Chapter 1

Describing Inverse Problems

1.1 FORMULATING INVERSE PROBLEMS

The starting place in most inverse problems is a description of the data. Since in most inverse problems the data are simply a list of numerical values, a vector provides a convenient means of their representation. If N measurements are performed in a particular experiment, for instance, one might consider these numbers as the elements of a vector $\mathbf{d}$ of length N.

The purpose of data analysis is to gain *knowledge* through systematic examination of data. While knowledge can take many forms, we assume here that it is primarily numerical in nature. We analyze data so to infer, as best we can, the values of numerical quantities—*model parameters*. Model parameters are chosen to be *meaningful*; that is, they are chosen to capture the essential character of the processes that are being studied. The model parameters can be represented as the elements of a vector $\mathbf{m}$, which is of length M

$$\begin{aligned} \text{data: } \mathbf{d} &= [d_1, d_2, d_3, d_4, \ldots, d_N]^{\mathrm{T}} \\ \text{model parameters: } \mathbf{m} &= [m_1, m_2, m_3, m_4, \ldots, m_M]^{\mathrm{T}} \end{aligned} \tag{1.1}$$

Here, T signifies transpose.

The basic statement of an inverse problem is that the model parameters and the data are in some way related. This relationship is called the *quantitative model* (or *model*, or *theory*, for short). Usually, the model takes the form of one or more formulas that the data and model parameters are expected to follow.

If, for instance, one were attempting to determine the density of an object, such as a rock, by measuring its mass and volume, there would be $N=2$ data—mass and volume (say, d_1 and d_2, respectively)—and $M=1$ unknown model parameter, density (say, m_1). The model would be the statement that density times volume equals mass, which can be written compactly by the vector equation $d_2 m_1 = d_1$. Note that the model parameter, density, is more meaningful than either mass or volume, in that it represents an intrinsic property of a substance that is related to its chemistry. The data—mass and volume—are easy to measure, but they are less fundamental because they depend on the size of the object, which is usually incidental.

Geophysical Data Analysis: Discrete Inverse Theory. DOI: 10.1016/B978-0-12-397160-9.00001-1

In more realistic situations, the data and model parameters are related in more complicated ways. Most generally, the data and model parameters might be related by one or more implicit equations such as

$$\begin{matrix} f_1(\mathbf{d},\mathbf{m}) = 0 \\ f_2(\mathbf{d},\mathbf{m}) = 0 \\ \vdots \\ f_L(\mathbf{d},\mathbf{m}) = 0 \end{matrix} \quad \text{or} \quad \mathbf{f}(\mathbf{d},\mathbf{m}) = 0 \tag{1.2}$$

where L is the number of equations. In the above example concerning the measuring of density, $L=1$ and $d_2m_1-d_1=0$ would constitute the one equation of the form $f_1(\mathbf{d}, \mathbf{m})=0$. These implicit equations, which can be compactly written as the vector equation $\mathbf{f}(\mathbf{d}, \mathbf{m})=0$, summarize what is known about how the measured data and the unknown model parameters are related. The purpose of inverse theory is to solve, or "invert," these equations for the model parameters, or whatever kinds of answers might be possible or desirable in any given situation.

No claims are made either that the equations $\mathbf{f}(\mathbf{d}, \mathbf{m})=0$ contain enough information to specify the model parameters uniquely or that they are even consistent. One of the purposes of inverse theory is to answer these kinds of questions and provide means of dealing with the problems that they imply. In general, $\mathbf{f}(\mathbf{d}, \mathbf{m})=0$ can consist of arbitrarily complicated (nonlinear) functions of the data and model parameters. In many problems, however, the equation takes on one of several simple forms. It is convenient to give names to some of these special cases, since they commonly arise in practical problems; we shall give them special consideration in later chapters.

1.1.1 Implicit Linear Form

The function $\mathbf{f}$ is linear in both data and model parameters and can therefore be written as the matrix equation

$$\mathbf{f}(\mathbf{d},\mathbf{m}) = 0 = \mathbf{F}\begin{bmatrix}\mathbf{d}\\ \mathbf{m}\end{bmatrix} = \mathbf{F}\mathbf{x} \tag{1.3}$$

where $\mathbf{F}$ is an $L \times (M+N)$ matrix and the vector $\mathbf{x}=[\mathbf{d}^T, \mathbf{m}^T]^T$ is a concatenation of $\mathbf{d}$ and $\mathbf{m}$, that is, $\mathbf{x}=[d_1, d_2, \ldots, d_N, m_1, m_2, \ldots, m_M]^T$.

1.1.2 Explicit Form

In many instances, it is possible to separate the data from the model parameters and thus to form $L=N$ equations that are linear in the data (but still nonlinear in the model parameters through a vector function $\mathbf{g}$).

$$\mathbf{f}(\mathbf{d},\mathbf{m}) = 0 = \mathbf{d} - \mathbf{g}(\mathbf{m}) \tag{1.4}$$

1.1.3 Explicit Linear Form

In the explicit linear form, the function **g** is also linear, leading to the $N \times M$ matrix equation (where $L = N$)

$$\mathbf{f}(\mathbf{d}, \mathbf{m}) = 0 = \mathbf{d} - \mathbf{Gm} \tag{1.5}$$

This form is equivalent to a special case of the matrix **F** in Section **1.1.1**:

$$\mathbf{F} = [\mathbf{I}, -\mathbf{G}] \tag{1.6}$$

1.2 THE LINEAR INVERSE PROBLEM

The simplest and best-understood inverse problems are those that can be represented with the explicit linear equation $\mathbf{Gm} = \mathbf{d}$. This equation, therefore, forms the foundation of the study of discrete inverse theory. As will be shown below, many important inverse problems that arise in the physical sciences involve precisely this equation. Others, while involving more complicated equations, can often be solved through linear approximations.

The matrix **G** is called the data kernel, in analogy to the theory of integral equations, in which the analogs of the data and model parameters are two continuous functions $d(x)$ and $m(x)$, where x is some independent variable. Continuous inverse theory lies between these two extremes, with discrete data but a continuous model function.

Discrete inverse theory:

$$d_i = \sum_{j=1}^{M} G_{ij} m_j \tag{1.7a}$$

Continuous inverse theory:

$$d_i = \int G_i(x)\, m(x)\, \mathrm{d}x \tag{1.7b}$$

Integral equation theory:

$$d(y) = \int G(y, x)\, m(x)\, \mathrm{d}x \tag{1.7c}$$

The main difference among discrete inverse theory, continuous inverse theory, and integral equation theory is whether the model m and data d are treated as continuous functions or discrete parameters. The data d_i in inverse theory are necessarily discrete, since inverse theory is concerned with deducing knowledge from observational data, which always has a discrete nature. Both continuous inverse problems and integral equations can be converted to discrete inverse problems by approximating the model $m(x)$ as a vector of its values at a set of M closely spaced points

$$\mathbf{m} = [m(x_1), m(x_2), m(x_3), \ldots, m(x_M)]^T \tag{1.8}$$

and the integral as a Riemann summation (or by some other quadrature formula).

1.3 EXAMPLES OF FORMULATING INVERSE PROBLEMS

1.3.1 Example 1: Fitting a Straight Line

Suppose that N temperature measurements T_i are made at times t_i in the atmosphere (Figure 1.1). The data are then a vector $\mathbf{d}$ of N measurements of temperature, where $\mathbf{d} = [T_1, T_2, T_3, \ldots, T_N]^T$. The times t_i are not, strictly speaking, data. Instead, they provide some auxiliary information that describes the geometry of the experiment. This distinction will be further clarified below.

Suppose that we assume a model in which temperature is a linear function of time: $T = a + bt$. The intercept a and slope b then form the two model parameters

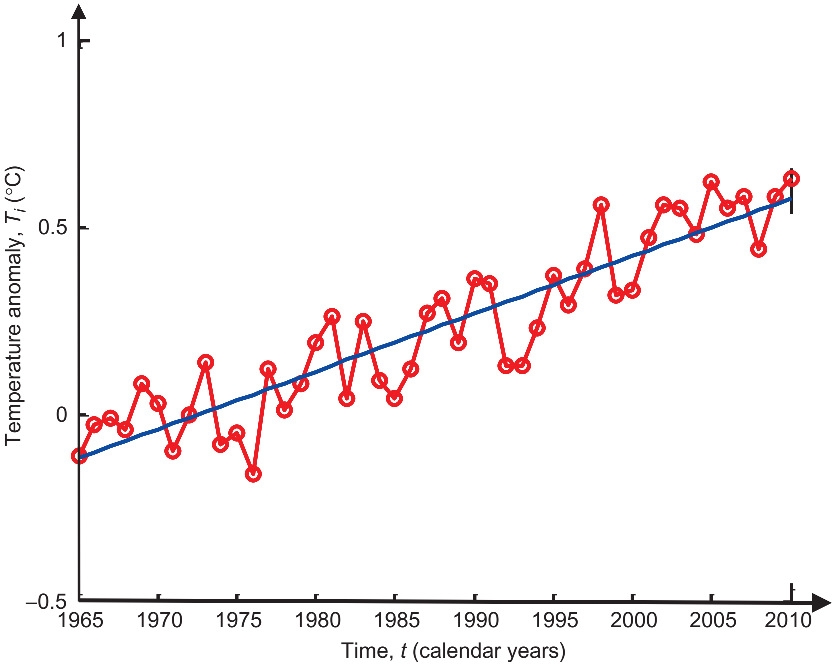

FIGURE 1.1 (Red) Average global temperature for the time period, 1965–2010. The inverse problem is to determine the rate of increase of temperature and its confidence interval. (Blue) Straight line fit to data. The slope of the line is 0.015 ± 0.002 (95%) °C/year. Data from Hansen et al. 2010. *MatLab* script gda01_01.

of the problem, $\mathbf{m} = [a, b]^T$. According to the model, each temperature observation must satisfy $T_i = a + bt_i$:

$$\begin{aligned} T_1 &= a + bt_1 \\ T_2 &= a + bt_2 \\ &\vdots \\ T_N &= a + bt_N \end{aligned} \tag{1.9}$$

These equations can be arranged as the matrix equation $\mathbf{d} = \mathbf{Gm}$

$$\begin{bmatrix} T_1 \\ T_2 \\ \vdots \\ T_N \end{bmatrix} = \begin{bmatrix} 1 & t_1 \\ 1 & t_2 \\ \vdots & \vdots \\ 1 & t_N \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} \tag{1.10}$$

In *MatLab*, the matrix **G** is computed as:

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

(*MatLab* script gda01_01)

1.3.2 Example 2: Fitting a Parabola

If the model in Example 1 is changed to assume a quadratic variation of temperature with depth of the form $T = a + bt + ct^2$, then a new model parameter c is added to the problem, and $\mathbf{m} = [a, b, c]^T$. The number of model parameters is now $M = 3$. The data and model parameters are supposed to satisfy

$$\begin{aligned} T_1 &= a + bt_1 + ct_1^2 \\ T_2 &= a + bt_2 + ct_2^2 \\ &\vdots \\ T_N &= a + bt_N + ct_N^2 \end{aligned} \tag{1.11}$$

These equations can be arranged into the matrix equation

$$\begin{bmatrix} T_1 \\ T_2 \\ \vdots \\ T_N \end{bmatrix} = \begin{bmatrix} 1 & t_1 & t_1^2 \\ 1 & t_2 & t_2^2 \\ \vdots & \vdots & \vdots \\ 1 & t_N & t_N^2 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} \tag{1.12}$$

This matrix equation has the explicit linear form $\mathbf{d} = \mathbf{Gm}$. Note that, although the equation is linear in the data and model parameters, it is not linear in the auxiliary variable t.

The equation has a very similar form to the equation of the previous example, which brings out one of the underlying reasons for employing matrix notation: it can often emphasize similarities between superficially different problems. In *MatLab*, the matrix **G** is computed as:

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

(*MatLab* script gda01_02)

Note the use of the element-wise power, signified “.ˆ,” to compute $t_i{}^2$.

1.3.3 Example 3: Acoustic Tomography

Suppose that a wall is assembled from a rectangular array of bricks (Figure 1.2) and that each brick is composed of a different type of clay. If the acoustic velocities of the different clays differ, one might attempt to distinguish the different kinds of bricks by measuring the travel time of sound across the various rows and columns of bricks in the wall. The data in this problem are $N=8$ measurements of travel times, $d=[T_1, T_2, T_3, \ldots, T_8]^{\mathrm{T}}$. The model assumes that each brick is composed of a uniform material and that the travel time of sound across each brick is proportional to the width and height of the brick. The proportionality factor is the brick's *slowness* s_i, thus giving $M=16$ model parameters $\mathbf{m}=[s_1, s_2, s_3, \ldots, s_{16}]^{\mathrm{T}}$, where the ordering is according to the numbering scheme of the figure. The data and model parameters are related by

$$\begin{aligned} \text{row } 1: T_1 &= hs_1 + hs_2 + hs_3 + hs_4 \\ \text{row } 2: T_2 &= hs_5 + hs_6 + hs_7 + hs_8 \\ &\vdots \\ \text{column } 4: T_8 &= hs_4 + hs_8 + hs_{12} + hs_{16} \end{aligned} \tag{1.13}$$

and the matrix equation is

$$\begin{bmatrix} T_1 \\ T_2 \\ \vdots \\ T_8 \end{bmatrix} = h \begin{bmatrix} 1&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&1&1&1&1&0&0&0&0&0&0&0&0 \\ \vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots \\ 0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&1 \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_{16} \end{bmatrix} \tag{1.14}$$

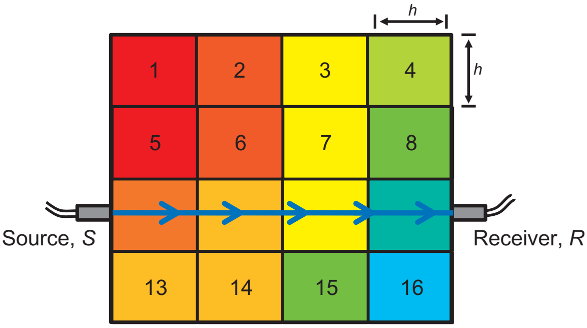

FIGURE 1.2 The travel time of acoustic waves (blue line) through the rows and columns of a square array of bricks is measured with acoustic source S and receiver R placed on the edges of the square. The inverse problem is to infer the acoustic properties of the bricks, here depicted by the colors. Although the overall pattern is spatially variable, individual bricks are assumed to be homogeneous.

Here, the bricks are assumed to be of width *and* height *h*. The *MatLab* code for constructing **G** is:

```
G=zeros(N,M);
for i = [1:4]
for j = [1:4]
    % measurements over rows
    k = (i-1)*4 + j;
    G(i,k)=h;
    % measurements over columns
    k = (j-1)*4 + i;
    G(i+4,k)=h;
end
end
```

(*MatLab* script gda01_03)

1.3.4 Example 4: X-ray Imaging

Tomography is the process of forming images of the interior of an object from measurements made along rays passed through that object (“tomo” comes from the Greek word for “slice”). The computerized tomography scanner is an X-ray imaging device that has revolutionized the diagnosis of brain tumors and many other medical conditions. The scanner solves an inverse problem for the X-ray opacity of body tissues using measurements of the amount of radiation absorbed from many crisscrossing beams of X-rays (Figure 1.3).

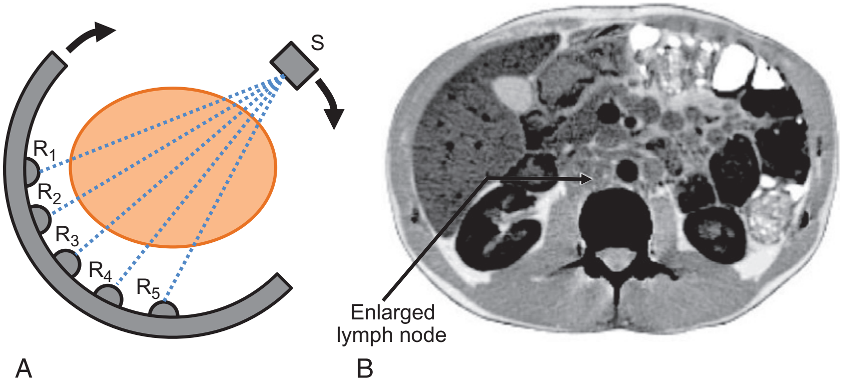

FIGURE 1.3 (A) An idealized computed tomography (CT) medical scanner measures the X-ray absorption along lines (blue) passing through the body of the patient (orange). After a set of measurements are made, the source *S* and receivers R_i are rotated, and the measurements are repeated so that data along many crisscrossing lines are collected. The inverse problem is to determine the X-ray opacity as a function of position in the body. (B) Actual CT image of a patient infected with *Mycobacterium genavense* (from de Lastours et al., 2008).

The basic physical model underlying this device is the idea that the intensity of X-rays diminishes with the distance traveled, at a rate proportional to the intensity of the X-ray beam, and an absorption coefficient that depends on the type of tissue:

$$\frac{\mathrm{d}I}{\mathrm{d}s} = -c(x,y)\,I \tag{1.15}$$

Here, I is the intensity of the beam, s the distance along the beam, and $c(x,y)$ the absorption coefficient, which varies with position. If the X-ray source has intensity I_0, then the intensity at the ith detector is

$$I_i = I_0 \exp\left\{-\int_{\text{beam}\,i} c(x,y)\,\mathrm{d}s\right\} \approx I_0\left\{1 - \int_{\text{beam}\,i} c(x,y)\,\mathrm{d}s\right\} \tag{1.16a}$$

$$I_0 - I_i = I_0 \int_{\text{beam}\,i} c(x,y)\,\mathrm{d}s \tag{1.16b}$$

Note that Equation (1.16a) is a nonlinear function of the unknown absorption coefficient $c(x,y)$ and that the absorption coefficient varies continuously along the beam. This is a nonlinear problem in continuous inverse theory. However, it can be linearized, for small net absorption, by approximating the exponential with the first two terms in its Taylor series expansion, that is, $\exp(-x) \approx 1 - x$.

We now convert this problem to a discrete inverse problem of the form $\mathbf{Gm} = \mathbf{d}$. We assume that the continuously varying absorption coefficient can be adequately represented by a grid of many small square boxes (or *pixels*), each of which has a constant absorption coefficient. With these pixels numbered 1 through M, the model parameters are then the vector $\mathbf{m} = [c_1, c_2, c_3, \ldots, c_M]^{\mathrm{T}}$. The integral can then be written as the sum

$$\Delta I_i = \frac{I_0 - I_i}{I_0} = \sum_{j=1}^{M} \Delta s_{ij} c_j \tag{1.17}$$

Here, the data $d_i = \Delta I_i$ represent the differences between the X-ray intensities at the source and at the detector, and $G_{ij} = \Delta s_{ij}$ is the distance the ith beam travels in the jth pixel.

The inverse problem can then be summarized by the matrix equation

$$\begin{bmatrix} \Delta I_1 \\ \Delta I_2 \\ \vdots \\ \Delta I_N \end{bmatrix} = \begin{bmatrix} \Delta s_{11} & \Delta s_{12} & \cdots & \Delta s_{1M} \\ \Delta s_{21} & \Delta s_{22} & \cdots & \Delta s_{2M} \\ \cdots & \cdots & \cdots & \cdots \\ \Delta s_{N1} & \Delta s_{N2} & \cdots & \Delta s_{NM} \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \cdots \\ c_M \end{bmatrix} \tag{1.18}$$

Since each beam passes through only a few of the many boxes, many of the Δs_{ij} are zero. Such a matrix is said to be *sparse*.

Computations with sparse matrices can be made extremely efficient by storing only the nonzero elements and by never explicitly multiplying or adding the zero elements (since the result is a foregone conclusion). However, special software support is necessary to gain this efficiency, since the computer must keep track of the zero elements. In *MatLab*, matrices need to be declared as sparse:

```
G = spalloc( N, M, MAXNONZEROELEMENTS);
```

(*MatLab* script gda01_03)

Once so defined, many normal matrix operations, including addition and multiplication, are efficiently computed without further user intervention. We will discuss this technique further in subsequent chapters, for its use makes practical the solving of very large inverse problems (say, with millions of model parameters). Further examples are given in Menke and Menke (2011).

1.3.5 Example 5: Spectral Curve Fitting

Not every inverse problem can be adequately represented by the discrete linear equation **Gm** = **d**. Consider, for example, a spectrogram containing a set of emission or absorption peaks, that vary with some auxiliary variable z (Figure 1.4). The positions f, area A, and width c of the peaks are of interest because they

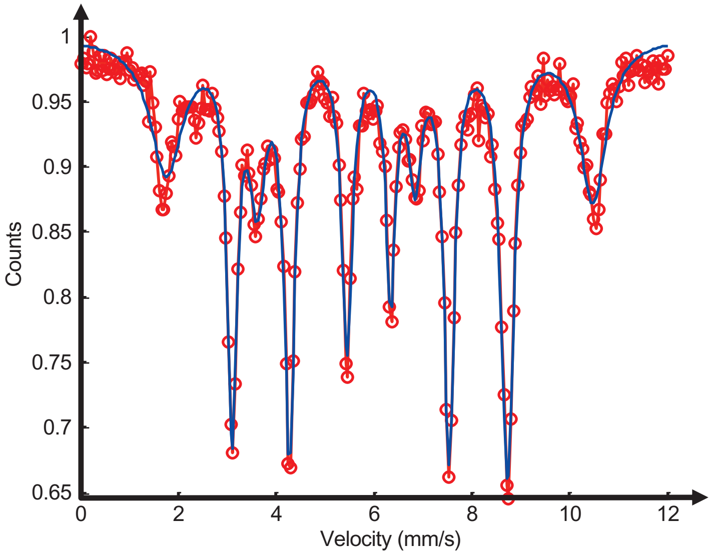

FIGURE 1.4 Example of a Mossbauer spectroscopy experiment performed by the *Spirit* rover on Martian soil. (Red) Absorption peaks reflect the concentration of different iron-bearing minerals in the soil. The inverse problem is to determine the position and area of each peak, which can be used to determine the concentration of the minerals. (Blue) The sum of ten Lorentzian curves fit to the data. Data courtesy of NASA and the University of Mainz. *MatLab* script gda01_04.

reflect the chemical composition of the sample. Denoting the shape of peak j as $p(z,f_j,A_j,c_j)$, the model is that the spectrum consists of a sum of q such peaks

$$d_i = \sum_{j=1}^{q} p(z_i,\, f_j,\, A_j,\, c_j) = \sum_{j=1}^{q} \frac{A_j c_j^2}{(z_i - f_j)^2 + c_j^2} \tag{1.19}$$

Here, the peak shape $p(z_i,f_j,A_j,c_j)$ is taken to be a *Lorentzian*. The data and model are therefore related by the *nonlinear* explicit equation $\mathbf{d}=\mathbf{g(m)}$, where $\mathbf{m}$ is a vector of the position, area, and width of each peak. This equation is inherently nonlinear.

1.3.6 Example 6: Factor Analysis

Another example of a nonlinear inverse problem is that of determining the composition of chemical end members on the basis of the chemistry of a suite of mixtures of the end members. Consider a simplified "ocean" (Figure 1.5) in which sediments are composed of mixtures of several chemically distinct rocks eroded from the continents. One expects the fraction of chemical j in the ith sediment sample S_{ij} to be related to the amount of end-member rock in sediment sample $i(C_{ik})$ and to the amount of the jth chemical in the end-member rock (F_{kj}) as

$$\begin{aligned} \begin{bmatrix} \text{sample} \\ \text{composition} \end{bmatrix} &= \sum_{\text{end members}} \begin{bmatrix} \text{amount of} \\ \text{end member} \end{bmatrix} \begin{bmatrix} \text{end member} \\ \text{composition} \end{bmatrix} \\ S_{ij} &= \sum_{k=1}^{p} C_{ik}\, F_{kj} \quad \text{or} \quad \mathbf{S} = \mathbf{CF} \end{aligned} \tag{1.20}$$

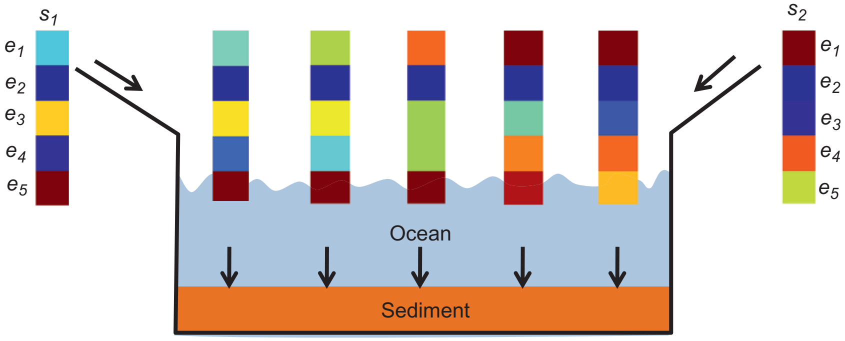

FIGURE 1.5 Sediment on the floor of this idealized ocean is a mixture of rocks eroded from several sources s_i. The sources are characterized by chemical elements, e_1 through e_5, depicted here with color bars. The chemical composition of the sediments is a simple mixture of the composition of the sources. The inverse problem is to determine the number and composition of sources from observations of the composition of the sediments. *MatLab* script gda01_05.

In a typical experiment, the number of end members p, the end-member composition $\mathbf{F}$, and the amount of end members in the samples $\mathbf{C}$ are all unknown model parameters. Since the data $\mathbf{S}$ are on one side of the equations, this problem is also of the explicit nonlinear type. Note that basically the problem is to factor a matrix $\mathbf{S}$ into two other matrices $\mathbf{C}$ and $\mathbf{F}$. This factoring problem is a well-studied part of the theory of matrices, and methods are available to solve it. As will be discussed in Chapter 10, this problem (which is often called *factor analysis*) is very closely related to the algebraic eigenvalue problem.

1.4 SOLUTIONS TO INVERSE PROBLEMS

We shall use the terms *solution* and *answer* to indicate broadly whatever information we are able to determine about the problem under consideration. As we shall see, there are many different points of view regarding what constitutes a solution to an inverse problem. Of course, one generally wants to know the numerical values of the model parameters (we call this kind of answer an *estimate* of the model parameters). Unfortunately, this issue is made complicated by the ubiquitous presence of measurement error and also by the possibility that some model parameters are not constrained by any observation. The solution of inverse problems rarely leads to exact information about the values of the model parameters. More typically, the practitioner of inverse theory is forced to make various compromises between the kind of information he or she actually wants and the kind of information that can in fact be obtained from any given data set. These compromises lead to other kinds of "answers" that are more abstract than simple estimates of the model parameters. Part of the practice of inverse theory is identifying what features of a solution are most valuable and making the compromises that emphasize these features. Some of the possible forms an "answer" to an inverse problem might take are described below.

1.4.1 Estimates of Model Parameters

The simplest kind of solution to an inverse problem is an estimate $\mathbf{m}^{\text{est}}$ of the model parameters. An estimate is simply a set of numerical values for the model parameters, $\mathbf{m}^{\text{est}} = [1.4, 2.9, \ldots, 1.0]^{\text{T}}$, for example. Estimates are generally the most useful kind of solution to an inverse problem. Nevertheless, in many situations, they can be very misleading. For instance, estimates in themselves give no insight into the quality of the solution. Depending on the structure of the particular problem, measurement errors might be averaged out (in which case the estimates might be meaningful) or amplified (in which case the estimates might be nonsense). In other problems, many solutions might exist. To single out arbitrarily only one of these solutions and call it $\mathbf{m}^{\text{est}}$ gives the false impression that a unique solution has been obtained.

1.4.2 Bounding Values

One remedy to the problem of defining the quality of an estimate is to state additionally some bounds that define its certainty. These bounds can be either absolute or probabilistic. Absolute bounds imply that the true value of the model parameter lies between two stated values, for example, $1.3 \leq m_1 \leq 1.5$. Probabilistic bounds imply that the estimate is likely to be between the bounds, with some given degree of certainty. For instance, $m_1^{\text{est}} = 1.4 \pm 0.1$ (95%) might mean that there is a 95% probability that the true value of the model parameter m_1^{true} lies between 1.3 and 1.5.

When they exist, bounding values can often provide the supplementary information needed to interpret properly the solution to an inverse problem. There are, however, many instances in which bounding values do not exist.

1.4.3 Probability Density Functions

A generalization of the stating of bounding values is the stating of the complete probability density function $p(\mathbf{m})$ for model parameters, either as an analytic function or as values on an M-dimensional grid. The usefulness of this technique depends, in part, on the complexity of $p(\mathbf{m})$. If the probability density functions $p(m_i)$ for an individual model parameter m_i has only one peak (Figure 1.6A), then it provides little more information than an estimate based on the position of the peak's center with error bounds based on the peak's shape. On the other hand, if the probability density function is very complicated (Figure 1.6C), it is basically uninterpretable (except in the sense that it implies that the model parameter cannot be well estimated). Only in those exceptional instances in which it has some intermediate complexity (Figure 1.6B) does it really provide information toward the solution of an inverse problem.

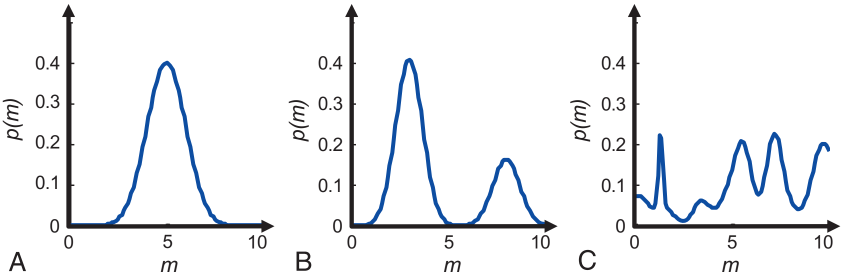

FIGURE 1.6 Three hypothetical probability density functions for a model parameter, m. (A) The first is so simple that its properties can be summarized by its central position, at $m=5$, and the width of its peak. (B) The second implies that the model parameter has two probable ranges of values, one near $m=3$ and the other near $m=8$. (C) The third is so complicated that it provides no easily interpretable information about the model parameter. *MatLab* script gda01_06.

1.4.4 Sets of Realizations of Model Parameters

Except in the well-understood Gaussian (Normal) case, which we will discuss later in the book, most probability density functions are exceedingly difficult to compute. A large set of realizations $\mathbf{m}^{(i)}$ of model parameter vectors drawn from $p(\mathbf{m})$ is somewhat easier to compute and can serve as an alternative. The set, itself, might be considered the solution to the inverse problem, since many of the properties of the probability density function can be inferred from it. However, this set might need to be extremely large to capture the properties of $p(\mathbf{m})$, especially when the number M of model parameters is large. Deriving useful knowledge from, say, a billion examples of possible **m**s is a challenging task.

1.4.5 Weighted Averages of Model Parameters

In many instances, it is possible to identify combinations or averages of the model parameters that are in some sense better determined than the model parameters themselves. For instance, given $\mathbf{m}=[m_1, m_2]^T$, it may turn out that $\langle m \rangle = 0.2m_1 + 0.8m_2$ is better determined than either m_1 or m_2. Unfortunately, one might not have the slightest interest in such an average, be it well determined or not, because it may not have physical significance. It may not contribute useful knowledge.

Averages *can* be of considerable interest when the model parameters represent a discretized version of some continuous function. If the weights are large only for a few physically adjacent parameters, then the average is said to be *localized*. The meaning of the average in such a case is that, although the data cannot resolve the model parameters at a particular point, they can resolve the average of the model parameters in the *neighborhood* of that point.

In the following chapters, we shall derive methods for determining each of these different kinds of solutions to inverse problems. We note here, however, that there is a great deal of underlying similarity between these types of "answers." In fact, it will turn out that the same numerical "answer" will be interpretable as any of several classes of solutions.

1.5 PROBLEMS

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. (A) Identify the data and model parameters in this problem. How many of each are there? (B) Write down the matrix **G** in the equation $\mathbf{d}=\mathbf{Gm}$ that relates the data to the model parameters. (C) How sparse is **G**? What percent of it is zero?

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. (A) Identify the data and model parameters in this problem. How many of each are there? (B) Write down the matrix **G** in the equation $\mathbf{d}=\mathbf{Gm}$ that relates the data to the model parameters. (C) How sparse is **G**? What percent of it is zero?

1.3 Write a *MatLab* script to compute **G** in the case of the cubic equation, $T=a+bz+cz^2+dz^3$. Assume that *11* zs are equally spaced from 0 to 10.

1.4 Let the data **d** be the running average of the model parameters, **m**, computed by averaging groups of three neighboring points; that is, $d_i=(m_{i-1}+m_i+m_{i+1})/3$. (A) What is the matrix **G** in the equation $\mathbf{d}=\mathbf{Gm}$ in this case? (B) What problems arise at the top and bottom rows of the matrix and how can you deal with them? (C) How sparse is **G**? What percent of it is zero?

1.5 Simplify Equation (1.20) by assuming that there is only one sample S_{1j} whose composition is measured. Consider the case where the composition of the p factors is known, but their proportions in the sample are unknown, and rewrite Equation (10.2) in the form $\mathbf{d}=\mathbf{Gm}$. Hint: You might start by taking the transpose of Equation (1.20).

REFERENCES

de Lastours, V., Guillemain, R., Mainardi, J.-L., Aubert, A., Chevalier, P., Lefort, A., Podglajen, I., 2008. Early diagnosis of disseminated Mycobacterium genavense infection. Emerg. Infect. Dis. 14 (2), 346.

Hansen, J., Ruedy, R., Sato, M., Lo, K., 2010. Global surface temperature change. Rev. Geophys. 48, RG4004, doi:10.1029/2010RG000345.

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