Chapter 7

Applications of Vector Spaces

7.1 MODEL AND DATA SPACES

So far, we have used vector notation for the data $\mathbf{d}$ and model parameters $\mathbf{m}$ mainly because it facilitates the algebra. The interpretations of vectors as geometrical quantities can also be used to gain an insight into the properties of inverse problems. We therefore introduce the idea of vector spaces containing $\mathbf{d}$ and $\mathbf{m}$, which we shall denote $S(\mathbf{d})$ and $S(\mathbf{m})$. Any particular choice of $\mathbf{m}$ and $\mathbf{d}$ is then represented as a vector in these spaces (Figure 7.1).

The linear equation $\mathbf{d} = \mathbf{G}\mathbf{m}$ can be interpreted as a mapping of vectors from $S(\mathbf{m})$ to $S(\mathbf{d})$ and its solution $\mathbf{m}^{\text{est}} = \mathbf{G}^{-g}\mathbf{d}$ as a mapping of vectors from $S(\mathbf{d})$ to $S(\mathbf{m})$.

One important property of a vector space, such as $S(\mathbf{m})$, is that its coordinate axes are arbitrary. Thus far we have been using axes parallel to the individual model parameters, but we recognize that we are by no means required to do so. Any set of vectors that spans the space will serve as coordinate axes. The Mth dimensional space $S(\mathbf{m})$ is spanned by any M vectors, say, $\mathbf{m}^{(i)}$, as long as these vectors are linearly independent. An arbitrary vector lying in $S(\mathbf{m})$, say $\mathbf{m}^*$, can be expressed as a sum of these M basis vectors, written as

$$\mathbf{m}^* = \sum_{i=1}^{M} \alpha_i \mathbf{m}^{(i)} \tag{7.1}$$

where the αs are the components of the vector $\mathbf{m}^*$ in the new coordinate system. If the $\mathbf{m}^{(i)}$s are linearly dependent, then the vectors $\mathbf{m}^{(i)}$ lie in a subspace, or *hyperplane*, of $S(\mathbf{m})$ and the expansion cannot be made (Figure 7.2).

We shall consider, therefore, transformations of the coordinate systems of the two spaces $S(\mathbf{m})$ and $S(\mathbf{d})$. Using $\mathbf{S}(\mathbf{m})$ as an example, if $\mathbf{m}$ is the representation of a vector in one coordinate system and $\mathbf{m}'$ its representation in another, we can write the transformation as

$$\mathbf{m}' = \mathbf{T}\mathbf{m} \quad \text{and} \quad \mathbf{m} = \mathbf{T}^{-1}\mathbf{m}' \tag{7.2}$$

where $\mathbf{T}$ is the transformation matrix. If the new basis vectors are still mutually orthogonal unit vectors, then $\mathbf{T}$ represents simple rotations or reflections of the coordinate axes. As we shall show below, however, it is sometimes convenient to choose a new set of basis vectors that are not unit vectors.

Geophysical Data Analysis: Discrete Inverse Theory. DOI: 10.1016/B978-0-12-397160-9.00007-2

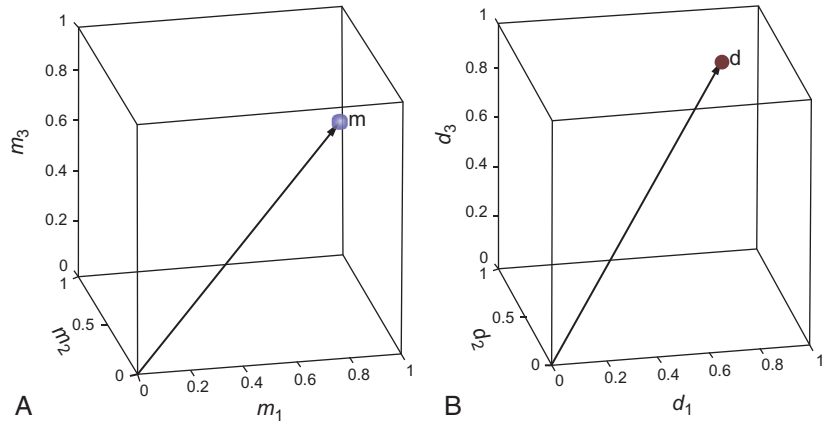

FIGURE 7.1 (A) The model parameters represented as a vector **m** in the M-dimensional space $S(\mathbf{m})$ of all possible model parameters. (B) The data represented as a vector **d** in the N-dimensional space $S(\mathbf{d})$ of all possible data. *MatLab* script gda07_01.

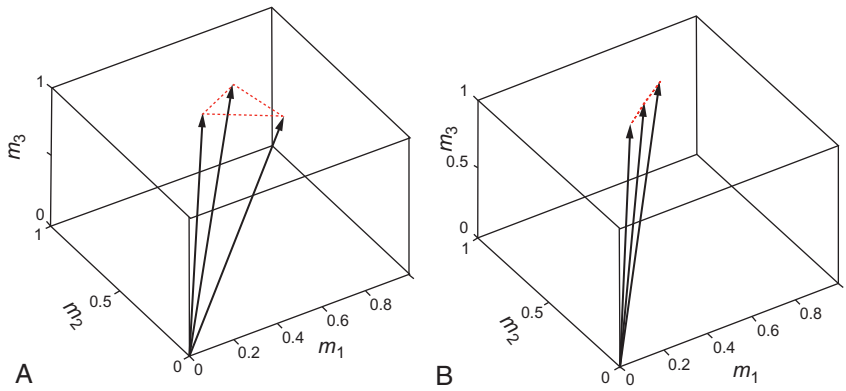

FIGURE 7.2 (A) These three vectors span the three-dimensional space $S(\mathbf{m})$. (B) These three vectors do not span the space as they all lie on the same plane. *MatLab* script gda07_02.

7.2 HOUSEHOLDER TRANSFORMATIONS

In this section, we shall show that the minimum length, least squares, and constrained least squares solutions can be found through simple transformations of the equation $\mathbf{Gm} = \mathbf{d}$. While the results are identical to the formulas derived in Chapter 3, the approach and emphasis are quite different and will enable us to visualize these solutions in a new way. We shall begin by considering a purely underdetermined linear problem $\mathbf{Gm} = \mathbf{d}$ with $M > N$. Suppose we want to find the minimum-length solution (the one that minimizes $L = \mathbf{m}^T\mathbf{m}$). We shall show that it is easy to find this solution by transforming the model parameters into a new coordinate system $\mathbf{m}' = \mathbf{Tm}$. The inverse problem becomes

$$\mathbf{d} = \mathbf{Gm} = \mathbf{GIm} = \{\mathbf{GT}^{-1}\}\{\mathbf{Tm}\} = \mathbf{G}'\mathbf{m}' \tag{7.3}$$

where $\mathbf{G}' = \mathbf{GT}^{-1}$ is the data kernel in the new coordinate system. The solution length becomes

$$L = \mathbf{m}^{\mathrm{T}}\mathbf{m} = \{\mathbf{T}^{-1}\mathbf{m}'\}^{\mathrm{T}}\{\mathbf{T}^{-1}\mathbf{m}'\} = \mathbf{m}'^{\mathrm{T}}\{\mathbf{T}^{-1\mathrm{T}}\mathbf{T}^{-1}\}\mathbf{m}' \tag{7.4}$$

Suppose that we could choose $\mathbf{T}$ so that $\{\mathbf{T}^{-1\mathrm{T}}\mathbf{T}^{-1}\} = \mathbf{I}$. The solution length would then have the same form in both coordinate systems, namely, the sum of squares of the vector elements. Minimizing $\mathbf{m}'^{\mathrm{T}}\mathbf{m}'$ would be equivalent to minimizing $\mathbf{m}^{\mathbf{T}}\mathbf{m}$. Transformations of this type that do not change the length of the vector components are called *unitary transformations*. They may be interpreted as rotations and reflections of the coordinate axes. We can see from Equation (7.4) that unitary transformations satisfy $\mathbf{T}^{\mathbf{T}} = \mathbf{T}^{-1}$.

Now suppose that we could also choose the transformation so that $\mathbf{G}'$ is the lower triangular matrix

$$\begin{bmatrix} G'_{11} & 0 & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ G'_{21} & G'_{22} & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ G'_{31} & G'_{32} & G'_{33} & 0 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \cdots & & & & \\ G'_{N1} & G'_{N2} & G'_{N3} & G'_{N4} & \cdots & G'_{NN} & 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} m'_1 \\ m'_2 \\ m'_3 \\ \vdots \\ m'_M \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ \vdots \\ d_N \end{bmatrix} \tag{7.5}$$

Notice that no matter what values we pick for $\{m_i^{\mathrm{est}}, i = N+1, M\}$, we cannot change the value of $\mathbf{G}'\mathbf{m}'$ as the last $M-N$ columns of $\mathbf{G}'$ are all zero. On the other hand, we can solve for the first N elements of $\mathbf{m}'^{\mathrm{est}}$ uniquely as

$$\begin{aligned} m_1^{\mathrm{est}} &= [d_1]/G'_{11} \\ m_2^{\mathrm{est}} &= \left[d_2 - G'_{21}m_1^{\mathrm{est}}\right]/G'_{22} \\ m_3^{\mathrm{est}} &= \left[d_3 - G'_{31}m_1^{\mathrm{est}} - G'_{32}m_2^{\mathrm{est}}\right]/G'_{33} \\ &\vdots \end{aligned} \tag{7.6}$$

This process is known as *back-solving*. As the first N elements of $\mathbf{m}'^{\mathrm{est}}$ are thereby determined, $\mathbf{m}'^{\mathrm{T}}\mathbf{m}'$ can be minimized by setting the remaining $m_i'^{\mathrm{est}}$ equal to zero. The solution in the original coordinate system is then $\mathbf{m}^{\mathrm{est}} = \mathbf{T}^{-1}\mathbf{m}'^{\mathrm{est}}$. As this solution satisfies the data exactly and minimizes the solution length, it is equal to the minimum-length solution (Section 3.7).

We have employed a transformation process that separates the determined and undetermined linear combinations of model parameters into two distinct groups so that we can deal with them separately. It is interesting to note that we can now easily determine the null vectors for the inverse problem. In the transformed coordinates they are the set of vectors whose first N elements are zero and whose last $M-N$ elements are zero except for one element. There are clearly $M-N$ such vectors, so we have established that there are never more

than M null vectors in a purely underdetermined problem. The null vectors can easily be transformed back into the original coordinate system by premultiplication by $\mathbf{T}^{-1}$. As all but one element of the transformed null vectors are zero, this operation just selects a column of $\mathbf{T}^{-1}$ (or, equivalently, a row of $\mathbf{T}$).

One transformation that can triangularize a matrix is called a *Householder transformation*. We shall discuss in Section 7.3 how it can be constructed. As we shall see below, Householder transformations have application to a wide variety of methods that employ the L_2 norm as a measure of size. These transformations provide an alternative method of solving such problems and additional insight into their structure.

The overdetermined linear inverse problem $\mathbf{Gm}=\mathbf{d}$ with $N>M$ can also be solved through the use of Householder transformations. In this case, we seek a solution that minimizes the prediction error $E=\mathbf{e}^{\mathrm{T}}\mathbf{e}$. We seek a transformation with two properties: it must operate on the transformed prediction error $\mathbf{e}'=\mathbf{Te}$ in such a way that minimizing $\mathbf{e}'^{\mathrm{T}}\mathbf{e}'$ is the same as minimizing $\mathbf{e}^{\mathrm{T}}\mathbf{e}$, and it must transform the data kernel into the upper-triangularized form. The transformed prediction error is

$$\mathbf{e}' = \mathbf{Te} = \mathbf{T}\{\mathbf{d}-\mathbf{Gm}\} = \mathbf{Td}-\mathbf{TGm} = \mathbf{d}' - \mathbf{G}'\mathbf{m} \tag{7.7}$$

where $\mathbf{d}'$ is the transformed data and $\mathbf{G}'$ is the transformed and triangularized data kernel

$$\begin{bmatrix} e'_1 \\ e'_2 \\ e'_3 \\ \vdots \\ e'_M \\ e'_{M+1} \\ \vdots \\ e'_N \end{bmatrix} = - \begin{bmatrix} G'_{11} & G'_{12} & G'_{13} & G'_{14} & \cdots & G'_{1M} \\ 0 & G'_{22} & G'_{23} & G'_{24} & \cdots & G'_{2M} \\ 0 & 0 & G'_{33} & G'_{34} & \cdots & G'_{3M} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & G'_{MM} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ \vdots \\ \vdots \\ \vdots \\ m_M \end{bmatrix} + \begin{bmatrix} d'_1 \\ d'_2 \\ d'_3 \\ \vdots \\ d'_M \\ d'_{M+1} \\ \vdots \\ d'_N \end{bmatrix} \tag{7.8}$$

We note that no matter what values we choose for $\mathbf{m}^{\mathrm{est}}$, we cannot alter the last $N-M$ elements of $\mathbf{e}'$ as the last $N-M$ rows of the transformed data kernel are zero. We can, however, set the first M elements of $\mathbf{e}'$ equal to zero by satisfying the first M equations $\mathbf{e}'=\mathbf{d}'-\mathbf{G}'\mathbf{m}=0$ exactly. As the top part of $\mathbf{G}'$ is triangular, we can use the back-solving technique described above. The total error is then the length of the last $N-M$ elements of $\mathbf{e}'$, written as

$$E = \sum_{i=M+1}^{N} \mathbf{e}_i'^2 \tag{7.9}$$

Again we use the Householder transformations to separate the problem into two parts: data that can be satisfied exactly and data that cannot be satisfied at all. The solution is chosen so that it minimizes the length of the prediction error, and the least square solution is thereby obtained.

Finally, we note that the constrained least squares problem can also be solved with Householder transformations. Suppose that we want to solve $\mathbf{Gm}=\mathbf{d}$ in the least squares sense except that we want the solution to obey p linear equality constraints of the form $\mathbf{Hm}=\mathbf{h}$. Because of the constraints, we do not have complete freedom in choosing the model parameters. We therefore employ Householder transformations to separate those linear combinations of $\mathbf{m}$ that are completely determined by the constraints from those that are completely undetermined. This process is precisely the same as the one used in the underdetermined problem and consists of finding a transformation, say, $\mathbf{T}$, that triangularizes $\mathbf{Hm}=\mathbf{h}$ as

$$\mathbf{h}=\mathbf{Hm}=\{\mathbf{HT}^{-1}\}\{\mathbf{Tm}\}=\mathbf{H}'\mathbf{m}' \tag{7.10}$$

The first p elements of $\mathbf{m}'^{\text{est}}$ are now completely determined and can be computed by back-solving the triangular system. The same transformation can be applied to $\mathbf{Gm}=\mathbf{d}$ to yield the transformed inverse problem $\mathbf{d}=\mathbf{Gm}=\{\mathbf{GT}^{-1}\}\{\mathbf{Tm}\}=\mathbf{G}'\mathbf{m}'$. But $\mathbf{G}'$ will not be triangular as the transformation was designed to triangularize $\mathbf{H}$, not $\mathbf{G}$. As the first p elements of $\mathbf{m}'^{\text{est}}$ have been determined by the constraints, we can partition $\mathbf{G}'$ into two submatrices $\mathbf{G}'=\left[\mathbf{G}'_1,\mathbf{G}'_2\right]$, where $\mathbf{G}'_1$ multiplies the p-determined model parameters and $\mathbf{G}'_2$ multiplies the as yet unknown $M-p$ model parameters

$$[\mathbf{G}'_1,\mathbf{G}'_2]\left[\left[m'^{\text{est}}_1\cdots m'^{\text{est}}_p\right],\left[m'^{\text{est}}_{p+1}\cdots m'^{\text{est}}_M\right]\right]^{\mathrm{T}}=\mathbf{d} \tag{7.11}$$

The equation can be rearranged into standard form by subtracting the part involving the already-determined model parameters:

$$\mathbf{G}'_2\left[m'^{\text{est}}_{p+1}\cdots m'^{\text{est}}_M\right]^{\mathrm{T}}=\mathbf{d}-\mathbf{G}'_1\left[m'^{\text{est}}_1\cdots m'^{\text{est}}_p\right]^{\mathrm{T}} \tag{7.12}$$

The equation is now a completely overdetermined one in the $M-p$ unknown model parameters and can be solved as described above. Finally, the solution is transformed back into the original coordinate system by $\mathbf{m}^{\text{est}}=\mathbf{T}^{-1}\mathbf{m}'^{\text{est}}$.

7.3 DESIGNING HOUSEHOLDER TRANSFORMATIONS

For a transformation to preserve length, it must be a unitary transformation (i.e., it must satisfy $\mathbf{T}^{\mathrm{T}}=\mathbf{T}^{-1}$). Any transformation of the form

$$\mathbf{T}=\mathbf{I}-\frac{2\mathbf{v}\mathbf{v}^{\mathrm{T}}}{\mathbf{v}^{\mathrm{T}}\mathbf{v}} \tag{7.13}$$

(where $\mathbf{v}$ is any vector) is a unitary transformation as

$$\mathbf{T}^{-1}\mathbf{T}=\mathbf{T}^{\mathrm{T}}\mathbf{T}=\left[\mathbf{I}-\frac{2\mathbf{v}\mathbf{v}^{\mathrm{T}}}{\mathbf{v}^{\mathrm{T}}\mathbf{v}}\right]^2=\mathbf{I}-\frac{4\mathbf{v}\mathbf{v}^{\mathrm{T}}}{\mathbf{v}^{\mathrm{T}}\mathbf{v}}+\frac{4\mathbf{v}\mathbf{v}^{\mathrm{T}}}{\mathbf{v}^{\mathrm{T}}\mathbf{v}}=\mathbf{I} \tag{7.14}$$

This can be shown to be the most general form of a unitary transformation. The problem is to find the vector $\mathbf{v}$ such that the transformation triangularizes a given matrix. To do so, we shall begin by finding a sequence of transformations, each of which converts to zeros either the elements beneath the main diagonal of one *column* of the matrix (for premultiplication of the transformation) or the elements to the right of the main diagonal of one *row* of the matrix (for postmultiplication by the transformation). The first i columns are converted to zeros by

$$\mathbf{T} = \mathbf{T}^{(i)}\mathbf{T}^{(i-1)}\mathbf{T}^{(i-2)}\cdots\mathbf{T}^{(1)} \tag{7.15}$$

In the overdetermined problem, applying M of these transformations produces an upper-triangularized matrix. In the $M=3$, $N=4$ case, the transformations proceed as

$$\begin{matrix} \mathbf{G} \\ \begin{bmatrix} x & x & x \\ x & x & x \\ x & x & x \\ x & x & x \end{bmatrix} \end{matrix} \rightarrow \begin{matrix} \mathbf{T}^{(1)}\mathbf{G} \\ \begin{bmatrix} x & x & x \\ 0 & x & x \\ 0 & x & x \\ 0 & x & x \end{bmatrix} \end{matrix} \rightarrow \begin{matrix} \mathbf{T}^{(2)}\mathbf{T}^{(1)}\mathbf{G} \\ \begin{bmatrix} x & x & x \\ 0 & x & x \\ 0 & 0 & x \\ 0 & 0 & x \end{bmatrix} \end{matrix} \rightarrow \begin{matrix} \mathbf{T}^{(3)}\mathbf{T}^{(2)}\mathbf{T}^{(1)}\mathbf{G} \\ \begin{bmatrix} x & x & x \\ 0 & x & x \\ 0 & 0 & x \\ 0 & 0 & 0 \end{bmatrix} \end{matrix} \tag{7.16}$$

where the xs symbolize nonzero matrix elements. Consider the ith column of $\mathbf{G}$, denoted by the vector $\mathbf{g}=[G_{1i}, G_{2i}, \ldots, G_{Ni}]^{\mathrm{T}}$. We want to construct a transformation such that

$$\mathbf{g}' = \mathbf{T}^{(i)}\mathbf{g} = \begin{bmatrix} G'_{1i} & G'_{2i} & \cdots & G'_{ii} & 0 & 0 & \cdots & 0 \end{bmatrix}^{\mathrm{T}} \tag{7.17}$$

Substituting the expression for the transformation yields

$$\mathbf{I}\mathbf{g} - \mathbf{I} - \left(\frac{2\mathbf{v}\mathbf{v}^{\mathrm{T}}}{\mathbf{v}^{\mathrm{T}}\mathbf{v}}\right)\mathbf{g} = \begin{bmatrix} G_{1i} \\ G_{2i} \\ \vdots \\ G_{Ni} \end{bmatrix} - \frac{2\mathbf{v}^{\mathrm{T}}\mathbf{g}}{\mathbf{v}^{\mathrm{T}}\mathbf{v}} \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_N \end{bmatrix} = \begin{bmatrix} G'_{1i} \\ G'_{2i} \\ \vdots \\ G_{ii} \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{7.18}$$

As the term $2\mathbf{v}^{\mathrm{T}}\mathbf{g}/\mathbf{v}^{\mathrm{T}}\mathbf{v}$ is a scalar, we can only zero the last $N-i$ elements of $\mathbf{g}'$ if $[2\mathbf{v}^{\mathrm{T}}\mathbf{g}/\mathbf{v}^{\mathrm{T}}\mathbf{v}][v_{i+1}, \ldots, v_N] = [G_{i+1,i}, \ldots, G_{Ni}]$. We therefore set $[v_{i+1}, \ldots, v_N] = [G_{i+1,i}, \ldots, G_{Ni}]$ and choose the other i elements of $\mathbf{v}$ so that the normalization is correct. As this is but a single constraint on i elements of $\mathbf{v}$, we have considerable flexibility in making the choice. It is convenient to choose the first $i-1$ elements of $\mathbf{v}$ as zero (this choice is both simple and causes the transformation to leave the first $i-1$ elements of $\mathbf{g}$ unchanged). This leaves the ith element of $\mathbf{v}$ to be determined by the constraint. It is easy to show that $v_i = G_{ii} - \alpha_i$, where

$$\alpha_i^2 = \sum_{j=1}^{N} G_{ji}^2 \tag{7.19}$$

(Although the sign of α_i is arbitrary, the choice is usually better than the other in terms of the numerical stability of computer algorithms.) The Householder transformation is then defined by the vector as

$$\mathbf{v} = [\,0 \;\; 0 \;\; \cdots \;\; 0 \;\; (G_{ii} - \alpha_i) \;\; G_{i+1,i} \;\; G_{i+2,i} \;\; \cdots \;\; G_{Ni}\,]^{\mathrm{T}} \tag{7.20}$$

Finally, we note that the $(i+1)$st Householder transformation does not destroy any of the zeros created by the ith, as long as we apply them in order of decreasing number of zeros. We can thus apply a succession of these transformations to triangularize an arbitrary matrix. The inverse transformations are also trivial to construct as they are just the transforms of the forward transformations.

7.4 TRANSFORMATIONS THAT DO NOT PRESERVE LENGTH

Suppose we want to solve the linear inverse problem $\mathbf{Gm} = \mathbf{d}$ in the sense of finding a solution $\mathbf{m}^{\mathrm{est}}$ that minimizes a weighted combination of prediction error and solution simplicity as

$$\text{minimize:} \quad E + L = \mathbf{e}^{\mathrm{T}}\mathbf{W}_{\mathrm{e}}\mathbf{e} + \mathbf{m}^{\mathrm{T}}\mathbf{W}_{\mathrm{m}}\mathbf{m} \tag{7.21}$$

It is possible to find transformations $\mathbf{m}' = \mathbf{T}_{\mathrm{m}}\mathbf{m}$ and $\mathbf{e}' = \mathbf{T}_{\mathrm{e}}\mathbf{e}$ which, although they do not preserve the length, nevertheless change it in precisely such a way that $E + L = \mathbf{e}'^{\mathrm{T}}\mathbf{e}' + \mathbf{m}'^{\mathrm{T}}\mathbf{m}'$ (Wiggins, 1972). The weighting factors are identity matrices in the new coordinate system.

Consider the weighted measure of length $L = \mathbf{m}^{\mathrm{T}}\mathbf{W}_{\mathrm{m}}\mathbf{m}$. If we could factor the weighting matrix into the product $\mathbf{W}_{\mathrm{m}} = \mathbf{T}_{\mathrm{m}}^{\mathrm{T}}\mathbf{T}_{\mathrm{m}}$, where $\mathbf{T}_{\mathrm{m}}$ is some matrix, then we could identify $\mathbf{T}_{\mathrm{m}}$ as a transformation of coordinates

$$L = \mathbf{m}^{\mathrm{T}}\mathbf{W}_{\mathrm{m}}\mathbf{m} = \mathbf{m}^{\mathrm{T}}\left\{\mathbf{T}_{\mathrm{m}}^{\mathrm{T}}\mathbf{T}_{\mathrm{m}}\right\}\mathbf{m} = \left\{\mathbf{T}_{\mathrm{m}}\mathbf{m}\right\}^{\mathrm{T}}\left\{\mathbf{T}_{\mathrm{m}}\mathbf{m}\right\} = \mathbf{m}'^{\mathrm{T}}\mathbf{m}' \tag{7.22}$$

This factorization can be accomplished in several ways. If we have built $\mathbf{W}_{\mathrm{m}}$ up from the $\mathbf{D}$ matrix, then $\mathbf{W}_{\mathrm{m}} = \mathbf{D}^{\mathrm{T}}\mathbf{D}$ and $\mathbf{T}_{\mathrm{m}} = \mathbf{D}$. If not, then we can rely upon the fact that $\mathbf{W}_{\mathrm{m}}$, being symmetric, has a symmetric square root, so that $\mathbf{W}_{\mathrm{m}} = \mathbf{W}_{\mathrm{m}}^{1/2}\mathbf{W}_{\mathrm{m}}^{1/2} = \mathbf{W}_{\mathrm{m}}^{1/2\mathrm{T}}\mathbf{W}_{\mathrm{m}}^{1/2}$, in which case $\mathbf{T} = \mathbf{W}_{\mathrm{m}}^{1/2}$. Then

$$\begin{array}{ccc}
\mathbf{m}' = \mathbf{W}_{\mathrm{m}}^{1/2}\mathbf{m} \text{ or } \mathbf{m}' = \mathbf{D}\mathbf{m} & & \mathbf{m} = \mathbf{W}_{\mathrm{m}}^{-1/2}\mathbf{m}' \text{ or } \mathbf{m} = \mathbf{D}^{-1}\mathbf{m}' \\
\mathbf{d}' = \mathbf{W}_{\mathrm{e}}^{1/2}\mathbf{d} & \text{and} & \mathbf{d} = \mathbf{W}_{\mathrm{e}}^{-1/2}\mathbf{d}' \\
\mathbf{G}' = \mathbf{W}_{\mathrm{e}}^{1/2}\mathbf{G}\mathbf{W}_{\mathrm{m}}^{-1/2} \text{ or } \mathbf{G}' = \mathbf{W}_{\mathrm{e}}^{1/2}\mathbf{G}\mathbf{D}^{-1} & & \mathbf{G} = \mathbf{W}_{\mathrm{e}}^{-1/2}\mathbf{G}'\mathbf{W}_{\mathrm{m}}^{1/2} \text{ or } \mathbf{G} = \mathbf{W}_{\mathrm{e}}^{-1/2}\mathbf{G}'\mathbf{D}
\end{array} \tag{7.23}$$

We have encountered this weighting before, in Equations (3.46) and (5.15). In fact, the damped least squares solution (Equation (3.35))

$$\mathbf{m}'^{\mathrm{est}} = [\mathbf{G}'^{\mathrm{T}}\mathbf{G}' + \varepsilon^2\mathbf{I}]^{-1}\mathbf{G}^{\mathrm{T}}\mathbf{d}' \tag{7.24}$$

transforms into the weighted damped least squares solution (Equation (3.45) with $\langle \mathbf{m} \rangle$ set to zero) under this transformation:

$$\mathbf{W}_m^{1/2}\mathbf{m}^{est} = \left[\mathbf{W}_m^{-1/2}\mathbf{G}^T\mathbf{W}_e^{1/2}\mathbf{W}_e^{1/2}\mathbf{G}\mathbf{W}_m^{-1/2} + \varepsilon^2\mathbf{I}\right]^{-1}\mathbf{W}_m^{-1/2}\mathbf{G}^T\mathbf{W}_e^{1/2}\mathbf{W}_e^{1/2}\mathbf{d}$$

or

$$\mathbf{m}^{est} = [\mathbf{G}^T\mathbf{W}_e\mathbf{G} + \varepsilon^2\mathbf{W}_m]^{-1}\mathbf{G}^T\mathbf{W}_e\mathbf{d} \tag{7.25}$$

The square root of a symmetric matrix $\mathbf{W}$ is defined via its eigenvalue decomposition. Let $\mathbf{\Lambda}$ and $\mathbf{U}$ be the eigenvalues and eigenvectors, respectively, of $\mathbf{W}$ so that $\mathbf{W} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$. Then

$$\begin{aligned}\mathbf{T} = \mathbf{W}^{1/2} &= \mathbf{U}\mathbf{\Lambda}^{1/2}\mathbf{U}^T \quad \text{so that } \mathbf{W}^{1/2T}\mathbf{W}^{1/2} = \mathbf{U}\mathbf{\Lambda}^{1/2}\mathbf{U}^T\mathbf{U}\mathbf{\Lambda}^{1/2}\mathbf{U}^T \\ &= \mathbf{U}\mathbf{\Lambda}^{1/2}\mathbf{\Lambda}^{1/2}\mathbf{U}^T = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T = \mathbf{W}\end{aligned} \tag{7.26}$$

Here we have used the fact that $\mathbf{U}^T\mathbf{U} = \mathbf{I}$. The *MatLab* command `sqrt(Wm)` correctly computes the square root of a square matrix, so the eigenvalue decomposition need not be used explicitly. Yet another possible definition of the transformation $\mathbf{T}$ is

$$\mathbf{T} = \mathbf{\Lambda}^{1/2}\mathbf{U}^T \tag{7.27}$$

as then $\mathbf{T}^T\mathbf{T} = \mathbf{U}\mathbf{\Lambda}^{1/2}\mathbf{\Lambda}^{1/2}\mathbf{U}^T = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T = \mathbf{W}$. The transformed inverse problem is then $\mathbf{G}'\mathbf{m}' = \mathbf{d}'$, where

$$\begin{aligned}\mathbf{m}' &= \left\{\mathbf{\Lambda}_m^{1/2}\mathbf{U}_m^T\right\}\mathbf{m} & \mathbf{m} &= \left\{\mathbf{U}_m\mathbf{\Lambda}_m^{-1/2}\right\}\mathbf{m}' \\ \mathbf{d}' &= \left\{\mathbf{\Lambda}_e^{1/2}\mathbf{U}_e^T\right\}\mathbf{d} \quad \text{and} & \mathbf{d} &= \left\{\mathbf{U}_e\mathbf{\Lambda}_e^{-1/2}\right\}\mathbf{d}' \\ \mathbf{G}' &= \left\{\mathbf{\Lambda}_e^{1/2}\mathbf{U}_e^T\right\}\mathbf{G}\left\{\mathbf{U}_m^T\mathbf{\Lambda}_m^{-1/2}\right\} & \mathbf{G} &= \left\{\mathbf{U}_e\mathbf{\Lambda}_e^{-1/2}\right\}\mathbf{G}'\left\{\mathbf{\Lambda}_m^{1/2}\mathbf{U}_m^T\right\}\end{aligned} \tag{7.28}$$

where $\mathbf{W}_e = \mathbf{U}_e\mathbf{\Lambda}_e\mathbf{U}_e^T$ and $\mathbf{W}_m = \mathbf{U}_m\mathbf{\Lambda}_m\mathbf{U}_m^T$. It is sometimes convenient to transform weighted L_2 problems into one or another of these two forms before proceeding with their solution.

7.5 THE SOLUTION OF THE MIXED-DETERMINED PROBLEM

The concept of vector spaces is particularly helpful in understanding the mixed-determined problem, in which some linear combinations of the model parameters are overdetermined and some are underdetermined.

If the problem is to some degree underdetermined, then the equation $\mathbf{G}\mathbf{m} = \mathbf{d}$ contains information about only some of the model parameters. We can think of these combinations as lying in a subspace $S_p(\mathbf{m})$ of the model parameters space. No information is provided about the part of the solution that lies in the rest of the space, which we shall call the *null space* $S_0(\mathbf{m})$.

The part of the $\mathbf{m}$ that lies in the null space is completely "unilluminated" by $\mathbf{Gm}=\mathbf{d}$, as the equation contains no information about these linear combinations of the model parameters.

On the other hand, if the problem is to some degree overdetermined, the product $\mathbf{Gm}$ may not be able to span $S(\mathbf{d})$ no matter what one chooses for $\mathbf{m}$. At best $\mathbf{Gm}$ may span a subspace $S_p(\mathbf{d})$ of the data space. Then no part of the data lying outside of this space, say, in $S_0(\mathbf{d})$, can be satisfied for any choice of the model parameters.

If the model parameters and data are divided into parts with subscript p that lie in the p spaces and parts with subscript 0 that lie in the null spaces, we can write $\mathbf{Gm}=\mathbf{d}$ as

$$\mathbf{G}[\mathbf{m}_p+\mathbf{m}_0]=[\mathbf{d}_p+\mathbf{d}_0] \tag{7.29}$$

The solution length is then

$$L=\mathbf{m}^{\mathrm{T}}\mathbf{m}=[\mathbf{m}_p+\mathbf{m}_0]^{\mathrm{T}}[\mathbf{m}_p+\mathbf{m}_0]=\mathbf{m}_p^{\mathrm{T}}\mathbf{m}_p+\mathbf{m}_0^{\mathrm{T}}\mathbf{m}_0 \tag{7.30}$$

(The cross terms $\mathbf{m}_p^{\mathrm{T}}\mathbf{m}_0$ and $\mathbf{m}_0^{\mathrm{T}}\mathbf{m}_p$ are zero as the vectors lie in different spaces.) Similarly, the prediction error is

$$E=[\mathbf{d}_p+\mathbf{d}_0-\mathbf{Gm}_p]^{\mathrm{T}}[\mathbf{d}_p+\mathbf{d}_0-\mathbf{Gm}_p]=[\mathbf{d}_p-\mathbf{Gm}_p]^{\mathrm{T}}[\mathbf{d}_p-\mathbf{Gm}_p]+\mathbf{d}_0^{\mathrm{T}}\mathbf{d}_0 \tag{7.31}$$

(as $\mathbf{Gm}_0=0$ and $\mathbf{d}_p$ and $\mathbf{d}_0$ lie in different spaces). We can now define precisely what we mean by a solution to the mixed-determined problem that minimizes prediction error while adding a minimum of *a priori* information: *a priori* information is added to specify only those linear combinations of the model parameters that reside in the null space $S_0(\mathbf{m})$, and the prediction error is reduced to only the portion in the null space $S_0(\mathbf{d})$ by satisfying $\mathbf{e}_p=[\mathbf{d}_p-\mathbf{Gm}_p]=0$ exactly. One possible choice of *a priori* information is $\mathbf{m}_0^{\mathrm{est}}=0$, which is sometimes called the "natural solution" of the mixed-determined problem. We note that when $\mathbf{Gm}=\mathbf{d}$ is purely underdetermined the natural solution is just the minimum-length solution, and when $\mathbf{Gm}=\mathbf{d}$ is purely overdetermined it is just the least squares solution.

One might be tempted to view the natural solution as *better* than, say, the damped least squares solution, as the prior information is applied only to the part of the solution that is in the null space and does not increase the prediction error E. However, such an assessment is not clear-cut. If the prior information is indeed accurate, it should be fully utilized, even if its use leads to a slightly larger prediction error. Anyway, given noisy measurements, two slightly different prediction errors will not be statistically distinguishable. This analysis emphasizes that the choice of the inversion method must be tailored carefully to the problem at hand.

7.6 SINGULAR-VALUE DECOMPOSITION AND THE NATURAL GENERALIZED INVERSE

The p and null subspaces of the linear problem can be easily identified through a type of eigenvalue decomposition of the data kernel that is called the *singular-value decomposition*. We shall derive this decomposition in Section 7.7, but first we shall state the result and demonstrate its usefulness.

Any $N \times M$ square matrix can be written as the product of three matrices (Penrose, 1955; Lanczos, 1961):

$$\mathbf{G} = \mathbf{U}\mathbf{\Lambda}\mathbf{V}^{\mathrm{T}} \tag{7.32}$$

The matrix $\mathbf{U}$ is an $N \times N$ matrix of eigenvectors that span the data space $S(\mathbf{d})$:

$$\mathbf{U} = \begin{bmatrix} \mathbf{u}^{(1)} & \mathbf{u}^{(2)} & \mathbf{u}^{(3)} & \cdots & \mathbf{u}^{N} \end{bmatrix} \tag{7.33}$$

where the $\mathbf{u}^{(i)}$s are the individual vectors. The vectors are orthogonal to one another and can be chosen to be of unit length so that $\mathbf{U}\mathbf{U}^{\mathrm{T}} = \mathbf{U}^{\mathrm{T}}\mathbf{U} = \mathbf{I}$ (where the identity matrix is $N \times N$). Similarly, $\mathbf{V}$ is an $M \times M$ matrix of eigenvectors that span the model parameter space $S(\mathbf{m})$ as

$$\mathbf{V} = \begin{bmatrix} \mathbf{v}^{(1)} & \mathbf{v}^{(2)} & \mathbf{v}^{(3)} & \cdots & \mathbf{v}^{M} \end{bmatrix} \tag{7.34}$$

Here the $\mathbf{v}^{(i)}$s are the individual orthonormal vectors so that $\mathbf{V}\mathbf{V}^{\mathrm{T}} = \mathbf{V}^{\mathrm{T}}\mathbf{V} = \mathbf{I}$ (the identity matrix being $M \times M$). The matrix $\mathbf{\Lambda}$ is an $N \times M$ diagonal eigenvalue matrix whose diagonal elements are nonnegative and are called *singular values*. In the $N = 4$, $M = 3$ case

$$\mathbf{\Lambda} = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \\ 0 & 0 & 0 \end{bmatrix} \tag{7.35}$$

The singular values are usually arranged in order of decreasing size. Some of the singular values may be zero. We therefore partition $\mathbf{\Lambda}$ into a submatrix $\mathbf{\Lambda}_p$ of p nonzero singular values and several zero matrices as

$$\mathbf{\Lambda} = \begin{bmatrix} \mathbf{\Lambda}_p & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{7.36}$$

where $\mathbf{\Lambda}_p$ is a $p \times p$ diagonal matrix. The decomposition then becomes $\mathbf{U}\mathbf{\Lambda}\mathbf{V}^{\mathrm{T}} = \mathbf{U}_p\mathbf{\Lambda}_p\mathbf{V}_p^{\mathrm{T}}$, where $\mathbf{U}_p$ and $\mathbf{V}_p$ consist of the first p columns of $\mathbf{U}$ and $\mathbf{V}$, respectively. The other portions of the eigenvector matrices are canceled by the zeros in $\mathbf{\Lambda}$. The matrix $\mathbf{G}$ contains no information about the subspaces spanned by these portions of the data and model eigenvectors, which we shall call $\mathbf{V}_0$ and $\mathbf{U}_0$, respectively. As we shall soon prove, these are precisely the same spaces as the p and null spaces defined in the previous section.

The data kernel is not a function of the null eigenvectors $\mathbf{V}_0$ and $\mathbf{U}_0$. The equation $\mathbf{d} = \mathbf{G}\mathbf{m} = \mathbf{U}_p\mathbf{\Lambda}_p\mathbf{V}_p^T\mathbf{m}$ contains no information about the part of the model parameters in the space spanned by $\mathbf{V}_0$ as the model parameters $\mathbf{m}$ are multiplied by $\mathbf{V}_p$ (which is orthogonal to everything in $\mathbf{V}_0$). The eigenvector $\mathbf{V}_p$, therefore, lies completely in $S_p(\mathbf{m})$, and $\mathbf{V}_0$ lies completely in $S_0(\mathbf{m})$. Similarly, no matter what value $\left\{\mathbf{\Lambda}_p\mathbf{V}_p^T\mathbf{m}\right\}$ attains, it can have no component in the space spanned by $\mathbf{U}_0$ as it is multiplied by $\mathbf{U}_p$ (and $\mathbf{U}_0$ and $\mathbf{U}_p$ are orthogonal). Therefore, $\mathbf{U}_p$ lies completely in $S_p(\mathbf{d})$ and $\mathbf{U}_0$ lies completely in $S_0(\mathbf{d})$.

We have demonstrated that the p and null spaces can be identified through the singular-value decomposition of the data kernel. The full spaces $S(\mathbf{m})$ and $S(\mathbf{d})$ are spanned by $\mathbf{V}$ and $\mathbf{U}$, respectively. The p spaces are spanned by the parts of the eigenvector matrices that have nonzero eigenvalues: $S_p(\mathbf{m})$ is spanned by $\mathbf{V}_p$ and $S_p(\mathbf{d})$ is spanned by $\mathbf{U}_p$. The remaining eigenvectors $\mathbf{V}_0$ and $\mathbf{U}_0$ span the null spaces $S_0(\mathbf{m})$ and $S_0(\mathbf{d})$. The p and null matrices are orthogonal and are normalized in the sense that $\mathbf{V}_p^T\mathbf{V}_p = \mathbf{U}_p^T\mathbf{U}_p = \mathbf{I}$, where $\mathbf{I}$ is $p \times p$ in size. However, as these matrices do not in general span the complete data and model spaces, $\mathbf{V}_p\mathbf{V}_p^T$ and $\mathbf{U}_p\mathbf{U}_p^T$ are not in general identity matrices.

The natural solution to the inverse problem can be constructed from the singular-value decomposition. This solution must have an $\mathbf{m}^{\text{est}}$ that has no component in $S_0(\mathbf{m})$ and a prediction error $\mathbf{e}$ that has no component in $S_p(\mathbf{d})$. We therefore consider the solution

$$\mathbf{m}^{\text{est}} = \mathbf{V}_p\mathbf{\Lambda}_p^{-1}\mathbf{U}_p^T\mathbf{d} \tag{7.37}$$

which is picked in analogy to the square matrix case. To show that $\mathbf{m}^{\text{est}}$ has no component in $S_0(\mathbf{m})$, we take the dot product of the equation with $\mathbf{V}_0$, which lies completely in $S_0(\mathbf{m})$, as

$$\mathbf{V}_0^T\mathbf{m}^{\text{est}} = \mathbf{V}_0^T\mathbf{V}_p\mathbf{\Lambda}_p^{-1}\mathbf{U}_p^T\mathbf{d} = \mathbf{0} \tag{7.38}$$

as $\mathbf{V}_0^T$ and $\mathbf{V}_p$ are orthogonal. To show that $\mathbf{e}$ has no component in $S_p(\mathbf{d})$, we take the dot product with $\mathbf{U}_p$ as

$$\begin{aligned}\mathbf{U}_p^T\mathbf{e} = \mathbf{U}_p^T[\mathbf{d} - \mathbf{G}\mathbf{m}^{\text{est}}] &= \mathbf{U}_p^T\left[\mathbf{d} - \mathbf{U}_p\mathbf{\Lambda}_p\mathbf{V}_p^T\mathbf{V}_p\mathbf{\Lambda}_p^{-1}\mathbf{U}_p^T\mathbf{d}\right] \\ &= \mathbf{U}_p^T\left[\mathbf{d} - \mathbf{U}_p\mathbf{U}_p^T\mathbf{d}\right] = \mathbf{U}_p^T\mathbf{d} - \mathbf{U}_p^T\mathbf{d} = \mathbf{0}\end{aligned} \tag{7.39}$$

as $\mathbf{V}_p^T\mathbf{V}_p = \mathbf{U}_p^T\mathbf{U}_p = \mathbf{\Lambda}_p\mathbf{\Lambda}_p^{-1} = \mathbf{I}$. The natural solution of the inverse problem is therefore shown to be

$$\mathbf{m}^{\text{est}} = \mathbf{V}_p\mathbf{\Lambda}_p^{-1}\mathbf{U}_p^T\mathbf{d} \tag{7.40}$$

We note that we can define a generalized inverse operator for the mixed-determined problem, the *natural generalized inverse* $\mathbf{G}^{-g} = \mathbf{V}_p\mathbf{\Lambda}_p^{-1}\mathbf{U}_p^T$. (This generalized inverse is so useful that it is sometimes referred to as *the* generalized inverse, although of course there are other generalized inverses that can be designed for the mixed-determined problem that embody other kinds of *a priori* information.)

In *MatLab*, the singular-value decomposition is computed as

```
[U, L, V] = svd(G);
lambda = diag(L);
```

(*MatLab* script gda07_03)

Here we have copied the diagonal elements of matrix `L` of singular values into the vector `lambda`. The submatrices $\mathbf{U}_p$ and $\mathbf{V}_p$ (with `p` an integer) are extracted via:

```
Up = U(:,1:p);
Vp = V(:,1:p);
lambdap = lambda(1:p);
```

(*MatLab* script gda07_03)

and the model parameters are estimated as

```
mest = Vp*((Up'*dobs)./lambdap);
```

(*MatLab* script gda07_03)

Note that the calculation is organized so that the matrices $\mathbf{U}_p^{\mathrm{T}}$ and $\mathbf{V}_p$ multiply vectors (as contrasted to matrices); the generalized inverse is never explicitly formed. The value of the integer `p` must be chosen so that zero eigenvalues are excluded from the estimate; else a division-by-zero error will occur. As discussed below, one often excludes near-zero eigenvalues as well, as the resulting solution (while not quite the natural solution) is often better behaved.

The natural generalized inverse has model resolution

$$\mathbf{R} = \mathbf{G}^{-g}\mathbf{G} = \left\{\mathbf{V}_p\boldsymbol{\Lambda}_p^{-1}\mathbf{U}_p^{\mathrm{T}}\right\}\left\{\mathbf{U}_p\boldsymbol{\Lambda}_p\mathbf{V}_p^{\mathrm{T}}\right\} = \mathbf{V}_p\mathbf{V}_p^{\mathrm{T}} \tag{7.41}$$

The model parameters will be perfectly resolved only if $\mathbf{V}_p$ spans the complete space of model parameters, that is, if there are no zero eigenvalues and $p \geq M$. The data resolution matrix is

$$\mathbf{N} = \mathbf{G}\mathbf{G}^{-g} = \left\{\mathbf{U}_p\boldsymbol{\Lambda}_p\mathbf{V}_p^{\mathrm{T}}\right\}\left\{\mathbf{V}_p\boldsymbol{\Lambda}_p^{-1}\mathbf{U}_p^{\mathrm{T}}\right\} = \mathbf{U}_p\mathbf{U}_p^{\mathrm{T}} \tag{7.42}$$

The data are only perfectly resolved if $\mathbf{U}_p$ spans the complete space of data and $p = N$. Finally, we note that if the data are uncorrelated with uniform variance ${\sigma_{\mathrm{d}}}^2$, the model covariance is

$$\begin{aligned}[\mathrm{cov}\,\mathbf{m}^{\mathrm{est}}] &= \mathbf{G}^{-g}[\mathrm{cov}\,\mathbf{d}]\mathbf{G}^{-g\mathrm{T}} = \sigma_{\mathrm{d}}^2\left\{\mathbf{V}_p\boldsymbol{\Lambda}_p^{-1}\mathbf{U}_p^{\mathrm{T}}\right\}\left\{\mathbf{V}_p\boldsymbol{\Lambda}_p^{-1}\mathbf{U}_p^{\mathrm{T}}\right\}^{\mathrm{T}} \\ &= {\sigma_{\mathrm{d}}}^2\mathbf{V}_p\boldsymbol{\Lambda}_p^{-2}\mathbf{V}_p^{\mathrm{T}}\end{aligned} \tag{7.43}$$

The covariance of the estimated model parameters is very sensitive to the smallest nonzero eigenvalue. (Note that forming the natural inverse corresponds to assuming that linear combinations of the *a priori* model parameters in the p space have infinite variance and that combinations in the null space have zero

variance and zero mean.) The covariance of the estimated model parameters, therefore, does not explicitly contain [cov **m**]. If one prefers a solution based on the natural inverse (but with the null vectors chosen to minimize the distance to a set of *a priori* model parameters with mean $\langle \mathbf{m} \rangle$ and covariance $[\text{cov } \mathbf{m}]_A$), it is appropriate to use the formula $\mathbf{m}^{\text{est}} = \mathbf{G}^{-g}\mathbf{d} + [\mathbf{I} - \mathbf{R}]\langle \mathbf{m} \rangle$, where $\mathbf{G}^{-g}$ is the natural inverse. The covariance of this estimate is now

$$[\text{cov } \mathbf{m}^{\text{est}}] = \mathbf{G}^{-g}[\text{cov } \mathbf{d}]\mathbf{G}^{-g\text{T}} + [\mathbf{I} - \mathbf{R}][\text{cov } \mathbf{m}]_A[\mathbf{I} - \mathbf{R}]^{\text{T}} \tag{7.44}$$

which is based on the usual rule for computing covariances.

To use the natural inverse one must be able to identify the number p, that is, to count the number of nonzero singular values. Plots of the sizes of the singular values against their index numbers (the *spectrum* of the data kernel) can be useful in this process. The value of p can be easily determined if the singular values fall into two clearly distinguishable groups, one nonzero and one zero (Figure 7.3A and B). In realistic inverse problems, however, the singular values often smoothly decline in size (Figure 7.3C and D), making it difficult to distinguish ones that are actually nonzero from ones that are zero but computed somewhat inaccurately owing to round-off error by the computer. Furthermore, if one chooses p so as to include these very small singular values, the solution variance will be very large, as it contains the factor $\mathbf{\Lambda}_p^{-2}$. One solution

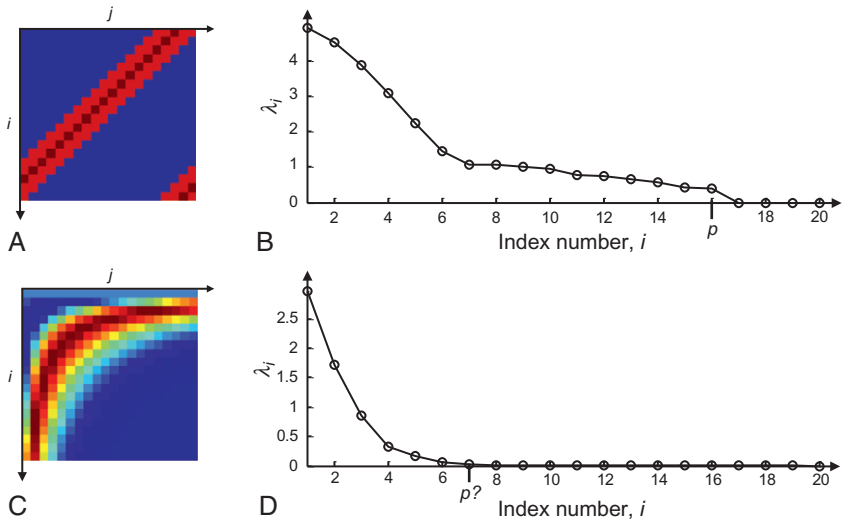

FIGURE 7.3 (A) Hypothetical data kernel G_{ij} for an inverse problem with $M = 20$ model parameters and $N = 20$ observations. (B) Corresponding singular values λ_i have a clear cutoff at $p = 16$. (C) Another hypothetical data kernel, also with $N = 20$ and $M = 20$. (D) Corresponding singular values do not have a clear cutoff, so the parameter, p, must be chosen in a more arbitrary fashion near $p \approx 7$. *MatLab* Script gda07_04.

to this problem is to pick some cutoff size for singular values and then consider any values smaller than this as equal to zero. This process artificially reduces the dimensions of $\mathbf{V}_p$ and $\mathbf{U}_p$ that are included in the generalized inverse. The resulting estimates of the model parameters are no longer exactly the natural solution. But, if only small singular values are excluded, the solution is generally close to the natural solution and possesses better variance. On the other hand, its model and data resolution are worse. We recognize that this trade-off is just another manifestation of the trade-off between resolution and variance discussed in Chapter 4.

As an example, we consider the problem $\mathbf{Gm} = \mathbf{d}$ illustrated in Figure 7.4A, which has the same data kernel as in Figure 7.3A. Although $\mathbf{G}$ is 20×20, the problem is mixed-determined, as $p = 16$ and four of the singular values are zero. In this case, the natural solution is very close to the true solution, presumably because, by coincidence, the true solution did not have much of its power in the null space. The natural solution is also close to the damped minimum-length solution (Equation (4.22)), which is not surprising, for both arise from similar minimization principles. The damped minimum-length solution minimizes a weighted sum of prediction error and solution length; the natural solution minimizes the prediction error and the component of the solution length in the null space.

Instead of choosing a sharp cutoff for the singular values, it is possible to include all the singular values while damping the smaller ones. We let $p = M$, but replace the reciprocals of all the singular values by $\lambda_i/(\varepsilon^2 + \lambda_i^2)$, where ε is some small number. This change has little effect on the larger singular values but prevents the smaller ones from leading to large variances. Of course, the solution is no longer the natural solution. While its variance is improved, its

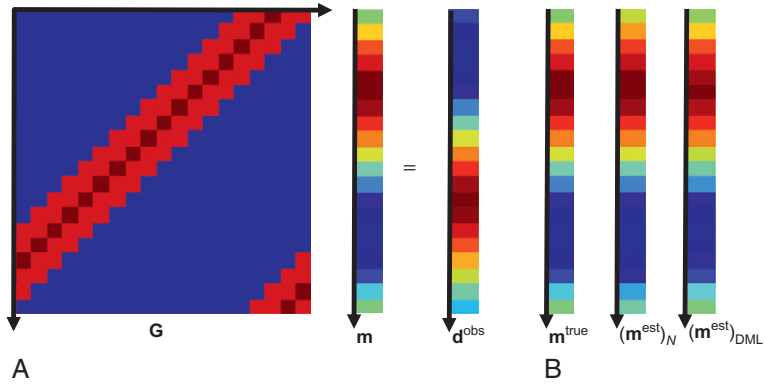

FIGURE 7.4 (A) Same linear problem as in Figure 7.3A, where $\mathbf{d}^{\text{obs}} = \mathbf{Gm}^{\text{true}} + \mathbf{n}$, with $\mathbf{n}$ uncorrelated Gaussian noise. (B) Corresponding solutions, true solutions $\mathbf{m}^{\text{true}}$, natural solution $(\mathbf{m}^{\text{est}})_N$, and damped minimum-length solution $(\mathbf{m}^{\text{est}})_{\text{DML}}$. *MatLab* Script gda07_03.

model and data resolution are degraded. In fact, this solution is precisely the damped least squares solution discussed in Chapter 3:

$$\begin{aligned}[\mathbf{G}^{\mathrm{T}}\mathbf{G}+\varepsilon^2\mathbf{I}]^{-1}\mathbf{G}^{\mathrm{T}} &= [\mathbf{V}\boldsymbol{\Lambda}\mathbf{U}^{\mathrm{T}}\mathbf{U}\boldsymbol{\Lambda}\mathbf{V}^{\mathrm{T}}+\varepsilon^2\mathbf{I}]^{-1}\mathbf{V}\boldsymbol{\Lambda}\mathbf{U}^{\mathrm{T}}\\ &= [\mathbf{V}\boldsymbol{\Lambda}^2\mathbf{V}^{\mathrm{T}}+\varepsilon^2\mathbf{V}\mathbf{I}\mathbf{V}^{\mathrm{T}}]^{-1}\mathbf{V}\boldsymbol{\Lambda}\mathbf{U}^{\mathrm{T}}\\ &= \mathbf{V}[\boldsymbol{\Lambda}^2+\varepsilon^2\mathbf{I}]^{-1}\mathbf{V}^{\mathrm{T}}\mathbf{V}\boldsymbol{\Lambda}\mathbf{U}^{\mathrm{T}}\\ &= \mathbf{V}\{[\boldsymbol{\Lambda}^2+\varepsilon^2\mathbf{I}]^{-1}\boldsymbol{\Lambda}\}\mathbf{U}^{\mathrm{T}}\end{aligned} \tag{7.45}$$

Here we rely upon the fact that if $\mathbf{M}=\mathbf{V}\mathbf{D}\mathbf{V}^{\mathrm{T}}$ is the eigenvalue decomposition of $\mathbf{M}$, then $\mathbf{M}^{-1}=\mathbf{V}\mathbf{D}^{-1}\mathbf{V}^{\mathrm{T}}$.

The damping of the singular values corresponds to the addition of *a priori* information that the model parameters are small. The precise value of the number used as the cutoff or damping parameter must be chosen by a trial-and-error process which weighs the relative merits of having a solution with small variance against those of having one that fits the data and is well resolved.

In Section 6.6, we discussed the problem of bounding nonunique averages of model parameters by incorporating *a priori* inequality constraints into the solution of the inverse problem. We see that the singular-value decomposition provides a simple way of identifying the null vectors of $\mathbf{G}\mathbf{m}=\mathbf{d}$. The general solution to the inverse problem (Wunsch and Minster, 1982)

$$\mathbf{m}^{\mathrm{gen}} = \mathbf{m}^{\mathrm{par}} + \sum_{i=1}^{p}\alpha_i\mathbf{m}^{\mathrm{null}(i)} \tag{7.46}$$

can be thought of as having the natural solution as its particular solution and a sum over the null eigenvectors as its null solution:

$$\mathbf{m}^{\mathrm{gen}} = \mathbf{V}_p\boldsymbol{\Lambda}_p^{-1}\mathbf{U}_p^{\mathrm{T}}\mathbf{d} + \mathbf{V}_0\boldsymbol{\alpha} \tag{7.47}$$

There are $q=M-p$ null vectors in the general solution, with the coefficients given by $\boldsymbol{\alpha}$. In Section 6.6, upper and lower bounds on localized averages of the solution were found by determining the α that maximizes (for the upper bound) or minimizes (for the lower bound) the localized average $\langle m\rangle = \mathbf{a}^{\mathrm{T}}\mathbf{m}$ with the constraint that $\mathbf{m}^{\mathrm{l}}\le\mathbf{m}\le\mathbf{m}^{\mathrm{u}}$, where $\mathbf{m}^{\mathrm{l}}$ and $\mathbf{m}^{u}$ are *a priori* bounds. The use of the natural solution guarantees that the prediction error is minimized in the L_2 sense.

The bounds on the localized average $\langle m\rangle = \mathbf{a}^{\mathrm{T}}\mathbf{m}$ should be treated with some skepticism, as they depend on a particular choice of the solution (in this case one that minimizes the L_2 prediction error). If the total error E increases only slightly as this solution is perturbed and if this additional error can be considered negligible, then the true bounds of the localized average will be larger than those given above. In principle, one can handle this problem by forsaking the eigenvalue decomposition and simply determining the $\mathbf{m}$ that extremizes $\langle m\rangle$ with the constraints that $\mathbf{m}^{\mathrm{l}}\le\mathbf{m}\le\mathbf{m}^{u}$ and that the total prediction error is less than some tolerable amount, say, E_M. For L_2 measures of the

prediction error, this is a very difficult nonlinear problem. However, if the error is measured under the L_1 norm, it can be transformed into a linear programming problem.

7.7 DERIVATION OF THE SINGULAR-VALUE DECOMPOSITION

The singular-value decomposition can be derived in many ways. We follow here the treatment of Lanczos (1961). For an alternate derivation, see Menke and Menke (2011, Section 8.2). We first form an $(N+M)\times(N+M)$ square symmetric matrix $\mathbf{S}$ from $\mathbf{G}$ and $\mathbf{G}^{\mathrm{T}}$ as

$$\mathbf{S} = \begin{bmatrix} 0 & \mathbf{G} \\ \mathbf{G}^{\mathrm{T}} & 0 \end{bmatrix} \tag{7.48}$$

From elementary linear algebra, we know that this matrix has $N+M$ real eigenvalues λ_i and a complete set of eigenvectors $\mathbf{w}^{(i)}$ which solve $\mathbf{S}\mathbf{w}^{(i)} = \lambda_i \mathbf{w}^{(i)}$. Partitioning $\mathbf{w}$ into a part $\mathbf{u}$ of length N and a part $\mathbf{v}$ of length M, we obtain

$$\mathbf{S}\mathbf{w}^{(i)} = \lambda_i \mathbf{w}^{(i)} \rightarrow \begin{bmatrix} 0 & \mathbf{G} \\ \mathbf{G}^{\mathrm{T}} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u}^{(i)} \\ \mathbf{v}^{(i)} \end{bmatrix} = \lambda_i \begin{bmatrix} \mathbf{u}^{(i)} \\ \mathbf{v}^{(i)} \end{bmatrix} \tag{7.49}$$

We shall now show that $\mathbf{u}^{(i)}$ and $\mathbf{v}^{(i)}$ are the same vectors as those defined in the previous section. We first note that the above equation implies that $\mathbf{G}\mathbf{v}^{(i)} = \lambda_i \mathbf{u}^{(i)}$ and $\mathbf{G}^{\mathrm{T}}\mathbf{u}^{(i)} = \lambda_i \mathbf{v}^{(i)}$. Suppose that there is a positive eigenvalue λ_i with eigenvector $[\mathbf{u}^{(i)}, \mathbf{v}^{(i)}]^{\mathrm{T}}$. Then we note that $-\lambda_i$ is also an eigenvalue with eigenvector $[-\mathbf{u}^{(i)}, \mathbf{v}^{(i)}]^{\mathrm{T}}$. If there are p positive eigenvalues, then there are $N+M-2p$ zero eigenvalues. Now by manipulating the above equations we obtain

$$\mathbf{G}^{\mathrm{T}}\mathbf{G}\mathbf{v}^{(i)} = \lambda_i^2 \mathbf{v}^{(i)} \quad \text{and} \quad \mathbf{G}\mathbf{G}^{\mathrm{T}}\mathbf{u}^{(i)} = \lambda_i^2 \mathbf{u}^{(i)} \tag{7.50}$$

As a symmetric matrix can have no more distinct eigenvectors than its dimension, we note that $p \leq \min(N, M)$. As both matrices are square and symmetric, there are M vectors $\mathbf{v}^{(i)}$ that form a complete orthogonal set $\mathbf{V}$ spanning $S(\mathbf{m})$ and N vectors $\mathbf{u}^{(i)}$ that form a complete orthogonal set $\mathbf{U}$ spanning $S(\mathbf{d})$. These include p of the $\mathbf{w}$ eigenvectors with distinct nonzero eigenvalues and remaining ones chosen from the eigenvectors with zero eigenvalues. The equation $\mathbf{G}\mathbf{v}^{(i)} = \lambda_i \mathbf{u}^{(i)}$ can be written in matrix form as $\mathbf{G}\mathbf{V} = \mathbf{U}\mathbf{\Lambda}$, where $\mathbf{\Lambda}$ is a diagonal matrix of the eigenvalues. Postmultiplying by $\mathbf{V}^{\mathrm{T}}$ gives the singular-value decomposition $\mathbf{G} = \mathbf{U}\mathbf{\Lambda}\mathbf{V}^{\mathrm{T}}$.

7.8 SIMPLIFYING LINEAR EQUALITY AND INEQUALITY CONSTRAINTS

The singular-value decomposition can be used to simplify linear constraints.

7.8.1 Linear Equality Constraints

Consider the problem of solving $\mathbf{Gm}=\mathbf{d}$ in the sense of finding a solution that minimizes the L_2 prediction error subject to the $K<M$ constraints that $\mathbf{Hm}=\mathbf{h}$. This problem can be reduced to the unconstrained problem $\mathbf{G}'\mathbf{m}'=\mathbf{d}'$ in $M'\leq M$ new model parameters. We first find the singular-value decomposition of the constraint matrix $\mathbf{H}=\mathbf{U}_p\mathbf{\Lambda}_p\mathbf{V}_p^{\mathrm{T}}$. If $p=K$, the constraints are consistent and determine p linear combinations of the unknowns. The general solution is then $\mathbf{m}=\mathbf{V}_p\mathbf{\Lambda}_p^{-1}\mathbf{U}_p^{\mathrm{T}}\mathbf{h}+\mathbf{V}_0\boldsymbol{\alpha}$, where $\boldsymbol{\alpha}$ is an arbitrary vector of length $M-p$ and is to be determined by minimizing the prediction error. Substituting this equation for $\mathbf{m}$ into $\mathbf{Gm}=\mathbf{d}$ and rearranging terms yields

$$\mathbf{GV}_0\boldsymbol{\alpha}=\mathbf{d}-\mathbf{GV}_p\mathbf{\Lambda}_p^{-1}\mathbf{U}_p^{\mathrm{T}}\mathbf{h} \tag{7.51}$$

This equation in the unknown coefficients $\boldsymbol{\alpha}$ now can be solved as an unconstrained least squares problem. We note that we have encountered this problem in a somewhat different form during the discussion of Householder transformations (Section 7.2). The main advantage of using the singular-value decomposition is that it provides a test of the constraint's consistency.

7.8.2 Linear Inequality Constraints

Consider the L_2 problem

$$\text{minimize} \quad \|\mathbf{d}-\mathbf{Gm}\|_2 \quad \text{subject to} \quad \mathbf{Hm}\geq\mathbf{h} \tag{7.52}$$

We shall show that as long as $\mathbf{Gm}=\mathbf{d}$ is in fact overdetermined, this problem can be reduced to the simpler problem (Lawson and Hanson, 1974):

$$\text{minimize} \quad \|\mathbf{m}'\|_2 \quad \text{subject to} \quad \mathbf{H}'\mathbf{m}'\geq\mathbf{h}' \tag{7.53}$$

To demonstrate this transformation, we form the singular-value decomposition of the data kernel $\mathbf{G}=\mathbf{U}_p\mathbf{\Lambda}_p\mathbf{V}_p^{\mathrm{T}}$. We first divide the data $\mathbf{d}$ into two parts $\mathbf{d}=\left[\mathbf{d}_p^{\mathrm{T}},\mathbf{d}_0^{\mathrm{T}}\right]^{\mathrm{T}}$ where $\mathbf{d}_p=\mathbf{U}_p^{\mathrm{T}}\mathbf{d}$ is in the $S_p(\mathbf{m})$ subspace and $\mathbf{d}_0=\mathbf{U}_0^{\mathrm{T}}\mathbf{d}$ is in the $S_0(\mathbf{m})$ subspace. The prediction error is then $E=\left\|\mathbf{d}^{\mathrm{obs}}-\mathbf{d}^{\mathrm{pre}}\right\|_2$ or

$$\begin{aligned}E&=\left\|\begin{bmatrix}\mathbf{U}_p^{\mathrm{T}}\mathbf{d}\\ \mathbf{U}_0^{\mathrm{T}}\mathbf{d}\end{bmatrix}-\begin{bmatrix}\mathbf{U}_p^{\mathrm{T}}\mathbf{U}_p\mathbf{\Lambda}_p\mathbf{V}_p^{\mathrm{T}}\mathbf{m}\\ 0\end{bmatrix}\right\|_2=\left\|\mathbf{d}_p-\mathbf{\Lambda}_p\mathbf{V}_p^{\mathrm{T}}\mathbf{m}\right\|_2+\|\mathbf{d}_0\|_2\\&=\|\mathbf{m}'\|_2+\|\mathbf{d}_0\|_2\end{aligned} \tag{7.54}$$

where $\mathbf{m}'=\mathbf{d}_p-\mathbf{\Lambda}_p\mathbf{V}_p^{\mathrm{T}}\mathbf{m}$. We note that minimizing $\|\mathbf{m}'\|_2$ is the same as minimizing E as the other term is a constant. Inverting this expression for the unprimed model parameters gives $\mathbf{m}=\mathbf{V}_p\mathbf{\Lambda}_p^{-1}\left[\mathbf{d}_p-\mathbf{m}'\right]=\mathbf{V}_p\mathbf{\Lambda}_p^{-1}\left[\mathbf{U}_p^{\mathrm{T}}\mathbf{d}-\mathbf{m}'\right]$. Substituting this expression into the constraint equation $\mathbf{Hm}\geq\mathbf{h}$ and rearranging terms yields

$$\left\{-\mathbf{H}\mathbf{V}_p\boldsymbol{\Lambda}_p^{-1}\right\}\mathbf{m}' \geq \left\{\mathbf{h} - \mathbf{H}\mathbf{V}_p\boldsymbol{\Lambda}_p^{-1}\mathbf{U}_p^{\mathrm{T}}\mathbf{d}\right\} \quad \text{or} \quad \mathbf{H}'\mathbf{m}' \geq \mathbf{h}' \tag{7.55}$$

which is in the desired form.

7.9 INEQUALITY CONSTRAINTS

We shall now consider the solution of L_2 minimization problems with inequality constraints of the form

$$\text{minimize} \quad \|\mathbf{d} - \mathbf{G}\mathbf{m}\|_2 \quad \text{subject to} \quad \mathbf{H}\mathbf{m} \geq \mathbf{h} \tag{7.56}$$

We first note that problems involving $=$ and $\leq$ constraints can be reduced to this form. Equality constraints can be removed by the transformation described in Section 7.8.1, and $\leq$ constraints can be removed by multiplication by -1 to change them into $\geq$ constraints. For this minimization problem to have any solution, the constraints $\mathbf{H}\mathbf{m} \geq \mathbf{h}$ must be consistent; there must be at least one $\mathbf{m}$ that satisfies all the constraints. We can view these constraints as defining a volume in $S(\mathbf{m})$. Each constraint defines a hyperplane that divides $S(\mathbf{m})$ into two half-spaces, one in which that constraint is satisfied (the *feasible half-space*) and the other in which it is violated (the *infeasible half-space*). The set of inequality constraints, therefore, defines a volume in $S(\mathbf{m})$ which might be zero, finite, or infinite in extent. If the region has zero volume, then no feasible solution exists (Figure 7.5B). If it has nonzero volume, then there is at least one solution that minimizes the prediction error (Figure 7.5A). This volume has the shape of a polyhedron as its boundary surfaces are planes. It can be shown that the polyhedron must be convex: it can have no reentrant angles or grooves.

The starting point for solving the L_2 minimization problem with inequality constraints is the Kuhn-Tucker theorem, which describes the properties that any solution to this problem must possess. For any $\mathbf{m}$ that minimizes

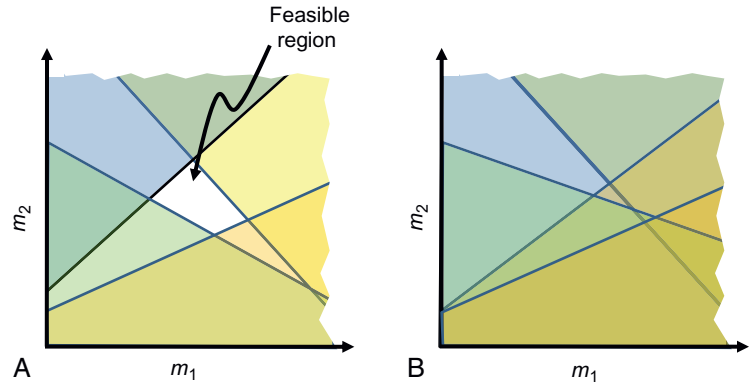

FIGURE 7.5 (A) Each linear inequality constraint divides $S(\mathbf{m})$ into two half-spaces, one infeasible (shaded) and the other feasible (white). Consistent constraints combine to form a convex, polygonal region within $S(\mathbf{m})$ (white) of feasible $\mathbf{m}$. (B) Inconsistent constraints do not form a feasible region.

$\|\mathbf{d} - \mathbf{Gm}\|_2$ subject to p constraints $\mathbf{Hm} \geq \mathbf{h}$, it is possible to find a vector $\mathbf{y}$ of length p such that

$$\frac{1}{2}\nabla E = \nabla[\mathbf{Hm}]\cdot\mathbf{y} \quad \text{or} \quad -\mathbf{G}^{\mathrm{T}}[\mathbf{d} - \mathbf{Gm}] = \mathbf{H}^{\mathrm{T}}\mathbf{y} \tag{7.57}$$

This equation states that the gradient of the error ∇E can be written as a linear combination of hyperplane normals $\mathbf{H}^{\mathrm{T}}$, with the combination specified by the vector $\mathbf{y}$. The Kuhn-Tucker theorem goes on to state that the elements of $\mathbf{y}$ are nonnegative; that $\mathbf{y}$ can be partitioned into two parts $\mathbf{y}_{\mathrm{E}}$ and $\mathbf{y}_{\mathrm{S}}$ (possibly requiring reordering of the constraints) that satisfy

$$\mathbf{y} = \begin{bmatrix} \mathbf{y}_{\mathrm{E}} > 0 \\ \mathbf{y}_{\mathrm{S}} = 0 \end{bmatrix} \quad \text{and} \quad \begin{matrix} \mathbf{H}_{\mathrm{E}}\mathbf{m} - \mathbf{h}_{\mathbf{E}} = 0 \\ \mathbf{H}_{\mathrm{S}}\mathbf{m} - \mathbf{h}_{\mathbf{S}} > 0 \end{matrix} \tag{7.58}$$

The first group of constraints is satisfied in the equality sense (thus the subscript E for "equality"). The rest are satisfied more loosely in the inequality sense (thus the subscript S for "slack").

The theorem indicates that any feasible solution $\mathbf{m}$ is the minimum solution only if the direction in which one would have to perturb $\mathbf{m}$ to decrease the total error E causes the solution to cross some constraint hyperplane and become infeasible. The direction of decreasing error is $-1/2\nabla\mathbf{E} = \mathbf{G}^{\mathrm{T}}[\mathbf{d} - \mathbf{Gm}]$. The constraint hyperplanes have normals $+\nabla[\mathbf{Hm}] = \mathbf{H}^{\mathrm{T}}$ which point into the feasible side. As $\mathbf{H}_{\mathrm{E}}\mathbf{m} - \mathbf{h}_{\mathrm{E}} = 0$, the solution lies exactly on the bounding hyperplanes of the $\mathbf{H}_{\mathrm{E}}$ constraints. As $\mathbf{H}_{\mathrm{S}}\mathbf{m} - \mathbf{h}_{\mathrm{S}} > 0$, it lies within the feasible volume of the $\mathbf{H}_{\mathrm{S}}$ constraints. An infinitesimal perturbation $\delta\mathbf{m}$ of the solution can, therefore, only violate the $\mathbf{H}_{\mathrm{E}}$ constraints. If it is not to violate these constraints, the perturbation must be made in the direction of feasibility so that it must be expressible as a nonnegative combination of hyperplane normals, that is, $\delta\mathbf{m}\cdot\nabla[\mathbf{Hm}] \geq 0$. On the other hand, if it is to decrease the total prediction error it must satisfy $\delta\mathbf{m}\cdot\nabla E \leq 0$. These two conditions are incompatible with the Kuhn-Tucker theorem, as dotting Equation (7.57) with $\delta\mathbf{m}$ yields $\delta\mathbf{m}\cdot 1/2\nabla E = \delta\mathbf{m}\cdot\nabla[\mathbf{Hm}]\cdot\mathbf{y} \geq 0$ as both $\delta\mathbf{m}\cdot\nabla[\mathbf{Hm}]$ and $\mathbf{y}$ are positive. These solutions are indeed minimum solutions to the constrained problem (Figure 7.6).

To demonstrate how the Kuhn-Tucker theorem can be used, we consider the simplified problem

$$\text{minimize} \quad E = \|\mathbf{d} - \mathbf{Gm}\|_2 \quad \text{subject to} \quad \mathbf{m} \geq \mathbf{0} \tag{7.59}$$

which is called *nonnegative least squares*. We find the solution using an iterative scheme of several steps (Lawson and Hanson, 1974):

Step 1. Start with an initial guess for $\mathbf{m}$. As $\mathbf{H} = \mathbf{I}$ each model parameter is associated with exactly one constraint. These model parameters can be separated into a set $\mathbf{m}_{\mathrm{E}}$ that satisfies the constraints in the equality sense and a set $\mathbf{m}_{\mathrm{S}}$ that satisfies the constraints in the inequality sense. The particular initial guess $\mathbf{m} = 0$ is clearly feasible and has all its elements in $\mathbf{m}_{\mathrm{E}}$.

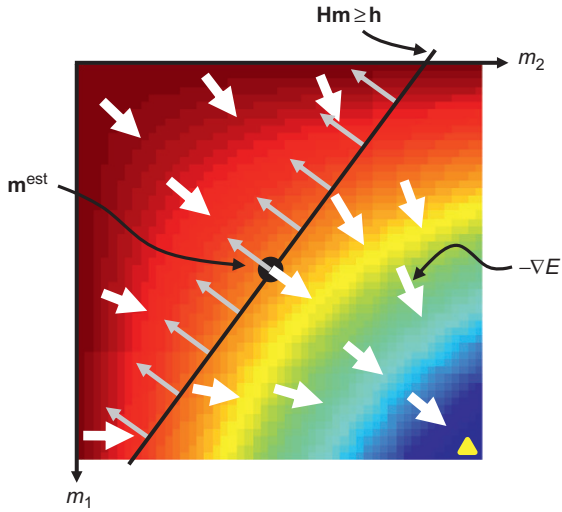

FIGURE 7.6 The error $E(\mathbf{m})$ (colors) has a single minimum (yellow triangle). The linear equality constraint, $\mathbf{Hm} \geq \mathbf{h}$, divides $S(\mathbf{m})$ into two half-spaces (black line, with gray arrows pointing into the feasible half-space). Solution (circle) lies on the boundary between the two half-spaces and therefore satisfies the constraint in the equality sense. At this point, the normal of the constraint hyperplane (gray arrow) is antiparallel to $-\nabla E$ (white arrows). *MatLab* script gda07_05.

Step 2. Any model parameter m_i in $\mathbf{m}_E$ that has associated with it a negative gradient $[\nabla E]_i$ can be changed both to decrease the error and to remain feasible. Therefore, if there is no such model parameter in $\mathbf{m}_E$, the Kuhn-Tucker theorem indicates that this $\mathbf{m}$ is the solution to the problem.

Step 3. If some model parameter m_i in $\mathbf{m}_E$ has a corresponding negative gradient, then the solution can be changed to decrease the prediction error. To change the solution, we select the model parameter corresponding to the most negative gradient and move it to the set $\mathbf{m}_S$. All the model parameters in $\mathbf{m}_S$ are now recomputed by solving the system $\mathbf{G}_S\mathbf{m}'_S = \mathbf{d}_S$ in the least squares sense. The subscript S on the matrix indicates that only the columns multiplying the model parameters in $\mathbf{m}_S$ have been included in the calculation. All the $\mathbf{m}_E$s are still zero. If the new model parameters are all feasible, then we set $\mathbf{m} = \mathbf{m}'$ and return to Step 2.

Step 4. If some of the elements of $\mathbf{m}'_S$ are infeasible, however, we cannot use this vector as a new guess for the solution. Instead, we compute the change in the solution $\delta\mathbf{m} = \mathbf{m}'_S - \mathbf{m}_S$ and add as much of this vector as possible to the solution $\mathbf{m}_S$ without causing the solution to become infeasible. We therefore replace $\mathbf{m}_S$ with the new guess $\mathbf{m}_S + \alpha\delta\mathbf{m}$, where $\alpha = \min_i\{m_{Si}/[m_{Si} - m'_{Si}]\}$ is the largest choice that can be made without some $\mathbf{m}_S$ becoming infeasible. At least one of the m_{Si}s has its constraint satisfied in the equality sense and must be moved back to $\mathbf{m}_E$. The process then returns to Step 3.

This algorithm contains two loops, one nested within the other. The outer loop successively moves model parameters from the group that is constrained to the group that minimizes the prediction error. The inner loop ensures that the addition of a variable to this latter group has not caused any of the constraints to be violated. Discussion of the convergence properties of this algorithm can be found in Lawson and Hanson (1974).

MatLab provides a function that implements nonnegative least squares

```
mest = lsqnonneg(G,dobs);
```

(*MatLab* script gda07_06)

As an example of its use, we analyze the problem of determining the mass distribution of an object from observations of its gravitational force (Figure 7.7). Mass is an inherently positive quantity, so the positivity constraint constitutes very accurate *a priori* information. The gravitational force d_i is measured at $N = 600$ points (x_i, y_i) above the object. The object (Figure 7.7A) is subdivided into a grid of 20×20 pixels, each of mass m_j and position (x_j, y_j). According to Newton's inverse-square law, the data kernel is $G_{ij} = \gamma \cos(\theta_{ij})/R_{ij}^2$, where γ is the gravitational constant, θ_{ij} is angle with respect to the vertical and R_{ij} is distance. Singular-value decomposition of $\mathbf{G}$ indicates that this mixed-determined problem is extremely nonunique, with only about 20 nonzero eigenvalues.

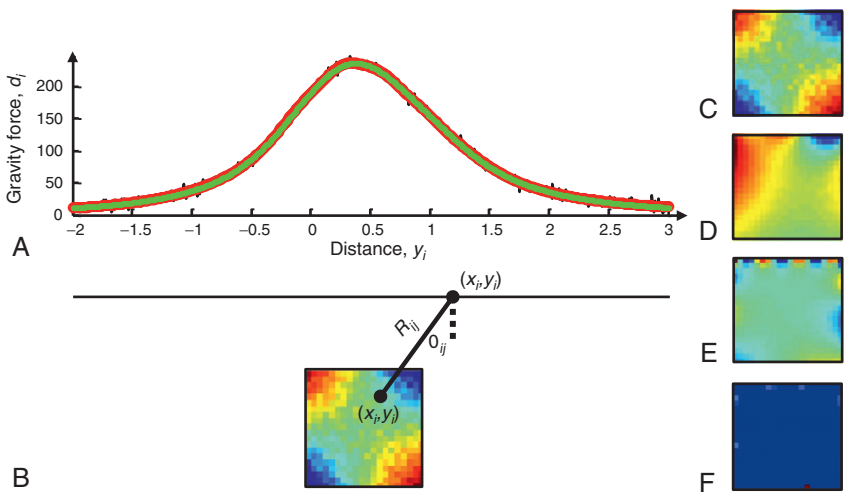

FIGURE 7.7 (A) The vertical force of gravity is measured on a horizontal line above a massive cube-shaped object. This cube contains a grid of 20×20 model parameters representing spatially varying density, $\mathbf{m}$ (colors). The equation $\mathbf{Gm} = \mathbf{d}$ embodies Newton's inverse-square law of gravity. (B) The $N = 600$ gravitational force observations $\mathbf{d}^{obs}$ (black curve) and the gravitational force predicted by the natural solution (red curve, $p = 15$) and nonnegative least squares (green curve). (C) True model. (D) Natural estimate of model, with $p = 4$. (E) Natural estimate of model, with $p = 15$. (F) Nonnegative estimate of model. *MatLab* script gda07_06.

Thus, the solution that one obtains depends critically on the type of *a priori* information that one employs. In this case, the natural solution (Figure 7.7E), which contains 136 negative masses, is completely different from the nonnegative solution (Figure 7.7F), which contains none. Nevertheless, both satisfy the data almost exactly (Figure 7.7A), with the prediction error E of the nonnegative solution about 1% larger than that of the natural solution. Ironically, neither the nonnegative solution nor the natural solution looks at all like the true solution (Figure 7.7C), but a heavily damped solution (Figure 7.7D) does.

The nonnegative least squares algorithm can also be used to solve the problem (Lawson and Hanson, 1974)

$$\text{minimize} \quad E = \|\mathbf{m}\|_2 \quad \text{subject to} \quad \mathbf{Hm} \geq \mathbf{h} \tag{7.60}$$

and, by virtue of the transformation described in Section 7.9.1, the completely general problem. The method consists of forming the $(M+1) \times p$ equation

$$\mathbf{G}'\mathbf{m}' = \mathbf{d}' = \begin{bmatrix} \mathbf{H}^{\mathrm{T}} \\ \mathbf{h}^{\mathrm{T}} \end{bmatrix} \mathbf{m}' = \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix} \tag{7.61}$$

and finding the $\mathbf{m}'$ that minimizes $\|\mathbf{d}' - \mathbf{G}'\mathbf{m}'\|_2$ subject to $\mathbf{m}' \geq 0$ by the nonnegative least squares algorithm described above. If the prediction error $\mathbf{e}' = \mathbf{d}' - \mathbf{G}'\mathbf{m}'$ is identically zero, then the constraints $\mathbf{Hm} \geq \mathbf{h}$ are inconsistent. Otherwise, the solution is $m_i = -e'_i/e'_{M+1}$ (which can also be written as $\mathbf{e}' = [-\mathbf{m}, 1]^{\mathrm{T}} e'_{M+1}$).

In *MatLab*, the solution is computed as

```
Gp = [H, h]';
dp = [zeros(1,length(H(1,:))), 1]';
mp = lsqnonneg(Gp,dp);
ep = dp - Gp*mp;
m = -ep(1:end-1)/ep(end);
```

(*MatLab* script gda07_07)

An example with $M = 2$ model parameters and $N = 3$ constraints is shown in Figure 7.8. The *MatLab* code for converting a *least squares with inequality constraints* problem into a *nonnegative least squares problem* using the procedure in Section 7.8.2 is:

```
[Up, Lp, Vp] = svd(G,0);
lambda = diag(Lp);
rlambda = 1./lambda;
Lpi = diag(rlambda);

% transformation 1
Hp = -H*Vp*Lpi;
hp = h + Hp*Up'*dobs;
```

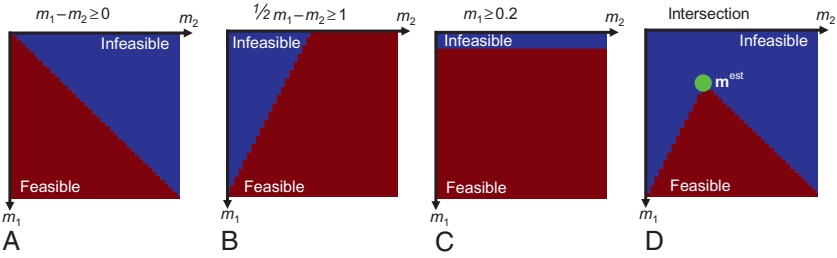

FIGURE 7.8 Exemplary solution of the problem of minimizing $\mathbf{m}^{\mathrm{T}}\mathbf{m}$ with $N = 3$ inequality constraints $\mathbf{Hm} \geq \mathbf{h}$. (A–C) Each constraint divides the (m_1, m_2) plane into two half-places, one feasible and the other infeasible. (D) The intersection of the three feasible half-planes is polygonal in shape. The solution $\mathbf{m}^{\mathrm{est}}$ (green circle) is the point in feasible area that is closest to the origin. Note that two of the three constrains are satisfied in the equality sense. *MatLab* script gda07_07.

```
% transformation 2
Gp = [Hp, hp]';
dp = [zeros(1,length(Hp(1,:))), 1]';
mpp = lsqnonneg(Gp,dp);
ep = dp - Gp*mpp;
mp = -ep(1:end-1)/ep(end);

% take mp back to m
mest = Vp*Lpi*(Up'*dobs-mp);
dpre = G*mest;
```

(*MatLab* script gda07_07)

Note that `svd()` is called a second argument, set to zero, which causes it to compute $\mathbf{U}_p$ rather than the default, the $\mathbf{U}$. An exemplary problem is illustrated in Figure 7.9.

We now demonstrate that this method does indeed solve the indicated problem (adapted from Lawson and Hanson, 1974). Step 1 is to show that e'_{M+1} is nonnegative. We first note that the gradient of the error satisfies $1/2\nabla E' = -\mathbf{G}'^{\mathrm{T}}[\mathbf{d}' - \mathbf{G}'\mathbf{m}'] = -\mathbf{G}'^{\mathrm{T}}\mathbf{e}'$, and that because of the Kuhn-Tucker theorem, $\mathbf{m}'$ and $\nabla E'$ satisfy

$$\begin{aligned} \mathbf{m}'_{\mathrm{E}} &= 0 \quad [\nabla E']_{\mathrm{E}} < 0 \\ \mathbf{m}'_{\mathrm{S}} &> 0 \quad [\nabla E']_{\mathrm{S}} = 0 \end{aligned} \tag{7.62}$$

Note that these conditions imply $\mathbf{m}'^{\mathrm{T}}\nabla E' = 0$. The length of the error E' is therefore

$$\begin{aligned} E' &= \mathbf{e}'^{\mathrm{T}}\mathbf{e}' = [\mathbf{d}' - \mathbf{G}'\mathbf{m}']^{\mathrm{T}}\mathbf{e}' = \mathbf{d}'^{\mathrm{T}}\mathbf{e}' - \mathbf{m}'^{\mathrm{T}}\mathbf{G}'^{\mathrm{T}}\mathbf{e}' \\ &= e'_{M+1} + {}^1\!/_2\mathbf{m}'^{\mathrm{T}}\nabla E' = e'_{M+1} \end{aligned} \tag{7.63}$$

as $\mathbf{d}'^{\mathrm{T}}\mathbf{e}' = [\mathbf{0},1]\,\mathbf{e}' = e'_{M+1}$. The error E' is necessarily a nonnegative quantity, so if it is not identically zero, then e'_{M+1} must be greater than zero.

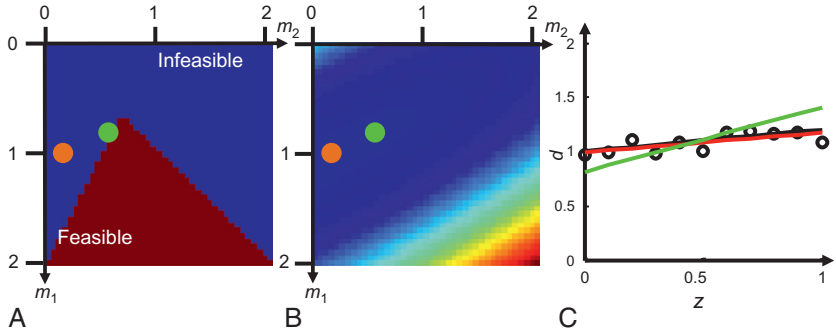

FIGURE 7.9 Problem of minimizing the prediction error E subject to inequality constraints, applied to the straight line problem. (A) Feasible region of the model parameters (the intercept m_1 and slope m_2 of the straight line). The unconstrained solution (orange circle) is outside the feasible region, but the constrained solution (green circle) is on its boundary. (B) The unconstrained solution (orange circle) is at the global minimum of the prediction error E, while the constrained solution is not. (C) Plot of the data $d(z)$ showing true data (black line), observed data (black circles), unconstrained prediction (red line), and constrained prediction (green line). *MatLab* script gda07_08.

Step 2 is to show that the solution satisfies the inequality constraints, $\mathbf{Hm}-\mathbf{h}\geq 0$. We start with the gradient of the error $\nabla E'$, which must have all nonnegative elements (or else the solution $\mathbf{m}'$ could be further minimized without violating the constraints $\mathbf{m}'\geq 0$):

$$0 \leq \frac{1}{2}\nabla E' = -\mathbf{G}'^{\mathrm{T}}\mathbf{e}' = -[\mathbf{H},\mathbf{h}][-\mathbf{m},1]^{\mathrm{T}}e'_{\mathrm{M}+1} = [\mathbf{Hm}-\mathbf{h}]e'_{\mathrm{M}+1} \qquad (7.64)$$

As $e'_{\mathrm{M}+1}>0$, we have $\mathbf{Hm}-\mathbf{h}\geq 0$.

Step 3 is to show that the solution minimizes $\|\mathbf{m}\|_2$. This follows from the Kuhn-Tucker condition that, at a valid solution, the gradient of the error ∇E be represented as a nonnegative combination of the rows of $\mathbf{H}$:

$$\nabla E = \nabla\|\mathbf{m}\|_2^2 = 2\mathbf{m} = -2\left[e'_1 \cdots e'_M\right]^{\mathrm{T}}/e'_{M+1} = 2\mathbf{H}^{\mathrm{T}}\left[m'_1 \cdots m'_M\right]^{\mathrm{T}}/e'_{M+1} \qquad (7.65)$$

Here we have used the fact that $\mathbf{e}'=\mathbf{d}'-\mathbf{G}'\mathbf{m}'$, together with the fact that the first M elements of $\mathbf{d}'$ are zero. Note that $\mathbf{m}'$ and e'_{M+1} are nonnegative, so the Kuhn-Tucker condition is satisfied.

Finally, we can also show that a feasible solution exists only when the error is not identically zero. Consider the contradiction, that the error is identically zero but that a feasible solution exists. Then

$$0 = \mathbf{e}'^{\mathrm{T}}\mathbf{e}'/e'_{M+1} = [-\mathbf{m}^{\mathrm{T}},1][\mathbf{d}'-\mathbf{G}'\mathbf{m}'] = 1+[\mathbf{Hm}-\mathbf{h}]^{\mathrm{T}}\mathbf{m}' \qquad (7.66)$$

As $\mathbf{m}' \geq 0$, the relationship $\mathbf{Hm} < \mathbf{h}$ is implied. This contradicts the constraint equations $\mathbf{Hm} \geq \mathbf{h}$ so that an identically zero error implies that no feasible solution exists and that the constraints are inconsistent.

7.10 PROBLEMS

7.1 This problem builds upon the discussion in Section 6.2. Use *MatLab*'s svd() function to compute the null vectors associated with the data kernel $\mathbf{G} = [1,1,1,1]$. (A) The null vectors are different from given in Equation (6.5). Why? Show that the two sets are equivalent.

7.2 Modify the weighted damped least squares "gap-filling" problem of Figure 3.10 (*MatLab* script gda03_09) so that the *a priori* information is applied only to the part of the solution in the null space. First, transform the weighted problem $\mathbf{Gm} = \mathbf{d}$ into an unweighted one $\mathbf{G}'\mathbf{m}' = \mathbf{d}'$ using the transformation given by Equation (7.23). Then, find the minimum-length solution (or the natural solution) $\mathbf{m}'^{\text{est}}$. Finally, transform back to obtain $\mathbf{m}^{\text{est}}$. How different is this solution from the one given in the figure? Compare the two prediction errors. In order to insure that $\mathbf{W}_e = \mathbf{D}^T\mathbf{D}$ has an inverse, which is required by the transformation, you should make $\mathbf{D}$ square by adding two rows, one at the top and the other at the bottom, that constrain the first and last model parameters to known values.

7.3 (A) Compute and plot the null vectors for the data kernel shown in Figure 7.4A. (B) Suppose that the elements of $\mathbf{m}^{\text{true}}$ are drawn from a uniform distribution between 0 and 1. How large a contribution do the null vectors make to the true solution $\mathbf{m}^{\text{true}}$? (Hint: Write the model parameters as a linear combination of all the eigenvectors $\mathbf{V}$ by writing $\mathbf{m}^{\text{true}} = \mathbf{Va}$, where $\mathbf{a}$ are coefficients, and then solving for $\mathbf{a}$. Then examine the size of the elements of $\mathbf{a}$. You may wish to refer to *MatLab* script gda07_03 for the definition of $\mathbf{G}$).

7.4 Consider fitting a cubic polynomial $d_i = m_1 + m_2 z_i + m_3 z_i^2 + m_4 z_i^3$ to $N = 20$ data d_i^{obs}, where the zs are evenly spaced on the interval (0,1), where $m_2 = m_3 = m_4 = 1$, and where m_1 is varied from -1 to 1, as described below. (A) Write a *MatLab* script that generates synthetic observed data (including Gaussian-distributed noise) for a particular choice of m_1, estimates the model parameters using both simple least squares and nonnegative least squares, plots the observed and predicted data, and outputs the total error. (B) Run a series of test cases for a range of values of m_1 and comment upon the results.

7.5 (A) Modify the *MatLab* gda07_07 script so that the last constraint is $m_1 \geq 1.2$. How do the feasible region and the solution change? (B) Modify the script to add a constraint $m_2 \geq 0.2$. How do the feasible region and the solution change?

REFERENCES

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

Lawson, C.L., Hanson, D.J., 1974. Solving Least Squares Problems. Prentice-Hall, Englewood Cliffs, New Jersey.

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

Penrose, R.A., 1955. A generalized inverse for matrices. Proc. Cambridge Phil. Soc. 51, 406–413.

Wiggins, R.A., 1972. The general linear inverse problem: implication of surface waves and free oscillations for Earth structure. Rev. Geophys. Space Phys. 10, 251–285.

Wunsch, C., Minster, J.F., 1982. Methods for box models and ocean circulation tracers: mathematical programming and non-linear inverse theory. J. Geophys. Res. 87, 5647–5662.