B}^{-1} = \mathbf{A}$.

The first 10 rules in this list are identical to rules of conventional algebra, and you should have little trouble in applying them. The rules involving transposes and inverses are new, but they can be mastered without too much trouble.

   Some students have difficulty with the following statements, which would appear to be true on the surface, but that are in fact **false** for at least some matrices.
   1. $\mathbf{AB} = \mathbf{BA}$.
   2. If $\mathbf{AB} = 0$, then $\mathbf{A} = 0$ or $\mathbf{B} = 0$.
   3. If $\mathbf{AB} = \mathbf{AC}$ and $\mathbf{A} \neq 0$, then $\mathbf{B} = \mathbf{C}$.

It is a worthwhile exercise to construct examples of 2 by 2 matrices for which each of these statements is false.

A.3. LINEAR INDEPENDENCE

Definition A.10 The vectors $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n$ are **linearly independent** if the system of equations

$$c_1\mathbf{v}_1 + c_2\mathbf{v}_2 + \cdots + c_n\mathbf{v}_n = \mathbf{0} \tag{A.42}$$

has only the trivial solution $\mathbf{c} = \mathbf{0}$. If there are multiple solutions, then the vectors are **linearly dependent**.

Determining whether a set of vectors is linearly independent is simple. Just solve the above system of equations (A.42).

●━━━━━━━━━━━━━━━━━━━━━━━━━━━━━

Example A.8

Let

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}. \tag{A.43}$$

Are the columns of $\mathbf{A}$ linearly independent vectors? To determine this we set up the system of equations $\mathbf{Ax} = \mathbf{0}$ in an augmented matrix, and then find the RREF,

$$\begin{bmatrix} 1 & 0 & -1 & | & 0 \\ 0 & 1 & 2 & | & 0 \\ 0 & 0 & 0 & | & 0 \end{bmatrix}. \tag{A.44}$$

The solutions are

$$\mathbf{x} = x_3 \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix}. \tag{A.45}$$

We can set $x_3 = 1$ and obtain the nonzero solution,

$$\mathbf{x} = \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix}. \tag{A.46}$$

Thus, the columns of $\mathbf{A}$ are linearly dependent.

───────────────────────────────

There are a number of important theoretical consequences of linear independence. For example, it can be shown that if the columns of an n by n matrix $\mathbf{A}$ are linearly independent, then $\mathbf{A}^{-1}$ exists, and the system of equations $\mathbf{Ax} = \mathbf{b}$ has a unique solution for every right–hand side $\mathbf{b}$ [95].

───────────────────────────────

A.4. SUBSPACES OF R^n

So far, we have worked with vectors of real numbers in the n-dimensional space R^n. There are a number of properties of R^n that make it convenient to work with vectors. First, the operation of vector addition always works. We can take any two vectors in R^n and add them together and get another vector in R^n. Second, we can multiply any vector in R^n by a scalar and obtain another vector in R^n. Finally, we have the $\mathbf{0}$ vector, with the property that for any vector $\mathbf{x}$, $\mathbf{x} + \mathbf{0} = \mathbf{0} + \mathbf{x} = \mathbf{x}$.

Definition A.11 A **subspace** W of R^n is a subset of R^n that satisfies the three properties:
1. If $\mathbf{x}$ and $\mathbf{y}$ are vectors in W, then $\mathbf{x} + \mathbf{y}$ is also a vector in W.
2. If $\mathbf{x}$ is a vector in W and s is any real scalar, then $s\mathbf{x}$ is also a vector in W.
3. The $\mathbf{0}$ vector is in W. A subspace of R^n is **nontrivial** if it contains vectors other than the $\mathbf{0}$ vector.

Example A.9

In R^3, the plane P defined by the equation

$$x_1 + x_2 + x_3 = 0 \tag{A.47}$$

is a subspace of R^n. To see this, note that if we take any two vectors in the plane and add them together, we get another vector in the plane. If we take a vector in this plane and multiply it by any scalar, we get another vector in the plane. Finally, $\mathbf{0}$ is a vector in the plane.

Subspaces are important because they provide an environment within which all of the rules of matrix-vector algebra apply. An especially important subspace of R^n that we will work with is the **null space** of an m by n matrix.

Definition A.12 Let $\mathbf{A}$ be an m by n matrix. The null space of $\mathbf{A}$, written $N(\mathbf{A})$, is the set of all vectors $\mathbf{x}$ such that $\mathbf{Ax} = \mathbf{0}$.

To show that $N(\mathbf{A})$ is actually a subspace of R^n, we need to show that:
1. If $\mathbf{x}$ and $\mathbf{y}$ are in $N(\mathbf{A})$, then $\mathbf{Ax} = \mathbf{0}$ and $\mathbf{Ay} = \mathbf{0}$. By adding these equations, we find that $\mathbf{A}(\mathbf{x} + \mathbf{y}) = \mathbf{0}$. Thus $\mathbf{x} + \mathbf{y}$ is in $N(\mathbf{A})$.
2. If $\mathbf{x}$ is in $N(\mathbf{A})$ and s is any scalar, then $\mathbf{Ax} = \mathbf{0}$. We can multiply this equation by s to get $s\mathbf{Ax} = \mathbf{0}$. Thus $\mathbf{A}(s\mathbf{x}) = \mathbf{0}$, and $s\mathbf{x}$ is in $N(\mathbf{A})$.
3. $\mathbf{A0} = \mathbf{0}$, so $\mathbf{0}$ is in $N(\mathbf{A})$.

Computationally, the null space of a matrix can be determined by solving the system of equations $\mathbf{Ax} = \mathbf{0}$.

Example A.10

Let

$$\mathbf{A} = \begin{bmatrix} 3 & 1 & 9 & 4 \\ 2 & 1 & 7 & 3 \\ 5 & 2 & 16 & 7 \end{bmatrix}. \tag{A.48}$$

To find the null space of **A**, we solve the system of equations $\mathbf{Ax} = \mathbf{0}$. To solve the equations, we put the system of equations into an augmented matrix,

$$
\left[\begin{array}{cccc|c}
3 & 1 & 9 & 4 & 0 \\
2 & 1 & 7 & 3 & 0 \\
5 & 2 & 16 & 7 & 0
\end{array}\right], \tag{A.49}
$$

and find the RREF,

$$
\left[\begin{array}{cccc|c}
1 & 0 & 2 & 1 & 0 \\
0 & 1 & 3 & 1 & 0 \\
0 & 0 & 0 & 0 & 0
\end{array}\right]. \tag{A.50}
$$

From the augmented matrix, we find that

$$
\mathbf{x} = x_3 \begin{bmatrix} -2 \\ -3 \\ 1 \\ 0 \end{bmatrix} + x_4 \begin{bmatrix} -1 \\ -1 \\ 0 \\ 1 \end{bmatrix}. \tag{A.51}
$$

Any vector in the null space can be written as a linear combination of the above vectors, so the null space is a two-dimensional plane within R^4.

Now, consider the problem of solving $\mathbf{Ax} = \mathbf{b}$, where

$$
\mathbf{b} = \begin{bmatrix} 22 \\ 17 \\ 39 \end{bmatrix} \tag{A.52}
$$

and one particular solution is

$$
\mathbf{p} = \begin{bmatrix} 1 \\ 2 \\ 1 \\ 2 \end{bmatrix}. \tag{A.53}
$$

We can take any vector in the null space of **A** and add it to this solution to obtain another solution. Suppose that $\mathbf{x}$ is in $N(\mathbf{A})$. Then

$$
\mathbf{A}(\mathbf{x} + \mathbf{p}) = \mathbf{Ax} + \mathbf{Ap}
$$
$$
\mathbf{A}(\mathbf{x} + \mathbf{p}) = \mathbf{0} + \mathbf{b}
$$
$$
\mathbf{A}(\mathbf{x} + \mathbf{p}) = \mathbf{b}.
$$

For example,

$$\mathbf{x} = \begin{bmatrix} 1 \\ 2 \\ 1 \\ 2 \end{bmatrix} + 2 \begin{bmatrix} -2 \\ -3 \\ 1 \\ 0 \end{bmatrix} + 3 \begin{bmatrix} -1 \\ -1 \\ 0 \\ 1 \end{bmatrix} \tag{A.54}$$

is also a solution to $\mathbf{Ax} = \mathbf{b}$.

In the context of inverse problems, the null space is critical because the presence of a nontrivial null space leads to nonuniqueness in the solution to a linear system of equations.

Definition A.13 A **basis** for a subspace W is a set of vectors $\mathbf{v}_1, \ldots, \mathbf{v}_p$ such that:
1. Any vector in W can be written as a linear combination of the basis vectors.
2. The basis vectors are linearly independent.

A particularly simple and useful basis is the **standard basis**.

Definition A.14 The **standard basis** for R^n is the set of vectors $\mathbf{e}_1, \ldots, \mathbf{e}_n$ such that the elements of $\mathbf{e}_i$ are all zero except for the *ith* element, which is 1.

Any nontrivial subspace W of R^n will have many different bases. For example, we can take any basis and multiply one of the basis vectors by 2 to obtain a new basis. It is possible to show that all bases for a subspace W have the same number of basis vectors [95].

Theorem A.2 *Let W be a subspace of R^n with basis $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_p$. Then all bases for W have p basis vectors, and p is the **dimension** of W.*

It can be shown that the procedure used in the above example always produces a basis for $N(\mathbf{A})$ [95]. A basis for the null space of a matrix can be found in MATLAB using the **null** command.

Definition A.15 Let $\mathbf{A}$ be an m by n matrix. The **column space** or **range** of $\mathbf{A}$ (written $R(\mathbf{A})$) is the set of all vectors $\mathbf{b}$ such that $\mathbf{Ax} = \mathbf{b}$ has at least one solution. In other words, the column space is the set of all vectors $\mathbf{b}$ that can be written as a linear combination of the columns of $\mathbf{A}$.

The range is important in the context of discrete linear inverse problems, because $R(\mathbf{G})$ consists of all vectors $\mathbf{d}$ for which there is a model $\mathbf{m}$ such that $\mathbf{Gm} = \mathbf{d}$.

To find the column space of a matrix, we consider what happens when we compute the RREF of $[\mathbf{A} \,|\, \mathbf{b}]$. In the part of the augmented matrix corresponding to the left-hand side of the equations we always get the same result, namely the RREF of $\mathbf{A}$. The

solution to the system of equations may involve some free variables, but we can always set these free variables to 0. Thus when we are able to solve $\mathbf{Ax} = \mathbf{b}$, we can solve the system of equations by using only variables corresponding to the pivot columns in the RREF of $\mathbf{A}$. In other words, if we can solve $\mathbf{Ax} = \mathbf{b}$, then we can write $\mathbf{b}$ as a linear combination of the pivot columns of $\mathbf{A}$. Note that these are columns from the original matrix $\mathbf{A}$, not columns from the RREF of $\mathbf{A}$. An orthonormal (see below) basis for the range of a matrix can be found in MATLAB using the **orth** command.

Example A.11

As in the previous example, let

$$\mathbf{A} = \begin{bmatrix} 3 & 1 & 9 & 4 \\ 2 & 1 & 7 & 3 \\ 5 & 2 & 16 & 7 \end{bmatrix}. \tag{A.55}$$

To find the column space of $\mathbf{A}$, note that the RREF of $\mathbf{A}$ is

$$\begin{bmatrix} 1 & 0 & 2 & 1 \\ 0 & 1 & 3 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \tag{A.56}$$

Thus, whenever we can solve $\mathbf{Ax} = \mathbf{b}$, we can find a solution in which x_3 and x_4 are 0. In other words, whenever there is a solution to $\mathbf{Ax} = \mathbf{b}$, we can write $\mathbf{b}$ as a linear combination of the first two columns of $\mathbf{A}$:

$$\mathbf{b} = x_1 \begin{bmatrix} 3 \\ 2 \\ 5 \end{bmatrix} + x_2 \begin{bmatrix} 1 \\ 1 \\ 2 \end{bmatrix}. \tag{A.57}$$

Since these two vectors are linearly independent and span $R(\mathbf{A})$, they form a basis for $R(\mathbf{A})$. The dimension of $R(\mathbf{A})$ is 2.

In finding the null space and range of a matrix $\mathbf{A}$, we found that the basis vectors for $N(\mathbf{A})$ corresponded to nonpivot columns of $\mathbf{A}$, while the basis vectors for $R(\mathbf{A})$ corresponded to pivot columns of $\mathbf{A}$. Since the matrix $\mathbf{A}$ had n columns, we obtain the following theorem.

Theorem A.3

$$dim\ N(\mathbf{A}) + dim\ R(\mathbf{A}) = n. \tag{A.58}$$

In addition to the null space and range of a matrix $\mathbf{A}$, we will often work with the null space and range of the transpose of $\mathbf{A}$. Since the columns of $\mathbf{A}^T$ are rows of $\mathbf{A}$, the

column space of $\mathbf{A}^T$ is also called the **row space** of $\mathbf{A}$. Since each row of $\mathbf{A}$ can be written as a linear combination of the nonzero rows of the RREF of $\mathbf{A}$, the nonzero rows of the RREF form a basis for the row space of $\mathbf{A}$. There are exactly as many nonzero rows in the RREF of $\mathbf{A}$ as there are pivot columns. Thus we have the following theorem.

Theorem A.4

$$dim(R(\mathbf{A}^T)) = dim\ R(\mathbf{A}). \tag{A.59}$$

Definition A.16 The **rank** of an m by n matrix $\mathbf{A}$ is the dimension of $R(\mathbf{A})$. If $\mathrm{rank}(\mathbf{A}) = \min(m, n)$, then $\mathbf{A}$ has **full rank**. If $\mathrm{rank}(\mathbf{A}) = m$, then $\mathbf{A}$ has **full row rank**. If $\mathrm{rank}(\mathbf{A}) = n$, then $\mathbf{A}$ has **full column rank**. If $\mathrm{rank}(\mathbf{A}) < \min(m, n)$, then $\mathbf{A}$ is **rank deficient**.

The rank of a matrix is readily found in MATLAB by using the **rank** command.

A.5. ORTHOGONALITY AND THE DOT PRODUCT

Definition A.17 Let $\mathbf{x}$ and $\mathbf{y}$ be two vectors in R^n. The **dot product** of $\mathbf{x}$ and $\mathbf{y}$ is

$$\mathbf{x} \cdot \mathbf{y} = \mathbf{x}^T \mathbf{y} = x_1 y_1 + x_2 y_2 + \cdots + x_n y_n. \tag{A.60}$$

Definition A.18 Let $\mathbf{x}$ be a vector in R^n. The **2–norm** or **Euclidean length** of $\mathbf{x}$ is

$$\|\mathbf{x}\|_2 = \sqrt{\mathbf{x}^T \mathbf{x}} = \sqrt{x_1^2 + x_2^2 + \cdots + x_n^2}. \tag{A.61}$$

Later we will introduce two other ways of measuring the "length" of a vector. The subscript 2 is used to distinguish this 2-norm from the other norms.

You may be familiar with an alternative definition of the dot product in which $\mathbf{x} \cdot \mathbf{y} = \|\mathbf{x}\|\|\mathbf{y}\| \cos(\theta)$, where θ is the angle between the two vectors. The two definitions are equivalent. To see this, consider a triangle with sides $\mathbf{x}$, $\mathbf{y}$, and $\mathbf{x} - \mathbf{y}$. See Figure A.1.
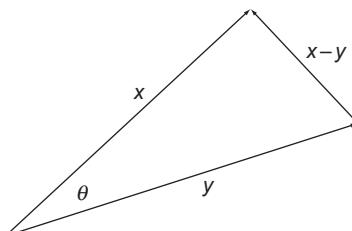
Figure A.1 Relationship between the dot product and the angle between two vectors.

The angle between sides $\mathbf{x}$ and $\mathbf{y}$ is θ. By the law of cosines,

$$\|\mathbf{x} - \mathbf{y}\|_2^2 = \|\mathbf{x}\|_2^2 + \|\mathbf{y}\|_2^2 - 2\|\mathbf{x}\|_2\|\mathbf{y}\|_2 \cos(\theta)$$

$$(\mathbf{x} - \mathbf{y})^T(\mathbf{x} - \mathbf{y}) = \mathbf{x}^T\mathbf{x} + \mathbf{y}^T\mathbf{y} - 2\|\mathbf{x}\|_2\|\mathbf{y}\|_2 \cos(\theta)$$

$$\mathbf{x}^T\mathbf{x} - 2\mathbf{x}^T\mathbf{y} + \mathbf{y}^T\mathbf{y} = \mathbf{x}^T\mathbf{x} + \mathbf{y}^T\mathbf{y} - 2\|\mathbf{x}\|_2\|\mathbf{y}\|_2 \cos(\theta)$$

$$-2\mathbf{x}^T\mathbf{y} = -2\|\mathbf{x}\|_2\|\mathbf{y}\|_2 \cos(\theta)$$

$$\mathbf{x}^T\mathbf{y} = \|\mathbf{x}\|_2\|\mathbf{y}\|_2 \cos(\theta).$$

We can also use this formula to compute the angle between two vectors:

$$\theta = \cos^{-1}\left(\frac{\mathbf{x}^T\mathbf{y}}{\|\mathbf{x}\|_2\|\mathbf{y}\|_2}\right). \tag{A.62}$$

Definition A.19 Two vectors $\mathbf{x}$ and $\mathbf{y}$ in R^n are **orthogonal**, or equivalently, **perpendicular** (written $\mathbf{x} \perp \mathbf{y}$), if $\mathbf{x}^T\mathbf{y} = 0$.

Definition A.20 A set of vectors $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_p$ is **orthogonal** if each pair of vectors in the set is orthogonal.

Definition A.21 Two subspaces V and W of R^n are **orthogonal** if every vector in V is perpendicular to every vector in W.

If $\mathbf{x}$ is in $N(\mathbf{A})$, then $\mathbf{A}\mathbf{x} = \mathbf{0}$. Since each element of the product $\mathbf{A}\mathbf{x}$ can be obtained by taking the dot product of a row of $\mathbf{A}$ and $\mathbf{x}$, $\mathbf{x}$ is perpendicular to each row of $\mathbf{A}$. Since $\mathbf{x}$ is perpendicular to all of the columns of $\mathbf{A}^T$, it is perpendicular to $R(\mathbf{A}^T)$. We have the following theorem.

Theorem A.5 *Let $\mathbf{A}$ be an m by n matrix. Then*

$$N(\mathbf{A}) \perp R(\mathbf{A}^T). \tag{A.63}$$

Furthermore,

$$N(\mathbf{A}) + R(\mathbf{A}^T) = R^n. \tag{A.64}$$

That is, any vector $\mathbf{x}$ in R^n can be written uniquely as $\mathbf{x} = \mathbf{p} + \mathbf{q}$, where $\mathbf{p}$ is in $N(\mathbf{A})$ and $\mathbf{q}$ is in $R(\mathbf{A}^T)$.

Definition A.22 A basis in which the basis vectors are orthogonal is an **orthogonal basis**. A basis in which the basis vectors are orthogonal and have length 1 is an **orthonormal basis**.

Definition A.23 An n by n matrix $\mathbf{Q}$ is **orthogonal** if the columns of $\mathbf{Q}$ are orthogonal and each column of $\mathbf{Q}$ has length 1.

With the requirement that the columns of an orthogonal matrix have length 1, using the term "orthonormal" would make logical sense. However, the definition of "orthogonal" given here is standard and we will not try to change standard usage.

Orthogonal matrices have a number of useful properties.

Theorem A.6 *If* $\mathbf{Q}$ *is an orthogonal matrix, then:*
1. $\mathbf{Q}^T\mathbf{Q} = \mathbf{Q}\mathbf{Q}^T = \mathbf{I}$. *In other words,* $\mathbf{Q}^{-1} = \mathbf{Q}^T$.
2. *For any vector* $\mathbf{x}$ *in* R^n, $\|\mathbf{Q}\mathbf{x}\|_2 = \|\mathbf{x}\|_2$.
3. *For any two vectors* $\mathbf{x}$ *and* $\mathbf{y}$ *in* R^n, $\mathbf{x}^T\mathbf{y} = (\mathbf{Q}\mathbf{x})^T(\mathbf{Q}\mathbf{y})$.

A problem that we will often encounter in practice is projecting a vector $\mathbf{x}$ onto another vector $\mathbf{y}$ or onto a subspace W to obtain a projected vector $\mathbf{p}$. See Figure A.2. We know that

$$\mathbf{x}^T\mathbf{y} = \|\mathbf{x}\|_2\|\mathbf{y}\|_2 \, \cos(\theta), \tag{A.65}$$

where θ is the angle between $\mathbf{x}$ and $\mathbf{y}$. Also,

$$\cos(\theta) = \frac{\|\mathbf{p}\|_2}{\|\mathbf{x}\|_2}. \tag{A.66}$$

Thus

$$\|\mathbf{p}\|_2 = \frac{\mathbf{x}^T\mathbf{y}}{\|\mathbf{y}\|_2}. \tag{A.67}$$

Since $\mathbf{p}$ points in the same direction as $\mathbf{y}$,

$$\mathbf{p} = \text{proj}_{\mathbf{y}}\mathbf{x} = \frac{\mathbf{x}^T\mathbf{y}}{\mathbf{y}^T\mathbf{y}}\mathbf{y}. \tag{A.68}$$

The vector $\mathbf{p}$ is called the **orthogonal projection** or simply the **projection** of $\mathbf{x}$ onto $\mathbf{y}$.
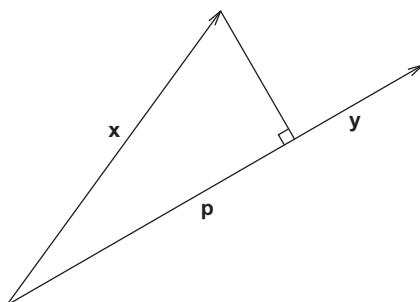
Figure A.2 The orthogonal projection of $\mathbf{x}$ onto $\mathbf{y}$.

Similarly, if W is a subspace of R^n with an orthogonal basis $\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_p$, then the **orthogonal projection of x onto** W is

$$\mathbf{p} = \operatorname{proj}_W \mathbf{x} = \frac{\mathbf{x}^T \mathbf{w}_1}{\mathbf{w}_1^T \mathbf{w}_1} \mathbf{w}_1 + \frac{\mathbf{x}^T \mathbf{w}_2}{\mathbf{w}_2^T \mathbf{w}_2} \mathbf{w}_2 + \cdots + \frac{\mathbf{x}^T \mathbf{w}_p}{\mathbf{w}_p^T \mathbf{w}_p} \mathbf{w}_p. \tag{A.69}$$

Note that this equation can be simplified considerably if the orthogonal basis vectors are also orthonormal. In that case, $\mathbf{w}_1^T \mathbf{w}_1, \mathbf{w}_2^T \mathbf{w}_2, \ldots, \mathbf{w}_p^T \mathbf{w}_p$ are all 1.

It is inconvenient that the projection formula requires an orthogonal basis. The **Gram–Schmidt orthogonalization process** can be used to turn any basis for a subspace of R^n into an orthogonal basis. We begin with a basis $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_p$. The process recursively constructs an orthogonal basis by taking each vector in the original basis and then subtracting off its projection on the space spanned by the previous vectors. The formulas are

$$\mathbf{w}_1 = \mathbf{v}_1$$

$$\mathbf{w}_2 = \mathbf{v}_2 - \frac{\mathbf{v}_1^T \mathbf{v}_2}{\mathbf{v}_1^T \mathbf{v}_1} \mathbf{v}_1 = \mathbf{v}_2 - \frac{\mathbf{w}_1^T \mathbf{v}_2}{\mathbf{w}_1^T \mathbf{w}_1} \mathbf{w}_1$$

$$\vdots$$

$$\mathbf{w}_p = \mathbf{v}_p - \frac{\mathbf{w}_1^T \mathbf{v}_p}{\mathbf{w}_1^T \mathbf{w}_1} \mathbf{w}_1 - \cdots - \frac{\mathbf{w}_p^T \mathbf{v}_p}{\mathbf{w}_p^T \mathbf{w}_p} \mathbf{w}_p. \tag{A.70}$$

Unfortunately, the Gram-Schmidt process is numerically unstable when applied to large bases. In MATLAB the command **orth** provides a numerically stable way to produce an orthogonal basis from a nonorthogonal basis. An important property of orthogonal projection is that the projection of $\mathbf{x}$ onto W is the point in W which is closest to $\mathbf{x}$. In the special case that $\mathbf{x}$ is in W, the projection of $\mathbf{x}$ onto W is $\mathbf{x}$.

Given an inconsistent system of equations $\mathbf{Ax} = \mathbf{b}$, it is often desirable to find an approximate solution. A natural measure of the quality of an approximate solution is the norm of the difference between $\mathbf{Ax}$ and $\mathbf{b}$, $\|\mathbf{Ax} - \mathbf{b}\|$. A solution that minimizes the 2-norm, $\|\mathbf{Ax} - \mathbf{b}\|_2$, is called a **least squares solution**, because it minimizes the sum of the squares of the residuals.

The least squares solution can be obtained by projecting $\mathbf{b}$ onto $R(\mathbf{A})$. This calculation requires us to first find an orthogonal basis for $R(\mathbf{A})$. There is an alternative approach that does not require finding an orthogonal basis. Let

$$\mathbf{Ax}_{ls} = \operatorname{proj}_{R(\mathbf{A})} \mathbf{b}. \tag{A.71}$$

Then, the difference between the projection (A.71) and $\mathbf{b}$, $\mathbf{Ax}_{ls} - \mathbf{b}$, will be perpendicular to $R(\mathbf{A})$ (Figure A.3). This orthogonality means that each of the columns of $\mathbf{A}$ will be orthogonal to $\mathbf{Ax}_{ls} - \mathbf{b}$. Thus

$$\mathbf{A}^T (\mathbf{Ax}_{ls} - \mathbf{b}) = 0 \tag{A.72}$$
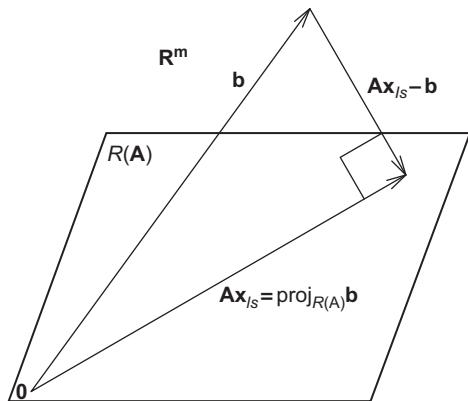
Figure A.3   Geometric conceptualization of the least squares solution to $\mathbf{Ax} = \mathbf{b}$. $\mathbf{b}$ generally lies in R^m, but $R(\mathbf{A})$ is generally a subspace of R^m. The least squares solution $\mathbf{x}_{ls}$ minimizes $\|\mathbf{Ax} - \mathbf{b}\|_2$.

or

$$\mathbf{A}^T \mathbf{A} \mathbf{x}_{ls} = \mathbf{A}^T \mathbf{b}. \tag{A.73}$$

This last system of equations is referred to as the **normal equations** for the least squares problem. It can be shown that if the columns of $\mathbf{A}$ are linearly independent, then the normal equations have exactly one solution for $\mathbf{x}_{ls}$ and this solution minimizes the sum of squared residuals [95].

A.6. EIGENVALUES AND EIGENVECTORS

Definition A.24   An n by n matrix $\mathbf{A}$ has an eigenvalue λ with an associated eigenvector $\mathbf{x}$ if $\mathbf{x}$ is not $\mathbf{0}$, and

$$\mathbf{Ax} = \lambda \mathbf{x}. \tag{A.74}$$

To find eigenvalues and eigenvectors, we rewrite the eigenvector equation (A.74) as

$$(\mathbf{A} - \lambda \mathbf{I})\mathbf{x} = \mathbf{0}. \tag{A.75}$$

To find nonzero eigenvectors, the matrix $\mathbf{A} - \lambda \mathbf{I}$ must be singular. This leads to the **characteristic equation**,

$$\det(\mathbf{A} - \lambda \mathbf{I}) = 0. \tag{A.76}$$

where det denotes the determinant. For small matrices (e.g., 2 by 2 or 3 by 3), it is relatively easy to solve (A.76) to find the eigenvalues. The eigenvalues can then be substituted into (A.75), and the resulting system can then be solved to find corresponding

eigenvectors. Note that the eigenvalues can, in general, be complex. For larger matrices, solving the characteristic equation becomes impractical and more sophisticated numerical methods are used. The MATLAB command **eig** can be used to find eigenvalues and eigenvectors of a matrix.

Suppose that we can find a set of n linearly independent eigenvectors, $\mathbf{v}_i$, of an n by n matrix $\mathbf{A}$ with associated eigenvalues λ_i. These eigenvectors form a basis for R^n. We can use the eigenvectors to **diagonalize** the matrix as

$$\mathbf{A} = \mathbf{P}\mathbf{\Lambda}\mathbf{P}^{-1}, \tag{A.77}$$

where

$$\mathbf{P} = \begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \ldots & \mathbf{v}_n \end{bmatrix}, \tag{A.78}$$

and $\mathbf{\Lambda}$ is a diagonal matrix of eigenvalues

$$\Lambda_{ii} = \lambda_i. \tag{A.79}$$

To see that this works, simply compute $\mathbf{AP}$:

$$\begin{aligned} \mathbf{AP} &= \mathbf{A} \begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \ldots & \mathbf{v}_n \end{bmatrix} \\ &= \begin{bmatrix} \lambda_1\mathbf{v}_1 & \lambda_2\mathbf{v}_2 & \ldots & \lambda_n\mathbf{v}_n \end{bmatrix} \\ &= \mathbf{P}\mathbf{\Lambda}. \end{aligned}$$

Thus, $\mathbf{A} = \mathbf{P}\mathbf{\Lambda}\mathbf{P}^{-1}$. Not all matrices are diagonalizable, because not all matrices have n linearly independent eigenvectors. However, there is an important special case in which matrices can always be diagonalized.

Theorem A.7 *If $\mathbf{A}$ is a real symmetric matrix, then $\mathbf{A}$ can be written as*

$$\mathbf{A} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^{-1} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T, \tag{A.80}$$

where $\mathbf{Q}$ is a real orthogonal matrix of eigenvectors of $\mathbf{A}$, and $\mathbf{\Lambda}$ is a real diagonal matrix of the eigenvalues of $\mathbf{A}$.

This **orthogonal diagonalization** of a real symmetric matrix $\mathbf{A}$ will be useful later on when we consider orthogonal factorizations of general matrices.

The eigenvalues of symmetric matrices are particularly important in the analysis of quadratic forms.

Definition A.25 A **quadratic form** is a function of the form

$$f(\mathbf{x}) = \mathbf{x}^T \mathbf{A}\mathbf{x}, \tag{A.81}$$

where $\mathbf{A}$ is a symmetric $n \times n$ matrix. The quadratic form $f(\mathbf{x})$ is **positive definite** if $f(x) \geq 0$ for all $\mathbf{x}$ and $f(\mathbf{x}) = 0$ only when $\mathbf{x} = \mathbf{0}$. The quadratic form is **positive**

semidefinite if $f(\mathbf{x}) \geq 0$ for all $\mathbf{x}$. Similarly, a symmetric matrix $\mathbf{A}$ is positive definite if the associated quadratic form $f(\mathbf{x}) = \mathbf{x}^T \mathbf{A} \mathbf{x}$ is positive definite. The quadratic form is **negative semidefinite** if $-f(\mathbf{x})$ is positive semidefinite. If $f(\mathbf{x})$ is neither positive semidefinite nor negative semidefinite, then $f(\mathbf{x})$ is **indefinite**.

Positive definite quadratic forms have an important application in analytic geometry. Let $\mathbf{A}$ be a symmetric and positive definite matrix. Then the region defined by the inequality

$$(\mathbf{x} - \mathbf{c})^T \mathbf{A}(\mathbf{x} - \mathbf{c}) \leq \delta \tag{A.82}$$

is an ellipsoidal volume, with its center at $\mathbf{c}$. We can diagonalize $\mathbf{A}$ as

$$\mathbf{A} = \mathbf{P} \mathbf{\Lambda} \mathbf{P}^{-1}, \tag{A.83}$$

where the columns of $\mathbf{P}$ are normalized eigenvectors of $\mathbf{A}$, and $\mathbf{\Lambda}$ is a diagonal matrix whose elements are the eigenvalues of $\mathbf{A}$. It can be shown that the ith eigenvector of $\mathbf{A}$ points in the direction of the ith semimajor axis of the ellipsoid, and the length of the ith semimajor axis is given by $\sqrt{\delta/\lambda_i}$ [95].

An important connection between positive semidefinite matrices and eigenvalues is the following theorem.

Theorem A.8 *A symmetric matrix $\mathbf{A}$ is positive semidefinite if and only if its eigenvalues are greater than or equal to 0. $\mathbf{A}$ is positive definite if and only if its eigenvalues are greater than 0.*

This provides a convenient way to check whether a symmetric matrix is positive semidefinite or positive definite.

The **Cholesky factorization** provides another way to determine whether a symmetric matrix is positive definite.

Theorem A.9 *Let $\mathbf{A}$ be an an n by n positive definite and symmetric matrix. Then $\mathbf{A}$ can be written uniquely as*

$$\mathbf{A} = \mathbf{R}^T \mathbf{R} = \mathbf{L} \mathbf{L}^T, \tag{A.84}$$

where $\mathbf{R}$ is a nonsingular upper-triangular matrix and $\mathbf{L} = \mathbf{R}^T$ is a nonsingular lower-triangular matrix. Note that $\mathbf{A}$ can be factored in this way if and only if it is positive definite.

The MATLAB command **chol** can be used to compute the Cholesky factorization.

A.7. VECTOR AND MATRIX NORMS

Although the conventional Euclidean length (A.61) is most commonly used, there are alternative ways to measure the length of a vector.

Definition A.26 Any measure of vector length satisfying the following four conditions is called a **norm**.

1. For any vector $\mathbf{x}$, $\|\mathbf{x}\| \geq 0$.
2. For any vector $\mathbf{x}$ and any scalar s, $\|s\mathbf{x}\| = |s|\|\mathbf{x}\|$.
3. For any vectors $\mathbf{x}$ and $\mathbf{y}$, $\|\mathbf{x} + \mathbf{y}\| \leq \|\mathbf{x}\| + \|\mathbf{y}\|$.
4. $\|\mathbf{x}\| = 0$ if and only if $\mathbf{x} = \mathbf{0}$.

If $\|\ \|$ satisfies conditions 1, 2, and 3, but does not satisfy condition 4, then $\|\ \|$ is called a **seminorm**.

Definition A.27 The p-**norm** of a vector in R^n is defined for $p \geq 1$ by

$$\|x\|_p = (|x_1|^p + |x_2|^p + \cdots + |x_n|^p)^{1/p}. \tag{A.85}$$

It can be shown that for any $p \geq 1$, (A.85) satisfies the conditions of Definition A.26 [53]. The conventional Euclidean length is just the 2-norm, but two other p-norms are also commonly used. The **1–norm** is the sum of the absolute values of the elements in $\mathbf{x}$. The ∞-**norm** is obtained by taking the limit as p goes to infinity. The ∞-norm is the maximum of the absolute values of the elements in $\mathbf{x}$. The MATLAB command **norm** can be used to compute the norm of a vector, and has options for the 1, 2, and infinity norms.

The 2-norm is particularly important because of its natural connection with dot products and projections. The projection of a vector onto a subspace is the point in the subspace that is closest to the vector as measured by the 2-norm. We have also seen in (A.73) that the problem of minimizing $\|\mathbf{Ax} - \mathbf{b}\|_2$ can be solved by computing projections or by using the normal equations. In fact, the 2-norm can be tied directly to the dot product by the formula

$$\|\mathbf{x}\|_2 = \sqrt{\mathbf{x}^T\mathbf{x}}. \tag{A.86}$$

The 1- and ∞-norms can also be useful in finding approximate solutions to over-determined linear systems of equations. To minimize the maximum of the residuals, we minimize $\|\mathbf{Ax} - \mathbf{b}\|_\infty$. To minimize the sum of the absolute values of the residuals, we minimize $\|\mathbf{Ax} - \mathbf{b}\|_1$. Unfortunately, these minimization problems are generally more difficult to solve than least squares problems.

Definition A.28 Any measure of the size or length of an m by n matrix that satisfies the following five properties can be used as a **matrix norm**.

1. For any matrix $\mathbf{A}$, $\|\mathbf{A}\| \geq 0$.
2. For any matrix $\mathbf{A}$ and any scalar s, $\|s\mathbf{A}\| = |s|\|\mathbf{A}\|$.
3. For any matrices $\mathbf{A}$ and $\mathbf{B}$, $\|\mathbf{A} + \mathbf{B}\| \leq \|\mathbf{A}\| + \|\mathbf{B}\|$.
4. $\|\mathbf{A}\| = 0$ if and only if $\mathbf{A} = \mathbf{0}$.
5. For any two matrices $\mathbf{A}$ and $\mathbf{B}$ of compatible sizes, $\|\mathbf{AB}\| \leq \|\mathbf{A}\|\|\mathbf{B}\|$.

Definition A.29 The *p*-**norm** of a matrix $\mathbf{A}$ is

$$\|\mathbf{A}\|_p = \max_{\|\mathbf{x}\|_p=1} \|\mathbf{Ax}\|_p, \tag{A.87}$$

where $\|\mathbf{x}\|_p$ and $\|\mathbf{Ax}\|_p$ are vector *p*-norms, while $\|\mathbf{A}\|_p$ is the matrix *p*-norm of $\mathbf{A}$.

Solving the maximization problem of (A.87) to determine a matrix *p*-norm could be extremely difficult. Fortunately, there are simpler formulas for the most commonly used matrix *p*-norms. See Exercises A.11, A.12, and C.4:

$$\|\mathbf{A}\|_1 = \max_j \sum_{i=1}^m |A_{i,j}| \tag{A.88}$$

$$\|\mathbf{A}\|_2 = \sqrt{\lambda_{\max}(\mathbf{A}^T\mathbf{A})} \tag{A.89}$$

$$\|\mathbf{A}\|_\infty = \max_i \sum_{j=1}^n |A_{i,j}|, \tag{A.90}$$

where $\lambda_{\max}(\mathbf{A}^T\mathbf{A})$ denotes the largest eigenvalue of $\mathbf{A}^T\mathbf{A}$.

Definition A.30 The **Frobenius norm** of an *m* by *n* matrix is given by

$$\|\mathbf{A}\|_F = \sqrt{\sum_{i=1}^m \sum_{j=1}^n A_{i,j}^2}. \tag{A.91}$$

Definition A.31 A matrix norm $\|\ \|_M$ and a vector norm $\|\ \|_V$ are **compatible** if

$$\|\mathbf{Ax}\|_V \le \|\mathbf{A}\|_M \|\mathbf{x}\|_V. \tag{A.92}$$

The matrix *p*-norm is by its definition compatible with the vector *p*-norm from which it was derived. It can also be shown that the Frobenius norm of a matrix is compatible with the vector 2-norm [109]. Thus the Frobenius norm is often used with the vector 2-norm.

In practice, the Frobenius norm, 1-norm, and ∞-norm of a matrix are easy to compute, while the 2-norm of a matrix can be difficult to compute for large matrices. The MATLAB **norm** command has options for computing these matrix norms.

A.8. THE CONDITION NUMBER OF A LINEAR SYSTEM

Suppose that we want to solve a system of *n* equations in *n* variables

$$\mathbf{Ax} = \mathbf{b}. \tag{A.93}$$

Suppose further that because of measurement errors in $\mathbf{b}$, we actually solve

$$\mathbf{A}\hat{\mathbf{x}} = \hat{\mathbf{b}}. \tag{A.94}$$

Can we get a bound on $\|\mathbf{x} - \hat{\mathbf{x}}\|$ in terms of $\|\mathbf{b} - \hat{\mathbf{b}}\|$? Starting with (A.93) and (A.94), we have

$$\mathbf{A}(\mathbf{x} - \hat{\mathbf{x}}) = \mathbf{b} - \hat{\mathbf{b}} \tag{A.95}$$

$$(\mathbf{x} - \hat{\mathbf{x}}) = \mathbf{A}^{-1}(\mathbf{b} - \hat{\mathbf{b}}) \tag{A.96}$$

$$\|\mathbf{x} - \hat{\mathbf{x}}\| = \|\mathbf{A}^{-1}(\mathbf{b} - \hat{\mathbf{b}})\| \tag{A.97}$$

$$\|\mathbf{x} - \hat{\mathbf{x}}\| \leq \|\mathbf{A}^{-1}\| \|\mathbf{b} - \hat{\mathbf{b}}\|. \tag{A.98}$$

This formula provides an absolute bound on the error in the solution. It is also worth–while to compute a relative error bound:

$$\frac{\|\mathbf{x} - \hat{\mathbf{x}}\|}{\|\mathbf{b}\|} \leq \frac{\|\mathbf{A}^{-1}\| \|\mathbf{b} - \hat{\mathbf{b}}\|}{\|\mathbf{b}\|} \tag{A.99}$$

$$\frac{\|\mathbf{x} - \hat{\mathbf{x}}\|}{\|\mathbf{A}\mathbf{x}\|} \leq \frac{\|\mathbf{A}^{-1}\| \|\mathbf{b} - \hat{\mathbf{b}}\|}{\|\mathbf{b}\|} \tag{A.100}$$

$$\|\mathbf{x} - \hat{\mathbf{x}}\| \leq \|\mathbf{A}\mathbf{x}\| \|\mathbf{A}^{-1}\| \frac{\|\mathbf{b} - \hat{\mathbf{b}}\|}{\|\mathbf{b}\|} \tag{A.101}$$

$$\|\mathbf{x} - \hat{\mathbf{x}}\| \leq \|\mathbf{A}\| \|\mathbf{x}\| \|\mathbf{A}^{-1}\| \frac{\|\mathbf{b} - \hat{\mathbf{b}}\|}{\|\mathbf{b}\|} \tag{A.102}$$

$$\frac{\|\mathbf{x} - \hat{\mathbf{x}}\|}{\|\mathbf{x}\|} \leq \|\mathbf{A}\| \|\mathbf{A}^{-1}\| \frac{\|\mathbf{b} - \hat{\mathbf{b}}\|}{\|\mathbf{b}\|}. \tag{A.103}$$

The relative error in $\mathbf{b}$ is measured by

$$\frac{\|\mathbf{b} - \hat{\mathbf{b}}\|}{\|\mathbf{b}\|}. \tag{A.104}$$

The relative error in $\mathbf{x}$ is measured by

$$\frac{\|\mathbf{x} - \hat{\mathbf{x}}\|}{\|\mathbf{x}\|}. \tag{A.105}$$

The constant

$$\text{cond}(\mathbf{A}) = \|\mathbf{A}\| \|\mathbf{A}^{-1}\| \tag{A.106}$$

is called the **condition number** of $\mathbf{A}$.

Note that nothing that we did in the calculation of the condition number depends on which norm we used. The condition number can be computed using the 1-norm,

2-norm, ∞-norm, or Frobenius norm. The MATLAB **cond** command can be used to find the condition number of a matrix using each of these norms.

The condition number provides an upper bound on how inaccurate the solution to a system of equations might be because of errors in the right-hand side. In some cases, the condition number greatly overestimates the error in the solution. As a practical matter, it is wise to assume that the error is of roughly the size predicted by the condition number. In practice, double-precision floating point arithmetic only allows us to store numbers to about 16 digits of precision. If the condition number is greater than 10^{16}, then by the above inequality, there may be no accurate digits in the computer solution to the system of equations. Systems of equations with very large condition numbers are called **ill–conditioned**.

It is important to understand that ill-conditioning is a property of the system of equations and not of the algorithm used to solve the system of equations. Ill-conditioning cannot be fixed simply by using a better algorithm. Instead, we must either increase the precision of our floating point representation or find a different, better conditioned system of equations to solve.

A.9. THE QR FACTORIZATION

Although the theory of linear algebra can be developed using the reduced row echelon form, there is an alternative computational approach that works better in practice. The basic idea is to compute factorizations of matrices that involve orthogonal, diagonal, and upper-triangular matrices. This alternative approach leads to algorithms that can quickly compute accurate solutions to linear systems of equations and least squares problems. In this section we introduce the QR factorization, which is one of the most widely used orthogonal matrix factorizations. Another factorization, the singular value decomposition (SVD), is introduced in Chapter 3.

Theorem A.10 *Let* **A** *be an m by n matrix.* **A** *can be written as*

$$\mathbf{A} = \mathbf{QR}, \tag{A.107}$$

where **Q** *is an m by m orthogonal matrix, and* **R** *is an m by n upper-triangular matrix. This is called the* **QR factorization of A**.

The MATLAB command **qr** can be used to compute the QR factorization of a matrix. In a common situation, **A** will be an m by n matrix with $m > n$, and the rank of **A** will be n. In this case, we can write

$$\mathbf{R} = \begin{bmatrix} \mathbf{R}_1 \\ \mathbf{0} \end{bmatrix}, \tag{A.108}$$

where $\mathbf{R}_1$ is n by n, and

$$\mathbf{Q} = [\mathbf{Q}_1 \quad \mathbf{Q}_2], \tag{A.109}$$

where $\mathbf{Q}_1$ is m by n and $\mathbf{Q}_2$ is m by $m - n$. In this case, the QR factorization has some important properties.

Theorem A.11 *Let $\mathbf{Q}$ and $\mathbf{R}$ be the QR factorization of an m by n matrix $\mathbf{A}$ with $m > n$ and rank($\mathbf{A}$) $= n$. Then,*
1. *The columns of $\mathbf{Q}_1$ are an orthonormal basis for $R(\mathbf{A})$.*
2. *The columns of $\mathbf{Q}_2$ are an orthonormal basis for $N(\mathbf{A}^T)$.*
3. *The matrix $\mathbf{R}_1$ is nonsingular.*

Now, suppose that we want to solve the least squares problem,

$$\min \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2. \tag{A.110}$$

Since multiplying a vector by an orthogonal matrix does not change its length, this is equivalent to

$$\min \|\mathbf{Q}^T(\mathbf{A}\mathbf{x} - \mathbf{b})\|_2. \tag{A.111}$$

But

$$\mathbf{Q}^T \mathbf{A} = \mathbf{Q}^T \mathbf{Q} \mathbf{R} = \mathbf{R}. \tag{A.112}$$

So, we have

$$\min \|\mathbf{R}\mathbf{x} - \mathbf{Q}^T \mathbf{b}\|_2 \tag{A.113}$$

or

$$\min \left\| \begin{matrix} \mathbf{R}_1 \mathbf{x} - \mathbf{Q}_1^T \mathbf{b} \\ \mathbf{0}\mathbf{x} - \mathbf{Q}_2^T \mathbf{b} \end{matrix} \right\|_2. \tag{A.114}$$

Whatever value of $\mathbf{x}$ we pick, we will probably end up with nonzero residuals because of the $\mathbf{0}\mathbf{x} - \mathbf{Q}_2^T \mathbf{b}$ part of the least squares problem. We cannot minimize the norm of this part of the vector. However, we can find an $\mathbf{x}$ that exactly solves $\mathbf{R}_1 \mathbf{x} = \mathbf{Q}_1^T \mathbf{b}$. Thus, we can minimize the least squares problem by solving the square system of equations,

$$\mathbf{R}_1 \mathbf{x} = \mathbf{Q}_1^T \mathbf{b}. \tag{A.115}$$

The advantage of solving this system of equations instead of the normal equations (A.73) is that the normal equations are typically much more badly conditioned than (A.115).

A.10. COMPLEX MATRICES AND VECTORS

Although nearly all of the mathematics in this textbook involves real numbers, complex numbers do appear in Chapter 8 when we consider the Fourier transform. We assume that the reader is already familiar with arithmetic involving complex numbers including addition, subtraction, multiplication, division, and complex exponentials. Most theorems of linear algebra extend trivially from real to complex vectors and matrices. In this section we briefly discuss our notation and some important differences between the real and complex cases.

Given a complex number $z = a + b\iota$, where ι is the $\sqrt{-1}$, the **complex conjugate** of z is $z^* = a - b\iota$. Note that the absolute value of z is

$$|z| = \sqrt{a^2 + b^2} = \sqrt{z^* z}. \tag{A.116}$$

The main difference between linear algebra on real vectors and complex vectors is in the definition of the dot product of two vectors. We define the dot product of two complex vectors $\mathbf{x}$ and $\mathbf{y}$ to be

$$\mathbf{x} \cdot \mathbf{y} = \mathbf{x}^{*^T} \mathbf{y}. \tag{A.117}$$

The advantage of this definition is that

$$\mathbf{x}^{*^T} \mathbf{x} = \sum_{k=1}^{n} x_k^* x_k = \sum_{k=1}^{n} |x_k|^2. \tag{A.118}$$

Thus we can then define the 2-norm of a complex vector by

$$\|x\|_2 = \sqrt{\mathbf{x}^{*^T} \mathbf{x}}. \tag{A.119}$$

The combination of taking the complex conjugate and transpose, called the **Hermitian transpose**, occurs so frequently that we denote this by

$$\mathbf{x}^H = \mathbf{x}^{*^T}. \tag{A.120}$$

Note that for a real vector, $\mathbf{x}$, the conjugate is simply $\mathbf{x}^* = \mathbf{x}$, so $\mathbf{x}^H = \mathbf{x}^T$. In MATLAB, the apostrophe denotes the Hermitian transpose.

In general, you will almost never go wrong by using the Hermitian transpose in any linear algebra computation involving complex numbers that would normally involve a transpose when working with real vectors and matrices. For example, if we want to minimize $\|\mathbf{Gm} - \mathbf{d}\|_2$, where $\mathbf{G}$, $\mathbf{m}$, and $\mathbf{d}$ are complex, we can solve the normal equations,

$$\mathbf{G}^H \mathbf{Gm} = \mathbf{G}^H \mathbf{d}. \tag{A.121}$$

A.11. LINEAR ALGEBRA IN SPACES OF FUNCTIONS

So far, we have considered only vectors in R^n. The concepts of linear algebra can be extended to other contexts. In general, as long as the objects that we want to consider can be multiplied by scalars and added together, and as long as they obey the laws of vector algebra, then we have a **vector space** in which we can practice linear algebra. If we can also define a vector product similar to the dot product, then we have what is called an **inner product space**, and we can define orthogonality, projections, and the 2-norm.

There are many different vector spaces used in various areas of science and mathematics. For our work in inverse problems, a very commonly used vector space is the space of functions defined on an interval $[a, b]$.

Multiplying a scalar times a function or adding two functions together clearly produces another function. In this space, the function $z(x) = 0$ takes the place of the **0** vector, since $f(x) + z(x) = f(x)$. Two functions $f(x)$ and $g(x)$ are linearly independent if the only solution to

$$c_1 f(x) + c_2 g(x) = z(x) \tag{A.122}$$

is $c_1 = c_2 = 0$.

We can define the dot product of two functions f and g to be

$$f \cdot g = \int_a^b f(x)g(x) \ dx. \tag{A.123}$$

Another commonly used notation for this dot product or **inner product** of f and g is

$$f \cdot g = \langle f, g \rangle. \tag{A.124}$$

It is easy to show that this inner product has all of the algebraic properties of the dot product of two vectors in R^n. A more important motivation for defining the dot product in this way is that it leads to a useful definition of the 2-norm of a function. Following our earlier formula that $\|x\|_2 = \sqrt{x^T x}$, we have

$$\|f\|_2 = \sqrt{\int_a^b f(x)^2 \ dx}. \tag{A.125}$$

Using this definition, the distance between two functions f and g is

$$\|f - g\|_2 = \sqrt{\int_a^b (f(x) - g(x))^2 \ dx}. \tag{A.126}$$

This measure is obviously 0 when $f(x) = g(x)$ everywhere, but can also be 0 when $f(x)$ and $g(x)$ differ at a finite or countably infinite set of points. The measure is only nonzero if $f(x)$ and $g(x)$ differ on an interval.

Using this inner product and norm, we can reconstruct the theory of linear algebra from R^n in our space of functions. This includes the concepts of orthogonality, projections, norms, and least squares solutions.

Definition A.32 Given a collection of functions $f_1(x), f_2(x), \ldots, f_m(x)$ in an inner product space, the **Gram matrix** of the functions is the $m \times m$ matrix $\mathbf{\Gamma}$, whose elements are given by

$$\mathbf{\Gamma}_{i, j} = f_i \cdot f_j. \tag{A.127}$$

The Gram matrix has several important properties. It is symmetric and positive semidefinite. If the functions are linearly independent, then the Gram matrix is also positive definite. Furthermore, the rank of $\mathbf{\Gamma}$ is equal to the size of the largest linearly independent subset of the functions $f_1(x), \ldots, f_m(x)$.

A.12. EXERCISES

1. Let $\mathbf{A}$ be an m by n matrix with n pivot columns in its RREF. Can the system of equations $\mathbf{Ax} = \mathbf{b}$ have infinitely many solutions?
2. If $\mathbf{C} = \mathbf{AB}$ is a 5 by 4 matrix, then how many rows does $\mathbf{A}$ have? How many columns does $\mathbf{B}$ have? Can you say anything about the number of columns in $\mathbf{A}$?
3. Suppose that $\mathbf{v}_1$, $\mathbf{v}_2$, and $\mathbf{v}_3$ are three vectors in R^3 and that $\mathbf{v}_3 = -2\mathbf{v}_1 + 3\mathbf{v}_2$. Are the vectors linearly dependent or linearly independent?
4. Let

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 2 & 2 & 1 & 3 \\ 4 & 6 & 7 & 11 \end{bmatrix}. \tag{A.128}$$

   Find bases for $N(\mathbf{A})$, $R(\mathbf{A})$, $N(\mathbf{A}^T)$, and $R(\mathbf{A}^T)$. What are the dimensions of the four subspaces?
5. Let $\mathbf{A}$ be an n by n matrix such that $\mathbf{A}^{-1}$ exists. What are $N(\mathbf{A})$, $R(\mathbf{A})$, $N(\mathbf{A}^T)$, and $R(\mathbf{A}^T)$?
6. Let $\mathbf{A}$ be any 9 by 6 matrix. If the dimension of the null space of $\mathbf{A}$ is 5, then what is the dimension of $R(\mathbf{A})$? What is the dimension of $R(\mathbf{A}^T)$? What is the rank of $\mathbf{A}$?
7. Suppose that a nonhomogeneous system of equations with four equations and six unknowns has a solution with two free variables. Is it possible to change the right-hand side of the system of equations so that the modified system of equations has no solutions?

8. Let W be the set of vectors $\mathbf{x}$ in R^4 such that $x_1 x_2 = 0$. Is W a subspace of R^4?

9. Let $\mathbf{v}_1$, $\mathbf{v}_2$, $\mathbf{v}_3$ be a set of three nonzero orthogonal vectors. Show that the vectors are also linearly independent.

10. Show that if $\mathbf{x} \perp \mathbf{y}$, then

$$\|\mathbf{x} + \mathbf{y}\|_2^2 = \|\mathbf{x}\|_2^2 + \|\mathbf{y}\|_2^2. \tag{A.129}$$

11. In this exercise, we will derive the formula (A.88) for the 1-norm of a matrix. Begin with the optimization problem

$$\|\mathbf{A}\|_1 = \max_{\|\mathbf{x}\|_1 = 1} \|\mathbf{A}\mathbf{x}\|_1. \tag{A.130}$$

 (a) Show that if $\|\mathbf{x}\|_1 = 1$, then

$$\|\mathbf{A}\mathbf{x}\|_1 \le \max_j \sum_{i=1}^m |A_{i,j}|. \tag{A.131}$$

 (b) Find a vector $\mathbf{x}$ such that $\|\mathbf{x}\|_1 = 1$, and

$$\|\mathbf{A}\mathbf{x}\|_1 = \max_j \sum_{i=1}^m |A_{i,j}|. \tag{A.132}$$

 (c) Conclude that

$$\|\mathbf{A}\|_1 = \max_{\|\mathbf{x}\|_1 = 1} \|\mathbf{A}\mathbf{x}\|_1 = \max_j \sum_{i=1}^m |A_{i,j}|. \tag{A.133}$$

12. Derive the formula (A.90) for the infinity norm of a matrix.

13. Let $\mathbf{A}$ be an m by n matrix.
 (a) Show that $\mathbf{A}^T \mathbf{A}$ is symmetric.
 (b) Show that $\mathbf{A}^T \mathbf{A}$ is positive semidefinite. Hint: Use the definition of positive semidefinite rather than trying to compute eigenvalues.
 (c) Show that if $\operatorname{rank}(\mathbf{A}) = n$, then the only solution to $\mathbf{A}\mathbf{x} = \mathbf{0}$ is $\mathbf{x} = \mathbf{0}$.
 (d) Use part c to show that if $\operatorname{rank}(\mathbf{A}) = n$, then $\mathbf{A}^T \mathbf{A}$ is positive definite.
 (e) Use part d to show that if $\operatorname{rank}(\mathbf{A}) = n$, then $\mathbf{A}^T \mathbf{A}$ is nonsingular.
 (f) Show that $N(\mathbf{A}^T \mathbf{A}) = N(\mathbf{A})$.

14. Show that

$$\operatorname{cond}(\mathbf{A}\mathbf{B}) \le \operatorname{cond}(\mathbf{A})\operatorname{cond}(\mathbf{B}). \tag{A.134}$$

15. Let $\mathbf{A}$ be a symmetric and positive definite matrix with Cholesky factorization,

$$\mathbf{A} = \mathbf{R}^T \mathbf{R}. \tag{A.135}$$

Show how the Cholesky factorization can be used to solve $\mathbf{Ax} = \mathbf{b}$ by solving two systems of equations, each of which has $\mathbf{R}$ or $\mathbf{R}^T$ as its matrix.

16. Let $P_3[0, 1]$ be the space of polynomials of degree less than or equal to 3 on the interval $[0, 1]$. The polynomials $p_1(x) = 1$, $p_2(x) = x$, $p_3(x) = x^2$, and $p_4(x) = x^3$ form a basis for $P_3[0, 1]$, but they are not orthogonal with respect to the inner product,

$$f \cdot g = \int_0^1 f(x)g(x) \ dx. \tag{A.136}$$

Use the Gram–Schmidt orthogonalization process to construct an orthogonal basis for $P_3[0, 1]$. Once you have your basis, use it to find the third-degree polynomial that best approximates $f(x) = e^{-x}$ on the interval $[0, 1]$ in the least squares sense.

A.13. NOTES AND FURTHER READING

Much of this material is typically covered in sophomore-level linear algebra courses, and there are an enormous number of textbooks at this level. One good introductory linear algebra textbook is [95]. At a slightly more advanced level, [109] and [152] are both excellent. The book by Strang and Borre [153] reviews linear algebra in the context of geodetic problems.

Fast and accurate algorithms for linear algebra computations are a somewhat more advanced and specialized topic. A classic reference is [53]. Other good books on this topic include [38] and [164].

The extension of linear algebra to spaces of functions is a topic in the subject of functional analysis. Unfortunately, most textbooks on functional analysis assume that the reader has a considerable mathematical background. One book that is reasonably accessible to readers with limited mathematical backgrounds is [102].

Review of Probability and Statistics

Synopsis

A brief review is given of the topics in classical probability and statistics that are used throughout the book. Connections between probability theory and its application to the analysis of data with random measurement errors are highlighted. Note that some very different philosophical interpretations of probability theory are discussed in Chapter 11.

B.1. PROBABILITY AND RANDOM VARIABLES

The mathematical theory of probability begins with an **experiment**, which has a set S of possible outcomes. We will be interested in **events** that are subsets A of S.

Definition B.1 The **probability function** P is a function defined on subsets of S with the following properties:

1. $P(S) = 1$.
2. For every event $A \subseteq S$, $P(A) \geq 0$.
3. If events $A_1, A_2, \ldots, A_n$ are pairwise mutually exclusive (i.e., impossible to both occur), so that $A_i \cap A_j$ is empty for all pairs i, j, then

$$P(\cup_{i=1}^{n} A_i) = \sum_{i=1}^{n} P(A_i). \tag{B.1}$$

The probability properties given above are fundamental to developing the mathematics of probability theory. However, applying this definition of probability to real–world situations frequently requires ingenuity.

●────────────────────────────────────

Example B.1

Consider the experiment of throwing a dart at a dart board. We will assume that our dart thrower is an expert who always hits the dart board. The sample space S consists of the points on the dart board. We can define an event A that consists of the points in the bullseye, so that $P(A)$ is the probability that the thrower hits the bullseye.

────────────────────────────────────

In practice, the outcome of an experiment is often a number rather than an event. Random variables are a useful generalization of the basic concept of probability.

Definition B.2 A **random variable** X is a function $X(s)$ that assigns a value to each outcome s in the sample space S.

Each time that we perform an experiment, we obtain a particular value of the random variable. These values are called **realizations** of the random variable.

●───

Example B.2

To continue our previous example, let X be the function that takes a point on the dart board and returns the associated score. Suppose that throwing the dart in the bullseye scores 50 points. Then for each point s in the bullseye, $X(s) = 50$.

───

In this book we deal frequently with experimental measurements that can include some random measurement error.

●───

Example B.3

Suppose that we measure the mass of an object five times to obtain the realizations $m_1 = 10.1$ kg, $m_2 = 10.0$ kg, $m_3 = 10.0$ kg, $m_4 = 9.9$ kg, and $m_5 = 10.1$ kg. We will assume that there is one true mass m, and that the measurements we obtained varied because of random measurement errors e_i, so that

$$m_1 = m + e_1, \quad m_2 = m + e_2, \quad m_3 = m + e_3, \quad m_4 = m + e_4, \quad m_5 = m + e_5. \qquad \text{(B.2)}$$

We can treat the measurement errors as realizations of a random variable E. Equivalently, since the true mass m is just a constant, we could treat the measurements $m_1, m_2, \ldots, m_5$ as realizations of a random variable M. In practice it makes little difference whether we treat the measurements or the measurement errors as random variables.

Note that in a Bayesian approach the mass m of the object would itself be a random variable. This is a viewpoint that we consider in Chapter 11.

───

The relative probability of realization values for a random variable can be characterized by a non-negative **probability density function (PDF)**, $f_X(x)$, with

$$P(X \le a) = \int_{-\infty}^{a} f_X(x) \, dx. \qquad \text{(B.3)}$$

Because a random variable always has some value,

$$\int_{-\infty}^{\infty} f_X(x) \ dx = 1. \tag{B.4}$$

The following definitions give some useful random variables that frequently arise in inverse problems.

Definition B.3 The **uniform** random variable on the the interval $[a, b]$ (Figure B.1) has the following PDF:

$$f_U(x) = \begin{cases} \frac{1}{b-a} & a \leq x \leq b \\ 0 & x < a \\ 0 & x > b. \end{cases} \tag{B.5}$$

Definition B.4 The **normal** or **Gaussian** random variable (Figure B.2) has the following PDF:

$$f_N(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}(x-\mu)^2/\sigma^2}. \tag{B.6}$$

The notation $N(\mu, \sigma^2)$ is used to denote a normal distribution with parameters μ and σ. The **standard normal** random variable, $N(0, 1)$, has $\mu = 0$ and $\sigma^2 = 1$.
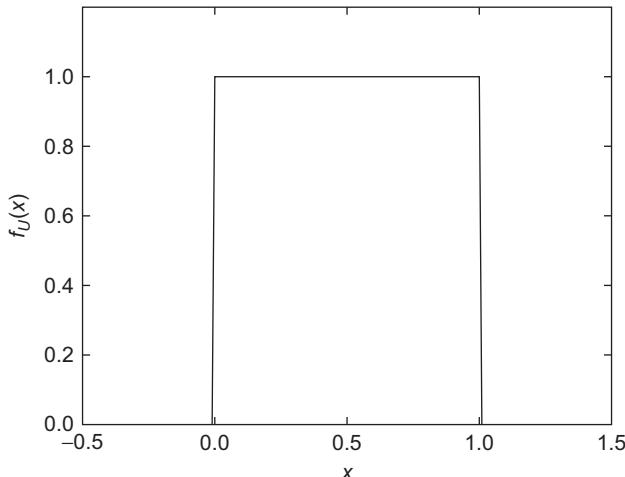
Figure B.1 The PDF for the uniform random variable on [0, 1].
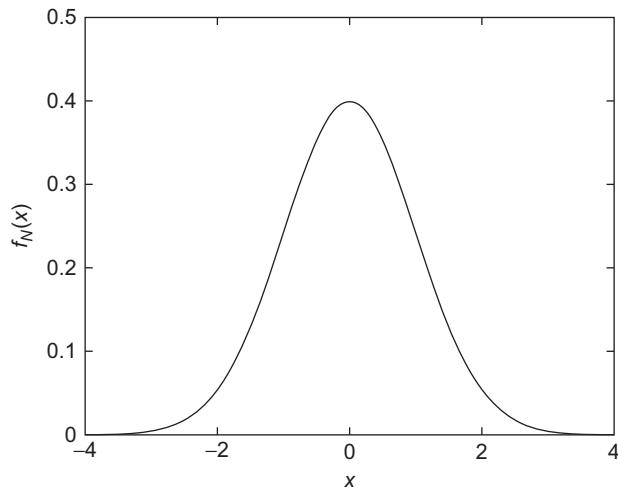
Figure B.2 The PDF of the standard normal random variable.
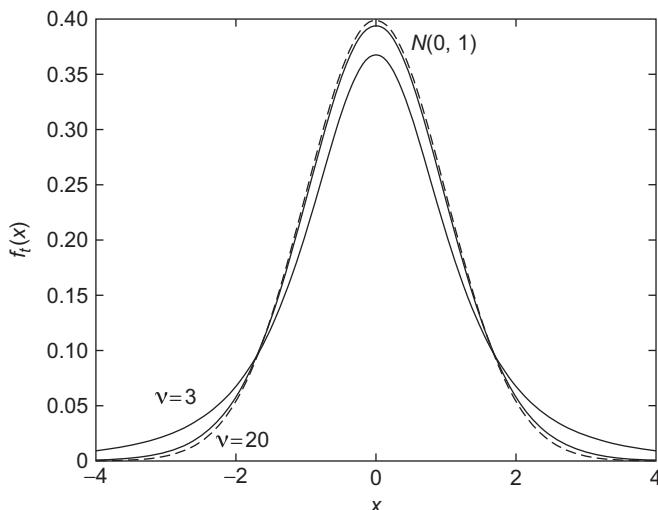
Figure B.3 The Student's t probability density function for $v = 3$ and $v = 20$. Dashed curve shows the normal distribution $N(0, 1)$ (Figure B 2).

Definition B.5 The **Student's t distribution** with v degrees of freedom (Figure B.3) has the PDF

$$f_t(x) = \frac{\Gamma((v + 1)/2)}{\Gamma(v/2)} \frac{1}{\sqrt{v\pi}} \left(1 + \frac{x^2}{v}\right)^{-(v+1)/2}, \tag{B.7}$$

where the **gamma function** is

$$\Gamma(x) = \int_0^\infty \xi^{x-1} e^{-\xi} \, d\xi. \tag{B.8}$$

The Student's t distribution is so named because W. S. Gosset used the pseudonym "Student" in publishing the first paper in which the distribution appeared. In the limit as ν goes to infinity, Student's t distribution approaches a standard normal distribution. However, for small values of ν, (B.7) has a greater percentage of extreme values than (B.6).

Definition B.6 The χ^2 random variable has the PDF (Figure B.4)

$$f_{\chi^2}(x) = \frac{1}{2^{\nu/2} \Gamma(\nu/2)} x^{\frac{1}{2}\nu - 1} e^{-x/2}, \tag{B.9}$$

where the parameter ν is the **number of degrees of freedom**.

Definition B.7 The **exponential** random variable (Figure B.5) has the PDF

$$f_{exp}(x) = \begin{cases} \lambda e^{-\lambda x} & x \geq 0 \\ 0 & x < 0. \end{cases} \tag{B.10}$$

Definition B.8 The **double–sided exponential** random variable (Figure B.6) has the PDF

$$f_{dexp}(x) = \frac{1}{\sigma \sqrt{2}} e^{-\sqrt{2}|x-\mu|/\sigma}. \tag{B.11}$$
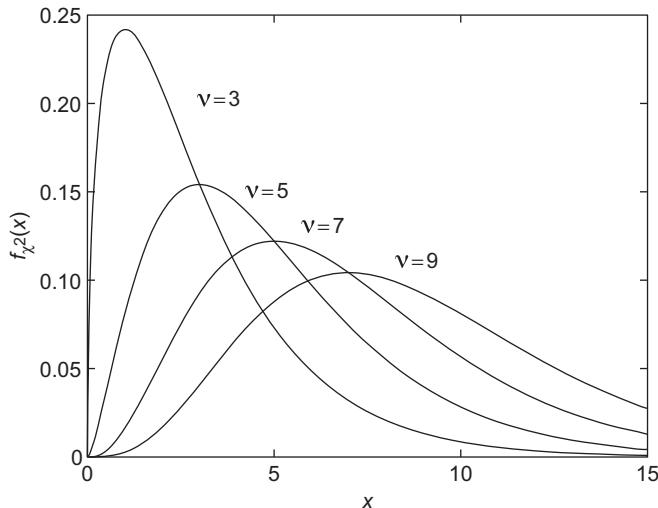
Figure B.4 The χ^2 PDF for several values of ν.
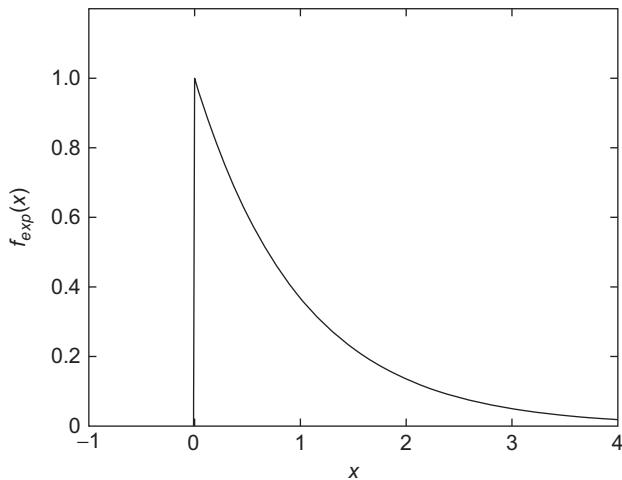
Figure B.5   The exponential PDF ($\lambda = 1$).
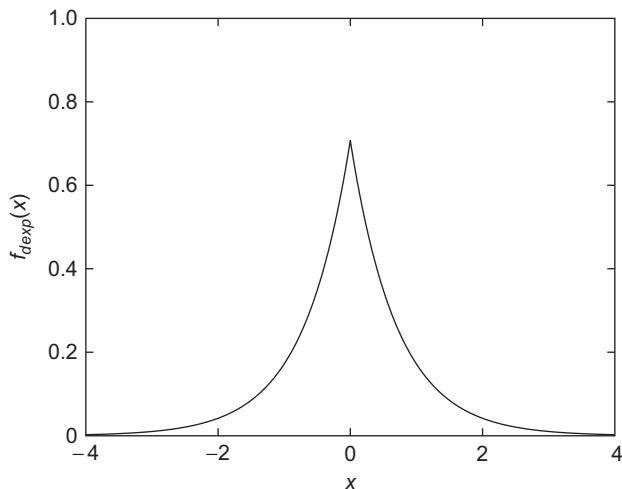
Figure B.6   The double-sided exponential PDF ($\mu = 0$, $\sigma = 1$).

It can be shown that for n independent random variables, X_i, with standard normal distributions, the random variable

$$Z = \sum_{i=1}^{n} X_i^2 \tag{B.12}$$

is a χ^2 random variable with $v = n$ degrees of freedom [47].

The **cumulative distribution function (CDF)** $F_X(a)$ of a one-dimensional random variable X is given by the definite integral of the associated PDF

$$F_X(a) = P(X \leq a) = \int_{-\infty}^{a} f_X(x) \ dx. \tag{B.13}$$

Note that $F_X(a)$ must lie in the interval $[0, 1]$ for all a, and is a nondecreasing function of a because of the unit area and non-negativity of the PDF.

For the uniform PDF on the unit interval, for example, the CDF is a ramp function,

$$F_U(a) = \int_{-\infty}^{a} f_u(z) \ dz \tag{B.14}$$

$$F_U(a) = \begin{cases} 0 & a \leq 0 \\ a & 0 \leq a \leq 1 \\ 1 & a > 1. \end{cases} \tag{B.15}$$

The PDF, $f_X(x)$, or CDF, $F_X(a)$, completely determine the properties of a random variable. The probability that a particular realization of X will lie within a general interval $[a, b]$ is

$$P(a \leq X \leq b) = P(X \leq b) - P(X \leq a) = F_X(b) - F_X(a) \tag{B.16}$$

$$= \int_{-\infty}^{b} f_X(x) \ dx - \int_{-\infty}^{a} f_X(x) \ dx = \int_{a}^{b} f_X(x) \ dx. \tag{B.17}$$

B.2. EXPECTED VALUE AND VARIANCE

Definition B.9 The **expected value** of a random variable X, denoted by $E[X]$ or $\mu(X)$, is

$$E[X] = \int_{-\infty}^{\infty} x \, f_X(x) \ dx. \tag{B.18}$$

In general, if $g(X)$ is some function of a random variable X, then

$$E[g(X)] = \int_{-\infty}^{\infty} g(x) f_X(x) \ dx. \tag{B.19}$$

Some authors use the term "mean" for the expected value of a random variable. We will reserve this term for the average of a set of data. Note that the expected value of

a random variable is not necessarily identical to the **mode** (the value with the largest value of $f(x)$) nor is it necessarily identical to the **median**, the value of x for which the value of the CDF is $F_X(x) = 1/2$.

● ───

Example B.4

The expected value of an $N(\mu, \sigma)$ random variable X is

$$E[X] = \int_{-\infty}^{\infty} x \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \, dx \tag{B.20}$$

$$= \int_{-\infty}^{\infty} \frac{1}{\sigma\sqrt{2\pi}} (x+\mu) e^{-\frac{x^2}{2\sigma^2}} \, dx \tag{B.21}$$

$$= \mu \int_{-\infty}^{\infty} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}} \, dx + \int_{-\infty}^{\infty} \frac{1}{\sigma\sqrt{2\pi}} x e^{-\frac{x^2}{2\sigma^2}} \, dx. \tag{B.22}$$

The first integral term is μ because the integral of the entire PDF is 1, and the second term is 0 because it is an odd function integrated over a symmetric interval. Thus,

$$E[X] = \mu. \tag{B.23}$$

───

Definition B.10 The **variance** of a random variable X, denoted by $\mathrm{Var}(X)$ or σ_X^2, is given by

$$\mathrm{Var}(X) = E[(X - E[X])^2]$$
$$= E[X^2] - E[X]^2$$
$$= \int_{-\infty}^{\infty} (x - E[X])^2 f_X(x) \, dx. \tag{B.24}$$

The **standard deviation** of X, often denoted σ_X, is

$$\sigma_X = \sqrt{\mathrm{Var}(X)}. \tag{B.25}$$

The variance and standard deviation serve as measures of the spread of the random variable about its expected value. Since the units of σ are the same as the units of μ, the standard deviation is generally more practical for reporting a spread measure. However, the variance has properties that make it more useful for certain calculations.

B.3. JOINT DISTRIBUTIONS

Definition B.11 If we have two random variables X and Y, they *may* have a **joint probability density function (JDF)**, $f(x, y)$, with

$$P(X \leq a \text{ and } Y \leq b) = \int_{-\infty}^{a} \int_{-\infty}^{b} f(x, y) \, dy \, dx. \tag{B.26}$$

If X and Y have a JDF, then we can use it to evaluate the expected value of a function of X and Y. The expected value of $g(X, Y)$ is

$$E[g(X, Y)] = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} g(x, y)f(x, y) \, dy \, dx. \tag{B.27}$$

Definition B.12 Two random variables X and Y are **independent** if a JDF exists and is defined by

$$f(x, y) = f_X(x)f_Y(y). \tag{B.28}$$

Definition B.13 If X and Y have a JDF, then the **covariance** of X and Y is

$$\text{Cov}(X, Y) = E[(X - E[X])(Y - E[Y])] = E[XY] - E[X]E[Y]. \tag{B.29}$$

If X and Y are independent, then $E[XY] = E[X]E[Y]$, and $\text{Cov}(X, Y) = 0$. However, if X and Y are dependent, it is still possible, given some particular distributions, for X and Y to have $\text{Cov}(X, Y) = 0$. If $\text{Cov}(X, Y) = 0$, X and Y are called **uncorrelated**.

Definition B.14 The **correlation** of X and Y is

$$\rho(X, Y) = \frac{\text{Cov}(X, Y)}{\sqrt{\text{Var}(X)\text{Var}(Y)}}. \tag{B.30}$$

Correlation is thus a scaled covariance.

Theorem B.1 *The following properties of Var, Cov, and correlation hold for any random variables X and Y and scalars s and a.*
1. *$Var(X) \geq 0$.*
2. *$Var(X + a) = Var(X)$.*
3. *$Var(sX) = s^2 Var(X)$.*
4. *$Var(X + Y) = Var(X) + Var(Y) + 2Cov(X, Y)$.*
5. *$Cov(X, Y) = Cov(Y, X)$.*
6. *$\rho(X, Y) = \rho(Y, X)$.*
7. *$-1 \leq \rho(X, Y) \leq 1$.*

The following example demonstrates the use of some of these properties.

Example B.5

Suppose that Z is a standard normal random variable. Let

$$X = \mu + \sigma Z. \tag{B.31}$$

Then

$$E[X] = E[\mu] + \sigma E[Z] \tag{B.32}$$

so

$$E[X] = \mu. \tag{B.33}$$

Also,

$$\text{Var}(X) = \text{Var}(\mu) + \sigma^2 \text{Var}(Z) = \sigma^2. \tag{B.34}$$

Thus, if we have a program to generate random numbers with the standard normal distribution, we can use it to generate normal random numbers with any desired expected value and standard deviation. The MATLAB command **randn** generates independent realizations of an $N(0, 1)$ random variable.

Example B.6

What is the CDF (or PDF) of the sum of two independent random variables $X + Y$? To see this, we write the desired CDF in terms of an appropriate integral over the JDF, $f(x, y)$, which gives (Figure B.7):

$$F_{X+Y}(z) = P(X + Y \leq z) \tag{B.35}$$

$$= \iint\limits_{x+y \leq z} f(x, y) \, dx \, dy \tag{B.36}$$

$$= \iint\limits_{x+y \leq z} f_X(x) f_Y(y) \, dx \, dy \tag{B.37}$$

$$= \int\limits_{-\infty}^{\infty} \int\limits_{-\infty}^{z-y} f_X(x) f_Y(y) \, dx \, dy \tag{B.38}$$

$$= \int\limits_{-\infty}^{\infty} \int\limits_{-\infty}^{z-y} f_X(x) \, dx \, f_Y(y) \, dy \tag{B.39}$$

$$= \int_{-\infty}^{\infty} F_X(z-y)f_Y(y) \; dy. \tag{B.40}$$

The associated PDF is

$$f_{X+Y}(z) = \frac{d}{dz} \int_{-\infty}^{\infty} F_X(z-y)f_Y(y) \; dy \tag{B.41}$$

$$= \int_{-\infty}^{\infty} \frac{d}{dz} F_X(z-y)f_Y(y) \; dy \tag{B.42}$$

$$= \int_{-\infty}^{\infty} f_X(z-y)f_Y(y) \; dy \tag{B.43}$$

$$= f_X(z) * f_Y(z). \tag{B.44}$$

Adding two independent random variables thus produces a new random variable that has a PDF given by the convolution of the PDFs of the two individual variables.
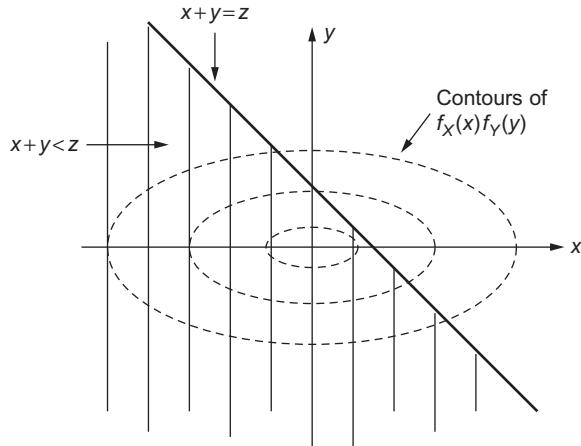
Figure B.7 Integration of a JDF for two independent random variables, X and Y, to evaluate the CDF of $Z = X + Y$.

The JDF can be used to evaluate the CDF or PDF arising from a general function of jointly distributed random variables. The process is identical to the previous example except that the specific form of the integral limits is determined by the specific function.

Example B.7

Consider the product of two independent, identically distributed, standard normal random variables,

$$Z = XY, \tag{B.45}$$

with a JDF given by

$$f(x, y) = f(x)f(y) = \frac{1}{2\pi\sigma^2}e^{-(x^2+y^2)/2\sigma^2}. \tag{B.46}$$

The CDF of Z is

$$F(z) = P(Z \le z) = P(XY \le z). \tag{B.47}$$

For $z \le 0$, this is the integral of the JDF over the exterior of the hyperbolas defined by $xy \le z \le 0$, while for $z \ge 0$, we integrate over the interior of the complementary hyperbolas $xy \le z \ge 0$. At $z = 0$, the integral covers exactly half of the (x, y) plane (the second and fourth quadrants) and, because of the symmetry of the JDF, has accumulated half of the probability, or $1/2$.

The integral is thus

$$F(z) = 2 \int_{-\infty}^{0} \int_{z/x}^{\infty} \frac{1}{2\pi\sigma^2}e^{-(x^2+y^2)/2\sigma^2} \, dy \, dx \quad (z \le 0) \tag{B.48}$$

and

$$F(z) = 1/2 + 2 \int_{-\infty}^{0} \int_{0}^{z/x} \frac{1}{2\pi\sigma^2}e^{-(x^2+y^2)/2\sigma^2} \, dy \, dx \quad (z \ge 0). \tag{B.49}$$

As in the previous example for the sum of two random variables, the PDF may be obtained from the CDF by differentiating with respect to z.

B.4. CONDITIONAL PROBABILITY

In some situations we will be interested in the probability of an event happening given that some other event has also happened.

Definition B.15 The **conditional probability** of A given that B has occurred is given by

$$P(A|B) = \frac{P(A \cap B)}{P(B)}. \tag{B.50}$$

Arguments based on conditional probabilities are often very helpful in computing probabilities. The key to such arguments is the **law of total probability**.

Theorem B.2 *Suppose that B_1, B_2, . . . , B_n are mutually disjoint and exhaustive events. That is, $B_i \cap B_j = \emptyset$ (the empty set) for $i \neq j$, and*

$$\cup_{i=1}^{n} B_i = S. \tag{B.51}$$

Then

$$P(A) = \sum_{i=1}^{n} P(A|B_i)P(B_i). \tag{B.52}$$

It is often necessary to reverse the order of conditioning in a conditional probability. Bayes' theorem provides a way to do this.

Theorem B.3 Bayes Theorem

$$P(B|A) = \frac{P(A|B)P(B)}{P(A)}. \tag{B.53}$$

⬤───

Example B.8

A screening test has been developed for a very serious but rare disease. If a person has the disease, then the test will detect the disease with probability 99%. If a person does not have the disease, then the test will give a false–positive detection with probability 1%. The probability that any individual in the population has the disease is 0.01%. Suppose that a randomly selected individual tests positive for the disease. What is the probability that this individual actually has the disease?

Let A be the event "the person tests positive." Let B be the event "the person has the disease." We then want to compute $P(B|A)$. By Bayes theorem,

$$P(B|A) = \frac{P(A|B)P(B)}{P(A)}. \tag{B.54}$$

We have that $P(A|B)$ is 0.99, and that $P(B)$ is 0.0001. To compute $P(A)$, we apply the law of total probability, considering separately the probability of a diseased individual testing positive and the probability of someone without the disease testing positive:

$$P(A) = 0.99 \cdot 0.0001 + 0.01 \cdot 0.9999 = 0.010098. \tag{B.55}$$

Thus

$$P(B|A) = \frac{0.99 \cdot 0.0001}{0.010098} = 0.0098. \tag{B.56}$$

In other words, even after a positive screening test, it is still unlikely that the individual will have the disease. The vast majority of those individuals who test positive will in fact not have the disease.

The concept of conditioning can be extended from simple events to distributions and expected values of random variables. If the distribution of X depends on the value of Y, then we can work with the **conditional PDF** $f_{X|Y}(x)$, the **conditional CDF** $F_{X|Y}(a)$, and the **conditional expected value** $E[X|Y]$.

In this notation, we can also specify a particular value of Y by using the notation $f_{X|Y=y}$, $F_{X|Y=y}$, or $E[X|Y=y]$. In working with conditional distributions and expected values, the following versions of the law of total probability can be very useful.

Theorem B.4 *Given two random variables X and Y, with the distribution of X depending on Y, we can compute*

$$P(X \le a) = \int_{-\infty}^{\infty} P(X \le a|Y = y)f_Y(y) \, dy \tag{B.57}$$

and

$$E[X] = \int_{-\infty}^{\infty} E[X|Y = y] \, f_Y(y) \, dy. \tag{B.58}$$

Example B.9

Let U be a random variable uniformly distributed on $(1, 2)$. Let X be an exponential random variable with parameter $\lambda = U$. We will find the expected value of X:

$$E[X] = \int_{1}^{2} E[X|U = u] \, f_U(u) \, du. \tag{B.59}$$

Since the expected value of an exponential random variable with parameter λ is $1/\lambda$, and the PDF of a uniform random variable on $(1, 2)$ is $f_U(u) = 1$,

$$E[X] = \int_{1}^{2} \frac{1}{u} du = \log 2. \tag{B.60}$$

B.5. THE MULTIVARIATE NORMAL DISTRIBUTION

Definition B.16 If the random variables $X_1, \ldots, X_n$ have a **multivariate normal distribution** (MVN), then the JDF is

$$f(\mathbf{x}) = \frac{1}{(2\pi)^{n/2}} \frac{1}{\sqrt{\det(\mathbf{C})}} e^{-(\mathbf{x}-\mu)^T \mathbf{C}^{-1}(\mathbf{x}-\mu)/2} \tag{B.61}$$

where $\mathbf{x} = [X_1, X_2, \ldots, X_n]^T$ and $\mu = [\mu_1, \mu_2, \ldots, \mu_n]^T$ is a vector containing the expected values along each of the coordinate directions of $X_1, \ldots, X_n$, and $\mathbf{C}$ contains the covariances between the random variables,

$$C_{i,j} = \text{Cov}(X_i, X_j). \tag{B.62}$$

Note that if $\mathbf{C}$ is singular, then the JDF involves a division by zero, and is simply not defined.

The vector μ and the covariance matrix $\mathbf{C}$ completely characterize the MVN distribution. There are other multivariate distributions that are not completely characterized by the expected values and covariance matrix.

Theorem B.5 *Let* $\mathbf{X}$ *be a multivariate normal random vector with expected values defined by the vector* μ *and covariance matrix* $\mathbf{C}$*, and let* $\mathbf{Y} = \mathbf{AX}$*. Then* $\mathbf{Y}$ *is also multivariate normal, with*

$$E[\mathbf{Y}] = \mathbf{A}\mu \tag{B.63}$$

and

$$\text{Cov}(\mathbf{Y}) = \mathbf{ACA}^T. \tag{B.64}$$

Theorem B.6 *If we have an n-dimensional MVN distribution* $\mathbf{X}$ *with covariance matrix* $\mathbf{C}$ *and expected value* μ*, and the covariance matrix is of full rank, then the quantity*

$$Z = (\mathbf{X} - \mu)^T \mathbf{C}^{-1}(\mathbf{X} - \mu) \tag{B.65}$$

has a χ^2 *distribution with n degrees of freedom.*

● ───

Example B.10
We can generate vectors that are realizations of an MVN distribution with a known mean, μ, and covariance matrix, $\mathbf{C}$, as follows.
1. Find the lower-triangular Cholesky factorization $\mathbf{C} = \mathbf{LL}^T$.
2. Let $\mathbf{Z}$ be a vector of n independent $N(0, 1)$ random numbers.
3. Let $\mathbf{X} = \mu + \mathbf{LZ}$.

We can easily show that this procedure produces the desired distribution. Because $E[\mathbf{Z}] = \mathbf{0}$, $E[\mathbf{X}] = \mu + \mathbf{L0} = \mu$. Also, since $\text{Cov}(\mathbf{Z}) = \mathbf{I}$ and $\text{Cov}(\mu) = \mathbf{0}$, $\text{Cov}(\mathbf{X}) = \text{Cov}(\mu + \mathbf{LZ}) = \text{Cov}(\mathbf{LZ}) = \mathbf{LIL}^T = \mathbf{C}$ using (B.64).

B.6. THE CENTRAL LIMIT THEOREM

Theorem B.7 *Let X_1, X_2, . . . , X_n be independent and identically distributed (IID) random variables with a finite expected value μ and variance σ^2. Let*

$$Z_n = \frac{X_1 + X_2 + \cdots + X_n - n\mu}{\sqrt{n}\sigma}. \tag{B.66}$$

In the limit as n approaches infinity, the distribution of Z_n approaches the standard normal distribution.

The central limit theorem shows why quasi-normally distributed random variables appear so frequently in nature; the sum of numerous independent random variables produces an approximately normal random variable, regardless of the distribution of the underlying IID variables. In particular, this is one reason that measurement errors are often normally distributed. As we will see in Chapter 2, having normally distributed measurement errors leads us to consider least squares solutions to parameter estimation and inverse problems.

B.7. TESTING FOR NORMALITY

Many of the statistical procedures that we will use assume that data are normally distributed. Fortunately, the statistical techniques that we describe are generally robust in the face of small deviations from normality. Large deviations from the normal distribution can cause problems. Thus, it is important to be able to examine a data set to see whether the distribution is approximately normal.

Plotting a histogram of the data provides a quick view of the distribution. The histogram should show a roughly "bell-shaped" distribution, symmetrical around a single peak. If the histogram shows that the distribution is obviously skewed, then it would be unwise to assume that the data are normally distributed.

The **Q-Q plot** provides a more precise graphical test of whether a set of data could have come from a particular distribution. The data points,

$$\mathbf{d} = [d_1, \ d_2, \ \ldots, \ d_n]^T, \tag{B.67}$$

are first sorted in numerical order from smallest to largest into a vector $\mathbf{y}$, which is plotted versus

$$x_i = F^{-1}((i - 0.5)/n) \quad (i = 1, 2, \ldots, n), \tag{B.68}$$

where $F(x)$ is the CDF of the distribution against which we wish to compare our observations.

If we are testing to see if the elements of $\mathbf{d}$ could have come from the normal distribution, then $F(x)$ is the CDF for the standard normal distribution:

$$F_N(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-\frac{1}{2}z^2} dz. \tag{B.69}$$

If the elements of $\mathbf{d}$ are normally distributed, the Q-Q plot points (y_i, x_i) will follow a straight line with a slope and intercept determined by the standard deviation and expected value, respectively, of the normal distribution that produced the data.

● ─────────────────────────────────────

Example B.11

Figure B.8 shows the histogram from a set of 1000 data points. The characteristic bell-shaped curve in the histogram might make it appear at first that these data are normally distributed. The sample mean is -0.01 and the sample standard deviation is 1.41.

Figure B.9 shows the Q-Q plot for these data, which makes it apparent that the data set contains substantially more extreme values than the normal distribution
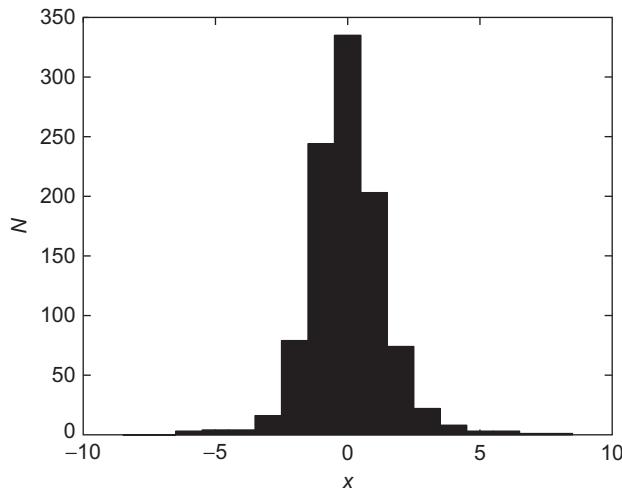
Figure B.8 Histogram of a sample data set.
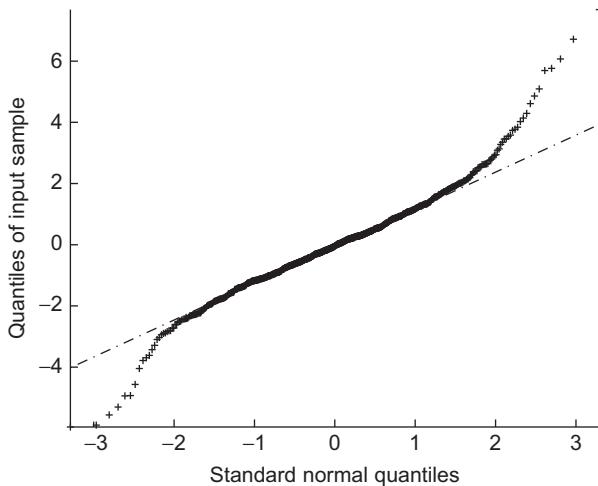
Figure B.9   *Q-Q* plot for the sample data set.

would predict. In fact, these data were generated according to a *t* distribution with 5 degrees of freedom, that has broader tails than the normal distribution (Figure B.3).

There are a number of statistical tests for normality. These include the Kolmogorov-Smirnov test, Anderson-Darling test, and Lilliefors test. Each produces probabilistic measures called **p-values**. A small *p*-value indicates that the observed data would be unlikely if the distribution were in fact normal, while a larger *p*-value is consistent with normality.

B.8. ESTIMATING MEANS AND CONFIDENCE INTERVALS

Given a collection of *n* noisy measurements m_1, m_2, . . . , m_n of some quantity of interest, how can we estimate the true value *m*, and how uncertain is this estimate? This is a classic problem in statistics.

We will assume first that the measurement errors are independent and normally distributed with expected value 0 and some unknown standard deviation σ. Equivalently, the measurements themselves are normally distributed with expected value *m* and standard deviation σ.

We begin by computing the measurement average,

$$\bar{m} = \frac{m_1 + m_2 + \cdots + m_n}{n}. \tag{B.70}$$

This **sample mean** $\bar{m}$ will serve as our estimate of m. We will also compute an estimate s of the standard deviation

$$s = \sqrt{\frac{\sum_{i=1}^{n}(m_i - \bar{m})^2}{n-1}}. \tag{B.71}$$

The key to our approach to estimating m is the following theorem.

Theorem B.8 *(The Sampling Theorem) Under the assumption that measurements are independent and normally distributed with expected value m and standard deviation σ, the random quantity*

$$t = \frac{m - \bar{m}}{s/\sqrt{n}} \tag{B.72}$$

has a **Student's t distribution** *with $n-1$ degrees of freedom.*

If we had the true standard deviation σ instead of the estimate s, then t would in fact be normally distributed with expected value 0 and standard deviation 1. This does not quite work out because we have used an estimate s of the standard deviation. For smaller values of n, the estimate s is less accurate, and the t distribution therefore has fatter tails than the standard normal distribution. As n becomes large, s becomes a better estimate of σ and it can be shown that the t distribution converges to a standard normal distribution [47].

Let $t_{n-1,0.975}$ be the 97.5%-tile of the t distribution and let $t_{n-1,0.025}$ be the 2.5%-tile of the t distribution. Then

$$P\left(t_{n-1,0.025} \leq \frac{m - \bar{m}}{s/\sqrt{n}} \leq t_{n-1,0.975}\right) = 0.95. \tag{B.73}$$

This can be rewritten as

$$P\left(\left(t_{n-1,0.025}s/\sqrt{n}\right) \leq (m - \bar{m}) \leq \left(t_{n-1,0.975}s/\sqrt{n}\right)\right) = 0.95. \tag{B.74}$$

We can construct the 95% **confidence interval** for m as the interval from $\bar{m} + t_{n-1,0.025}s/\sqrt{n}$ to $\bar{m} + t_{n-1,0.975}s/\sqrt{n}$. Because the t distribution is symmetric, this can also be written as $\bar{m} - t_{n-1,0.975}s/\sqrt{n}$ to $\bar{m} + t_{n-1,0.975}s/\sqrt{n}$.

As we have seen, there is a 95% probability that when we construct the confidence interval, that interval will contain the true mean, m. Note that we have not said that, given a particular set of data and the resulting confidence interval, there is a 95% probability that m is in the confidence interval. The semantic difficulty here is that m is not a random variable, but is rather some true fixed quantity that we are estimating; the measurements $m_1, m_2, \ldots, m_n$, and the calculated $\bar{m}$, s, and confidence interval are the random quantities.

Example B.12

Suppose that we want to estimate the mass of an object and obtain the following ten measurements of the mass (in grams):

$$9.98 \quad 10.07 \quad 9.94 \quad 10.22 \quad 9.98$$
$$10.01 \quad 10.11 \quad 10.01 \quad 9.99 \quad 9.92 \tag{B.75}$$

The sample mean is $\bar{m} = 10.02$ g. The sample standard deviation is $s = 0.0883$ grams. The 97.5%-tile of the t distribution with $n - 1 = 9$ degrees of freedom is (from a t-table or function) 2.262. Thus our 95% confidence interval for the mean is

$$\left[\bar{m} - 2.262 s/\sqrt{n}, \ \bar{m} + 2.262 s/\sqrt{n}\,\right] \text{ g.} \tag{B.76}$$

Substituting the values for $\bar{m}$, s, and n, we get an interval of

$$\left[10.02 - 2.262 \cdot 0.0883/\sqrt{10}, \ 10.02 + 2.262 \cdot 0.0883/\sqrt{10}\right] \text{g} \tag{B.77}$$

or

$$[9.96, \ 10.08] \text{ g.} \tag{B.78}$$

The above procedure for constructing a confidence interval for the mean using the t distribution was based on the assumption that the measurements were normally distributed. In situations where the data are not normally distributed, this procedure can fail in a very dramatic fashion (Exercise B.9). However, it may be safe to generate an approximate confidence interval using this procedure if (1) the number n of data is large (50 or more) or (2) the distribution of the data is not strongly skewed and n is at least 15.

B.9. HYPOTHESIS TESTS

In some situations we want to test whether a set of normally distributed data could reasonably have come from a normal distribution with expected value μ_0. Applying the sampling theorem, we see that if our data did come from a normal distribution with expected value μ_0, then there would be a 95% probability that

$$t_{\text{obs}} = \frac{\mu_0 - \bar{m}}{s/\sqrt{n}} \tag{B.79}$$

would lie in the interval

$$[F_t^{-1}(0.025), \ F_t^{-1}(0.975)] = [t_{n-1,0.025}, \ t_{n-1,0.975}], \tag{B.80}$$

and only a 5% probability that t would lie outside this interval. Equivalently, there is only a 5% probability that $|t_{\text{obs}}| \geq t_{n-1,0.975}$.

This leads to the **t-test**: If $|t_{\text{obs}}| \geq t_{n-1,0.975}$, then we reject the hypothesis that $\mu = \mu_0$. On the other hand, if $|t_{\text{obs}}| < t_{n-1,0.975}$, then we cannot reject the hypothe-

sis that $\mu = \mu_0$. Although the 95% confidence level is traditional, we can also perform the t-test at a 99% or some other confidence level. In general, if we want a confidence level of $1 - \alpha$, then we compare $|t_{obs}|$ to $t_{n-1,1-\alpha/2}$.

In addition to reporting whether or not a set of data passes a t-test, it is good practice to report the associated t-test p-value. The p-value associated with a t-test is the largest value of α for which the data passes the t-test. Equivalently, it is the probability that we could have gotten a greater t value than we have observed, given that all of our assumptions are correct.

Example B.13

Consider the following data:

$$
\begin{array}{ccccc}
1.2944 & -0.3362 & 1.7143 & 2.6236 & 0.3082 \\
1.8580 & 2.2540 & -0.5937 & -0.4410 & 1.5711
\end{array}
\tag{B.81}
$$

These appear to be roughly normally distributed, with a mean that seems to be larger than 0. We will test the hypothesis $\mu = 0$. The t statistic is

$$
t_{obs} = \frac{\mu_0 - \bar{m}}{s/\sqrt{n}},
\tag{B.82}
$$

which, for this data set, is

$$
t_{obs} = \frac{0 - 1.0253}{1.1895/\sqrt{10}} \approx -2.725.
\tag{B.83}
$$

Because $|t_{obs}|$ is larger than $t_{9,0.975} = 2.262$, we reject the hypothesis that these data came from a normal distribution with expected value 0 at the 95% confidence level.

The t-test (or any other statistical test) can fail in two ways. First, it could be that the hypothesis that $\mu = \mu_0$ is true, but our particular data set contained some unlikely values and failed the t-test. Rejecting the hypothesis when it is in fact true is called a **type I error**. We can control the probability of a type I error by decreasing α.

The second way in which the t-test can fail is more difficult to control. It could be that the hypothesis $\mu = \mu_0$ was false, but the sample mean was close enough to μ_0 to pass the t-test. In this case, we have a **type II error**. The probability of a type II error depends very much on how close the true mean is to μ_0. If the true mean $\mu = \mu_1$ is very close to μ_0, then a type II error is quite likely. If the true mean $\mu = \mu_1$ is very far from μ_0, then a type II error will be less likely. Given a particular alternative hypothesis, $\mu = \mu_1$, we call the probability of a type II error $\beta(\mu_1)$, and call the probability of not making a type II error $(1 - \beta(\mu_1))$ the **power** of the test. We can estimate $\beta(\mu_1)$ by repeatedly generating sets of n random numbers with $\mu = \mu_1$ and performing the hypothesis test on the sets of random numbers (Exercise B.10).

The results of a hypothesis test should always be reported with care. It is important to discuss and justify any assumptions (such as the normality assumption made in the t-test) underlying the test. The p-value should always be reported along with whether the hypothesis was rejected. If the hypothesis was not rejected and some particular alternative hypothesis is available, it is good practice to estimate the power of the hypothesis test against this alternative hypothesis. Confidence intervals for the mean should be reported along with the results of a hypothesis test.

It is important to distinguish between the statistical significance of a hypothesis test and the actual magnitude of any difference between the observed mean and the hypothesized mean. For example, with very large n it is nearly always possible to achieve statistical significance at the 95% confidence level, even though the observed mean may differ from the hypothesis by only 1% or less.

B.10. EXERCISES

1. Compute the expected value and variance of a uniform random variable in terms of the parameters a and b.
2. Compute the CDF of an exponential random variable with parameter λ. You may find the MATLAB library function **exprand.m** to be useful here.
3. Show that

$$\mathrm{Cov}(aX, \, Y) = a\mathrm{Cov}(X, \, Y) \qquad (B.84)$$

   and that

$$\mathrm{Cov}(X + Y, \, Z) = \mathrm{Cov}(X, \, Z) + \mathrm{Cov}(Y, \, Z). \qquad (B.85)$$

4. Show that the PDF for the sum of two independent uniform random variables on $[a, \, b] = [0, \, 1]$ is

$$f(x) = \begin{cases} 0 & (x \le 0) \\ x & (0 \le x \le 1) \\ 2 - x & (1 \le x \le 2) \\ 0 & (x \ge 2). \end{cases} \qquad (B.86)$$

5. Suppose that X and Y are independent random variables. Use conditioning to find a formula for the CDF of $X + Y$ in terms of the PDFs and CDFs of X and Y.
6. Suppose that $\mathbf{x} = (X_1, \, X_2)^T$ is a vector composed of two random variables with a multivariate normal distribution with expected value μ and covariance matrix $\mathbf{C}$, and that $\mathbf{A}$ is a 2 by 2 matrix. Use properties of expected value and covariance to show that $\mathbf{y} = \mathbf{A}\mathbf{x}$ has expected value $\mathbf{A}\mu$ and covariance $\mathbf{A}\mathbf{C}\mathbf{A}^T$.
7. Suppose that $\mathbf{x}$ is a multivariate normal random variable with mean vector μ and covariance matrix $\mathbf{C}$. Use Theorem B.5 to show that the *ith* component of $\mathbf{x}$, $\mathbf{x}_i$, is

normally distributed with mean μ_i and variance $\mathbf{C}_{i,i}$. Hint: Let $\mathbf{A} = \mathbf{e}_i^T$, where $\mathbf{e}_i$ is the *ith* column of the identity matrix.

8. Consider the following data, which we will assume are drawn from a normal distribution.

$$
\begin{array}{ccccc}
-0.4326 & -1.6656 & 0.1253 & 0.2877 & -1.1465 \\
1.1909 & 1.1892 & -0.0376 & 0.3273 & 0.1746
\end{array}
$$

Find the sample mean and standard deviation. Use these to construct a 95% confidence interval for the mean. Test the hypothesis $H_0 : \mu = 0$ at the 95% confidence level. What do you conclude? What was the corresponding *p*-value?

9. Using MATLAB, repeat the following experiment 1000 times. Generate five exponentially distributed random numbers from the exponential probability density function (B.10) with means $\mu = 1/\lambda = 10$. You may find the library function **exprand** to be useful here. Use (B.74) to calculate a 95% confidence interval for the 1000 mean determinations. How many times out of the 1000 experiments did the 95% confidence interval cover the expected value of 10? What happens if you instead generate 50 exponentially distributed random numbers at a time? Discuss your results.

10. Using MATLAB, repeat the following experiment 1000 times. Use the **randn** function to generate a set of 10 normally distributed random numbers with expected value 10.5 and standard deviation 1. Perform a *t*-test of the hypothesis $\mu = 10$ at the 95% confidence level. How many type II errors were committed? What is the approximate power of the *t*-test with $n = 10$ against the alternative hypothesis $\mu = 10.5$? Discuss your results.

11. Using MATLAB, repeat the following experiment 1000 times. Generate five exponentially distributed random numbers with expected value 10. You may find the library function **exprand** to be useful here. Take the average of the five random numbers. Plot a histogram and a *Q*-*Q* plot of the 1000 averages that you computed. Are the averages approximately normally distributed? Explain why or why not. What would you expect to happen if you took averages of 50 exponentially distributed random numbers at a time? Try it and discuss the results.

B.11. NOTES AND FURTHER READING

Most of the material in this Appendix can be found in virtually any introductory textbook in probability and statistics. Some recent textbooks include [4, 29]. The multivariate normal distribution is a somewhat more advanced topic that is often ignored in introductory courses. Searle [139] has a good discussion of the multivariate normal distribution and its properties. Numerical methods for probability and statistics are a specialized topic. Two standard references include [89, 157].

This page intentionally left blank

Review of Vector Calculus

Synopsis

A review of key vector calculus topics, including the gradient, Hessian, Jacobian, Taylor's theorem, and Lagrange multipliers.

C.1. THE GRADIENT, HESSIAN, AND JACOBIAN

In vector calculus, the familiar first and second derivatives of a single-variable function are generalized to operate on vectors.

Definition C.1 Given a scalar-valued function with a vector argument, $f(\mathbf{x})$, the **gradient** of f is

$$\nabla f(\mathbf{x}) = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{bmatrix}. \tag{C.1}$$

The vector $\nabla f(\mathbf{x})$ has an important geometric interpretation in that it points in the direction in which $f(\mathbf{x})$ increases most rapidly at the point $\mathbf{x}$.

Recall from single-variable calculus that if a function f is continuously differentiable, then a point x^* can only be a minimum or maximum point of f if $f'(x)|_{x=x^*} = 0$. Similarly in vector calculus, if $f(\mathbf{x})$ is continuously differentiable, then a point $\mathbf{x}^*$ can only be a minimum or maximum point if $\nabla f(\mathbf{x}^*) = \mathbf{0}$. In more than one dimension, a point $\mathbf{x}^*$ where $\nabla f(\mathbf{x}^*) = \mathbf{0}$ can also be a **saddle point**. Any point where $\nabla f(\mathbf{x}^*) = \mathbf{0}$ is called a **critical point**.

Definition C.2 Given a scalar-valued function of a vector, $f(\mathbf{x})$, the **Hessian** of f is a square matrix of partial derivatives given by

$$\mathbf{H}(f(\mathbf{x})) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}. \tag{C.2}$$

If f is twice continuously differentiable, the Hessian is symmetric. It is common in mathematics to write the Hessian using the operator ∇^2, but this sometimes leads to confusion with another vector calculus operator, the Laplacian.

Theorem C.1 *If $f(\mathbf{x})$ is a twice continuously differentiable function, and $\mathbf{H}(f(\mathbf{x}_0))$ is a positive semidefinite matrix, then $f(\mathbf{x})$ is a **convex function** at $\mathbf{x}_0$. If $\mathbf{H}(f(\mathbf{x}_0))$ is positive definite, then $f(\mathbf{x})$ is **strictly convex** at $\mathbf{x}_0$.*

This theorem can be used to check whether a critical point is a minimum of f. If $\mathbf{x}^*$ is a critical point of f and $\mathbf{H}(f(\mathbf{x}^*))$ is positive definite, then f is convex at $\mathbf{x}^*$, and $\mathbf{x}^*$ is thus a local minimum of f.

It will be necessary to compute derivatives of quadratic forms.

Theorem C.2 *Let $f(\mathbf{x}) = \mathbf{x}^T \mathbf{A} \mathbf{x}$, where $\mathbf{A}$ is an n by n symmetric matrix. Then*

$$\nabla f(\mathbf{x}) = 2\mathbf{A}\mathbf{x} \tag{C.3}$$

and

$$\mathbf{H}(f(\mathbf{x})) = 2\mathbf{A}. \tag{C.4}$$

Definition C.3 Given a vector-valued function of a vector, $\mathbf{F}(\mathbf{x})$, where

$$\mathbf{F}(\mathbf{x}) = \begin{bmatrix} f_1(\mathbf{x}) \\ f_2(\mathbf{x}) \\ \vdots \\ f_m(\mathbf{x}) \end{bmatrix}, \tag{C.5}$$

the **Jacobian** of $\mathbf{F}$ is

$$\mathbf{J}(\mathbf{x}) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \frac{\partial f_m}{\partial x_2} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}. \tag{C.6}$$

Some authors use the notation $\nabla \mathbf{F}(\mathbf{x})$ for the Jacobian. Note that the rows of $\mathbf{J}(\mathbf{x})$ are the gradients (C.1) of the functions $f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_m(\mathbf{x})$.

C.2. TAYLOR'S THEOREM

In the calculus of single-variable functions, Taylor's theorem produces an infinite series for $f(x + \Delta x)$ in terms of $f(x)$ and its derivatives. Taylor's theorem can be extended to a function of a vector $f(\mathbf{x})$, but in practice, derivatives of order higher than 2 are extremely inconvenient. The following form of Taylor's theorem is often used in optimization theory.

Theorem C.3 *Suppose that $f(\mathbf{x})$ and its first and second partial derivatives are continuous. For any vectors $\mathbf{x}$ and $\Delta\mathbf{x}$, there is a vector $\mathbf{c}$, with $\mathbf{c}$ on the line between $\mathbf{x}$ and $\mathbf{x} + \Delta\mathbf{x}$, such that*

$$f(\mathbf{x} + \Delta\mathbf{x}) = f(\mathbf{x}) + \nabla f(\mathbf{x})^T \Delta\mathbf{x} + \frac{1}{2}\Delta\mathbf{x}^T \mathbf{H}(f(\mathbf{c}))\Delta\mathbf{x}. \qquad (C.7)$$

This form of **Taylor's theorem with remainder term** is useful in many proofs. However, in computational work there is no way to determine $\mathbf{c}$. For that reason, when $\Delta\mathbf{x}$ is a small perturbation, we often make use of the approximation

$$f(\mathbf{x} + \Delta\mathbf{x}) \approx f(\mathbf{x}) + \nabla f(\mathbf{x})^T \Delta\mathbf{x} + \frac{1}{2}\Delta\mathbf{x}^T \mathbf{H}(f(\mathbf{x}))\Delta\mathbf{x}. \qquad (C.8)$$

An even simpler version of Taylor's theorem, called the **mean value theorem**, uses only the first derivative.

Theorem C.4 *Suppose that $f(\mathbf{x})$ and its first partial derivatives are continuous. For any vectors $\mathbf{x}$ and $\Delta\mathbf{x}$ there is a vector $\mathbf{c}$, with $\mathbf{c}$ on the line between $\mathbf{x}$ and $\mathbf{x} + \Delta\mathbf{x}$ such that*

$$f(\mathbf{x} + \Delta\mathbf{x}) = f(\mathbf{x}) + \nabla f(\mathbf{c})^T \Delta\mathbf{x}. \qquad (C.9)$$

We will make use of a truncated version of (C.8):

$$f(\mathbf{x} + \Delta\mathbf{x}) \approx f(\mathbf{x}) + \nabla f(\mathbf{x})^T \Delta\mathbf{x}. \qquad (C.10)$$

By applying (C.10) to each of the functions $f_1(\mathbf{x})$, $f_2(\mathbf{x})$, . . . , $f_m(\mathbf{x})$, we obtain the approximation

$$\mathbf{F}(\mathbf{x} + \Delta\mathbf{x}) \approx \mathbf{F}(\mathbf{x}) + \mathbf{J}(\mathbf{x})\Delta\mathbf{x}. \qquad (C.11)$$

C.3. LAGRANGE MULTIPLIERS

The method of **Lagrange multipliers** is an important technique for solving optimization problems of the form:

$$\begin{aligned} \min f(\mathbf{x}) \\ g(\mathbf{x}) = 0, \end{aligned} \qquad (C.12)$$

where the scalar-valued function of a vector argument, $f(\mathbf{x})$, is called the **objective function**.

Figure C.1 shows a general situation. The solid contour represents the set of points where the (nonconstant) function $g(\mathbf{x}) = 0$, and the dashed contours are those of another function $f(\mathbf{x})$ that has a minimum as indicated. Moving along the $g(\mathbf{x}) = 0$ contour, we can trace out the curve $\mathbf{x}(t)$, parameterized by the variable $t \geq 0$, where $g(\mathbf{x}(t)) = 0$ and t increases as we progress counter-clockwise. At any point $\mathbf{x}(t)$ on the contour, the gradient of $g(\mathbf{x}(t))$ must be perpendicular to the contour because the function is constant along this curve. Note that in Figure C.1, $g(\mathbf{x})$ increases in the outward direction relative to the contour, so the gradient of $g(\mathbf{x})$ will be outward.

By the chain rule,

$$f'(\mathbf{x}(t)) = \mathbf{x}'(t)^T \nabla f(\mathbf{x}(t)), \tag{C.13}$$

where $\mathbf{x}'(t)$ is the counter-clockwise tangent to the contour $g(x) = 0$. For the point $\mathbf{x}_1 = \mathbf{x}(t_1)$ in Figure C.1, $\nabla f(\mathbf{x}_1)$ and $\mathbf{x}'(t_1)$ are at an obtuse angle, and their dot product $f'(\mathbf{x}_1)$ (C.13) will therefore be negative. Thus, $f(\mathbf{x})$ is decreasing as we move counter-clockwise around the contour $g(\mathbf{x}) = 0$ from $\mathbf{x}_1$, and $\mathbf{x}_1$ cannot satisfy (C.12).

In Figure C.2, for the point $\mathbf{x}_0 = \mathbf{x}(t_0)$, $\nabla f(\mathbf{x}_0)$ is perpendicular to the curve $g(\mathbf{x}) = 0$. In this case, by (C.13), $f'(\mathbf{x}_0) = 0$, and the point $\mathbf{x}_0$ may or may not be a minimum for $f(\mathbf{x})$ along the contour. Figure C.2 shows that a point $\mathbf{x}_0$ on the curve $g(\mathbf{x}) = 0$ can only be a possible minimum point for $f(\mathbf{x})$ if $\nabla g(\mathbf{x}_0)$ and $\nabla f(\mathbf{x}_0)$ are parallel or antiparallel. A point where this occurs is called a **stationary point**.
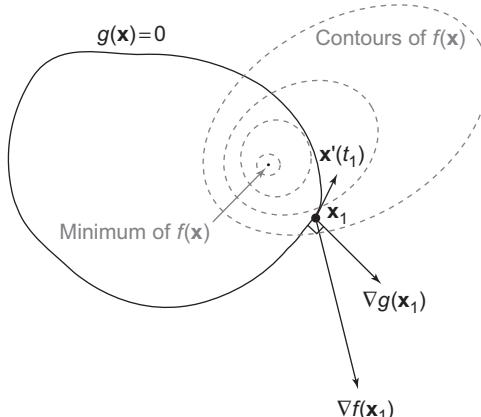
Figure C.1 The situation at a point $\mathbf{x}_1 = \mathbf{x}(t_1)$ along the contour $g(\mathbf{x}) = 0$ that is not a minimum of $f(\mathbf{x})$ and thus does not satisfy (C.12).

Finding a stationary point is necessary, but not sufficient, for finding a minimum of $f(\mathbf{x})$ along the contour $g(\mathbf{x}) = 0$, because such a point may be a minimum, maximum, or saddle point. Furthermore, a problem may have several local minima. Thus it is necessary to examine the behavior of $f(\mathbf{x})$ at all stationary points to find a global minimum.

Theorem C.5 *(C.12) can only be satisfied at a point $\mathbf{x}_0$ where*

$$\nabla f(\mathbf{x}_0) + \lambda \nabla g(\mathbf{x}_0) = \mathbf{0} \tag{C.14}$$

for some λ. λ is called a **Lagrange multiplier**.

The Lagrange multiplier condition can be extended to problems of the form

$$\min f(\mathbf{x}) \\ g(\mathbf{x}) \le 0. \tag{C.15}$$

Since points along the curve $g(\mathbf{x}) = 0$ are still feasible in (C.15), (C.14) must still hold true. However, there is an additional restriction. Suppose that $\nabla g(\mathbf{x}_0)$ and $\nabla f(\mathbf{x}_0)$ both point in the outward direction, as in Figure C.2. In this case, we can move in the opposite direction, into the feasible region to decrease $f(\mathbf{x})$ (e.g., in the situation depicted in Figure C.2, the solution to (C.15) is simply the indicated minimum of $f(\mathbf{x})$). Thus, a point $\mathbf{x}_0$ satisfying (C.14) cannot satisfy (C.15) unless the gradients of $g(\mathbf{x}_0)$ and $f(\mathbf{x}_0)$ point in opposite directions.

Theorem C.6 *(C.15) can only be satisfied at a point $\mathbf{x}_0$ where*

$$\nabla f(\mathbf{x}_0) + \lambda \nabla g(\mathbf{x}_0) = \mathbf{0} \tag{C.16}$$

for some Lagrange multiplier $\lambda > 0$.
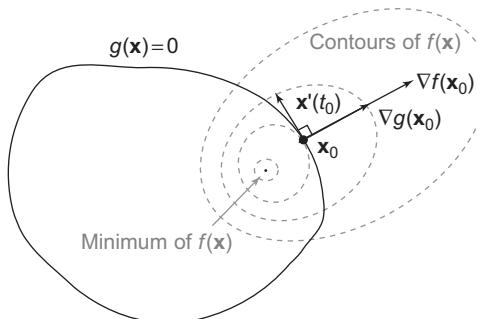
Figure C.2 The situation at a point $\mathbf{x}_0 = \mathbf{x}(t_0)$ along the contour $g(\mathbf{x}) = 0$ that is a minimum of $f(\mathbf{x})$ and thus satisfies (C.12). Note that $\nabla g(\mathbf{x}_0)$ and $\nabla f(\mathbf{x}_0)$ are parallel.

●━━━

Example C.1

Consider a simple example in two variables where $f(\mathbf{x})$ defines linear contours and $g(\mathbf{x}) = 0$ defines a unit circle

$$\min x_1 + x_2 \\ x_1^2 + x_2^2 - 1 \leq 0. \tag{C.17}$$

The Lagrange multiplier condition is

$$\begin{bmatrix} 1 \\ 1 \end{bmatrix} + \lambda \begin{bmatrix} 2x_1 \\ 2x_2 \end{bmatrix} = \mathbf{0}. \tag{C.18}$$

One stationary point solution to this nonlinear system of equations is $x_1 = 0.7071$, $x_2 = 0.7071$, with $\lambda = -0.7071$. This is the maximum of $f(\mathbf{x})$ subject to $g(\mathbf{x}) \leq 0$. The second solution to (C.18) is $x_1 = -0.7071$, $x_2 = -0.7071$, with $\lambda = 0.7071$. Because this is the only solution with $\lambda > 0$, so that $\nabla f(\mathbf{x})$ and $\nabla g(\mathbf{x})$ are antiparallel, this solves the minimization problem.

──

Note that (C.16) is (except for the non-negativity constraint on λ) the necessary condition for a minimum point of the unconstrained minimization problem,

$$\min f(\mathbf{x}) + \lambda g(\mathbf{x}). \tag{C.19}$$

Here the parameter λ can be adjusted so that, for the optimal solution, $\mathbf{x}^*$, $g(\mathbf{x}^*) \leq 0$. We will make frequent use of this technique to convert constrained optimization problems into unconstrained optimization problems.

──

C.4. EXERCISES

1. Let

$$f(\mathbf{x}) = x_1^2 x_2^2 - 2x_1 x_2^2 + x_2^2 - 3x_1^2 x_2 + 12x_1 x_2 - 12x_2 + 6. \tag{C.20}$$

Find the gradient, $\nabla f(\mathbf{x})$, and Hessian, $\mathbf{H}(f(\mathbf{x}))$. What are the critical points of f? Which of these are minima and maxima of f?

2. Find a Taylor's series approximation for $f(\mathbf{x} + \Delta\mathbf{x})$, where

$$f(\mathbf{x}) = e^{-(x_1 + x_2)^2} \tag{C.21}$$

is near the point

$$\mathbf{x} = \begin{bmatrix} 2 \\ 3 \end{bmatrix}. \tag{C.22}$$

3. Use the method of Lagrange multipliers to solve the problem,

$$\min 2x_1 + x_2$$
$$4x_1^2 + 3x_2^2 - 5 \leq 0. \tag{C.23}$$

4. Derive the formula (A.89) for the 2-norm of a matrix. Begin with the maximization problem,

$$\max_{\|\mathbf{x}\|_2 = 1} \|\mathbf{A}\mathbf{x}\|_2^2. \tag{C.24}$$

Note that we have squared $\|\mathbf{A}\mathbf{x}\|_2$. We will take the square root at the end of the problem.

i. Using the formula $\|\mathbf{x}\|_2 = \sqrt{\mathbf{x}^T\mathbf{x}}$, rewrite the above maximization problem without norms.

ii. Use the Lagrange multiplier method to find a system of equations that must be satisfied by any stationary point of the maximization problem.

iii. Explain how the eigenvalues and eigenvectors of $\mathbf{A}^T\mathbf{A}$ are related to this system of equations. Express the solution to the maximization problem in terms of the eigenvalues and eigenvectors of $\mathbf{A}^T\mathbf{A}$.

iv. Use this solution to get $\|\mathbf{A}\|_2$.

5. Derive the normal equations (2.3) using vector calculus, by letting

$$f(\mathbf{m}) = \|\mathbf{G}\mathbf{m} - \mathbf{d}\|_2^2 \tag{C.25}$$

and minimizing $f(\mathbf{m})$. Note that in problems with many least squares solutions, all of the least squares solutions will satisfy the normal equations.

i. Rewrite $f(\mathbf{m})$ as a dot product and then expand the expression.

ii. Find $\nabla f(\mathbf{m})$.

iii. Set $\nabla f(\mathbf{m}) = \mathbf{0}$, and obtain the normal equations.

C.5. NOTES AND FURTHER READING

Basic material on vector calculus can be found in many calculus textbooks. However, more advanced topics, such as Taylor's theorem for functions of a vector, are often skipped in basic texts. The material in this chapter is particularly important in optimization, and can often be found in associated references [58, 105, 119].

This page intentionally left blank

Glossary of Notation

- α, β, γ, ...: Scalars.
- a, b, c, ...: Scalar-valued functions or scalars.
- $\mathbf{a}$, $\mathbf{b}$, $\mathbf{c}$, ...: Column vectors.
- a_i: ith element of vector $\mathbf{a}$.
- A, B, C, ...: Scalar-valued functions or random variables.
- $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$, ...: Fourier transforms.
- $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, ...: Vector-valued functions or matrices.
- $\mathbf{A}_{i,\cdot}$: ith row of matrix $\mathbf{A}$.
- $\mathbf{A}_{\cdot,i}$: ith column of matrix $\mathbf{A}$.
- $A_{i,j}$: (i,j)th element of matrix $\mathbf{A}$.
- $\mathbf{A}^{-1}$: Inverse of matrix $\mathbf{A}$.
- $\mathbf{A}^{T}$: Transpose of matrix $\mathbf{A}$.
- $\mathbf{x}^{*}$: Complex conjugate of matrix $\mathbf{x}$.
- $\mathbf{A}^{H}$: Complex conjugate transpose of matrix $\mathbf{A}$.
- $\mathbf{a} \odot \mathbf{b}$: Vector constructed by element-by-element multiplication of vectors $\mathbf{a}$ and $\mathbf{b}$.
- $\mathbf{a} \oslash \mathbf{b}$: Vector constructed by element-by-element division of vector $\mathbf{a}$ by $\mathbf{b}$.
- $\mathbf{A}^{-T}$: Transpose of matrix $\mathbf{A}^{-1}$.
- R^n: Space of n-dimensional real vectors.
- $N(\mathbf{A})$: Null space of matrix $\mathbf{A}$.
- $R(\mathbf{A})$: Range of matrix $\mathbf{A}$.
- rank$(\mathbf{A})$: Rank of matrix $\mathbf{A}$.
- Tr$(\mathbf{A})$: Trace of matrix $\mathbf{A}$.
- $\|\mathbf{x}\|$: Norm of vector $\mathbf{x}$. A subscript is used to specify the 1-norm, 2-norm, or infinity norm.
- $\|\mathbf{A}\|$: Norm of matrix $\mathbf{A}$. A subscript is used to specify the 1-norm, 2-norm, or infinity norm.
- $\mathbf{G}^{\dagger}$: Generalized inverse of matrix $\mathbf{G}$.
- $\mathbf{m}_{\dagger}$: Generalized inverse solution $\mathbf{m}_{\dagger} = \mathbf{G}^{\dagger}\mathbf{d}$.

- $\mathbf{G}^\sharp$: A regularized generalized inverse of matrix $\mathbf{G}$.
- $E[X]$: Expected value of random variable X.
- $\bar{\mathbf{a}}$: Mean value of elements in vector $\mathbf{a}$.
- $N(\mu, \sigma^2)$: Normal probability distribution with expected value μ and variance σ^2.
- $\text{Cov}(X, Y)$: Covariance of random variables X and Y.
- $\text{Cov}(\mathbf{x})$: Matrix of covariances of elements of vector $\mathbf{x}$.
- $\rho(X, Y)$: Correlation between random variables X and Y.
- $\text{Var}(X)$: Variance of random variable X.
- $f(\mathbf{d}|\mathbf{m})$: Conditional probability density for $\mathbf{d}$, conditioned on a particular model $\mathbf{m}$.
- $L(\mathbf{m}|\mathbf{d})$: Likelihood function for a model $\mathbf{m}$, given a particular data vector $\mathbf{d}$.
- σ: Standard deviation.
- σ^2: Variance.
- $\nabla f(\mathbf{x})$: Gradient of function $f(\mathbf{x})$.
- $\mathbf{J}(\mathbf{x})$: Jacobian of vector-valued function, $\mathbf{F}(\mathbf{x})$.
- $\mathbf{H}(f(\mathbf{x}))$: Hessian of scalar-valued function $f(\mathbf{x})$.

BIBLIOGRAPHY

[1] R. C. Aster. On projecting error ellipsoids. *Bulletin of the Seismological Society of America*, 78(3): 1373–1374, 1988.

[2] G. Backus and F. Gilbert. Uniqueness in the inversion of inaccurate gross earth data. *Philosophical Transactions of the Royal Society A*, 266:123–192, 1970.

[3] Z. Bai and J. W. Demmel. Computing the generalized singular value decomposition. *SIAM Journal on Scientific Computing*, 14:1464–1486, 1993.

[4] L. J. Bain and M. Englehardt. *Introduction to Probability and Mathematical Statistics*, 2nd ed. Brooks/Cole, Pacific Grove, CA, 2000.

[5] R. Barrett, M. Berry, T. F. Chan, J. Demmel, J. Donato, V. Eijkhout, R. Pozo, C. Romine, and H. van der Vorst. *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods, 2nd Ed.* SIAM, Philadelphia, 1994.

[6] I. Barrowdale and F. D. K. Roberts. Solution of an overdetermined system of equations in the l_1 norm. *Communications of the ACM*, 17(6):319–326, 1974.

[7] D. M. Bates and D. G. Watts. *Nonlinear Regression Analysis and Its Applications*. Wiley, New York, 1988.

[8] J. Baumeister. *Stable Solution of Inverse Problems*. Vieweg, Braunschweig, 1987.

[9] C. Bekas, E. Kokiopoulou, and Y. Saad. An estimator for the diagonal of a matrix. *Applied Numerical Mathematics*, 57(11–12):1214–1229, 2005.

[10] A. Ben-Israel and T. N. E. Greville. *Generalized Inverses*, 2nd ed. Springer-Verlag, New York, 2003.

[11] A. Ben-Tal and A. Nemirovski. *Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications*. SIAM, Philadelphia, 2001.

[12] J. G. Berryman. Analysis of approximate inverses in tomography. I. Resolution analysis. *Optimization and Engineering*, 1(1):87–115, 2000.

[13] J. G. Berryman. Analysis of approximate inverses in tomography. II. Iterative inverses. *Optimization and Engineering*, 1(4):437–473, 2000.

[14] M. Bertero and P. Boccacci. *Introduction to Inverse Problems in Imaging*. Institute of Physics, London, 1998.

[15] J. T. Betts. *Practical Methods for Optimal Control Using Nonlinear Programming*. SIAM, Philadelphia, 2001.

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

[17] C. G. E. Boender and H. E. Romeijn. Stochastic methods. In R. Horst and P. M. Pardalos, eds., *Handbook of Global Optimization*, pages 829–869. Kluwer Academic Publishers, Dordrecht, 1995.

[18] P. T. Boggs, J. R. Donaldson, R. H. Byrd, and R. B. Schnabel. ODRPACK software for weighted orthogonal distance regression. *ACM Transactions on Mathematical Software*, 15(4):348–364, 1989. Available at http://www.netlib.org/odrpack/.

[19] L. Boschi. Measures of resolution in global body wave tomography. *Geophysical Research Letters*, 30:SDE 2-1-SDE 2-4, 2003.

[20] R. Bracewell. *The Fourier Transform and Its Applications*, 3rd ed. McGraw-Hill, Boston, 2000.

[21] A. M. Bruckstein, D. L. Donoho, and M. Elad. From sparse solutions of systems of equations to sparse modeling of signals and images. *SIAM Review*, 51(1):34–81, 2009.

[22] D. Calvetti and E. Somersalo. *Introduction to Bayesian Scientific Computing: Ten Lectures on Subjective Computing*. Springer, New York, 2007.

[23] S. L. Campbell and C. D. Meyer, Jr. *Generalized Inverses of Linear Transformations*. Dover, Mineola, NY, 1991.

[24] E. J. Candès, J. K. Romberg, and T. Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, 52(2): 489–509, 2006.

[25] E. J. Candès, J. K. Romberg, and T. Tao. Stable signal recovery from incomplete and inaccurate measurements. *Communications on Pure and Applied Mathematics*, 59(8):1207–1223, 2006.

[26] E. J. Candès and T. Tao. Near-optimal signal recovery from random projections: Universal encoding strategies? *IEEE Transactions on Information Theory*, 52(12):5406–5425, 2006.

[27] E. J. Candès and M. B. Wakin. An introduction to compressive sampling. *IEEE Signal Processing Magazine*, 25(2):21–30, 2008.

[28] P. Carrion. *Inverse Problems and Tomography in Acoustics and Seismology*. Penn, Atlanta, GA, 1987.

[29] G. Casella and R. L. Berger. *Statistical Inference*, 2nd ed. Duxbury, Pacific Grove, CA, 2002.

[30] Y. Censor and S. A. Zenios. *Parallel Optimization: Theory, Algorithms, and Applications*. Oxford University Press, New York, 1997.

[31] J. F. Claerbout and F. Muir. Robust modeling with erratic data. *Geophysics*, 38(5):826–844, 1973.

[32] T. F. Coleman and Y. Li. A globally and quadratically convergent method for linear l_1 problems. *Mathematical Programming*, 56:189–222, 1992.

[33] S. C. Constable, R. L. Parker, and C. G. Constable. Occam's inversion: A practical algorithm for generating smooth models from electromagnetic sounding data. *Geophysics*, 52(3):289–300, 1987.

[34] G. Corliss, C. Faure, A. Griewank, L. Hascoet, and U. Naumann. *Automatic Differentiation of Algorithms*. Springer-Verlag, Berlin, 2002.

[35] R. T. Cox. *Algebra of Probable Inference*. Johns Hopkins University Press, Baltimore, 2002.

[36] P. Craven and G. Wahba. Smoothing noisy data with spline functions: Estimating the correct degree of smoothing by the method of generalized cross-validation. *Numerische Mathematik*, 31:377–403, 1979.

[37] M. M. Deal and G. Nolet. Comment on estimation of resolution and covariance for large matrix inversions by J. Zhang and G. A. McMechan. *Geophysical Journal International*, 127(1):245–250, 1996.

[38] J. W. Demmel. *Applied Numerical Linear Algebra*. SIAM, Philadelphia, 1997.

[39] J. E. Dennis, Jr. and R. B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. SIAM, Philadelphia, 1996.

[40] N. R. Draper and H. Smith. *Applied Regression Analysis*, 3rd ed. Wiley, New York, 1998.

[41] C. Eckart and G. Young. A principal axis transformation for non-Hermitian matrices. *Bulletin of the American Mathematical Society*, 45:118–121, 1939.

[42] A. W. F. Edwards. *Likelihood*. Johns Hopkins University Press, Baltimore, 1992.

[43] L. El Ghaoui and H. Lebret. Robust solutions to least-squares problems with uncertain data. *SIAM Journal on Matrix Analysis and Applications*, 18(4):1035–1064, 1997.

[44] H. W. Engl. Regularization methods for the stable solution of inverse problems. *Surveys on Mathematics for Industry*, 3:71–143, 1993.

[45] H. W. Engl, M. Hanke, and A. Neubauer. *Regularization of Inverse Problems*. Kluwer Academic Publishers, Boston, 1996.

[46] R. M. Errico. What is an adjoint model? *Bulletin of the American Meteorological Society*, 78(11):2577–2591, 1997.

[47] M. Evans, N. Hasting, and B. Peacock. *Statistical Distributions*, 2nd ed. John Wiley & Sons, New York, 1993.

[48] S. N. Evans and P. B. Stark. Inverse problems as statistics. *Inverse Problems*, 18:R1–R43, 2002.

[49] J. G. Ferris and D. B. Knowles. *The Slug-Injection Test for Estimating the Coefficient of Transmissibility of an Aquifer*. In U.S. Geological Survey Water Supply Paper 1536-I, pages 299–304. U.S. Geological Survey, Washington, DC, 1963.

[50] A. Frommer and P. Maass. Fast CG-based methods for Tikhonov-Phillips regularization. *SIAM Journal on Scientific Computing*, 20(5):1831–1850, 1999.

[51] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin. *Bayesian Data Analysis*, 2nd ed. Chapman & Hall/CRC Press, Boca Raton, FL, 2003.

[52] M. Th. van Genuchten. A closed-form equation for predicting the hydraulic conductivity of unsaturated soils. *Soil Science Society of America Journal*, 44:892–898, 1980.

[53] G. H. Golub and C. F. van Loan. *Matrix Computations*, 3rd ed. Johns Hopkins University Press, Baltimore, 1996.

[54] G. H. Golub and U. von Matt. Generalized cross-validation for large-scale problems. *Journal of Computational and Graphical Statistics*, 6(1):1–34, 1997.

[55] G. H. Golub and D. P. O'Leary. Some history of the conjugate gradient and Lanczos methods. *SIAM Review*, 31(1):50–102, 1989.

[56] W. P. Gouveia and J. A. Scales. Resolution of seismic waveform inversion: Bayes versus Occam. *Inverse Problems*, 13(2):323–349, 1997.

[57] A. Griewank. *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*. SIAM, Philadelphia, 2000.

[58] I. Griva, S. G. Nash, and A. Sofer. *Linear and Nonlinear Programming*, 2nd ed. SIAM, Philadelphia, 2008.

[59] C. W. Groetsch. *Inverse Problems in the Mathematical Sciences*. Vieweg, Braunschweig, 1993.

[60] D. Gubbins. *Time Series Analysis and Inverse Theory for Geophysicists*. Cambridge University Press, Cambridge, 2004.

[61] M. Hanke. *Conjugate Gradient Type Methods for Ill-Posed Problems*. CRC Press, New York, 1995.

[62] P. C. Hansen. Relations between SVD and GSVD of discrete regularization problems in standard and general form. *Linear Algebra and Its Applications*, 141:165–176, 1990.

[63] P. C. Hansen. Analysis of discrete ill-posed problems by means of the L-curve. *SIAM Review*, 34(4):561–580, 1992.

[64] P. C. Hansen. Regularization tools: A MATLAB package for analysis and solution of discrete ill-posed problems. *Numerical Algorithms*, 6(I–II):1–35, 1994. http://www.imm.dtu.dk/documents/users/pch/Regutools/regutools.html.

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

[66] P. C. Hansen. Deconvolution and regularization with Toeplitz matrices. *Numerical Algorithms*, 29:323–378, 2002.

[67] P. C. Hansen. *Discrete Inverse Problems: Insight and Algorithms*. Society for Industrial and Applied Mathematics, Philadelphia, 2010.

[68] J. M. H. Hendrickx. Bosque del Apache soil data. Personal communication, 2003.

[69] J. M. H. Hendrickx, B. Borchers, J. D. Rhoades, D. L. Corwin, S. M. Lesch, A. C. Hilgendorf, and J. Schlue. Inversion of soil conductivity profiles from electromagnetic induction measurements: Theory and experimental verification. *Soil Science Society of America Journal*, 66(3):673–685, 2002.

[70] G. T. Herman. *Image Reconstruction from Projections*. Academic Press, San Francisco, 1980.

[71] M. R. Hestenes. Conjugacy and gradients. In Stephen G. Nash, ed., *A History of Scientific Computing*, pages 167–179. ACM Press, New York, 1990.

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

[73] J. A. Hildebrand, J. M. Stevenson, P. T. C. Hammer, M. A. Zumberge, R. L. Parker, C. J. Fox, and P. J. Meis. A sea–floor and sea–surface gravity survey of axial volcano. *Journal of Geophysical Research*, 95(B8):12751–12763, 1990.

[74] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, Cambridge, 1985.

[75] R. Horst and P. M. Pardalos. *Handbook of Global Optimization*. Kluwer Academic Publishers, Dordrecht, 1995.

[76] R. Horst, P. M. Pardalos, and N. V. Thoai. *Introduction to Global Optimization*. Kluwer Academic Publishers, Dordrecht, 1995.

[77] P. J. Huber. *Robust Statistical Procedures*, 2nd ed. SIAM, Philadelphia, 1996.

[78] S. von Huffel and J. Vandewalle. *The Total Least Squares Problem: Computational Aspects and Analysis*. SIAM, Philadelphia, 1991.

[79] H. Iyer and K. Hirahara (eds.). *Seismic Tomography*. Chapman & Hall, New York and London, 1993.

[80] E. T. Jaynes. *Probability Theory: The Logic of Science*. Cambridge University Press, Cambridge, 2003.

[81] H. Jeffreys. *Theory of Probability*. Oxford University Press, New York, 1998.

[82] W. H. Jeffreys, M. J. Fitzpatrick, and B. E. McArthur. Gaussfit—a system for least squares and robust estimation. *Celestial Mechanics*, 41(1–4):39–49, 1987. Available at http://clyde.as.utexas.edu/Gaussfit.html.

[83] J. Kaipio and E. Somersalo. *Statistical and Computational Inverse Problems*. Springer, New York, 2004.

[84] A. C. Kak and M. Slaney. *Principles of Computerized Tomographic Imaging*. SIAM, Philadelphia, 2001.

[85] L. Kaufman and A. Neumaier. PET regularization by envelope guided conjugate gradients. *IEEE Transactions on Medical Imaging*, 15(3):385–389, 1996.

[86] L. Kaufman and A. Neumaier. Regularization of ill-posed problems by envelope guided conjugate gradients. *Journal of Computational and Graphical Statistics*, 6(4):451–463, 1997.

[87] C. T. Kelley. *Iterative Methods for Solving Linear and Nonlinear Equations*. SIAM, Philadelphia, 1995.

[88] C. T. Kelley. *Solving Nonlinear Equations with Newton's Method*. SIAM, Philadelphia, 2003.

[89] W. J. Kennedy, Jr. and J. E. Gentle. *Statistical Computing*. Marcel Dekker, New York, 1980.

[90] A. Kirsch. *An Introduction to the Mathematical Theory of Inverse Problems*. Springer-Verlag, New York, 1996.

[91] F. J. Klopping, G. Peter, D. S. Robertson, K. A. Berstis, R. E. Moose, and W. E. Carter. Improvements in absolute gravity observations. *Journal of Geophysical Research*, 96(B5):8295–8303, 1991.

[92] C. Lanczos. Solutions of systems of linear equations by minimized iterations. *Journal of Research, National Bureau of Standards*, 49:33–53, 1952.

[93] C. Lanczos. *Linear Differential Operators*. Dover, Mineola, NY, 1997.

[94] C. L. Lawson and R. J. Hanson. *Solving Least Squares Problems*. SIAM, Philadelphia, 1995.

[95] D. C. Lay. *Linear Algebra and Its Applications*, 3rd ed. Addison-Wesley, Boston, 2003.

[96] T. Lay and T. Wallace. *Modern Global Seismology*. Academic Press, San Diego, 1995.

[97] J.-J. Leveque, L. Rivera, and G. Wittlinger. On the use of the checker-board test to assess the resolution of tomographic inversions. *Geophysical Journal International*, 115(1):313–318, 1993.

[98] Z.-P. Liang and P. C. Lauterbur. *Principles of Magnetic Resonance Imaging: A Signal Processing Perspective*. IEEE Press, New York, 2000.

[99] L. R. Lines, ed. *Inversion of Geophysical Data*. Society of Exploration Geophysicists, Tulsa, OK, 1988.

[100] T.-W. Lo and P. Inderwiesen. *Fundamentals of Seismic Tomography*. Society of Exploration Geophysicists, Tulsa, OK, 1994.

[101] C. F. van Loan. Generalizing the singular value decomposition. *SIAM Journal on Numerical Analysis*, 13:76–83, 1976.

[102] D. G. Luenberger. *Optimization by Vector Space Methods*. John Wiley and Sons, New York, 1969.

[103] J. MacCarthy, B. Borchers, and R. Aster. Estimating the model resolution matrix diagonal and generalized cross-validation for large geophysical inverse problems. *Journal of Geophysical Research*, 2011. In press.

[104] D. Mackenzie. Compressed sensing makes every pixel count. In *What's Happening in the Mathematical Sciences*. American Mathematical Society, Providence, RI, 2009.

[105] W. H. Marlow. *Mathematics for Operations Research*. Dover, Mineola, NY, 1993.

[106] P. J. McCarthy. Direct analytic model of the L-curve for Tikhonov regularization parameter selection. *Inverse Problems*, 19:643–663, 2003.

[107] P. R. McGillivray and D. Oldenburg. Methods for calculating Frechet derivatives and sensitivities for the non-linear inverse problem: A comparative study. *Geophysical Prospecting*, 38:499–524, 1990.

[108] W. Menke. *Geophysical Data Analysis: Discrete Inverse Theory*, volume 45 of *International Geophysics Series*, rev. ed. Academic Press, San Diego, 1989.

[109] C. D. Meyer. *Matrix Analysis and Applied Linear Algebra*. SIAM, Philadelphia, 2000.

[110] E. H. Moore. On the reciprocal of the general algebraic matrix. *Bulletin of the American Mathematical Society*, 26:394–395, 1920.

[111] J. J. More, B. S. Garbow, and K. E. Hillstrom. User guide for MINPACK-1. Technical Report ANL-80-74, Argonne National Laboratory, 1980.

[112] V. A. Morozov. *Methods for Solving Incorrectly Posed Problems*. Springer-Verlag, New York, 1984.

[113] R. H. Myers. *Classical and Modern Regression with Applications*, 2nd ed. PWS Kent, Boston, 1990.

[114] F. Natterer. *The Mathematics of Computerized Tomography*. SIAM, Philadelphia, 2001.

[115] F. Natterer and F. Wübbeling. *Mathematical Methods in Image Reconstruction*. SIAM, Philadelphia, 2001.

[116] A. Neumaier. Solving ill-conditioned and singular linear systems: A tutorial on regularization. *SIAM Review*, 40(3):636–666, 1998.

[117] R. Neupauer, B. Borchers, and J. L. Wilson. Comparison of inverse methods for reconstructing the release history of a groundwater contamination source. *Water Resources Research*, 36(9):2469–2475, 2000.

[118] I. Newton. *The Principia, Mathematical Principles of Natural Philosophy*, trans. by I. B. Cohen and A. Whitman. University of California Press, Berkeley, 1999.

[119] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer–Verlag, New York, 2006.

[120] G. Nolet. Solving or resolving inadequate and noisy tomographic systems. *Journal of Computational Physics*, 61(3):463–482, 1985.

[121] G. Nolet, ed. *Seismic Tomography with Applications in Global Seismology and Exploration Geophysics*. D. Reidel, Boston, 1987.

[122] S. J. Osher and R. P. Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*. Springer–Verlag, New York, 2002.

[123] C. C. Paige and M. A. Saunders. Algorithm 583 LSQR: Sparse linear equations and least-squares problems. *ACM Transactions on Mathematical Software*, 8(2):195–209, 1982.

[124] C. C. Paige and M. A. Saunders. LSQR: An algorithm for sparse linear equations and sparse least squares. *ACM Transactions on Mathematical Software*, 8(1):43–71, 1982.

[125] R. L. Parker. A theory of ideal bodies for seamount magnetism. *Journal of Geophysical Research*, 96(B10):16101–16112, 1991.

[126] R. L. Parker. *Geophysical Inverse Theory*. Princeton University Press, Princeton, NJ, 1994.

[127] R. L. Parker and M. K. McNutt. Statistics for the one-norm misfit measure. *Journal of Geophysical Research*, 85:4429–4430, 1980.

[128] R. L. Parker and M. A. Zumberge. An analysis of geophysical experiments to test Newton's law of gravity. *Nature*, 342:29–32, 1989.

[129] R. Penrose. A generalized inverse for matrices. *Mathematical Proceedings of the Cambridge Philosophical Society*, 51:406–413, 1955.

[130] S. Portnoy and R. Koenker. The Gaussian hare and the Laplacian tortoise: Computability of squared-error versus absolute-error estimators. *Statistical Science*, 12:279–296, 1997.

[131] M. B. Priestley. *Spectral Analysis and Time Series*. Academic Press, London, 1983.

[132] W. Rison, R. J. Thomas, P. R. Krehbiel, T. Hamlin, and J. Harlin. A GPS-based three-dimensional lightning mapping system: Initial observations in central New Mexico. *Geophysical Research Letters*, 26(23):3573–3576, 1999.

[133] C. D. Rodgers. *Inverse Methods for Atmospheric Sounding: Theory and Practice*. Word Scientific Publishing, Singapore, 2000.

[134] W. Rudin. *Real and Complex Analysis*, 3rd ed. McGraw-Hill, New York, 1987.

[135] Y. Saad. *Iterative Methods for Sparse Linear Systems*, 2nd ed. SIAM, Philadelphia, 2003.

[136] L. J. Savage. *The Foundation of Statistics*, 2nd ed. Dover, Mineola, NY, 1972.

[137] J. Scales and M. Smith. DRAFT: Geophysical inverse theory. http://landau.Mines.EDU/samizdat/inverse_theory/, 1997.

[138] J. A. Scales, A. Gersztenkorn, and S. Treitel. Fast lp solution of large, sparse, linear systems: Application to seismic travel time tomography. *Journal of Computational Physics*, 75(2):314–333, 1988.

[139] S. R. Searle. *Matrix Algebra Useful for Statistics*. Wiley-Interscience, New York, 2006.

[140] M. K. Sen and P. L. Stoffa. *Global Optimization Methods in Geophysical Inversion*. Number 4 in Advances in Exploration Geophysics. Elsevier, New York, 1995.

[141] C. B. Shaw, Jr. Improvement of the resolution of an instrument by numerical solution of an integral equation. *Journal of Mathematical Analysis and Applications*, 37:83–112, 1972.

[142] J. R. Shewchuk. An introduction to the Conjugate Gradient Method without the agonizing pain. Technical report, School of Computer Science, Carnegie Mellon University, 1994. http://www.cs.cmu.edu/jrs/jrspapers.html.

[143] D. S. Sivia and J. Skilling. *Data Analysis: A Bayesian Tutorial*, 2nd ed. Oxford University Press, New York, 2006.

[144] T. H. Skaggs and Z. J. Kabala. Recovering the release history of a groundwater contaminant. *Water Resources Research*, 30(1):71–79, 1994.

[145] T. H. Skaggs and Z. J. Kabala. Recovering the history of a groundwater contaminant plume: Method of quasi-reversibility. *Water Resources Research*, 31:2669–2673, 1995.

[146] A. Van der Sluis and H. A. Van der Vorst. Numerical solution of large, sparse linear algebraic systems arising from tomographic problems. In G. Nolet, ed., *Seismic Tomography*, pages 49–83. D. Reidel, Hingham, MA, 1987.

[147] W. Spakman and G. Nolet. Imaging algorithms, accuracy and resolution in delay time tomography. In M. J. R. Wortel, N. J. Vlaar, G. Nolet, and S. A. P. L. Cloetingh, eds., *Mathematical Geophysics: A Survey of Recent Developments in Seismology and Geodynamics*, pages 155–187. D. Reidel, Dordrecht, 1988.

[148] P. B. Stark and R. L. Parker. Velocity bounds from statistical estimates of $\tau(p)$ and $x(p)$. *Journal of Geophysical Research*, 92(B3):2713–2719, 1987.

[149] P. B. Stark and R. L. Parker. Correction to "velocity bounds from statistical estimates of $\tau(p)$ and $x(p)$." *Journal of Geophysical Research*, 93:13821–13822, 1988.

[150] P. B. Stark and R. L. Parker. Bounded-variable least-squares: An algorithm and applications. *Computational Statistics*, 10(2):129–141, 1995.

[151] G. W. Stewart. On the early history of the singular value decomposition. *SIAM Review*, 35:551–566, 1993.

[152] G. Strang. *Linear Algebra and Its Applications*, 3rd ed. Harcourt Brace Jovanovich, San Diego, 1988.

[153] G. Strang and K. Borre. *Linear Algebra, Geodesy, and GPS*. Wellesley–Cambridge Press, Wellesley, MA, 1997.

[154] N. Z. Sun. *Inverse Problems in Groundwater Modeling*. Kluwer Academic Publishers, Boston, 1984.

[155] A. Tarantola. *Inverse Problem Theory and Methods for Model Parameter Estimation*. SIAM, Philadelphia, 2004.

[156] A. Tarantola and B. Valette. Inverse problems = quest for information. *Journal of Geophysics*, 50(3):159–170, 1982.

[157] R. A. Thisted. *Elements of Statistical Computing*. Chapman and Hall, New York, 1988.

[158] C. Thurber. Hypocenter velocity structure coupling in local earthquake tomography. *Physics of the Earth and Planetary Interiors*, 75(1–3):55–62, 1992.

[159] C. Thurber and K. Aki. Three-dimensional seismic imaging. *Annual Review of Earth and Planetary Sciences*, 15:115–139, 1987.

[160] C. Thurber and J. Ritsema. Theory and observations—seismic tomography and inverse methods. In *Treatise on Geophysics*, pages 323–360. Elsevier, Amsterdam, 2007.

[161] A. N. Tikhonov and V. Y. Arsenin. *Solutions of Ill-Posed Problems*. Halsted Press, New York, 1977.

[162] A. N. Tikhonov and A. V. Goncharsky, eds. *Ill-Posed Problems in the Natural Sciences*. MIR Publishers, Moscow, 1987.

[163] J. Trampert and J.-J. Lévêque. Simultaneous iterative reconstruction technique: Physical interpretation based on the generalized least squares solution. *Journal of Geophysical Research*, 95(B8):12553–12559, 1990.

[164] L. N. Trefethen and D. Bau. *Numerical Linear Algebra*. SIAM, Philadelphia, 1997.

[165] J. Tromp, C. Tape, and Q. Liu. Seismic tomography, adjoint methods, time reversal, and banana-doughnut kernels. *Geophysical Journal International*, 160:195–216, 2005.

[166] S. Twomey. *Introduction to the Mathematics of Inversion in Remote Sensing and Indirect Measurements*. Dover, Mineola, NY, 1996.

[167] J. Um and C. Thurber. A fast algorithm for 2-point seismic ray tracing. *Bulletin of the Seismological Society of America*, 77(3):972–986, 1987.

[168] C. R. Vogel. Non-convergence of the L-curve regularization parameter selection method. *Inverse Problems*, 12:535–547, 1996.

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

[170] G. Wahba. *Spline Models for Observational Data*. SIAM, Philadelphia, 1990.

[171] G. A. Watson. Approximation in normed linear spaces. *Journal of Computational and Applied Mathematics*, 121(1–2):1–36, 2000.

[172] G. M. Wing. *A Primer on Integral Equations of the First Kind: The Problem of Deconvolution and Unfolding*. SIAM, Philadelphia, 1991.

[173] H. Zhang and C. H. Thurber. Estimating the model resolution matrix for large seismic tomography problems based on Lanczos bidiagonalization with partial reorthogonalization. *Geophysical Journal International*, 170(1):337–345, 2007.

[174] J. Zhang and G. A. McMechan. Estimation of resolution and covariance for large matrix inversions. *Geophysical Journal International*, 121(2):409–426, 1995.

INDEX

Note: Bold entries signify the introduction and/or definition of a topic.

This page intentionally left blank

This page intentionally left blank

This page intentionally left blank

This page intentionally left blank

This page intentionally left blank

This page intentionally left blank