Chapter 3

Solution of the Linear, Gaussian Inverse Problem, Viewpoint 1: The Length Method

3.1 THE LENGTHS OF ESTIMATES

The simplest of methods for solving the linear inverse problem $\mathbf{Gm} = \mathbf{d}$ is based on measures of the size, or length, of the estimated model parameters $\mathbf{m}^{\text{est}}$ and of the predicted data $\mathbf{d}^{\text{pre}} = \mathbf{Gm}^{\text{est}}$.

To see that measures of length can be relevant to the solution of inverse problems, consider the simple problem of fitting a straight line to data (Figure 3.1). This problem is often solved by the so-called method of least squares. In this method, one tries to pick the model parameters (intercept and slope) so that the predicted data are as close as possible to the observed data. For each observation, one defines a prediction error, or misfit, $e_i = d_i^{\text{obs}} - d_i^{\text{pre}}$. The best-fit line is then the one with model parameters that lead to the smallest overall error E, defined as

$$E = \sum_{i=1}^{N} e_i^2 = \mathbf{e}^{\text{T}}\mathbf{e} \tag{3.1}$$

The total error E (the sum of the squares of the individual errors) is exactly the squared Euclidean length of the vector $\mathbf{e}$, or $E = \mathbf{e}^{\text{T}}\mathbf{e}$.

The method of least squares estimates the solution of an inverse problem by finding the model parameters that minimize a particular measure of the length of the prediction error, $\mathbf{e} = \mathbf{d}^{\text{obs}} - \mathbf{d}^{\text{pre}}$, namely, its Euclidean length. As will be detailed below, it is the simplest of the methods that use measures of length as the guiding principle in solving an inverse problem.

3.2 MEASURES OF LENGTH

Note that although the Euclidean length is one way of quantifying the size or length of a vector, it is by no means the only possible measure. For instance, one could equally well quantify length by summing the absolute values of the elements of the vector.

Geophysical Data Analysis: Discrete Inverse Theory. DOI: 10.1016/B978-0-12-397160-9.00003-5

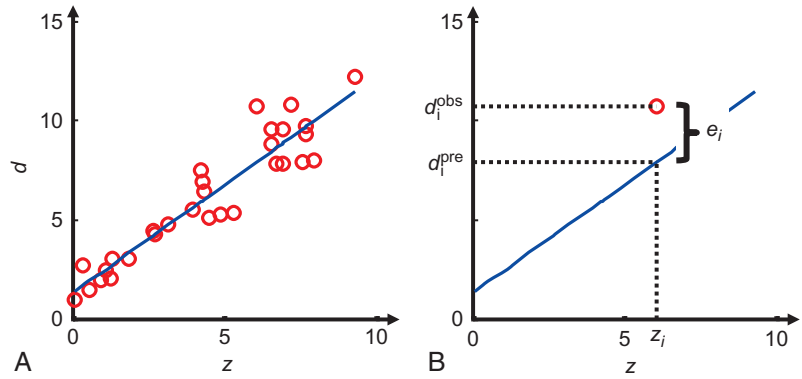

FIGURE 3.1 (A) Least squares fitting of a straight line to (z, d) data. (B) The error e_i for each observation is the difference between the observed and predicted datum: $e_i = d_i^{obs} - d_i^{pre}$. *MatLab* script gda03_01.

The term *norm* is used to refer to some measure of length or size and is indicated by a set of double vertical bars; that is, $||\mathbf{e}||$ is the norm of the vector $\mathbf{e}$. The most commonly employed norms are those based on the sum of some power of the elements of a vector and are given the name L_n, where n is the power:

$$L_1 \text{ norm:} \quad ||\mathbf{e}||_1 = \left[\sum_i |e_i|^1\right] \tag{3.2a}$$

$$L_2 \text{ norm:} \quad ||\mathbf{e}||_2 = \left[\sum_i |e_i|^2\right]^{1/2} \tag{3.2b}$$

$$L_n \text{ norm:} \quad ||\mathbf{e}||_n = \left[\sum_i |e_i|^n\right]^{1/n} \tag{3.2c}$$

Successively higher norms give the largest element of $\mathbf{e}$ successively larger weight. The limiting case of $n \to \infty$ gives nonzero weight to only the largest element (Figure 3.2); therefore, it is equivalent to the selection of the vector element with largest absolute value as the measure of length and is written as

$$L_\infty \text{ norm:} \quad ||\mathbf{e}||_\infty = \max_i |e_i| \tag{3.2d}$$

The method of least squares uses the L_2 norm to quantify length. It is appropriate to inquire why this, and not some other choice of norm, is used. The answer involves the way in which one chooses to weight data *outliers* that fall far from the average trend (Figure 3.3). If the data are very accurate, then the fact that one prediction falls far from its observed value is important. A high-order norm is

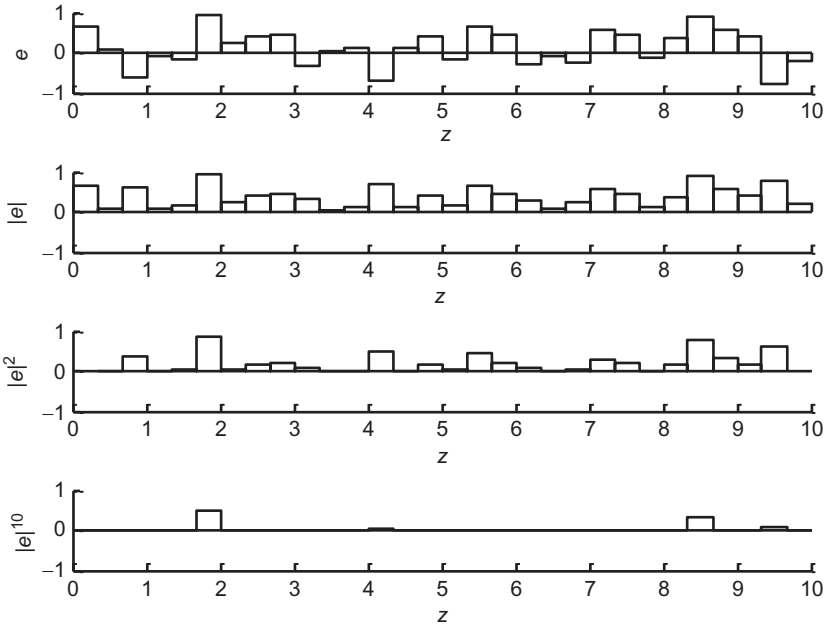

FIGURE 3.2 Hypothetical prediction error, $e_i(z_i)$, and its absolute value, raised to the powers of, 1, 2, and 10. While most elements of $|e_i|$ are numerically significant, only a few elements of $|e_i|^{10}$ are. *MatLab* script gda03_02.

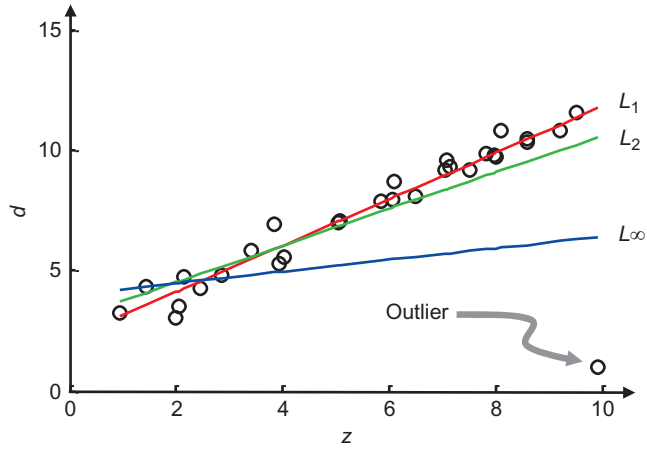

FIGURE 3.3 Straight line fits to (z, d) pairs where the error is measured under the L_1, L_2 and L_∞ norms. The L_1 norm gives the least weight to the one outlier. *MatLab* script gda03_03.

used, since it weights the larger errors preferentially. On the other hand, if the data are expected to scatter widely about the trend, then no significance can be placed upon a few large prediction errors. A low-order norm is used, since it gives more equal weight to errors of different sizes.

As will be discussed in more detail later, the L_2 norm implies that the data obey Gaussian statistics. Gaussians are rather short-tailed functions, so it is appropriate to place considerable weight on any data that have a large prediction error.

The likelihood of an observed datum falling far from the trend depends on the shape of the distribution for that datum. Long-tailed distributions imply many scattered (improbable) points. Short-tailed distributions imply very few scattered points (Figure 3.4). The choice of a norm, therefore, implies an assertion that the data obey a particular type of statistics.

Even though many measurements have approximately Gaussian statistics, most data sets generally have a few outliers that are wildly improbable. The occurrence of these points demonstrates that the assumption of Gaussian statistics is in error, especially in the tails of the distribution. If one applies least squares to this kind of problem, the estimates of the model parameters can be completely erroneous. Least squares weights large errors so heavily that even one "bad" data point can completely throw off the result. In these situations, methods based on the L_1 norm give more reliable estimates. (Methods that can tolerate a few bad data are said to be *robust*.)

Matrix norms can be defined in a manner similar to vector norms (see Equation (3.3d)). Vector and matrix norms obey the following relationships:

Vector norms:

$$||\mathbf{x}|| > 0 \text{ as long as } \mathbf{x} \neq 0 \tag{3.3a}$$

$$||a\mathbf{x}|| = |a|||\mathbf{x}|| \tag{3.3b}$$

$$||\mathbf{x} + \mathbf{y}|| \leq ||\mathbf{x}|| + ||\mathbf{y}|| \tag{3.3c}$$

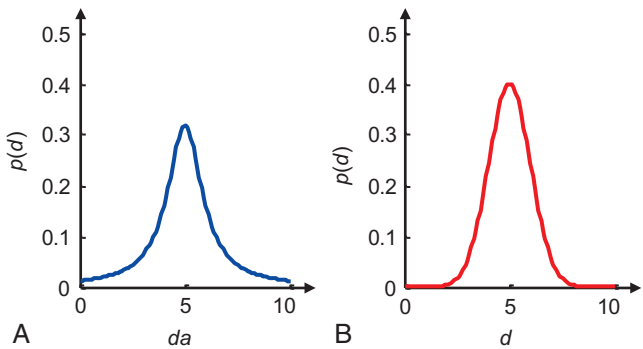

FIGURE 3.4 (A) Long-tailed probability density function. (B) Short-tailed probability density function. *MatLab* script gda03_04.

Matrix norms:

$$||\mathbf{A}||_2 = \left(\sum_{i=1}^{N} \sum_{j=1}^{N} A_{ij}^2 \right)^{1/2} \tag{3.3d}$$

$$||c\mathbf{A}|| = |c| \, ||\mathbf{A}|| \tag{3.3e}$$

$$||\mathbf{Ax}|| \leq ||\mathbf{A}|| \, ||\mathbf{x}|| \tag{3.3f}$$

$$||\mathbf{AB}|| \leq ||\mathbf{A}|| \, ||\mathbf{B}|| \tag{3.3g}$$

$$||\mathbf{A} + \mathbf{B}|| \leq ||\mathbf{A}|| + ||\mathbf{B}|| \tag{3.3h}$$

Equations (3.3c) and (3.3h) are called *triangle inequalities* because of their similarity to Pythagoras's law for right triangles.

3.3 LEAST SQUARES FOR A STRAIGHT LINE

The elementary problem of fitting a straight line to data illustrates the basic procedures applied in this technique. The model is the assertion that the data can be described by the linear equation $d_i = m_1 + m_2 z_i$. Note that there are two model parameters, $M = 2$, and that typically there are many more than two data, $N > M$. Since a line is defined by precisely two points, it is clearly impossible to choose a straight line that passes through every one of the data, except in the instance that they all lie precisely on the same straight line. Collinearity rarely occurs when measurements are influenced by noise.

As we shall discuss in more detail below, the fact that the equation $d_i = m_1 + m_2 z_i$ cannot be satisfied for every i means that the inverse problem is *overdetermined*; that is, it has no solution for which $\mathbf{e} = 0$. One therefore seeks values of the model parameters that solve $d_i = m_1 + m_2 z_i$ approximately, where the goodness of the approximation is defined by the error

$$E = \mathbf{e}^T \mathbf{e} = \sum_{i=1}^{N} (d_i - m_1 - m_2 z_i)^2 \tag{3.4}$$

This problem is then the elementary calculus problem of locating the minimum of the function $E(m_1, m_2)$ and is solved by setting the derivatives of E to zero and solving the resulting equations.

$$\begin{aligned} \frac{\partial E}{\partial m_1} &= \frac{\partial}{\partial m_1} \sum_{i=1}^{N} [d_i - m_1 - m_2 z_i]^2 = 2Nm_1 + 2m_2 \sum_{i=1}^{N} z_i - 2 \sum_{i=1}^{N} d_i = 0 \\ \frac{\partial E}{\partial m_2} &= \frac{\partial}{\partial m_2} \sum_{i=1}^{N} [d_i - m_1 - m_2 z_i]^2 = 2m_1 \sum_{i=1}^{N} z_i + 2m_2 \sum_{i=1}^{N} z_i^2 - 2 \sum_{i=1}^{N} z_i d_i = 0 \end{aligned} \tag{3.5}$$

These two equations are then solved simultaneously for m_1 and m_2, yielding the classic formulas for the least squares fitting of a line.

3.4 THE LEAST SQUARES SOLUTION OF THE LINEAR INVERSE PROBLEM

Least squares can be extended to the general linear inverse problem in a very straightforward manner. Again, one computes the derivative of the error E with respect to one of the model parameters, say, m_q, and sets the result to zero. The error E is

$$E = \mathbf{e}^{\mathrm{T}}\mathbf{e} = (\mathbf{d} - \mathbf{G}\mathbf{m})^{\mathrm{T}}(\mathbf{d} - \mathbf{G}\mathbf{m}) = \sum_{i=1}^{N}\left[d_i - \sum_{j=1}^{M} G_{ij}m_j\right]\left[d_i - \sum_{k=1}^{M} G_{ik}m_k\right] \tag{3.6}$$

Note that the indices on the sums within the parentheses are different dummy variables, to prevent confusion. Multiplying out the terms and reversing the order of the summations lead to

$$E = \sum_{j=1}^{M}\sum_{k=1}^{M} m_j m_k \sum_{i=1}^{N} G_{ij}G_{ik} - 2\sum_{j=1}^{M} m_j \sum_{i=1}^{N} G_{ij}d_i + \sum_{i=1}^{N} d_i d_i \tag{3.7}$$

The derivatives $\partial E/\partial m_q$ are now computed. Performing this differentiation term by term gives

$$\begin{aligned}\frac{\partial}{\partial m_q}\left[\sum_{j=1}^{M}\sum_{k=1}^{M} m_j m_k \sum_{i=1}^{N} G_{ij}G_{ik}\right] &= \sum_{j=1}^{M}\sum_{k=1}^{M}[\delta_{jq}m_k + m_j\delta_{kq}]\sum_{i=1}^{N} G_{ij}G_{ik} \\ &= 2\sum_{k=1}^{M} m_k \sum_{i=1}^{N} G_{iq}G_{ik}\end{aligned} \tag{3.8}$$

for the first term. Since the model parameters are independent variables, derivatives of the form $\partial m_i/\partial m_j$ are either unity, when $i = j$, or zero, when $i \neq j$. Thus, $\partial m_i/\partial m_j$ is just the Kronecker delta δ_{ij} (see Section I.4) and the formula containing it can be simplified trivially. The second term gives

$$-2\frac{\partial}{\partial m_q}\left[\sum_{j=1}^{M} m_j \sum_{i=1}^{N} G_{ij}d_i\right] = -2\sum_{j=1}^{M}\delta_{jq}\sum_{i=1}^{N} G_{ij}d_i = -2\sum_{i=1}^{N} G_{iq}d_i \tag{3.9}$$

Since the third term does not contain any ms, it is zero as

$$\frac{\partial}{\partial m_q}\left[\sum_{i=1}^{N} d_i d_i\right] = 0 \tag{3.10}$$

Combining the three terms gives

$$\frac{\partial E}{\partial m_q} = 0 = 2\sum_{k=1}^{M} m_k \sum_{i=1}^{N} G_{iq} G_{ik} - 2\sum_{i=1}^{N} G_{iq} d_i \tag{3.11}$$

Writing this equation in matrix notation yields

$$\mathbf{G}^{\mathrm{T}}\mathbf{G}\mathbf{m} - \mathbf{G}^{\mathrm{T}}\mathbf{d} = 0 \tag{3.12}$$

Note that the quantity $\mathbf{G}^{\mathrm{T}}\mathbf{G}$ is a square $M \times M$ matrix and that it multiplies a vector $\mathbf{m}$ of length M. The quantity $\mathbf{G}^{\mathrm{T}}\mathbf{d}$ is also a vector of length M. This equation is therefore a square matrix equation for the unknown model parameters. Presuming that $[\mathbf{G}^{\mathrm{T}}\mathbf{G}]^{-1}$ exists (an important question that we shall return to later), we have the following estimate for the model parameters:

$$\mathbf{m}^{\mathrm{est}} = [\mathbf{G}^{\mathrm{T}}\mathbf{G}]^{-1}\mathbf{G}^{\mathrm{T}}\mathbf{d} \tag{3.13}$$

which is the least squares solution to the inverse problem $\mathbf{G}\mathbf{m} = \mathbf{d}$.

When the dimensions of $\mathbf{G}$ are small (say, N and M less than a few hundred), the least squares solution is computed as

```
mest = (G'*G)\(G'*d);
```

(*MatLab* script gda03_05)

However, for larger problems, the computational cost of computing $\mathbf{G}^{\mathrm{T}}\mathbf{G}$ can be prohibitive. Furthermore, $\mathbf{G}^{\mathrm{T}}\mathbf{G}$ is rarely as sparse as $\mathbf{G}$ itself. In this case, an iterative matrix solver, such as the *biconjugate gradient algorithm*, is preferred. This algorithm only requires products of the form $\mathbf{G}^{\mathrm{T}}\mathbf{G}\mathbf{v}$, where $\mathbf{v}$ is a vector constructed by the algorithm, and this product can be performed as $\mathbf{G}^{\mathrm{T}}(\mathbf{G}\mathbf{v})$ so that $\mathbf{G}^{\mathrm{T}}\mathbf{G}$ is never explicitly calculated (e.g., Menke and Menke, 2011, their Section 5.8). *MatLab* provides a `bicg()` function, which is as follows:

```
tol = 1e-6;
maxit = 3*N;
mest2 = bicg(@leastsquaresfcn, G'*dobs, tol, maxit);
```

(*MatLab* script gda03_05)

The algorithm involves iteratively improving a solution, with the iterations terminating when the error is less than a tolerance `tol` or when `maxit` iterations have been performed (whichever comes first). The first argument is a *handle* to a user-provided `leastsquaresfcn()` function that calculates $\mathbf{G}^{\mathrm{T}}(\mathbf{G}\mathbf{v})$:

```
function y = leastsquaresfcn(v,transp_flag)
global G;
temp = G*v;
y = G'*temp;
return
```

(*MatLab* script gda03_05)

This function is stored in the file `leastsquaresfcn.m`. In order for this function to perform correctly, *MatLab* must be instructed that the **G** being used within it is the same as is defined in the main script. This is accomplished by placing the commands

```
clear G;
global G;
```

(*MatLab* script gda03_05)

at the beginning of the main script. Although this algorithm will work for any **G**, it is particularly useful when **G** has been defined as sparse using the `spalloc()` function. An example is given in *MatLab* script gda03_06.

3.5 SOME EXAMPLES

3.5.1 The Straight Line Problem

In the straight line problem, the model is $d_i = m_1 + m_2 z_i$, so the equation $\mathbf{Gm} = \mathbf{d}$ has the form

$$\begin{bmatrix} 1 & z_1 \\ 1 & z_2 \\ \vdots & \vdots \\ 1 & z_N \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_N \end{bmatrix} \tag{3.14}$$

In *MatLab*, the matrix **G** can be created with the command

```
G=[ones(N,1), z];
```

(*MatLab* script gda03_05)

The matrix products required by the least squares solution are

$$\mathbf{G}^{\mathrm{T}}\mathbf{G} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ z_1 & z_2 & \cdots & z_N \end{bmatrix} \begin{bmatrix} 1 & z_1 \\ 1 & z_2 \\ \vdots & \vdots \\ 1 & z_N \end{bmatrix} = \begin{bmatrix} N & \sum_{i=1}^{N} z_i \\ \sum_{i=1}^{N} z_i & \sum_{i=1}^{N} z_i^2 \end{bmatrix} \tag{3.15}$$

and

$$\mathbf{G}^{\mathrm{T}}\mathbf{d} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ z_1 & z_2 & \cdots & z_N \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_N \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{N} d_i \\ \sum_{i=1}^{N} d_i z_i \end{bmatrix} \tag{3.16}$$

This gives the least squares solution

$$\mathbf{m}^{\text{est}} = [\mathbf{G}^{\mathrm{T}}\mathbf{G}]^{-1}\mathbf{G}^{\mathrm{T}}\mathbf{d} = \begin{bmatrix} N & \sum_{i=1}^{N} z_i \\ \sum_{i=1}^{N} z_i & \sum_{i=1}^{N} z_i^2 \end{bmatrix}^{-1} \begin{bmatrix} \sum_{i=1}^{N} d_i \\ \sum_{i=1}^{N} d_i z_i \end{bmatrix} \tag{3.17}$$

3.5.2 Fitting a Parabola

The problem of fitting a parabola is a trivial generalization of fitting a straight line (Figure 3.4). Now the model is $d_i = m_1 + m_2 z_i + m_3 z_i^2$, so the equation $\mathbf{Gm} = \mathbf{d}$ has the form

$$\begin{bmatrix} 1 & z_1 & z_1^2 \\ 1 & z_2 & z_2^2 \\ \vdots & \vdots & \vdots \\ 1 & z_N & z_N^2 \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ m_3 \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_N \end{bmatrix} \tag{3.18}$$

In *MatLab*, the matrix $\mathbf{G}$ can be created with the command

```
G=[ones(N,1), z, z.^2];
```

(*MatLab* script gda03_07)

The matrix products required by the least squares solution are as follows:

$$\begin{aligned} \mathbf{G}^{\mathrm{T}}\mathbf{G} &= \begin{bmatrix} 1 & 1 & \cdots & 1 \\ z_1 & z_2 & \cdots & z_N \\ z_1^2 & z_N^2 & \cdots & z_N^2 \end{bmatrix} \begin{bmatrix} 1 & z_1 & z_1^2 \\ 1 & z_2 & z_2^2 \\ \vdots & \vdots & \vdots \\ 1 & z_N & z_N^2 \end{bmatrix} \\ &= \begin{bmatrix} N & \sum_{i=1}^{N} z_i & \sum_{i=1}^{N} z_i^2 \\ \sum_{i=1}^{N} z_i & \sum_{i=1}^{N} z_i^2 & \sum_{i=1}^{N} z_i^3 \\ \sum_{i=1}^{N} z_i^2 & \sum_{i=1}^{N} z_i^3 & \sum_{i=1}^{N} z_i^4 \end{bmatrix} \end{aligned} \tag{3.19}$$

and

$$\mathbf{G}^{\mathrm{T}}\mathbf{d} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ z_1 & z_2 & \cdots & z_N \\ z_1^2 & z_N^2 & \cdots & z_N^2 \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_N \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{N} d_i \\ \sum_{i=1}^{N} z_i d_i \\ \sum_{i=1}^{N} z_i^2 d_i \end{bmatrix} \tag{3.20}$$

giving the least squares solution

$$\mathbf{m}^{\text{est}} = [\mathbf{G}^{\text{T}}\mathbf{G}]^{-1}\mathbf{G}^{\text{T}}\mathbf{d} = \begin{bmatrix} N & \sum_{i=1}^{N} z_i & \sum_{i=1}^{N} z_i^2 \\ \sum_{i=1}^{N} z_i & \sum_{i=1}^{N} z_i^2 & \sum_{i=1}^{N} z_i^3 \\ \sum_{i=1}^{N} z_i^2 & \sum_{i=1}^{N} z_i^3 & \sum_{i=1}^{N} z_i^4 \end{bmatrix}^{-1} \begin{bmatrix} \sum_{i=1}^{N} d_i \\ \sum_{i=1}^{N} z_i d_i \\ \sum_{i=1}^{N} z_i^2 d_i \end{bmatrix} \tag{3.21}$$

An example of using a quadratic fit to examine Kepler's third law is shown in Figure 3.5.

3.5.3 Fitting a Plane Surface

To fit a plane surface, two auxiliary variables, say, x and y, are needed. The model is

$$d_i = m_1 + m_2 x_i + m_3 y_i \tag{3.22}$$

so the equation $\mathbf{Gm} = \mathbf{d}$ has the form

$$\begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ \vdots & \vdots & \vdots \\ 1 & x_N & y_N \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ m_3 \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_N \end{bmatrix} \tag{3.23}$$

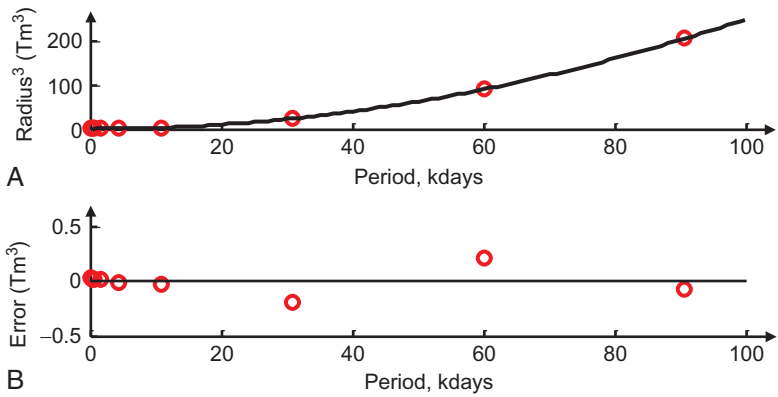

FIGURE 3.5 Test of Kepler's third law, which states that the cube of the orbital radius of a planet equals the square of its orbital period. (A) Data (red circles) for our solar system are least squares fit with a quadratic formula $d_i = m_1 + m_2 z_i + m_3 z_i^2$, where d_i is radius cubed and z_i is period. (B) Error of fit. A separate graph is used so that the error can be plotted at a meaningful scale. Data courtesy of Wikipedia. *MatLab* script gda03_07.

In *MatLab*, the matrix **G** can be created with the command

```
G=[ones(N,1), x, y];
```

(*MatLab* script gda03_08)

forming the matrix products $\mathbf{G}^{\mathrm{T}}\mathbf{G}$

$$\mathbf{G}^{\mathrm{T}}\mathbf{G} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ x_1 & x_2 & \cdots & x_N \\ y_1 & y_2 & \cdots & y_N \end{bmatrix} \begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ \vdots & \vdots & \vdots \\ 1 & x_N & y_N \end{bmatrix}$$

$$= \begin{bmatrix} N & \sum_{i=1}^{N} x_i & \sum_{i=1}^{N} y_i \\ \sum_{i=1}^{N} x_i & \sum_{i=1}^{N} x_i^2 & \sum_{i=1}^{N} x_i y_i \\ \sum_{i=1}^{N} y_i & \sum_{i=1}^{N} x_i y_i & \sum_{i=1}^{N} y_i^2 \end{bmatrix} \tag{3.24}$$

and

$$\mathbf{G}^{\mathrm{T}}\mathbf{d} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ x_1 & x_2 & \cdots & x_N \\ y_1 & y_2 & \cdots & y_N \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_N \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{N} d_i \\ \sum_{i=1}^{N} x_i d_i \\ \sum_{i=1}^{N} y_i d_i \end{bmatrix} \tag{3.25}$$

giving the least squares solution

$$\mathbf{m}^{\mathrm{est}} = [\mathbf{G}^{\mathrm{T}}\mathbf{G}]^{-1}\mathbf{G}^{\mathrm{T}}\mathbf{d} = \begin{bmatrix} N & \sum_{i=1}^{N} x_i & \sum_{i=1}^{N} y_i \\ \sum_{i=1}^{N} x_i & \sum_{i=1}^{N} x_i^2 & \sum_{i=1}^{N} x_i y_i \\ \sum_{i=1}^{N} y_i & \sum_{i=1}^{N} x_i y_i & \sum_{i=1}^{N} x_i y_i^2 \end{bmatrix}^{-1} \begin{bmatrix} \sum_{i=1}^{N} d_i \\ \sum_{i=1}^{N} x_i d_i \\ \sum_{i=1}^{N} y_i d_i \end{bmatrix} \tag{3.26}$$

An example of using a planar fit to examine earthquakes on a geologic fault is shown in Figure 3.6.

3.6 THE EXISTENCE OF THE LEAST SQUARES SOLUTION

The least squares solution arose from consideration of an inverse problem that had no exact solution. Since there was no exact solution, we chose to do the next best thing: to estimate the solution by those values of the model parameters that gave the best approximate solution (where "best" meant minimizing the L_2 prediction error). By writing a single formula $\mathbf{m}^{\mathrm{est}} = [\mathbf{G}^{\mathrm{T}}\mathbf{G}]^{-1}\mathbf{G}^{\mathrm{T}}\mathbf{d}$, we implicitly assumed that there

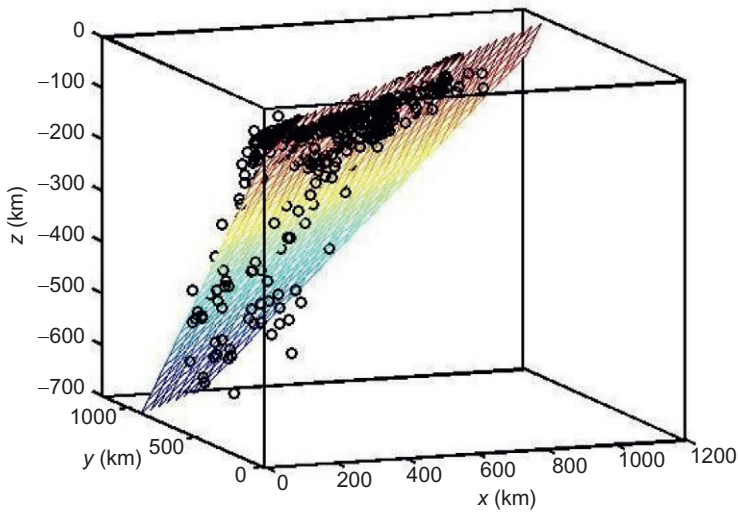

FIGURE 3.6 (Circles) Earthquakes in the Kurile subduction zone, northwest Pacific Ocean. The x-axis points north and the y-axis east. The earthquakes scatter about a dipping planar surface (colored grid), determined using least squares. Data courtesy of the U.S. Geological Survey. *MatLab* script gda03_08.

was only one such "best" solution. As we shall prove later, least squares fails if the number of solutions that give the same minimum prediction error is greater than one.

To see that least squares fails for problems with nonunique solutions, consider the straight line problem with only one data point (Figure 3.7). It is clear that this problem is nonunique; many possible lines can pass through the point, and each has zero prediction error. The solution then contains the following expression:

$$[\mathbf{G}^{\mathrm{T}}\mathbf{G}]^{-1} = \begin{bmatrix} N & \sum_{i=1}^{N} z_i \\ \sum_{i=1}^{N} z_i & \sum_{i=1}^{N} z_i^2 \end{bmatrix}^{-1} \rightarrow \begin{bmatrix} 1 & z_1 \\ z_1 & z_1^2 \end{bmatrix}^{-1} \tag{3.27}$$

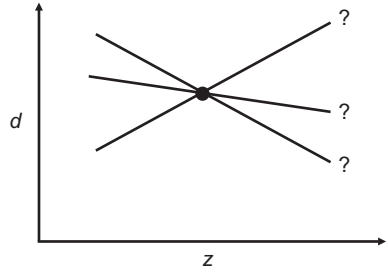

FIGURE 3.7 An infinity of different lines can pass through a single point. The prediction error for each is $E = 0$.

The inverse of a matrix is proportional to the reciprocal of the determinant of the matrix so that

$$[\mathbf{G}^{\mathrm{T}}\mathbf{G}]^{-1} \propto \left(\det\begin{bmatrix} 1 & z_1 \\ z_1 & z_1^2 \end{bmatrix}\right)^{-1} = \frac{1}{z_1^2 - z_1^2} \tag{3.28}$$

This expression clearly is singular. The formula for the least squares solution fails.

The question of whether the equation $\mathbf{Gm} = \mathbf{d}$ provides enough information to specify uniquely the model parameters serves as a basis for classifying inverse problems. A classification system based on this criterion is discussed in Sections 3.6.1–3.6.3.

3.6.1 Underdetermined Problems

When the equation $\mathbf{Gm} = \mathbf{d}$ does not provide enough information to determine uniquely all the model parameters, the problem is said to be *underdetermined*. As we saw in the example above, this can happen if there are several solutions that have zero prediction error. From elementary linear algebra, we know that underdetermined problems occur when there are more unknowns than data, that is, when $M > N$. We must note, however, that there is no special reason why the prediction error must be zero for an underdetermined problem. Frequently, the data uniquely determine some of the model parameters but not others. For example, consider the acoustic experiment in Figure 3.8. Since no measurements are made of the acoustic slowness in the second brick, it is clear that this model parameter is completely unconstrained by the data. In contrast, the acoustic slowness of the first brick is *overdetermined*, since in the presence of measurement noise, no choice of s_1 can satisfy the data exactly. The equation describing this experiment is

$$h\begin{bmatrix} 1 & 0 \\ 1 & 0 \\ \vdots & \vdots \\ 1 & 0 \end{bmatrix}\begin{bmatrix} s_1 \\ s_2 \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_N \end{bmatrix} \tag{3.29}$$

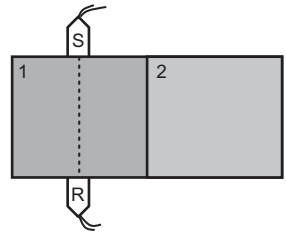

FIGURE 3.8 An acoustic travel time experiment with source S and receiver R in a medium consisting of two discrete bricks. Because of poor experimental geometry, the acoustic waves (dashed line) pass only through brick 1. The slowness of brick 2 is completely underdetermined.

where s_i is the slowness in the ith brick, h the brick width, and the d_i the measurements of travel time. If one were to attempt to solve this problem with least squares, one would find that the term $[\mathbf{G}^T\mathbf{G}]^{-1}$ is singular. Even though $M < N$, the problem is still underdetermined since the data kernel has a very poor structure. Although this is a rather trivial case in which only some of the model parameters are underdetermined, in realistic experiments, the problem arises in more subtle forms.

We shall refer to underdetermined problems that have nonzero prediction error as *mixed-determined problems*, to distinguish them from *purely underdetermined problems* that have zero prediction error.

3.6.2 Even-Determined Problems

In even-determined problems, there is exactly enough information to determine the model parameters. There is only one solution, and it has zero prediction error.

3.6.3 Overdetermined Problems

When there is too much information contained in the equation $\mathbf{Gm} = \mathbf{d}$ for it to possess an exact solution, we speak of it as being *overdetermined*. This is the case in which we can employ least squares to select a "best" approximate solution. Overdetermined problems typically have more data than unknowns, that is, $N > M$, although for the reasons discussed above it is possible to have problems that are to some degree overdetermined even when $N < M$ and to have problems that are to some degree underdetermined even when $N > M$.

To deal successfully with the full range of inverse problems, we shall need to be able to characterize whether an inverse problem is under- or overdetermined (or some combination of the two). We shall develop quantitative methods for making this characterization in Chapter 7. For the moment, we assume that it is possible to characterize the problem intuitively on the basis of the kind of experiment the problem represents.

3.7 THE PURELY UNDERDETERMINED PROBLEM

Suppose that an inverse problem $\mathbf{Gm} = \mathbf{d}$ has been identified as one that is purely underdetermined. For simplicity, assume that there are fewer equations than unknown model parameters, that is, $N < M$, and that there are no inconsistencies in these equations. It is therefore possible to find more than one solution for which the prediction error E is zero. (In fact, we shall show that underdetermined linear inverse problems have an infinite number of such solutions.) Although the data provide information about the model parameters, they do not provide enough to determine them uniquely.

We must have some means of singling out precisely one of the infinite number of solutions with zero prediction error E to obtain a unique solution $\mathbf{m}^{\text{est}}$ to

the inverse problem. To do this, we must add to the problem some information not contained in the equation $\mathbf{Gm} = \mathbf{d}$. This extra information is called *a priori* information (Jackson, 1979). *A priori* information can take many forms, but in each case, it quantifies expectations about the character of the solution that are not based on the actual data.

For instance, in the case of fitting a straight line through a single data point, one might have the expectation that the line also passes through the origin. This *a priori* information now provides enough information to solve the inverse problem uniquely, since two points (one datum, one *a priori*) determine a line.

Another example of *a priori* information concerns expectations that the model parameters possess a given sign or lie in a given range. For instance, suppose the model parameters represent density at different points in the earth. Even without making any measurements, one can state with certainty that the density is everywhere positive, since density is an inherently positive quantity. Furthermore, since the interior of the earth can reasonably be assumed to be rock, its density must have values in some range known to characterize rock, say, between 1000 and 10,000 kg/m^3. If one can use this *a priori* information when solving the inverse problem, it may greatly reduce the range of possible solutions—or even cause the solution to be unique.

There is something unsatisfying about having to add *a priori* information to an inverse problem to single out a solution. Where does this information come from, and how certain is it? There are no firm answers to these questions. In certain instances, one might be able to identify reasonable *a priori* assumptions; in other instances, one might not. Clearly, the importance of the *a priori* information depends greatly on the *use* one plans for the estimated model parameters. If one simply wants one example of a solution to the problem, the choice of *a priori* information is unimportant. However, if one wants to develop arguments that depend on the uniqueness of the estimates, the validity of the *a priori* assumptions is of paramount importance. These problems are the price one must pay for estimating the model parameters of a nonunique inverse problem. As will be shown in Chapter 6, there are other kinds of "answers" to inverse problems that do not depend on *a priori* information (localized averages, for example). However, these "answers" invariably are not as easily interpretable as estimates of model parameters.

The first kind of *a priori* assumption we shall consider is the expectation that the solution to the inverse problem is *simple*, where the notion of simplicity is quantified by some measure of the length of the solution. One such measure is simply the Euclidean length of the solution, $L = \mathbf{m}^{\mathrm{T}}\mathbf{m} = \sum m_i^2$. A solution is therefore defined to be simple if it is small when measured under the L_2 norm. Admittedly, this measure is perhaps not a particularly realistic measure of simplicity. It can be useful occasionally, and we shall describe shortly how it can be generalized to more realistic measures. One instance in which solution length may be realistic is when the model parameters describe the velocity of various points in a moving fluid. The length L is then a measure of the kinetic energy of

the fluid. In certain instances, it may be appropriate to find that velocity field in the fluid that has the smallest possible kinetic energy of those solutions satisfying the data.

We pose the following problem: Find the $\mathbf{m}^{\text{est}}$ that minimizes $L = \mathbf{m}^{\text{T}}\mathbf{m} = \sum m_i^2$ subject to the constraint that $\mathbf{e} = \mathbf{d} - \mathbf{G}\mathbf{m} = 0$. This problem can easily be solved by the method of Lagrange multipliers (see Section 14.1). We minimize the function as

$$\Phi(\mathbf{m}) = L + \sum_{i=1}^{N} \lambda_i e_i = \sum_{i=1}^{M} m_i^2 + \sum_{i=1}^{N} \lambda_i \left[d_i - \sum_{j=1}^{M} G_{ij} m_j \right] \tag{3.30}$$

with respect to m_q, where λ_i are the Lagrange multipliers. Taking the derivatives yields

$$\frac{\partial \Phi}{\partial m_q} = \sum_{i=1}^{M} 2 \frac{\partial m_i}{\partial m_q} m_i - \sum_{i=1}^{N} \lambda_i \sum_{j=1}^{M} G_{ij} \frac{\partial m_j}{\partial m_q} = 2m_q - \sum_{i=1}^{N} \lambda_i G_{iq} \tag{3.31}$$

Setting this result to zero and rewriting it in matrix notation yield the equation $2\mathbf{m} = \mathbf{G}^{\text{T}}\boldsymbol{\lambda}$, which must be solved along with the constraint equation $\mathbf{G}\mathbf{m} = \mathbf{d}$. Plugging the first equation into the second gives $\mathbf{d} = \mathbf{G}\mathbf{m} = \mathbf{G}[\mathbf{G}^{\text{T}}\boldsymbol{\lambda}/2]$. We note that the matrix $\mathbf{G}\mathbf{G}^{\text{T}}$ is a square $N \times N$ matrix. If its inverse exists, we can then solve this equation for the Lagrange multipliers, $\boldsymbol{\lambda} = 2[\mathbf{G}\mathbf{G}^{\text{T}}]^{-1}\mathbf{d}$. Then inserting this expression into the first equation yields the solution

$$\mathbf{m}^{\text{est}} = \mathbf{G}^{\text{T}}[\mathbf{G}\mathbf{G}^{\text{T}}]^{-1}\mathbf{d} \tag{3.32}$$

We shall discuss the conditions under which this solution exists later. As we shall see, one condition is that the equation $\mathbf{G}\mathbf{m} = \mathbf{d}$ be purely underdetermined—that it contain no inconsistencies.

3.8 MIXED-DETERMINED PROBLEMS

Most inverse problems that arise in practice are neither completely overdetermined nor completely underdetermined. For instance, in the X-ray tomography problem, there may be one box through which several rays pass (Figure 3.9A). The X-ray opacity of this box is clearly overdetermined. On the other hand, there may be boxes that have been missed entirely (Figure 3.9B). These boxes are completely undetermined. There may also be boxes that cannot be individually resolved because every ray that passes through one also passes through an equal distance of the other (Figure 3.9C). These boxes are also underdetermined, since only their mean opacity is determined.

Ideally, we would like to sort the unknown model parameters into two groups: those that are overdetermined and those that are underdetermined. Actually, to do this, we need to form a new set of model parameters that are linear combinations of the old. For example, in the two-box problem above, the average

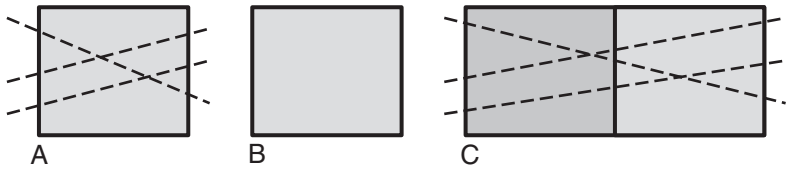

FIGURE 3.9 (A) The X-ray opacity of the brick is overdetermined, since measurements of X-ray intensity are made along three different paths (dashed lines). (B) The opacity is underdetermined, since no measurements have been made. (C) The average opacity of these two bricks is overdetermined, but since each path has an equal length in either brick, the individual opacities are underdetermined.

opacity $m_1' = \frac{1}{2}(m_1 + m_2)$ is completely overdetermined, whereas the difference in opacity $m_2' = \frac{1}{2}(m_1 - m_2)$ is completely underdetermined. We want to perform this partitioning from an arbitrary equation $\mathbf{Gm} = \mathbf{d} \rightarrow \mathbf{G}'\mathbf{m}' = \mathbf{d}'$, where $\mathbf{m}'$ is partitioned into an upper part $\mathbf{m}'^{\mathrm{o}}$ that is overdetermined and a lower part $\mathbf{m}'^{\mathrm{u}}$ that is underdetermined:

$$\begin{bmatrix} \mathbf{G}'^{\mathrm{o}} & 0 \\ 0 & \mathbf{G}'^{\mathrm{u}} \end{bmatrix} \begin{bmatrix} \mathbf{m}'^{\mathrm{o}} \\ \mathbf{m}'^{\mathrm{u}} \end{bmatrix} = \begin{bmatrix} \mathbf{d}'^{\mathrm{o}} \\ \mathbf{d}'^{\mathrm{u}} \end{bmatrix} \tag{3.33}$$

If this can be achieved, we could determine the overdetermined model parameters by solving the upper equations in the least squares sense and determine the underdetermined model parameters by finding those that have minimum L_2 solution length. In addition, we would have found a solution that added as little *a priori* information to the inverse problem as possible.

This partitioning process can be accomplished through singular-value decomposition of the data kernel, a process that we shall discuss in Chapter 7. Since it is a relatively time-consuming process, we first examine an approximate process that works if the inverse problem is not too underdetermined.

Instead of partitioning $\mathbf{m}$, suppose that we determine a solution that minimizes some combination Φ of the prediction error and the solution length for the unpartitioned model parameters:

$$\Phi(\mathbf{m}) = E + \varepsilon^2 L = \mathbf{e}^{\mathrm{T}}\mathbf{e} + \varepsilon^2 \mathbf{m}^{\mathrm{T}}\mathbf{m} \tag{3.34}$$

where the weighting factor ε^2 determines the relative importance given to the prediction error and solution length. If ε is made large enough, this procedure will clearly minimize the underdetermined part of the solution. Unfortunately, it also tends to minimize the overdetermined part of the solution. As a result, the solution will not minimize the prediction error E and will not be a very good estimate of the true model parameters. If ε is set to zero, the prediction error will be minimized, but no *a priori* information will be provided to single out the underdetermined model parameters. It may be possible, however, to find some compromise value for ε that will approximately minimize E while approximately minimizing the length of the underdetermined part of the

solution. There is no simple method of determining what this compromise ε should be (without solving the partitioned problem); it must be determined by trial and error. By minimizing $\mathbf{\Phi}(\mathbf{m})$ with respect to the model parameters in a manner exactly analogous to the least squares derivation, we obtain

$$[\mathbf{G}^{\mathrm{T}}\mathbf{G} + \varepsilon^2\mathbf{I}]\mathbf{m}^{\mathrm{est}} = \mathbf{G}^{\mathrm{T}}\mathbf{d} \quad \text{or} \quad \mathbf{m}^{\mathrm{est}} = [\mathbf{G}^{\mathrm{T}}\mathbf{G} + \varepsilon^2\mathbf{I}]^{-1}\mathbf{G}^{\mathrm{T}}\mathbf{d} \tag{3.35}$$

This estimate of the model parameters is called the *damped least squares* solution. The concept of error has been generalized to include not only prediction error but also error in fitting the *a priori* information (that the solution length is zero, in this case). The underdeterminacy of the inverse problem is said to have been damped.

3.9 WEIGHTED MEASURES OF LENGTH AS A TYPE OF *A PRIORI* INFORMATION

There are many instances in which $L = \mathbf{m}^{\mathrm{T}}\mathbf{m}$ is not a very good measure of solution simplicity. For instance, suppose that one were solving an inverse problem for density fluctuations in the ocean. One may not want to find a solution that is smallest in the sense of closest to zero but one that is smallest in the sense that it is closest to some other value, such as the average density of sea water. The obvious generalization of L is then

$$L = (\mathbf{m} - \langle\mathbf{m}\rangle)^{\mathrm{T}}(\mathbf{m} - \langle\mathbf{m}\rangle) \tag{3.36}$$

where $\langle\mathbf{m}\rangle$ is the *a priori* value of the model parameters (a known typical value of sea water, in this case).

Sometimes the whole idea of length as a measure of simplicity is inappropriate. For instance, one may feel that a solution is simple if it is flat or if it is smooth. These measures may be particularly appropriate when the model parameters represent a discretized continuous function such as density or X-ray opacity. One may have the expectation that these parameters vary only slowly with position. Fortunately, properties such as *flatness* can be easily quantified by measures that are generalizations of length. For example, the flatness of a continuous function of space can be quantified by the norm of its first derivative, which is a measure of *steepness* (the opposite of flatness). For discrete model parameters, one can use the difference between physically adjacent model parameters as approximations of a derivative. The steepness $\mathbf{l}$ of a vector $\mathbf{m}$ is then

$$\mathbf{l} = \frac{1}{\Delta x}\begin{bmatrix} -1 & 1 & & & \\ & -1 & 1 & & \\ & & \ddots & \ddots & \\ & & & -1 & 1 \end{bmatrix}\begin{bmatrix} m_1 \\ m_2 \\ \vdots \\ m_M \end{bmatrix} = \mathbf{D}\mathbf{m} \tag{3.37}$$

where $\mathbf{D}$ is the *steepness matrix*. Other methods of simplicity can also be represented by a matrix multiplying the model parameters. For instance, solution smoothness can be implemented by quantifying the *roughness* (the opposite of smoothness) by the second derivative. The matrix multiplying the model parameters would then have rows containing $(\Delta x)^{-2}[\cdots 1 \; -2 \; 1 \cdots]$. The overall steepness or roughness of the solution is then just the length

$$L = \mathbf{l}^{\mathrm{T}}\mathbf{l} = [\mathbf{Dm}]^{\mathrm{T}}[\mathbf{Dm}] = \mathbf{m}^{\mathrm{T}}\mathbf{D}^{\mathrm{T}}\mathbf{Dm} = \mathbf{m}^{\mathrm{T}}\mathbf{W}_{\mathrm{m}}\mathbf{m} \tag{3.38}$$

The matrix $\mathbf{W}_{\mathrm{m}} = \mathbf{D}^{\mathrm{T}}\mathbf{D}$ can be interpreted as a weighting factor that enters into the calculation of the length of the vector $\mathbf{m}$. Note, however, that $||\mathbf{m}||^2_{\mathrm{weighted}} = \mathbf{m}^{\mathrm{T}}\mathbf{W}_{\mathrm{m}}\mathbf{m}$ is *not* a proper norm, since it violates the positivity condition given in Equation (3.3a); that is, $||\mathbf{m}||^2_{\mathrm{weighted}} = 0$ for some nonzero vectors (such as the constant vector). This behavior usually poses no insurmountable problems, but it can cause solutions based on minimizing this norm to be nonunique.

The measure of solution simplicity can therefore be generalized to

$$L = [\mathbf{m} - \langle\mathbf{m}\rangle]^{\mathrm{T}}\mathbf{W}_{\mathrm{m}}[\mathbf{m} - \langle\mathbf{m}\rangle] \tag{3.39}$$

By suitably choosing the *a priori* model vector $\langle\mathbf{m}\rangle$ and the weighting matrix $\mathbf{W}_{\mathrm{m}}$, we can quantify a wide variety of measures of simplicity.

Weighted measures of the prediction error can also be useful. Frequently some observations are made with more accuracy than others. In this case, one would like the prediction error e_i of the more accurate observations to have a greater weight in the quantification of the overall error E than the inaccurate observations. To accomplish this weighting, we define a generalized prediction error

$$E = \mathbf{e}^{\mathrm{T}}\mathbf{W}_{\mathrm{e}}\mathbf{e} \tag{3.40}$$

where the matrix $\mathbf{W}_{\mathrm{e}}$ defines the relative contribution of each individual error to the total prediction error. Normally we would choose this matrix to be diagonal. For example, if $N = 5$ and the third observation is known to be twice as accurately determined as the others, one might use

$$\mathbf{W}_{\mathrm{e}} = \begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & & 2 & & \\ & & & 1 & \\ & & & & 1 \end{bmatrix} \tag{3.41}$$

The inverse problem solutions stated above can then be modified to take into account these new measures of prediction error and solution simplicity. The derivations are substantially the same as for the unweighted cases but the algebra is lengthy.

3.9.1 Weighted Least Squares

If the equation $\mathbf{Gm}=\mathbf{d}$ is completely overdetermined, then one can estimate the model parameters by minimizing the generalized prediction error $E=\mathbf{e}^{\mathrm{T}}\mathbf{W}_{\mathrm{e}}\mathbf{e}$. This procedure leads to the solution

$$\mathbf{m}^{\mathrm{est}} = [\mathbf{G}^{\mathrm{T}}\mathbf{W}_{\mathrm{e}}\mathbf{G}]^{-1}\mathbf{G}^{\mathrm{T}}\mathbf{W}_{\mathrm{e}}\mathbf{d} \tag{3.42}$$

3.9.2 Weighted Minimum Length

If the equation $\mathbf{Gm}=\mathbf{d}$ is completely underdetermined, then one can estimate the model parameters by choosing the solution that is simplest, where simplicity is defined by the generalized length $L = [\mathbf{m} - \langle\mathbf{m}\rangle]^{\mathrm{T}}\mathbf{W}_{\mathrm{m}}[\mathbf{m} - \langle\mathbf{m}\rangle]^{\mathrm{T}}$. This procedure leads to the solution

$$\mathbf{m}^{\mathrm{est}} = \langle\mathbf{m}\rangle + \mathbf{W}_{\mathrm{m}}^{-1}\mathbf{G}^{\mathrm{T}}[\mathbf{G}\mathbf{W}_{\mathrm{m}}^{-1}\mathbf{G}^{\mathrm{T}}]^{-1}[\mathbf{d} - \mathbf{G}\langle\mathbf{m}\rangle] \tag{3.43}$$

3.9.3 Weighted Damped Least Squares

If the equation $\mathbf{Gm}=\mathbf{d}$ is slightly underdetermined, it can be solved by minimizing a combination of prediction error and solution length, $\Phi(\mathbf{m})=E+\varepsilon^2 L$ (Franklin, 1970; Jackson, 1972; Jordan and Franklin, 1971). The parameter ε is chosen by trial and error to yield a solution that has a reasonably small prediction error. The equation for the solution, obtained by minimizing Φ with respect to $\mathbf{m}$, is then

$$[\mathbf{G}^{\mathrm{T}}\mathbf{W}_{\mathrm{e}}\mathbf{G} + \varepsilon^2\mathbf{W}_{\mathrm{m}}]\mathbf{m}^{\mathrm{est}} = \mathbf{G}^{\mathrm{T}}\mathbf{W}_{\mathrm{e}}\mathbf{d} + \varepsilon^2\mathbf{W}_{\mathrm{m}}\langle\mathbf{m}\rangle \tag{3.44}$$

or

$$\mathbf{m}^{\mathrm{est}} = [\mathbf{G}^{\mathrm{T}}\mathbf{W}_{\mathrm{e}}\mathbf{G} + \varepsilon^2\mathbf{W}_{\mathrm{m}}]^{-1}\left[\mathbf{G}^{\mathrm{T}}\mathbf{W}_{\mathrm{e}}\mathbf{d} + \varepsilon^2\mathbf{W}_{\mathrm{m}}\langle\mathbf{m}\rangle\right] \tag{3.45}$$

This equation appears to be rather complicated. However, it can be vastly simplified by noting that it is equivalent to solving the equation

$$\mathbf{F}\mathbf{m}^{\mathrm{est}} = \mathbf{f} \quad \text{with} \quad \mathbf{F} = \begin{bmatrix} \mathbf{W}_{\mathrm{e}}^{1/2}\mathbf{G} \\ \varepsilon\mathbf{D} \end{bmatrix} \quad \text{and} \quad \mathbf{f} = \begin{bmatrix} \mathbf{W}_{\mathrm{e}}^{1/2}\mathbf{d} \\ \varepsilon\mathbf{D}\langle\mathbf{m}\rangle \end{bmatrix} \quad \text{and} \quad \mathbf{W}_{\mathrm{m}} = \mathbf{D}^{\mathrm{T}}\mathbf{D} \tag{3.46}$$

by simple least squares; that is, the equation $\mathbf{F}^{\mathrm{T}}\mathbf{F}\mathbf{m}^{\mathrm{est}}=\mathbf{F}^{\mathrm{T}}\mathbf{f}$, when multiplied out, is identical to Equation (3.45). As explained previously, the weight matrix $\mathbf{W}_{\mathrm{e}}$ is typically diagonal. In that case, its square root, $\mathbf{W}_{\mathrm{e}}^{1/2}$, is also diagonal with elements that are the square roots of the corresponding elements of $\mathbf{W}_{\mathrm{e}}$.

Equation (3.46) has a very simple interpretation: its top row is the data equation $\mathbf{Gm}^{\mathrm{est}}=\mathbf{d}$, with both sides multiplied by the weight matrix $\mathbf{W}_{\mathrm{e}}^{1/2}$, and its bottom row is the *a priori* equation, $\mathbf{m}^{\mathrm{est}}=\langle\mathbf{m}\rangle$, with both sides multiplied by the *a priori* matrix, $\varepsilon\mathbf{D}$. Note that the data and *a priori* information play completely symmetric roles in this equation.

Equation (3.46) is extremely well suited to computations, especially if a sparse matrix is used for $\mathbf{F}$. As an example, suppose that $\mathbf{m}$ represents the values of a function $m(z)$ at evenly spaced zs, but that only a few of these ms have been observed. The data equation is then just $m_i = d_j$, where the indices i and j "match up" the observation with the corresponding model parameter. The ith row of the data kernel matrix $\mathbf{G}$ is all zero, except for a single one in the jth column. Since the observations are insufficient to determine all the model parameters, we add *a priori* information of smoothness using a roughness matrix $\mathbf{D}$. Each row of $\mathbf{D}$ is mostly zeros, except for the sequence $[1\ -2\ 1]$, with the -2 centered on the model parameter whose second derivative is being computed. We can only form $M-2$ of these rows, since computing the second derivative of m_1 or m_M would require model parameters off the ends of $\mathbf{m}$. We choose to add *a priori* information of flatness at these two points, with a steepness matrix $\mathbf{D}$ with rows containing the sequence $[-1\ 1]$. In both the roughness and steepness case, the vector $\mathbf{D}\langle\mathbf{m}\rangle$ is taken to be zero, since the solution is taken to be smooth and flat. This leads to an equation $\mathbf{Fm} = \mathbf{f}$ of the form

$$\mathbf{F} = \begin{bmatrix} & 1 & & & & & & & & & & \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ & & & & & & 1 & & & & & \\ - & - & - & - & - & - & - & - & - & - & - & \\ a & -2a & a & & & & & & & & & \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ & & & & & & & & a & -2a & a & \\ - & - & - & - & - & - & - & - & - & - & - & \\ -b & b & & & & & & & & & & \\ & & & & & & & & & -b & b & \end{bmatrix} \quad \text{and} \quad \mathbf{f} = \begin{bmatrix} d_1 \\ \vdots \\ d_N \\ - \\ 0 \\ \vdots \\ 0 \\ - \\ 0 \\ 0 \end{bmatrix} \tag{3.47}$$

Here $a = \varepsilon(\Delta x)^{-2}$ and $b = \varepsilon(\Delta x)^{-1}$. This equation can be solved using the biconjugate gradient algorithm, using the *MatLab* code. An example is shown in Figure 3.10.

```
tol=1e-6;
maxit=3*M;
mest=bicg( @weightedleastsquaresfcn, F'*f, tol, maxit );
```

(*MatLab* script gda03_08)

The function `weightedleastsquaresfcn()`, which performs the multiplication $\mathbf{F}^{\mathrm{T}}(\mathbf{Fv})$, is similar to the `leastsquaresfcn()` discussed previously.

Equation (3.45) can be manipulated into another useful form by subtracting $[\mathbf{G}^{\mathrm{T}}\mathbf{W}_{\mathrm{e}}\mathbf{G} + \varepsilon^2\mathbf{W}_{\mathrm{m}}]\ \langle\mathbf{m}\rangle$ from both sides of the equation and rearranging to obtain

$$[\mathbf{m}^{\mathrm{est}} - \langle\mathbf{m}\rangle] = [\mathbf{G}^{\mathrm{T}}\mathbf{W}_{\mathrm{e}}\mathbf{G} + \varepsilon^2\mathbf{W}_{\mathrm{m}}]^{-1}\mathbf{G}^{\mathrm{T}}\mathbf{W}_{\mathrm{e}}[\mathbf{d} - \mathbf{G}\langle\mathbf{m}\rangle] \tag{3.48}$$

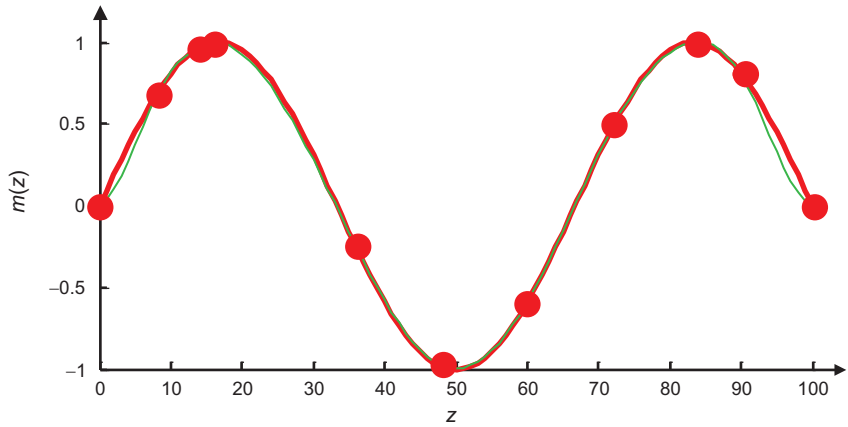

FIGURE 3.10 Example of weighted damped least squares. (Red curve) The true model, sampled with $\Delta z = 1$, is a sinusoid. (Red circles) The data are the model observed at just a few points. (Green curve) The estimated model is reconstructed from the data using the *a priori* information of smoothness in the interior of the (0, 100) interval and flatness at its ends. *MatLab* script gda03_09.

This equation is of the form

$$\begin{aligned} &\Delta\mathbf{m} = \mathbf{M}\Delta\mathbf{d} \\ &\text{with} \quad \Delta\mathbf{d} = \mathbf{d} - \mathbf{G}\langle\mathbf{m}\rangle \quad \text{and} \quad \Delta\mathbf{m} = [\mathbf{m}^{\text{est}} - \langle\mathbf{m}\rangle] \\ &\text{and} \quad \mathbf{M} = [\mathbf{G}^{\mathrm{T}}\mathbf{W}_{\mathrm{e}}\mathbf{G} + \varepsilon^2\mathbf{W}_{\mathrm{m}}]^{-1}\mathbf{G}^{\mathrm{T}}\mathbf{W}_{\mathrm{e}} \end{aligned} \tag{3.49}$$

This form emphasizes that the *deviation* $\Delta\mathbf{m}$ of the estimated solution from the *a priori* value is a linear function of the deviation $\Delta\mathbf{d}$ of the data from the value predicted by the *a priori* model.

Finally, we note that an alternative form of $\mathbf{M}$ in Equation (3.49), reminiscent of the minimum-length solution, is

$$\mathbf{M} = \mathbf{W}_{\mathrm{m}}^{-1}\mathbf{G}^{\mathrm{T}}[\mathbf{G}\mathbf{W}_{\mathrm{m}}^{-1}\mathbf{G}^{\mathrm{T}} + \varepsilon^2\mathbf{W}_{\mathrm{e}}^{-1}]^{-1} \tag{3.50}$$

The equivalence can be demonstrated by equating the two forms of $\mathbf{M}$, premultiplying by $[\mathbf{G}^{\mathrm{T}}\mathbf{W}_{\mathrm{e}}\mathbf{G} + \varepsilon^2\mathbf{W}_{\mathrm{m}}]$ and postmultiplying by $\left[\mathbf{G}\mathbf{W}_{\mathrm{m}}^{-1}\mathbf{G}^{\mathrm{T}} + \varepsilon^2\mathbf{W}_{\mathrm{e}}^{-1}\right]$. In both instances, one must take care to ascertain whether the inverses actually exist. Depending on the choice of the weighting matrices, sufficient *a priori* information may or may not have been added to the problem to damp the underdeterminacy.

3.10 OTHER TYPES OF *A PRIORI* INFORMATION

One commonly encountered type of *a priori* information is the knowledge that some function of the model parameters equals a constant. Linear equality constraints of the form $\mathbf{Hm} = \mathbf{h}$ are particularly easy to implement. For example,

one such linear constraint requires that the mean of the model parameters must equal some value h_1:

$$\mathbf{Hm} = \frac{1}{M}[1 \quad 1 \quad \cdots \quad 1]\begin{bmatrix} m_1 \\ m_2 \\ \vdots \\ m_M \end{bmatrix} = [h_1] = \mathbf{h} \tag{3.51}$$

Another such constraint requires that a particular model parameter, m_k, equals a given value

$$\mathbf{Hm} = [0 \quad \cdots \quad 0 \quad 1 \quad 0 \quad \cdots \quad 0]\begin{bmatrix} m_1 \\ \vdots \\ m_k \\ \vdots \\ m_M \end{bmatrix} = [\langle h_k \rangle] = \mathbf{h} \tag{3.52}$$

One problem that frequently arises is to solve an inverse problem $\mathbf{Gm} = \mathbf{d}$ in the least squares sense with the *a priori* constraint that linear relationships between the model parameters of the form $\mathbf{Hm} = \mathbf{h}$ are satisfied exactly.

One way to implement this constraint is to use weighted damped least squares (Equation 3.46), with $\mathbf{D} = \mathbf{H}$ and $\mathbf{D}\langle \mathbf{m} \rangle = \mathbf{h}$, and with the weighting factor ε chosen to be very large, so that the *a priori* equations are given much more weight than the data equations (Lanczos, 1961). This method is well suited for computation, but it does require the value of ε to be chosen with some care—too big and the solution will suffer from numerical noise; too small and the constraints will be only very approximately satisfied.

Another method of implementing the constraints is through the use of Lagrange multipliers. One minimizes $E = \mathbf{e}^{\mathrm{T}}\mathbf{e}$ with the constraint that $\mathbf{Hm} - \mathbf{h} = 0$ by forming the function

$$\Phi(m) = \sum_{i=1}^{N}\left[\sum_{j=1}^{M} G_{ij}m_j - d_i\right]^2 + 2\sum_{i=1}^{p}\lambda_i\left[\sum_{j=1}^{M} H_{ij}m_j - h_i\right] \tag{3.53}$$

(where there are p constraints and $2\lambda_i$ are the Lagrange multipliers) and setting its derivatives with respect to the model parameters to zero as

$$\frac{\partial \Phi(\mathbf{m})}{\partial m_q} = 2\sum_{i=1}^{M} m_i \sum_{j=1}^{N} G_{jq}G_{ji} - 2\sum_{i=1}^{N} G_{iq}d_i + 2\sum_{i=1}^{p}\lambda_i H_{iq} \tag{3.54}$$

This equation must be solved simultaneously with the constraint equations $\mathbf{Hm} = \mathbf{h}$ to yield the estimated solution. These equations, in matrix form, are

$$\begin{bmatrix} \mathbf{G}^{\mathrm{T}}\mathbf{G} & \mathbf{H}^{\mathrm{T}} \\ \mathbf{H} & 0 \end{bmatrix}\begin{bmatrix} \mathbf{m} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{G}^{\mathrm{T}}\mathbf{d} \\ \mathbf{h} \end{bmatrix} \tag{3.55}$$

Although these equations can be manipulated to yield an explicit formula for $\mathbf{m}^{\text{est}}$, it is often more convenient to solve directly this $M+p$ system of equations for M estimates of model parameters and p Lagrange multipliers by premultiplying by the inverse of the square matrix.

3.10.1 Example: Constrained Fitting of a Straight Line

Consider the problem of fitting the straight line $d_i = m_1 + m_2 z_i$ to data, where one has *a priori* information that the line must pass through the point (z', d') (Figure 3.11). There are two model parameters: intercept m_1 and slope m_2. The $p=1$ constraint is that $d' = m_1 + m_2 z'$, or

$$\mathbf{Hm} = \begin{bmatrix} 1 & z' \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \end{bmatrix} = [d'] = \mathbf{h} \tag{3.56}$$

Using the $\mathbf{G}^{\mathrm{T}}\mathbf{G}$ and $\mathbf{G}^{\mathrm{T}}\mathbf{d}$ computed in Section 3.5.1, the solution is

$$\begin{bmatrix} m_1^{\text{est}} \\ m_2^{\text{est}} \\ \lambda_1 \end{bmatrix} = \begin{bmatrix} N & \sum_{i=1}^{N} z_i & 1 \\ \sum_{i=1}^{N} z_i & \sum_{i=1}^{N} z_i^2 & z' \\ 1 & z' & 0 \end{bmatrix}^{-1} \begin{bmatrix} \sum_{i=1}^{N} d_i \\ \sum_{i=1}^{N} z_i d_i \\ d' \end{bmatrix} \tag{3.57}$$

Another kind of *a priori* constraint is the *linear inequality constraint*, which we can write as $\mathbf{Hm} \geq \mathbf{h}$ (the inequality being interpreted component by component). Note that this form can also include $\leq$ inequalities by multiplying the inequality relation by -1. This kind of *a priori* constraint has application to problems in which the model parameters are inherently positive quantities,

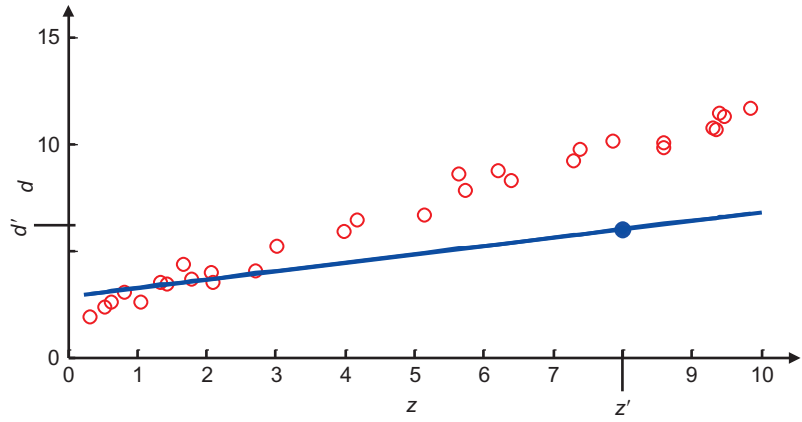

FIGURE 3.11 Least squares fitting of a straight line to (z, d) data, where the line is constrained to pass through the point $(z', d') = (8, 6)$. *MatLab* script gda03_10.

$m_i > 0$, and to other cases when the solution is known to possess some kind of bounds. One could therefore propose a new kind of constrained least squares solution of overdetermined problems, one that minimizes the error subject to the given inequality constraints. *A priori* inequality constraints also have application to underdetermined problems. One can find the smallest solution that solves both $\mathbf{Gm} = \mathbf{d}$ and $\mathbf{Hm} \geq \mathbf{h}$. These problems can be solved in a straightforward fashion, which will be discussed in Chapter 7.

3.11 THE VARIANCE OF THE MODEL PARAMETER ESTIMATES

The data invariably contain noise that causes errors in the estimates of the model parameters. We can calculate how this measurement error *maps* into errors in $\mathbf{m}^{\text{est}}$ by noting that all of the formulas derived above for estimates of the model parameters are linear functions of the data, of the form $\mathbf{m}^{\text{est}} = \mathbf{Md} + \mathbf{v}$, where $\mathbf{M}$ is some matrix and $\mathbf{v}$ some vector. Therefore, if we assume that the data have a distribution characterized by some covariance matrix [cov $\mathbf{d}$], the estimates of the model parameters have a distribution characterized by a covariance matrix [cov $\mathbf{m}$] $= \mathbf{M}$[cov $\mathbf{d}$]$\mathbf{M}^{\mathbf{T}}$. The covariance of the solution can therefore be calculated in a straightforward fashion. If the data are uncorrelated and of equal variance σ_{d}^2, then very simple formulas are obtained for the covariance of some of the more simple inverse problem solutions.

The simple least squares solution $\mathbf{m}^{\text{est}} = [\mathbf{G}^{\mathrm{T}}\mathbf{G}]^{-1}\mathbf{G}^{\mathrm{T}}\mathbf{d}$ has covariance

$$[\text{cov}\,\mathbf{m}] = \left[[\mathbf{G}^{\mathrm{T}}\mathbf{G}]^{-1}\mathbf{G}^{\mathrm{T}}\right]\sigma_{\text{d}}^2\mathbf{I}\left[[\mathbf{G}^{\mathrm{T}}\mathbf{G}]^{-1}\mathbf{G}^{\mathrm{T}}\right]^{\mathrm{T}} = \sigma_{\text{d}}^2[\mathbf{G}^{\mathrm{T}}\mathbf{G}]^{-1} \tag{3.58}$$

and the simple minimum-length solution $\mathbf{m}^{\text{est}} = \mathbf{G}^{\mathrm{T}}[\mathbf{G}\mathbf{G}^{\mathrm{T}}]^{-1}\mathbf{d}$ has covariance

$$[\text{cov}\,\mathbf{m}] = \left[\mathbf{G}^{\mathrm{T}}[\mathbf{G}\mathbf{G}^{\mathrm{T}}]^{-1}\right]\sigma_{\text{d}}{}^2\mathbf{I}\left[\mathbf{G}^{\mathrm{T}}[\mathbf{G}\mathbf{G}^{\mathrm{T}}]^{-1}\right]^{\mathrm{T}} = \sigma_{\text{d}}^2\mathbf{G}^{\mathrm{T}}[\mathbf{G}\mathbf{G}^{\mathrm{T}}]^{-2}\mathbf{G} \tag{3.59}$$

An important issue is how to arrive at an estimate of the variance of the data σ_{d}^2 that can be used in these equations. One possibility is to base it upon knowledge about the inherent accuracy of the measurement process, in which case it is termed an *a priori variance*. For instance, if lengths are being measured with a ruler with 1 mm divisions, the estimate $\sigma_{\text{d}} \approx 1/2$ mm would be reasonable. Another possibility is to base the estimate upon the size distribution of prediction errors $\mathbf{e}$ determined by fitting a model to the data, in which case it is termed an *a posteriori* variance. A reasonable estimate, whose theoretical justification will be discussed in Chapter 5, is

$$\sigma_{\text{d}}{}^2 \approx \frac{1}{N-M}\sum_{i=1}^{N} e_i^2 \tag{3.60}$$

This formula is essentially the mean-squared error $N^{-1}\sum_{i=1}^{N} e_i^2$, except that N has been replaced by $N-M$ to account for the ability of a model with M parameters

to exactly fit M data. A posteriori estimates are usually overestimates because inaccuracies in the model contribute to the size of the prediction error.

The least squares rule for error propagation, $[\text{cov}\,\mathbf{m}] = \sigma_d^2[\mathbf{G}^T\mathbf{G}]^{-1}$, indicates that the model parameters can be correlated and can be of unequal variance even when the data are uncorrelated and are of equal variance. Whether observational error is attenuated or amplified by the inversion process is critically dependent upon the structure of the data kernel $\mathbf{G}$. In the problem for the mean of N data, discussed above, observational error is attenuated, but this desirable behavior is not common to all inverse problems (Figure 3.12).

3.12 VARIANCE AND PREDICTION ERROR OF THE LEAST SQUARES SOLUTION

If the prediction error $E(\mathbf{m}) = \mathbf{e}^T\mathbf{e}$ of an overdetermined problem has a very sharp minimum in the vicinity of the estimated solution $\mathbf{m}^{est}$, we would expect that the solution is well determined in the sense that it has small variance. Small errors in determining the shape of $E(\mathbf{m})$ due to random fluctuations in the data lead to only small errors in $\mathbf{m}^{est}$. Conversely, if $E(\mathbf{m})$ has a broad minimum, we expect that $\mathbf{m}^{est}$ has a large variance. Since the curvature of a function is a measure of the sharpness of its minimum, we expect that the variance of the solution is related to the curvature of $E(\mathbf{m})$ at its minimum, which in turn depends on the structure of the data kernel $\mathbf{G}$ (Figure 3.13).

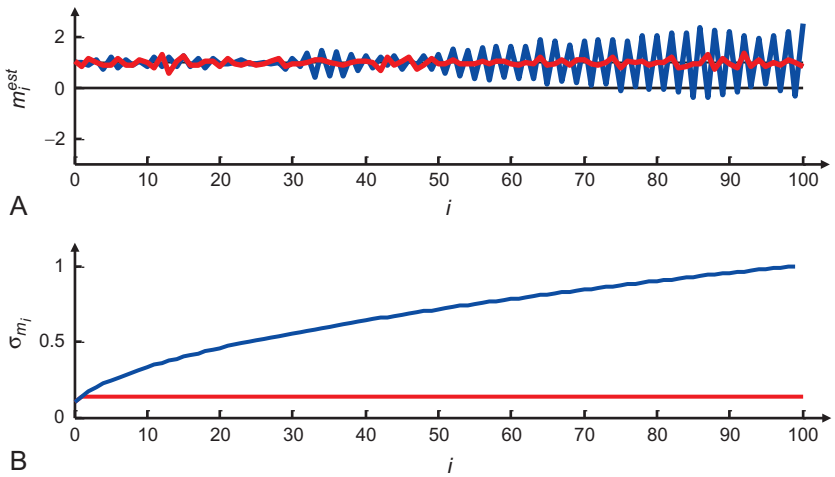

FIGURE 3.12 Two hypothetical experiments to measure the weight m_i of each of 100 bricks. In experiment 1 (red), the bricks are accumulated on a scale so that observation d_i is the sum of the weight of the first i bricks. In experiment 2 (blue), the first brick is weighed, and then subsequently, pairs or bricks (the first and the second, the second and the third, and so forth). (A) Least squares solution for weights m_i. (B) Corresponding error σ_{m_i}. Note that the first experiment has the lower error. *MatLab* script gda03_11.

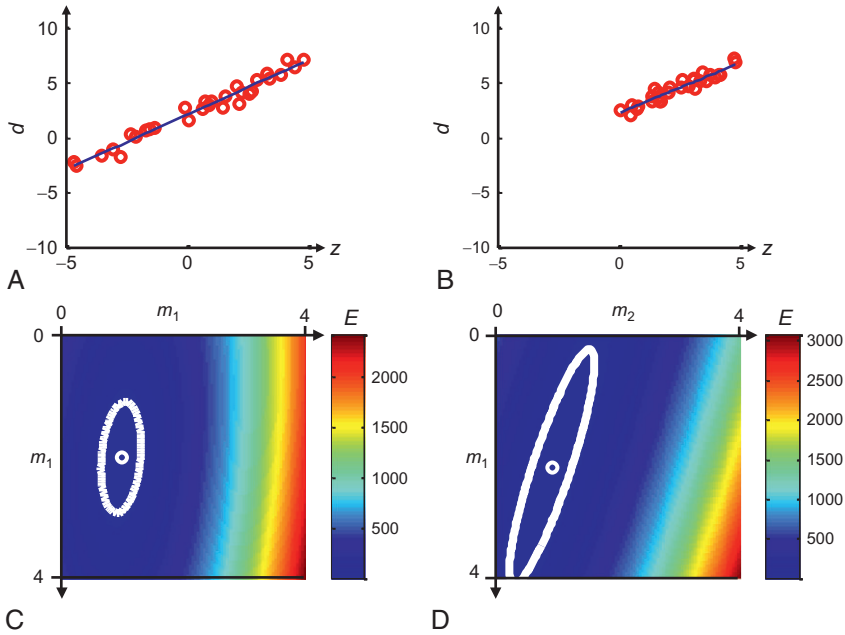

FIGURE 3.13 (A) Least squares fitting of a straight line (blue) to (z, d) data (red). (C) The best estimate of the model parameters ($m_1^{\text{est}}, m_2^{\text{est}}$) (white circle) occurs at the minimum of the error surface $E(m_1, m_2)$, which is a function of model parameters, intercept m_1 and slope m_2. The minimum is surrounded by a region of low error (white ellipse) that corresponds to lines that fit "almost as well" as the best estimate. The variance of the estimate is related to the size of the ellipse. In this example, the ellipse is narrowest in the m_2 direction, indicating that the slope m_2 is determined more accurately than intercept m_1. The geometry of the experiment, and not the overall level of observational error, determines the shape of the ellipse, as can be seen from the example in (B) and (D). The tilt of the ellipse indicates that the intercept and slope are negatively correlated. *MatLab* script gda03_12.

The curvature of the prediction error can be measured by its second derivative, as we can see by computing how small changes in the model parameters change the prediction error. Expanding the prediction error in a Taylor series about its minimum and keeping up to second-order terms give

$$\Delta E = E(\mathbf{m}) - E(\mathbf{m}^{\text{est}}) = [\mathbf{m} - \mathbf{m}^{\text{est}}]^{\text{T}} \left[\frac{1}{2}\frac{\partial^2 E}{\partial \mathbf{m}^2}\right]_{\mathbf{m}=\mathbf{m}^{\text{est}}} [\mathbf{m} - \mathbf{m}^{\text{est}}] \qquad (3.61)$$

Here the matrix $\frac{\partial^2 E}{\partial \mathbf{m}^2}$ has elements $\frac{\partial^2 E}{\partial m_i \partial m_j}$. Note that the first-order term is zero, since the expansion is made at a minimum. The second derivative can also be computed directly from the expression

$$E(\mathbf{m}) = ||\mathbf{d} - \mathbf{G}\mathbf{m}||_2^2 = \sum_{i=1}^{N} d_i^2 - 2\sum_{i=1}^{N} d_i \sum_{j=1}^{M} G_{ij} m_j + \sum_{i=1}^{N}\sum_{j=1}^{M} G_{ij} m_j \sum_{k=1}^{M} G_{ik} m_k \qquad (3.62)$$

which gives

$$\frac{\partial^2 E}{\partial m_p \partial m_q} = 2\sum_{i=1}^{N}\sum_{j=1}^{M} G_{ij}\frac{\partial m_j}{\partial m_p}\sum_{k=1}^{M} G_{ik}\frac{\partial m_k}{\partial m_q} = 2\sum_{i=1}^{N} G_{ip}G_{iq} \quad \text{or} \quad \left[\frac{1}{2}\frac{\partial^2 E}{\partial \mathbf{m}^2}\right] = \mathbf{G}^{\mathrm{T}}\mathbf{G} \tag{3.63}$$

The covariance of the least squares solution (assuming uncorrelated data all with equal variance σ_{d}^2) is therefore

$$[\mathrm{cov}\,\mathbf{m}] = \sigma_{\mathrm{d}}^2[\mathbf{G}^{\mathrm{T}}\mathbf{G}]^{-1} = \sigma_{\mathrm{d}}^2\left[\frac{1}{2}\frac{\partial^2 E}{\partial \mathbf{m}^2}\right]^{-1}_{\mathbf{m}=\mathbf{m}^{\mathrm{est}}} \tag{3.64}$$

The prediction error $E = \mathbf{e}^{\mathrm{T}}\mathbf{e}$ is the sum of squares of Gaussian data minus a constant. It is, therefore, a random variable with a χ^2 distribution with $N-M$ degrees of freedom, which has mean $(N-M)\sigma_{\mathrm{d}}^2$ and variance $2(N-M)\sigma_{\mathrm{d}}^4$. (The degrees of freedom are reduced by M since the model can force M linear combinations of the e_i to zero.) We can use the standard deviation of E

$$\sigma_E = [2(N-M)]^{1/2}{\sigma_{\mathrm{d}}}^2 \tag{3.65}$$

in the expression for variance as

$$[\mathrm{cov}\,\mathbf{m}] = {\sigma_{\mathrm{d}}}^2[\mathbf{G}^{\mathrm{T}}\mathbf{G}]^{-1} = \frac{\sigma_E}{[2(N-M)]^{1/2}}\left[\frac{1}{2}\frac{\partial^2 E}{\partial \mathbf{m}^2}\right]^{-1}_{\mathbf{m}=\mathbf{m}^{\mathrm{est}}} \tag{3.66}$$

The covariance [cov $\mathbf{m}$] can be interpreted as being controlled either by the variance of the data times a measure of how error in the data is mapped into error in the model parameters or by the standard deviation of the total prediction error times a measure of the curvature of the prediction error at its minimum.

The methods of solving inverse problems that have been discussed in this chapter emphasize the data and model parameters themselves. The method of least squares estimates the model parameters with smallest prediction length. The method of minimum-length estimates the simplest model parameters. The ideas of data and model parameters are very concrete and straightforward, and the methods based on them are simple and easily understood. Nevertheless, this viewpoint tends to obscure an important aspect of inverse problems: that the nature of the problems depends more on the *relationship* between the data and model parameters than on the data or model parameters themselves. It should, for instance, be possible to tell a well-designed experiment from a poor one without knowing what the numerical values of the data or model parameters are, or even the range in which they fall. In the next chapter, we will begin to explore this kind of problem.

3.13 PROBLEMS

3.1. Show that the equations worked out for the straight line problem in Equation (3.5) have the solution given in Equation (3.17).

3.2. This problem builds on Problem 1.1. Suppose that you determine the masses of 100 objects by weighing the first, then weighing the first and second together, and then weighing the rest in triplets: the first, second, and third; the second, third, and fourth; and so forth. Write a *MatLab* script that (A) randomly assigns masses m_i^{true} to the objects in the range of 0–1 kg; (B) builds the appropriate data kernel $\mathbf{G}$; (C) creates synthetic observed data $\mathbf{d}^{\text{obs}} = \mathbf{Gm} + \mathbf{n}$, where $\mathbf{n}$ is a vector of Gaussian random numbers with zero mean and $\sigma_{\text{d}} = 0.01$ kg; (D) solves the inverse problem by simple least squares; (E) estimates the variance of each of the estimated model parameters $\mathbf{m}^{\text{est}}$; and (F) counts up the number of estimated model parameters that are within $\pm 2\sigma_{\text{m}}$ of their true value. (G) Make a plot of σ_{m} as a function of the index of the model parameter. Does it decline, remain constant, or grow?

3.3. This problem builds on Problem 1.2. Suppose that you determine the height of 50 objects by measuring the first, and then stacking the second on top of the first and measuring their combined height, stacking the third on top of the first two and measuring their combined height, and so forth. Write a *MatLab* script that (A) randomly assigns heights m_i^{true} to the objects in the range of 0–1 m; (B) builds the appropriate data kernel $\mathbf{G}$; (C) creates synthetic observed data $\mathbf{d}^{\text{obs}} = \mathbf{Gm} + \mathbf{n}$, where $\mathbf{n}$ is a vector of Gaussian random numbers with zero mean and $\sigma_{\text{d}} = 0.01$ m; (D) solves the inverse problem by simple least squares; (E) estimates the variance of each of the estimated model parameters $\mathbf{m}^{\text{est}}$; and (F) counts up the number of estimated model parameters that are within $\pm 2\sigma_m$ of their true value. (G) Make a plot of σ_{m} as a function of the index of the model parameter. Does it decline, remain constant, or grow?

3.4. This problem builds on Problem 1.3, which considers the cubic equation, $d_i = m_1 + m_2 z_i + m_3 z_i^2 + m_4 z_i^3$. Write a *MatLab* script that (A) computes a vector $\mathbf{z}$ with $N = 11$ elements equally spaced from 0 to 1; (B) randomly assigns the elements of $\mathbf{m}^{\text{true}}$ in the range of -1 to 1; (C) builds the appropriate data kernel $\mathbf{G}$; (D) creates synthetic observed data $\mathbf{d}^{\text{obs}} = \mathbf{Gm} + \mathbf{n}$, where $\mathbf{n}$ is a vector of Gaussian random numbers with zero mean and $\sigma_{\text{d}} = 0.05$; (E) solves the inverse problem by simple least squares; (F) calculates the predicted data, $\mathbf{d}^{\text{pre}} = \mathbf{Gm}^{\text{est}}$; and (G) plots d_i^{obs} and d_i^{pre}. Comment upon the results.

3.5. This problem builds on Problem 3.4. Modify your solution of Problem 3.4 by adding the constraint that the predicted data pass through the point $(z', d') = (5, 0)$. Comment upon the results.

REFERENCES

Franklin, J.N., 1970. Well-posed stochastic extensions of ill-posed linear problems. J. Math. Anal. Appl. 31, 682–716.

Jackson, D.D., 1972. Interpretation of inaccurate, insufficient and inconsistent data. Geophys. J. R. Astron. Soc. 28, 97–110.

Jackson, D.D., 1979. The use of a priori data to resolve non-uniqueness in linear inversion. Geophys. J. R. Astron. Soc. 57, 137–157.

Jordan, T.H., Franklin, J.N., 1971. Optimal solutions to a linear inverse problem in geophysics. Proc. Natl. Acad. Sci. USA 68, 291–293.

Lanczos, C., 1961. Linear Differential Operators. Van Nostrand-Reinhold, Princeton, NJ.

Menke, W., Menke, J., 2011. Environmental Data Analysis with MatLab. Academic Press, Elsevier Inc, Oxford, UK 263pp.