4

RANK DEFICIENCY AND ILL-CONDITIONING

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

4.1 THE SVD AND THE GENERALIZED INVERSE

A method of analyzing and solving least squares problems that is of particular interest in ill-conditioned and/or rank-deficient systems is the **singular value decomposition**, or **SVD**. In the SVD [49, 89, 155] an m by n matrix $\mathbf{G}$ is factored into

$$\mathbf{G} = \mathbf{U}\mathbf{S}\mathbf{V}^T \tag{4.1}$$

where

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

The SVD matrices can be computed in MATLAB with the **svd** command. It can be shown that every matrix has a singular value decomposition [49].

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

first p singular values are nonzero, we can partition $\mathbf{S}$ as

$$\mathbf{S} = \begin{bmatrix} \mathbf{S}_p & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{4.2}$$

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

$$\mathbf{G} = [\mathbf{U}_{\cdot,1}, \mathbf{U}_{\cdot,2}, \ldots, \mathbf{U}_{\cdot,m}] \begin{bmatrix} \mathbf{S}_p & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} [\mathbf{V}_{\cdot,1}, \mathbf{V}_{\cdot,2}, \ldots, \mathbf{V}_{\cdot,n}]^T \tag{4.3}$$

$$= [\mathbf{U}_p, \mathbf{U}_0] \begin{bmatrix} \mathbf{S}_p & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} [\mathbf{V}_p, \mathbf{V}_0]^T \tag{4.4}$$

where $\mathbf{U}_p$ denotes the first p columns of $\mathbf{U}$, $\mathbf{U}_0$ denotes the last $m - p$ columns of $\mathbf{U}$, $\mathbf{V}_p$ denotes the first p columns of $\mathbf{V}$, and $\mathbf{V}_0$ denotes the last $n - p$ columns of $\mathbf{V}$. Because the last $m - p$ columns of $\mathbf{U}$ and the last $n - p$ columns of $\mathbf{V}$ in (4.4) are multiplied by zeros in $\mathbf{S}$, we can simplify the SVD of $\mathbf{G}$ into its **compact form**

$$\mathbf{G} = \mathbf{U}_p \mathbf{S}_p \mathbf{V}_p^T. \tag{4.5}$$

For any vector $\mathbf{y}$ in the range of $\mathbf{G}$, applying (4.5) gives

$$\mathbf{y} = \mathbf{G}\mathbf{x} \tag{4.6}$$

$$= \mathbf{U}_p \left(\mathbf{S}_p \mathbf{V}_p^T \mathbf{x} \right). \tag{4.7}$$

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

$$\mathbf{y} = \sum_{i=1}^{p} z_i \mathbf{U}_{\cdot,i}. \tag{4.8}$$

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

Since $\mathbf{U}$ is an orthogonal matrix, the columns of $\mathbf{U}$ form an orthonormal basis for R^m. We have already shown in (4.8) that the p columns of $\mathbf{U}_p$ form an orthonormal basis for $R(\mathbf{G})$. By Theorem A.5, $N(\mathbf{G}^T) + R(\mathbf{G}) = R^m$, so the remaining $m - p$ columns of $\mathbf{U}_0$ form an orthonormal basis for the null space of $\mathbf{G}^T$. We will sometimes refer to $N(\mathbf{G}^T)$ as the **data null space**. Similarly, because $\mathbf{G}^T = \mathbf{V}_p \mathbf{S}_p \mathbf{U}_p^T$, the columns of $\mathbf{V}_p$ form an orthonormal basis for $R(\mathbf{G}^T)$ and the columns of $\mathbf{V}_0$ form an orthonormal basis for $N(\mathbf{G})$. We will sometimes refer to $N(\mathbf{G})$ as the **model null space**.

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

$$\mathbf{V}^T\mathbf{V}_{\cdot,i} = \mathbf{e}_i. \tag{4.9}$$

Thus

$$\mathbf{G}\mathbf{V}_{\cdot,i} = \mathbf{U}\mathbf{S}\mathbf{V}^T\mathbf{V}_{\cdot,i} \tag{4.10}$$

$$= \mathbf{U}\mathbf{S}\mathbf{e}_i \tag{4.11}$$

$$= s_i\mathbf{U}_{\cdot,i} \tag{4.12}$$

and

$$\mathbf{G}^T\mathbf{U}_{\cdot,i} = \mathbf{V}\mathbf{S}^T\mathbf{U}^T\mathbf{U}_{\cdot,i} \tag{4.13}$$

$$= \mathbf{V}\mathbf{S}^T\mathbf{e}_i \tag{4.14}$$

$$= s_i\mathbf{V}_{\cdot,i}. \tag{4.15}$$

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

$$\mathbf{G}\mathbf{G}^T\mathbf{U}_{\cdot,i} = \mathbf{G}s_i\mathbf{V}_{\cdot,i} \tag{4.16}$$

$$= s_i\mathbf{G}\mathbf{V}_{\cdot,i} \tag{4.17}$$

$$= s_i^2\mathbf{U}_{\cdot,i}. \tag{4.18}$$

Similarly,

$$\mathbf{G}^T\mathbf{G}\mathbf{V}_{\cdot,i} = s_i^2\mathbf{V}_{\cdot,i}. \tag{4.19}$$

These relations show that we could, in theory, compute the SVD by finding the eigenvalues and eigenvectors of $\mathbf{G}^T\mathbf{G}$ and $\mathbf{G}\mathbf{G}^T$. In practice, more efficient specialized algorithms are used [31, 49, 167].

The SVD can be used to compute a generalized inverse of $\mathbf{G}$, called the **Moore–Penrose pseudoinverse**, because it has desirable inverse properties originally identified by Moore and Penrose [102, 125]. The generalized inverse is

$$\mathbf{G}^\dagger = \mathbf{V}_p\mathbf{S}_p^{-1}\mathbf{U}_p^T. \tag{4.20}$$

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

Using (4.20), we define the pseudoinverse solution to be

$$\mathbf{m}_\dagger = \mathbf{G}^\dagger \mathbf{d} \tag{4.21}$$

$$= \mathbf{V}_p \mathbf{S}_p^{-1} \mathbf{U}_p^T \mathbf{d}. \tag{4.22}$$

Among the desirable properties of (4.22) is that $\mathbf{G}^\dagger$, and hence $\mathbf{m}_\dagger$, always exist. In contrast, the inverse of $\mathbf{G}^T\mathbf{G}$ that appears in the normal equations (2.3) does not exist when $\mathbf{G}$ is not of full column rank. We will shortly show that $\mathbf{m}_\dagger$ is a least squares solution.

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

1. Both the model and data null spaces, $N(\mathbf{G})$ and $N(\mathbf{G}^T)$ are trivial (only include the zero vector). $\mathbf{U}_p = \mathbf{U}$ and $\mathbf{V}_p = \mathbf{V}$ are square orthogonal matrices, so that $\mathbf{U}_p^T = \mathbf{U}_p^{-1}$, and $\mathbf{V}_p^T = \mathbf{V}_p^{-1}$. Equation (4.22) gives

$$\mathbf{G}^\dagger = \mathbf{V}_p \mathbf{S}_p^{-1} \mathbf{U}_p^T \tag{4.23}$$

$$= (\mathbf{U}_p \mathbf{S}_p \mathbf{V}_p^T)^{-1} \tag{4.24}$$

$$= \mathbf{G}^{-1} \tag{4.25}$$

which is the matrix inverse for a square full rank matrix where $m = n = p$. The solution is unique, and the data are fit exactly.

2. $N(\mathbf{G})$ is nontrivial, but $N(\mathbf{G}^T)$ is trivial. $\mathbf{U}_p^T = \mathbf{U}_p^{-1}$ and $\mathbf{V}_p^T \mathbf{V}_p = \mathbf{I}_p$. $\mathbf{G}$ applied to the generalized inverse solution 4.21 gives

$$\mathbf{G}\mathbf{m}_\dagger = \mathbf{G}\mathbf{G}^\dagger \mathbf{d} \tag{4.26}$$

$$= \mathbf{U}_p \mathbf{S}_p \mathbf{V}_p^T \mathbf{V}_p \mathbf{S}_p^{-1} \mathbf{U}_p^T \mathbf{d} \tag{4.27}$$

$$= \mathbf{U}_p \mathbf{S}_p \mathbf{I}_p \mathbf{S}_p^{-1} \mathbf{U}_p^T \mathbf{d} \tag{4.28}$$

$$= \mathbf{d}. \tag{4.29}$$

The data are fit exactly but the solution is nonunique, because of the existence of the nontrivial model null space $N(\mathbf{G})$. Since $\mathbf{m}_\dagger$ is an exact solution to $\mathbf{G}\mathbf{m} = \mathbf{d}$, it is also a least-squares solution.

We need to characterize the least squares solutions to $\mathbf{G}\mathbf{m} = \mathbf{d}$. If $\mathbf{m}$ is any least squares solution, then it satisfies the normal equations. This is shown in Exercise C.5.

$$(\mathbf{G}^T\mathbf{G})\mathbf{m} = \mathbf{G}^T\mathbf{d}. \tag{4.30}$$

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

$$(\mathbf{G}^T\mathbf{G})\mathbf{m}_\dagger = \mathbf{G}^T\mathbf{d}. \tag{4.31}$$

Subtracting (4.30) from (4.31), we find that

$$(\mathbf{G}^T\mathbf{G})(\mathbf{m}_\dagger - \mathbf{m}) = \mathbf{0}. \tag{4.32}$$

Thus $\mathbf{m}_\dagger - \mathbf{m}$ lies in $N(\mathbf{G}^T\mathbf{G})$. In Exercise A.17f it is shown that $N(\mathbf{G}^T\mathbf{G}) = N(\mathbf{G})$. This implies that $\mathbf{m}_\dagger - \mathbf{m}$ lies in $N(\mathbf{G})$.

The general solution is thus the sum of $\mathbf{m}_\dagger$ and an arbitrary vector in $N(\mathbf{G})$ that can be written as a linear combination of the basis vectors for $N(\mathbf{G})$:

$$\mathbf{m} = \mathbf{m}_\dagger + \mathbf{m}_0 \tag{4.33}$$

$$= \mathbf{m}_\dagger + \sum_{i=p+1}^{n} \alpha_i \mathbf{V}_{\cdot,i}. \tag{4.34}$$

Because the columns of $\mathbf{V}$ are orthonormal, the square of the 2-norm of a general solution $\mathbf{m}$ is

$$\|\mathbf{m}\|_2^2 = \|\mathbf{m}_\dagger\|_2^2 + \sum_{i=p+1}^{n} \alpha_i^2 \geq \|\mathbf{m}_\dagger\|_2^2 \tag{4.35}$$

where we have equality only if all of the model null space coefficients α_i are zero. The generalized inverse solution is thus a **minimum length solution.**

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

$$\mathbf{m}_\dagger = \mathbf{V}_p\mathbf{S}_p^{-1}\mathbf{U}_p^T\mathbf{d} \tag{4.36}$$

$$= \mathbf{V}_p\mathbf{S}_p\mathbf{U}_p^T\mathbf{U}_p\mathbf{S}_p^{-2}\mathbf{U}_p^T\mathbf{d} \tag{4.37}$$

$$= \mathbf{G}^T(\mathbf{U}_p\mathbf{S}_p^{-2}\mathbf{U}_p^T)\mathbf{d} \tag{4.38}$$

$$= \mathbf{G}^T(\mathbf{G}\mathbf{G}^T)^{-1}\mathbf{d}. \tag{4.39}$$

In practice it is better to compute a solution using the SVD than to use (4.39) because of numerical accuracy issues.

3. $N(\mathbf{G})$ is trivial but $N(\mathbf{G}^T)$ is nontrivial and $R(\mathbf{G})$ is a strict subset of R^m. Here

$$\mathbf{G}\mathbf{m}_\dagger = \mathbf{U}_p\mathbf{S}_p\mathbf{V}_p^T\mathbf{V}_p\mathbf{S}_p^{-1}\mathbf{U}_p^T\mathbf{d} \tag{4.40}$$

$$= \mathbf{U}_p\mathbf{U}_p^T\mathbf{d}. \tag{4.41}$$

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

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

$$(\mathbf{G}^T\mathbf{G})^{-1} = (\mathbf{V}_p\mathbf{S}_p\mathbf{U}_p^T\mathbf{U}_p\mathbf{S}_p\mathbf{V}_p^T)^{-1} \tag{4.42}$$

$$= (\mathbf{V}_p\mathbf{S}_p^2\mathbf{V}_p^T)^{-1} \tag{4.43}$$

$$= \mathbf{V}_p\mathbf{S}_p^{-2}\mathbf{V}_p^T \tag{4.44}$$

and

$$\mathbf{m}_\dagger = \mathbf{G}^\dagger\mathbf{d} \tag{4.45}$$

$$= \mathbf{V}_p\mathbf{S}_p^{-1}\mathbf{U}_p^T\mathbf{d} \tag{4.46}$$

$$= \mathbf{V}_p\mathbf{S}_p^{-2}\mathbf{V}_p^T\mathbf{V}_p\mathbf{S}_p\mathbf{U}_p^T\mathbf{d} \tag{4.47}$$

$$= (\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T\mathbf{d}. \tag{4.48}$$

This solution is unique, but cannot fit general data exactly. As with (4.39), it is better in practice to use the generalized inverse solution than to use (4.48) because of numerical accuracy issues.

4. Both $N(\mathbf{G}^T)$ and $N(\mathbf{G})$ are nontrivial and p is less than both m and n. In this case, the generalized inverse solution encapsulates the behavior of both of the two previous cases, minimizing both $\|\mathbf{Gm} - \mathbf{d}\|_2$ and $\|\mathbf{m}\|_2$.
As in case 3,

$$\mathbf{Gm}_\dagger = \mathbf{U}_p\mathbf{S}_p\mathbf{V}_p^T\mathbf{V}_p\mathbf{S}_p^{-1}\mathbf{U}_p^T\mathbf{d} \tag{4.49}$$

$$= \mathbf{U}_p\mathbf{U}_p^T\mathbf{d} \tag{4.50}$$

$$= \text{proj}_{R(\mathbf{G})}\mathbf{d}. \tag{4.51}$$

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

As in case 2 we can write the model and its norm using (4.34) and (4.35). Thus $\mathbf{m}_\dagger$ is the least squares solution of minimum length.

We have shown that the generalized inverse provides an inverse solution (4.22) that always exists, is both least squares and minimum length, and properly accommodates the rank and dimensions of $\mathbf{G}$. Relationships between the subspaces $R(\mathbf{G})$, $N(\mathbf{G}^T)$, $R(\mathbf{G}^T)$, $N(\mathbf{G})$ and the operators $\mathbf{G}$ and $\mathbf{G}^\dagger$ are schematically depicted in Figure 4.1. Table 4.1 summarizes the SVD and its properties.

The existence of a nontrivial model null space (one that includes more than just the zero vector) is at the heart of solution nonuniqueness. There are an infinite number of solutions that will fit the data equally well, because model components in $N(\mathbf{G})$ have no effect on data fit. To select a particular preferred solution from this infinite set thus requires more constraints (such as minimum length or smoothing constraints) than are encoded in the matrix $\mathbf{G}$.

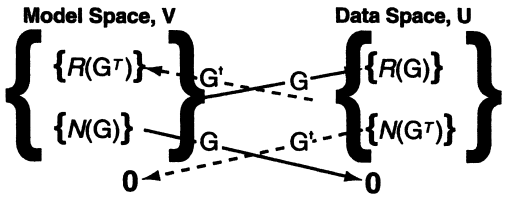

Figure 4.1 SVD model and data space mappings, where $\mathbf{G}^\dagger$ is the generalized inverse. $N(\mathbf{G}^T)$ and $N(\mathbf{G})$ are the data and model null spaces, respectively.

Table 4.1 Summary of the SVD and its associated scalars and matrices.

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

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

$$\mathbf{d}_0 = \sum_{i=p+1}^{m} \beta_i \mathbf{U}_{\cdot,i}. \tag{4.52}$$

The generalized inverse operating on such a data vector gives

$$\mathbf{m}_\dagger = \mathbf{V}_p\mathbf{S}_p^{-1}\mathbf{U}_p^T\mathbf{d}_0 \tag{4.53}$$

$$= \mathbf{V}_p\mathbf{S}_p^{-1}\sum_{i=p+1}^{n} \beta_i \mathbf{U}_p^T\mathbf{U}_{\cdot,i} \tag{4.54}$$

$$= \mathbf{0} \tag{4.55}$$

because the $\mathbf{U}_{\cdot,i}$ are orthogonal. $N(\mathbf{G}^T)$ is a subspace of R^m consisting of all vectors $\mathbf{d}_0$ that have no influence on the generalized inverse model, $\mathbf{m}_\dagger$. If $p < n$ there are an infinite number of potential data sets that will produce the same model when (4.22) is applied.

4.2 COVARIANCE AND RESOLUTION OF THE GENERALIZED INVERSE SOLUTION

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

In Section 2.2, we found that, under the assumption of independent and normally distributed measurement errors, the least squares solution was an unbiased estimator of the true model, and that the estimated model parameters had a multivariate normal distribution with covariance

$$\text{Cov}(\mathbf{m}_{L_2}) = \sigma^2(\mathbf{G}^T\mathbf{G})^{-1}. \tag{4.56}$$

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

$$\text{Cov}(\mathbf{m}_\dagger) = \mathbf{G}^\dagger \text{Cov}(\mathbf{d})(\mathbf{G}^\dagger)^T \tag{4.57}$$

$$= \sigma^2 \mathbf{G}^\dagger (\mathbf{G}^\dagger)^T \tag{4.58}$$

$$= \sigma^2 \mathbf{V}_p \mathbf{S}_p^{-2} \mathbf{V}_p^T \tag{4.59}$$

$$= \sigma^2 \sum_{i=1}^{p} \frac{V_{\cdot,i} V_{\cdot,i}^T}{s_i^2}. \tag{4.60}$$

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

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

$$\mathbf{m}_\dagger = \mathbf{G}^\dagger \mathbf{G}\mathbf{m}. \tag{4.61}$$

We would obviously like to get back our original model so that $\mathbf{m}_\dagger = \mathbf{m}$. Since the original model may have had a nonzero projection onto the model null space $N(\mathbf{G})$, $\mathbf{m}_\dagger$ will not in

general be equal to $\mathbf{m}$. The **model resolution matrix** is

$$\mathbf{R}_{\mathrm{m}} = \mathbf{G}^{\dagger}\mathbf{G} \tag{4.62}$$

$$= \mathbf{V}_p\mathbf{S}_p^{-1}\mathbf{U}_p^T\mathbf{U}_p\mathbf{S}_p\mathbf{V}_p^T \tag{4.63}$$

$$= \mathbf{V}_p\mathbf{V}_p^T. \tag{4.64}$$

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

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

$$E[\mathbf{m}_{\dagger}] = E[\mathbf{G}^{\dagger}\mathbf{d}] \tag{4.65}$$

$$= \mathbf{G}^{\dagger}E[\mathbf{d}] \tag{4.66}$$

$$= \mathbf{G}^{\dagger}\mathbf{G}\mathbf{m}_{\text{true}} \tag{4.67}$$

$$= \mathbf{R}_{\mathrm{m}}\mathbf{m}_{\text{true}}. \tag{4.68}$$

Thus the bias in the generalized inverse solution is

$$E[\mathbf{m}_{\dagger}] - \mathbf{m}_{\text{true}} = \mathbf{R}_{\mathrm{m}}\mathbf{m}_{\text{true}} - \mathbf{m}_{\text{true}} \tag{4.69}$$

$$= (\mathbf{R}_{\mathrm{m}} - \mathbf{I})\mathbf{m}_{\text{true}} \tag{4.70}$$

where

$$\mathbf{R}_{\mathrm{m}} - \mathbf{I} = \mathbf{V}_p\mathbf{V}_p^T - \mathbf{V}\mathbf{V}^T \tag{4.71}$$

$$= -\mathbf{V}_0\mathbf{V}_0^T. \tag{4.72}$$

Notice that as p increases, $\mathbf{R}_{\mathrm{m}}$ approaches $\mathbf{I}$. Equations (4.60) and (4.72) reveal an important trade-off associated with the value of p. As p increases, the variance in the generalized inverse solution increases (4.60), but bias decreases.

We can formulate a bound on the norm of the bias (4.70):

$$\|E[\mathbf{m}_{\dagger}] - \mathbf{m}_{\text{true}}\| \leq \|\mathbf{R}_{\mathrm{m}} - \mathbf{I}\|\|\mathbf{m}_{\text{true}}\|. \tag{4.73}$$

Computing $\|\mathbf{R}_{\mathrm{m}} - \mathbf{I}\|$ can give us some idea of how much bias has been introduced by the generalized inverse solution. The bound is not very useful, since we typically have no *a priori* knowledge of $\|\mathbf{m}_{\text{true}}\|$.

In practice, the model resolution matrix is commonly used in two different ways. First, we can examine the diagonal elements of $\mathbf{R}_m$. Diagonal elements that are close to one correspond to parameters for which we can claim good resolution. Conversely, if any of the diagonal elements are small, then the corresponding model parameters will be poorly resolved. Second, we can multiply $\mathbf{R}_m$ times a particular test model $\mathbf{m}$ to see how that model would be resolved in the inverse solution. This strategy is called a **resolution test**. One commonly used test model is a **spike model**, which is a vector with all zero elements, except for a single entry which is one. Multiplying $\mathbf{R}_m$ times a spike model effectively picks out the corresponding column of the resolution matrix. These columns of the resolution matrix are called **resolution kernels**. These are similar to the averaging kernels in the method of Backus and Gilbert (see Section 3.5).

We can multiply $\mathbf{G}^\dagger$ and $\mathbf{G}$ in the opposite order from (4.64) to obtain the **data space resolution matrix**, $\mathbf{R}_d$:

$$\mathbf{d}_\dagger = \mathbf{G}\mathbf{m}_\dagger \tag{4.74}$$

$$= \mathbf{G}\mathbf{G}^\dagger\mathbf{d} \tag{4.75}$$

$$= \mathbf{R}_d\mathbf{d} \tag{4.76}$$

where

$$\mathbf{R}_d = \mathbf{U}_p\mathbf{S}_p\mathbf{V}_p^T\mathbf{V}_p\mathbf{S}_p^{-1}\mathbf{U}_p^T \tag{4.77}$$

$$= \mathbf{U}_p\mathbf{U}_p^T. \tag{4.78}$$

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

Note that model and data space resolution matrices (4.64) and (4.78) do *not* depend on specific data or models, but are exclusively properties of $\mathbf{G}$. They reflect the physics and geometry of a problem, and can thus be assessed during the design phase of an experiment.

4.3 INSTABILITY OF THE GENERALIZED INVERSE SOLUTION

The generalized inverse solution $\mathbf{m}_\dagger$ has zero projection onto $N(\mathbf{G})$. However, it may include terms involving column vectors in $\mathbf{V}_p$ with very small nonzero singular values. In analyzing the generalized inverse solution it is useful to examine the **singular value spectrum**, which is simply the range of singular values. Small singular values cause the generalized inverse solution to be extremely sensitive to small amounts of noise in the data. As a practical matter, it can also be difficult to distinguish between zero singular values and extremely small singular values. We can quantify the instabilities created by small singular values by recasting the generalized inverse solution to make the effect of small singular values explicit. We start with the formula for the generalized inverse solution

$$\mathbf{m}_\dagger = \mathbf{V}_p\mathbf{S}_p^{-1}\mathbf{U}_p^T\mathbf{d}. \tag{4.79}$$

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

$$\mathbf{U}_p^T\mathbf{d} = \begin{bmatrix} (\mathbf{U}_{\cdot,1})^T\mathbf{d} \\ (\mathbf{U}_{\cdot,2})^T\mathbf{d} \\ \vdots \\ (\mathbf{U}_{\cdot,p})^T\mathbf{d} \end{bmatrix}. \tag{4.80}$$

When we left-multiply $\mathbf{S}_p^{-1}$ times (4.80), we obtain

$$\mathbf{S}_p^{-1}\mathbf{U}_p^T\mathbf{d} = \begin{bmatrix} \dfrac{(\mathbf{U}_{\cdot,1})^T\mathbf{d}}{s_1} \\ \dfrac{(\mathbf{U}_{\cdot,2})^T\mathbf{d}}{s_2} \\ \vdots \\ \dfrac{(\mathbf{U}_{\cdot,p})^T\mathbf{d}}{s_p} \end{bmatrix}. \tag{4.81}$$

Finally, when we left-multiply $\mathbf{V}_p$ times (4.81), we obtain a linear combination of the columns of $\mathbf{V}_p$ that can be written as

$$\mathbf{m}_\dagger = \mathbf{V}_p\mathbf{S}_p^{-1}\mathbf{U}_p^T\mathbf{d} = \sum_{i=1}^{p} \frac{\mathbf{U}_{\cdot,i}^T\mathbf{d}}{s_i}\mathbf{V}_{\cdot,i}. \tag{4.82}$$

In the presence of random noise, $\mathbf{d}$ will generally have a nonzero projection onto each of the directions specified by the columns of $\mathbf{U}$. The presence of a very small s_i in the denominator of (4.82) can thus give us a very large coefficient for the corresponding model space basis vector $\mathbf{V}_{\cdot,i}$, and these basis vectors can dominate the solution. In the worst case, the generalized inverse solution is just a noise amplifier, and the answer is practically useless. A measure of the instability of the solution is the **condition number**. Note that the condition number considered here for an m by n matrix is a generalization of the condition number for an n by n matrix in (A.109), and that the two formulations are equivalent when $m = n$.

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

$$\mathbf{m}_\dagger - \mathbf{m}'_\dagger = \mathbf{G}^\dagger(\mathbf{d} - \mathbf{d}') \tag{4.83}$$

and

$$\|\mathbf{m}_\dagger - \mathbf{m}'_\dagger\|_2 \leq \|\mathbf{G}^\dagger\|_2\|\mathbf{d} - \mathbf{d}'\|_2. \tag{4.84}$$

From (4.82), it is clear that the largest difference in the inverse models will occur when $\mathbf{d} - \mathbf{d}'$ is in the direction $\mathbf{U}_{\cdot,p}$. If

$$\mathbf{d} - \mathbf{d}' = \alpha \mathbf{U}_{\cdot,p} \tag{4.85}$$

then

$$\|\mathbf{d} - \mathbf{d}'\|_2 = \alpha. \tag{4.86}$$

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

$$\mathbf{m}_\dagger - \mathbf{m}'_\dagger = \frac{\alpha}{s_p} \mathbf{V}_{\cdot,p} \tag{4.87}$$

with

$$\|\mathbf{m}_\dagger - \mathbf{m}'_\dagger\|_2 = \frac{\alpha}{s_p}. \tag{4.88}$$

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

$$\|\mathbf{m}_\dagger - \mathbf{m}'_\dagger\|_2 \leq \frac{1}{s_p} \|\mathbf{d} - \mathbf{d}'\|_2. \tag{4.89}$$

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

$$\|\mathbf{m}_\dagger\|_2 \geq \frac{1}{s_1} \|\mathbf{d}\|_2. \tag{4.90}$$

Combining these inequalities, we obtain

$$\frac{\|\mathbf{m}_\dagger - \mathbf{m}'_\dagger\|_2}{\|\mathbf{m}_\dagger\|_2} \leq \frac{s_1}{s_p} \frac{\|\mathbf{d} - \mathbf{d}'\|_2}{\|\mathbf{d}\|_2}. \tag{4.91}$$

The bound (4.91) is applicable to pseudoinverse solutions, regardless of what value of p we use. If we decrease p and thus eliminate model space vectors associated with small singular values, the solution becomes more stable. However, this stability comes at the expense of reducing the dimension of the subspace of R^n where the solution lies. As a result, the model resolution matrix for the stabilized solution obtained by decreasing p becomes less like the identity matrix, and the fit to the data worsens.

The condition number of $\mathbf{G}$ is the coefficient in (4.91)

$$\text{cond}(\mathbf{G}) = \frac{s_1}{s_k} \tag{4.92}$$

where $k = \min(m, n)$. The MATLAB command **cond** can be used to compute (4.92). If $\mathbf{G}$ is of full rank, and we use all of the singular values in the pseudoinverse solution ($p = k$), then

the condition number is exactly (4.92). If **G** is of less than full rank, then the condition number is effectively infinite. As with the model and data resolution matrices [(4.64) and (4.78)], the condition number is a property of **G** that can be computed in the design phase of an experiment before any data are collected.

A condition that insures solution stability and arises naturally from consideration of (4.82) is the **discrete Picard condition** [59]. The discrete Picard condition is satisfied when the dot products of the columns of **U** and the data vector decay to zero more quickly than the singular values, s_i. Under this condition, we should not see instability due to small singular values. The discrete Picard condition can be assessed by plotting the ratios of $\mathbf{U}_i^T\mathbf{d}$ to s_i across the singular value spectrum.

If the discrete Picard condition is not satisfied, we may still be able to recover a useful model by truncating (4.82) at some highest term $p' < p$, to produce a **truncated SVD** or **TSVD** solution. One way to decide when to truncate (4.82) in this case is to apply the **discrepancy principle**. In the discrepancy principle, we pick the smallest value of p' so that the model fits the data to some tolerance based on the length of the residual vector

$$\|\mathbf{G}_w\mathbf{m} - \mathbf{d}_w\|_2 \leq \delta \tag{4.93}$$

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

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

A TSVD solution will not fit the data as well as solutions that do include the model space basis vectors with small singular values. Perhaps surprisingly, this is an example of the general approach for solving ill-posed problems with noise. If we fit the data vector exactly or nearly exactly, we are in fact **overfitting** the data and may be letting the noise control major features of the model.

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

4.4 AN EXAMPLE OF A RANK-DEFICIENT PROBLEM

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

■ **Example 4.1** Using the SVD, let us revisit the straight ray path tomography example that we considered earlier in Example 1.6 (see Figure 4.2). We introduced a rank-deficient system in which we were constraining a nine-parameter slowness model with eight travel time observations:

$$\mathbf{Gm} = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ \sqrt{2} & 0 & 0 & 0 & \sqrt{2} & 0 & 0 & 0 & \sqrt{2} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sqrt{2} \end{bmatrix} \begin{bmatrix} s_{11} \\ s_{12} \\ s_{13} \\ s_{21} \\ s_{22} \\ s_{23} \\ s_{31} \\ s_{32} \\ s_{33} \end{bmatrix} = \begin{bmatrix} t_1 \\ t_2 \\ t_3 \\ t_4 \\ t_5 \\ t_6 \\ t_7 \\ t_8 \end{bmatrix}. \tag{4.94}$$

The eight singular values of **G** are, numerically evaluated,

$$\text{diag}(\mathbf{S}) = \begin{bmatrix} 3.180 \\ 2.000 \\ 1.732 \\ 1.732 \\ 1.732 \\ 1.607 \\ 0.553 \\ 4.230 \times 10^{-16} \end{bmatrix}. \tag{4.95}$$

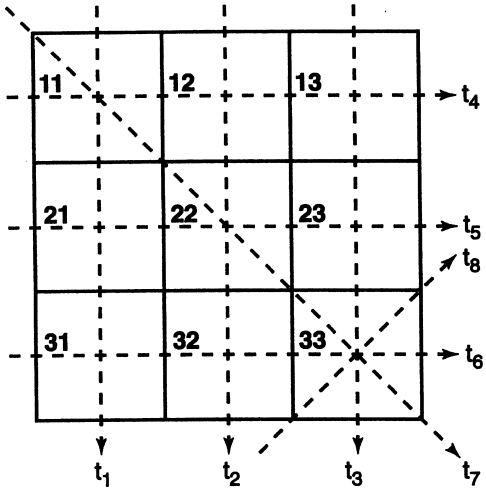

Figure 4.2 A simple tomography example (revisited).

The smallest singular value, s_8, is nonzero in numerical evaluation only because of round-off error in the SVD algorithm. It is zero in an analytical solution. s_8 is clearly effectively zero relative to the other singular values. The ratio of the largest to smallest nonzero singular values is about 6, and the generalized inverse solution (4.82) will thus be stable in the presence of noise. Because rank(**G**) $= p = 7$, the problem is rank-deficient. The model null space, $N(\mathbf{G})$, is spanned by the two orthonormal vectors that form the eighth and ninth columns of **V**:

$$\mathbf{V}_0 = [\mathbf{V}_{.,8}, \mathbf{V}_{.,9}] = \begin{bmatrix} -0.136 & -0.385 \\ 0.385 & -0.136 \\ -0.249 & 0.521 \\ -0.385 & 0.136 \\ 0.136 & 0.385 \\ 0.249 & -0.521 \\ 0.521 & 0.249 \\ -0.521 & -0.249 \\ 0.000 & 0.000 \end{bmatrix}. \tag{4.96}$$

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

$$\text{reshape}(\mathbf{V}_{.,8}, 3, 3) = \begin{bmatrix} -0.136 & -0.385 & 0.521 \\ 0.385 & 0.136 & -0.521 \\ -0.249 & 0.249 & 0.000 \end{bmatrix} \tag{4.97}$$

$$\text{reshape}(\mathbf{V}_{.,9}, 3, 3) = \begin{bmatrix} -0.385 & 0.136 & 0.249 \\ -0.136 & 0.385 & -0.249 \\ 0.521 & -0.521 & 0.000 \end{bmatrix}. \tag{4.98}$$

See Figures 4.3 and 4.4.

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

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

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

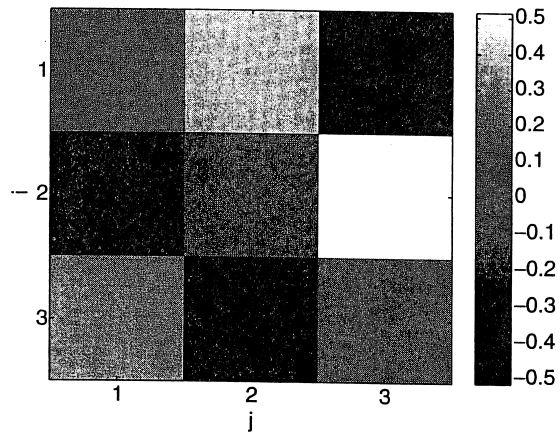

Figure 4.3 Image of the null space model $\mathbf{V}_{.,8}$.

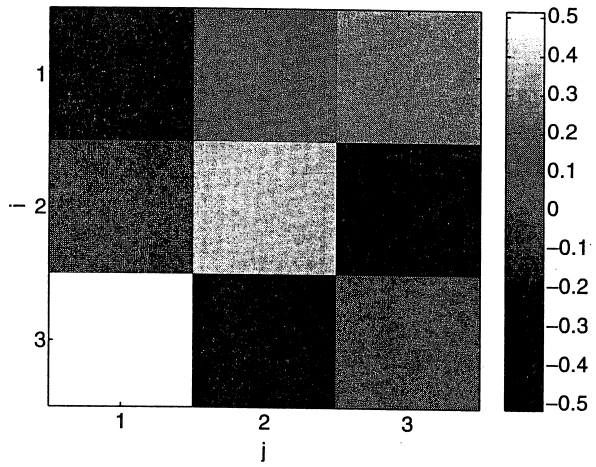

Figure 4.4 Image of the null space model $\mathbf{V}_{.,9}$.

clearly affect the predicted data, and any vector in the model null space must have a value of 0 in m_9.

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

$$\mathbf{U}_0 = \mathbf{U}_{.,8} = \begin{bmatrix} -0.408 \\ -0.408 \\ -0.408 \\ 0.408 \\ 0.408 \\ 0.408 \\ 0.000 \\ 0.000 \end{bmatrix}. \tag{4.99}$$

Recall that, even for noise-free data, we will not recover a general $\mathbf{m}_{\text{true}}$ in a rank-deficient problem using (4.22), but will instead recover a "smeared" model $\mathbf{R}_{\text{m}}\mathbf{m}_{\text{true}}$. Because $\mathbf{R}_{\text{m}}$ for a rank-deficient problem is itself rank-deficient, this smearing is irreversible. The full $\mathbf{R}_{\text{m}}$ matrix dictates precisely how this smearing occurs. The elements of $\mathbf{R}_{\text{m}}$ for this example are shown in Figure 4.5.

Examining the entire n by n model resolution matrix becomes cumbersome in large problems. The n diagonal elements of $\mathbf{R}_{\text{m}}$ can be examined more easily to provide basic information on how well recovered each model parameter will be. The reshaped diagonal of $\mathbf{R}_{\text{m}}$ from Figure 4.5 is

$$\text{reshape}(\text{diag}(\mathbf{R}_{\text{m}}), 3, 3) = \begin{bmatrix} 0.833 & 0.833 & 0.667 \\ 0.833 & 0.833 & 0.667 \\ 0.667 & 0.667 & 1.000 \end{bmatrix}. \tag{4.100}$$

See Figure 4.6.

Figure 4.6 and (4.100) tell us that m_9 is perfectly resolved, but that we can expect loss of resolution (and hence smearing of the true model into other blocks) for all of the other solution parameters.

We next assess the smoothing effects of limited model resolution by performing a resolution test using synthetic data for a test model of interest. The resolution test assesses the recovery of the test model by examining the corresponding inverse solution. One synthetic model that is commonly used in resolution tests is uniformly zero except for a single perturbed model element. Examining the inverse recovery using data generated by such a model is commonly referred to as a **spike** or **impulse** resolution test. For this example, consider the spike model consisting of the vector with its fifth element equal to one and zeros elsewhere. This model is shown in Figure 4.7. Forward modeling gives the predicted data set for $\mathbf{m}_{\text{test}}$:

$$\mathbf{d}_{\text{test}} = \mathbf{G}\mathbf{m}_{\text{test}} = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 0 & 0 & \sqrt{2} & 0 \end{bmatrix}^T, \tag{4.101}$$

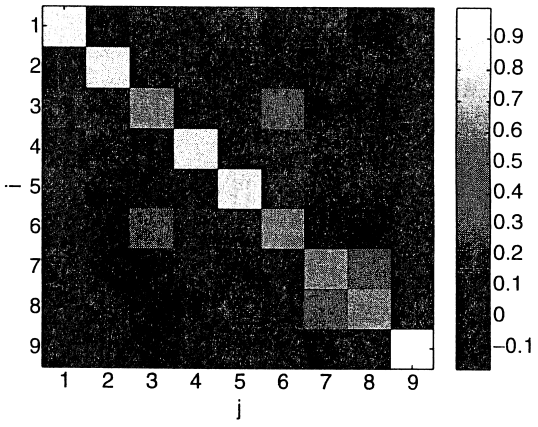

Figure 4.5 The model resolution matrix elements, $\text{R}_{\text{m}i,j}$ for the generalized inverse solution.

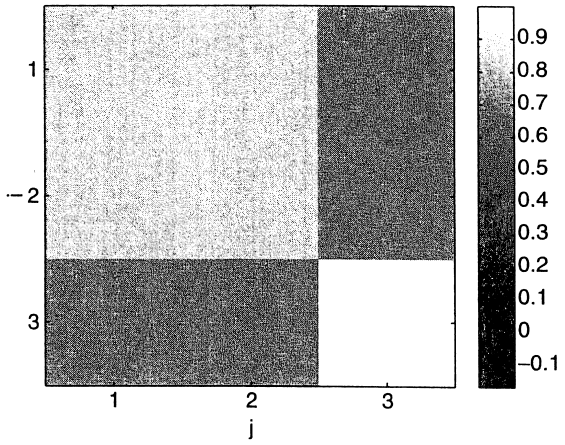

Figure 4.6 Diagonal elements of the resolution matrix plotted in their respective geometric model locations.

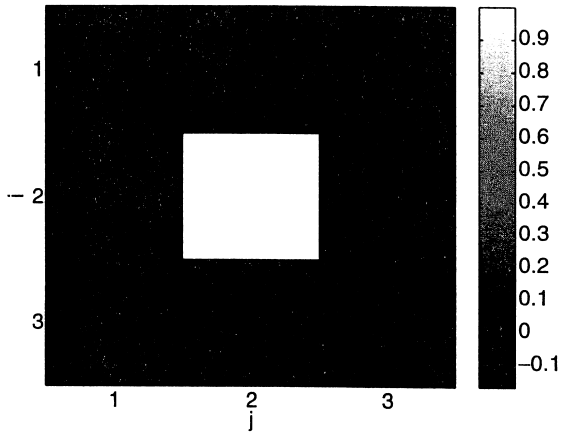

Figure 4.7 A spike test model.

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

$$\text{reshape}(\mathbf{m}_\dagger, 3, 3) = \begin{bmatrix} 0.167 & 0 & -0.167 \\ 0 & 0.833 & 0.167 \\ -0.167 & 0.167 & 0.000 \end{bmatrix}. \tag{4.102}$$

See Figure 4.8. The recovered model in this spike test shows that limited resolution causes information about the central block slowness to smear into some, but not all, of the adjacent blocks *even for noise-free data*, with the exact form of the smearing dictated by the model resolution matrix.

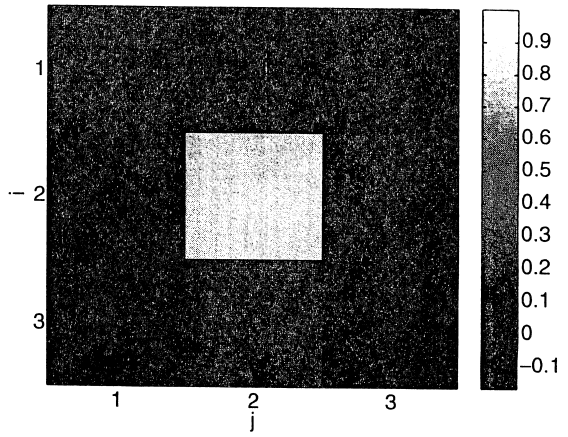

Figure 4.8 The generalized inverse solution for the noise-free spike test.

It is important to reemphasize that the ability to recover the true model in practice is affected both by the bias caused by limited resolution, which is a characteristic of the matrix $\mathbf{G}$ and hence applies even to noise-free data, and by the mapping of any data noise into the model parameters. In practice, the error due to noise in the data can also be very significant. ■

4.5 DISCRETE ILL-POSED PROBLEMS

In many problems the singular values decay gradually toward zero and do not show an obvious jump between nonzero and zero singular values. This happens frequently when we discretize Fredholm integral equations of the first kind as in Chapter 3. In particular, as we increase the number of points in the discretization, we typically find that $\mathbf{G}$ becomes more and more poorly conditioned. Discrete inverse problems such as these cannot formally be called ill-posed, because the condition number remains finite although very large. We will refer to these as **discrete ill-posed problems**.

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

In addition to the general pattern of singular values which decay to 0, discrete ill-posed problems are typically characterized by differences in the character of the singular vectors $\mathbf{V}_{\cdot,j}$ [59]. For large singular values, the corresponding singular vectors are smooth, while for smaller singular values, the corresponding singular vectors may be highly oscillatory. These oscillations become apparent in the generalized inverse solution as more singular values and vectors are included.

When we attempt to solve such a problem with the TSVD, it is difficult to decide where to truncate (4.82). If we truncate the sum too early, then our solution will lack details that correspond to model vectors associated with the smaller singular values. If we include too many of the terms, then the solution becomes unstable in the presence of noise. In particular we can expect that more oscillatory components of the generalized inverse solution may be most strongly affected by noise [59]. Regularization is required to address this fundamental issue.

■ **Example 4.2** Consider an inverse problem where we have a physical process (e.g., seismic ground motion) recorded by a linear instrument of limited bandwidth (e.g., a vertical seismometer). The response of such a device is commonly characterized by an **instrument impulse response**, which is the response of the system to a delta function input. Consider the instrument impulse response

$$g(t) = \begin{cases} g_0 t e^{-t/T_0} & (t \geq 0) \\ 0 & (t < 0) \end{cases} \tag{4.103}$$

Figure 4.9 shows the displacement response of a critically damped seismometer with a characteristic time constant T_0 to a unit area (1 $\text{m/s}^2 \cdot \text{s}$) impulsive ground acceleration input, where g_0 is a gain constant. Assuming that the displacement of the seismometer is electronically converted to output volts, we conveniently choose g_0 to be $T_0 e^{-1}$ $\text{V/m} \cdot \text{s}$ to produce a 1 V maximum output value for the impulse response, and $T_0 = 10$ s.

The seismometer output (or seismogram), $v(t)$, is a voltage record given by the convolution of the true ground acceleration, $m_{\text{true}}(t)$, with (4.103):

$$v(t) = \int_{-\infty}^{\infty} g(t - \tau)\, m_{\text{true}}(\tau)\, d\tau. \tag{4.104}$$

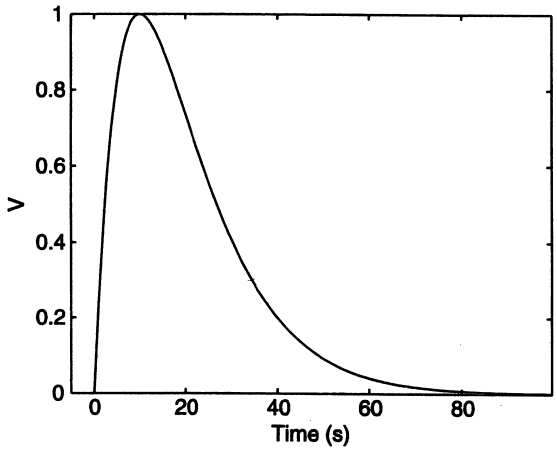

Figure 4.9 Example instrument response; seismometer output voltage in response to a unit area ground acceleration impulse.

We are interested in the inverse **deconvolution** operation that will remove the smoothing effect of $g(t)$ in (4.104) and allow us to recover the true ground acceleration m_{true}.

Discretizing (4.104) using the midpoint rule with a time interval Δt, we obtain

$$\mathbf{d} = \mathbf{G}\mathbf{m} \tag{4.105}$$

where

$$G_{i,j} = \begin{cases} (t_i - t_j)e^{-(t_i - t_j)/T_0}\Delta t & (t_j \geq t_i) \\ 0 & (t_j < t_i) \end{cases}. \tag{4.106}$$

The rows of $\mathbf{G}$ in (4.106) are time reversed, and the columns of $\mathbf{G}$ are non-time-reversed, sampled versions of the impulse response $g(t)$, lagged by i and j, respectively. Using a time interval of $[-5, 100]$ s, outside of which (4.103) and any model, $\mathbf{m}$, of interest are assumed to be very small or zero, and a discretization interval of $\Delta t = 0.5$ s, we obtain a discretized m by n system matrix $\mathbf{G}$ with $m = n = 210$.

The singular values of $\mathbf{G}$ are all nonzero and range from about 25.3 to 0.017, giving a condition number of ≈ 1480, and showing that this discretization has produced a discrete system that is mildly ill-posed. See Figure 4.10. However, adding noise at the level of 1 part in 1000 will be sufficient to make the generalized inverse solution unstable. The reason for the large condition number can be seen by examining successive rows of $\mathbf{G}$, which are nearly but not quite identical, with

$$\frac{\mathbf{G}_{i,\cdot}\mathbf{G}_{i+1,\cdot}^T}{\|\mathbf{G}_{i,\cdot}\|_2\|\mathbf{G}_{i+1,\cdot}\|_2} \approx 0.999. \tag{4.107}$$

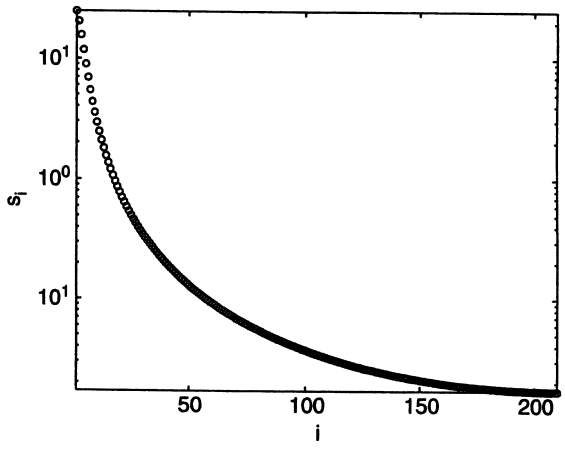

Figure 4.10 Singular values for the discretized convolution matrix.

Now, consider a true ground acceleration signal that consists of two acceleration pulses with widths of $\sigma = 2$ s, centered at $t = 8$ s and $t = 25$ s (Figure 4.11):

$$m_{\text{true}}(t) = e^{-(t-8)^2/(2\sigma^2)} + 0.5e^{-(t-25)^2/(2\sigma^2)}. \tag{4.108}$$

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

$$\mathbf{d}_{\text{true}} = \mathbf{G}\mathbf{m}_{\text{true}} \tag{4.109}$$

and a second data set with independent $N(0, (0.05\text{ V})^2)$ noise added. The data set with noise is shown in Figure 4.12.

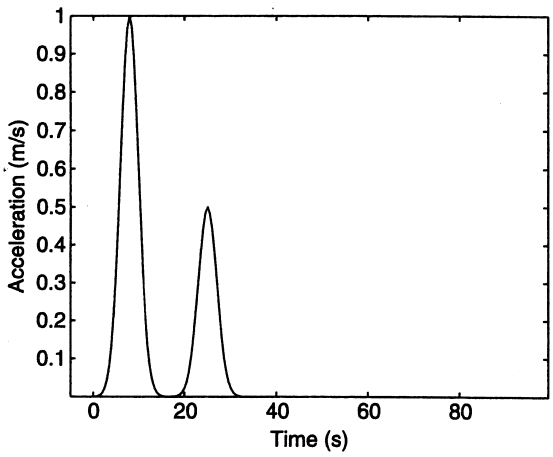

Figure 4.11 The true model.

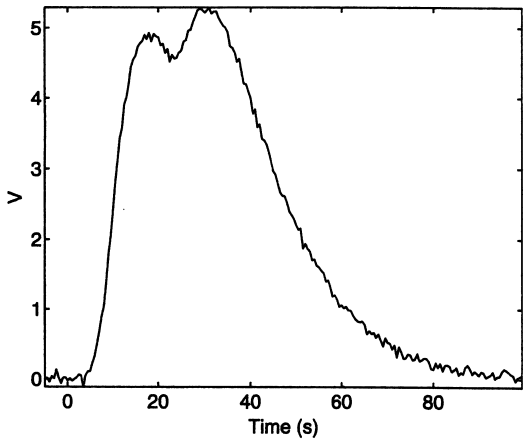

Figure 4.12 Predicted data from the true model plus independent $N(0, (0.05\text{ V})^2)$ noise.

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

$$\mathbf{m} = \mathbf{V}\mathbf{S}^{-1}\mathbf{U}^T\mathbf{d}_{\text{true}} \tag{4.110}$$

is shown in Figure 4.13. The model fits its noiseless data vector, $\mathbf{d}_{\text{true}}$, perfectly, and is essentially identical to the true model.

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

$$\mathbf{m} = \mathbf{V}\mathbf{S}^{-1}\mathbf{U}^T(\mathbf{d}_{\text{true}} + \boldsymbol{\eta}) \tag{4.111}$$

is shown in Figure 4.14.

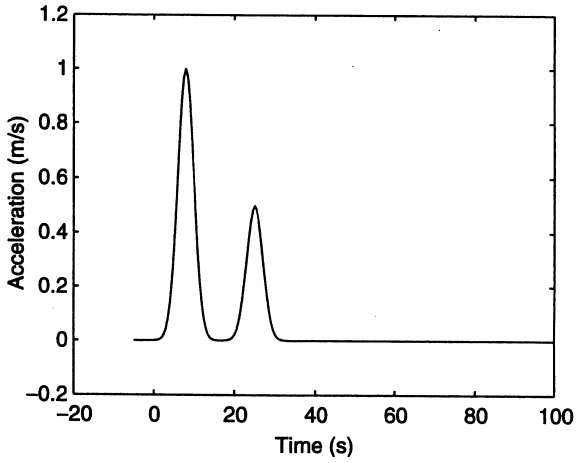

Figure 4.13 Generalized inverse solution using all 210 singular values for the noise-free data.

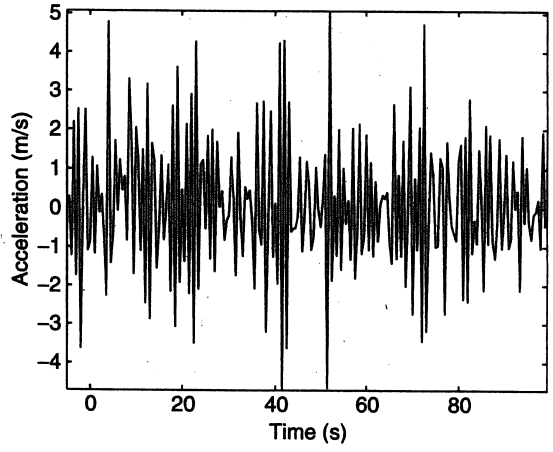

Figure 4.14 Generalized inverse solution using all 210 singular values for the noisy data of Figure 4.12.

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

Can a useful model be recovered by the truncated SVD? Using the discrepancy principle as our guide and selecting a range of solutions with varying p', we can in fact obtain an appropriate solution when we keep $p' = 26$ columns in $\mathbf{V}$. See Figure 4.15.

Essential features of the true model are resolved in the solution of Figure 4.15, but the solution technique introduces oscillations and loss of resolution. Specifically, we see that the widths of the inferred pulses are somewhat wider, and the inferred amplitudes somewhat less, than those of the true ground acceleration. These effects are both hallmarks of limited resolution, as characterized by a nonidentity model resolution matrix. An image of the model resolution matrix in Figure 4.16 shows a finite-width central band and oscillatory side lobes.

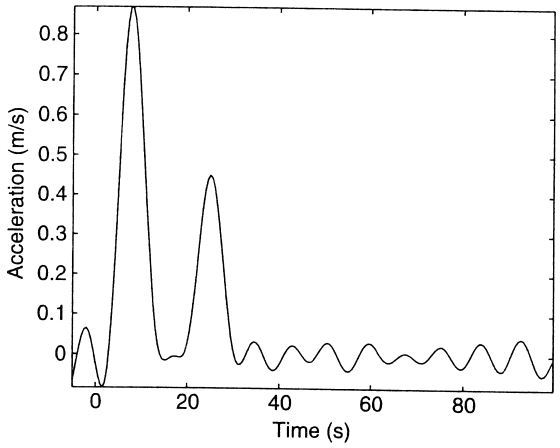

Figure 4.15 Solution using the 26 largest singular values for noisy data shown in Figure 4.12.

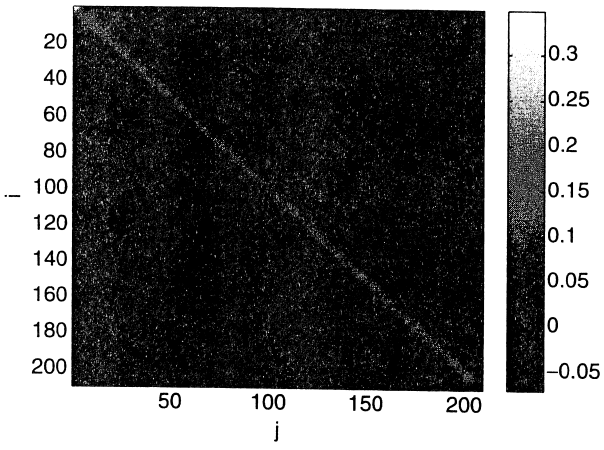

Figure 4.16 The model resolution matrix elements $R_{m_{i,j}}$ for the truncated SVD solution including the 26 largest singular values.

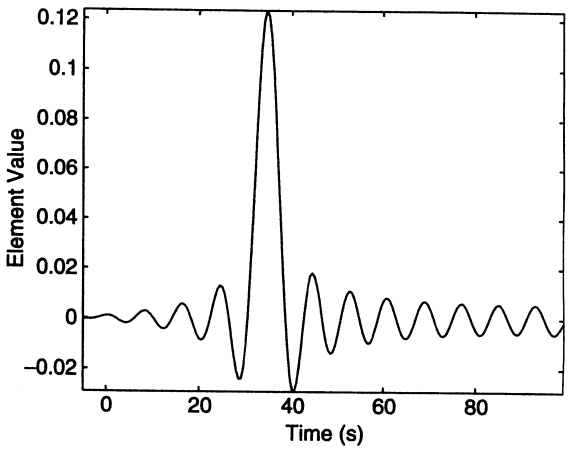

Figure 4.17 A column from the model resolution matrix $\mathbf{R}_m$ for the truncated SVD solution including the 26 largest singular values.

A typical (80th) column of the model resolution matrix quantifies the smearing of the true model into the recovered model for the choice of the $p = 26$ inverse operator. See Figure 4.17. The smoothing is over a characteristic width of about 5 seconds, which is why our recovered model, although it does a decent job of rejecting noise, underestimates the amplitude and narrowness of the true model. The oscillatory behavior of the resolution matrix is attributable to our abrupt truncation of the model space. Each of the n columns of $\mathbf{V}$ is an oscillatory model basis function, with $j - 1$ zero crossings, where j is the column number.

When we truncate (4.82) after 26 terms to stabilize the inverse solution, we place a limit on the most oscillatory model space basis vectors that we will allow in our solution. This truncation gives us a model, and model resolution, that contain oscillatory structure with around $p - 1 = 25$ zero crossings. We will examine this perspective further in Chapter 8, where issues associated with highly oscillatory model basis functions will be revisited in the context of Fourier theory. ■

■ **Example 4.3** Recall the Shaw problem from Example 3.2. The MATLAB Regularization Tools contains a routine **shaw** that computes the $\mathbf{G}$ matrix and an example model and data for this problem [58]. We computed the $\mathbf{G}$ matrix for $n = 20$ and examined the singular values. Figure 4.18 shows the singular value spectrum, which is characterized by very rapid singular value decay to zero in an exponential fashion.

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

The 18th column of $\mathbf{V}$, which corresponds to the smallest nonzero singular value, is shown in Figure 4.19. In contrast, the first column of $\mathbf{V}$, which corresponds to the largest singular value, represents a much smoother model. See Figure 4.20. This behavior is typical of discrete ill-posed problems.

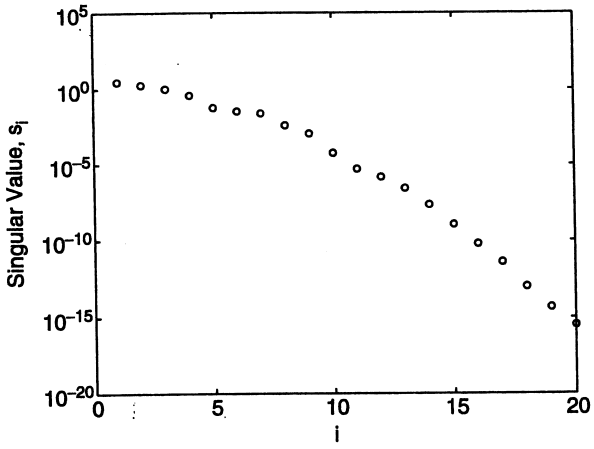

Figure 4.18 Singular values of G for the Shaw example ($n = 20$).

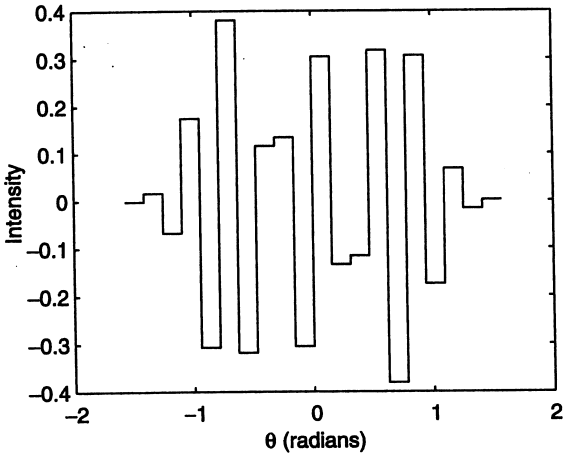

Figure 4.19 $\mathbf{V}_{\cdot,18}$.

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

$$m_i = \begin{cases} 1 & i = 10 \\ 0 & \text{otherwise.} \end{cases} \tag{4.112}$$

See Figure 4.21. We use the model to obtain noise-free data and then apply the generalized inverse (4.22) with various values of p to obtain TSVD inverse solutions. The corresponding data are shown in Figure 4.22. If we compute the generalized inverse from these data using MATLAB's double-precision algorithms, we get fairly good recovery of (4.112). See Figure 4.23.

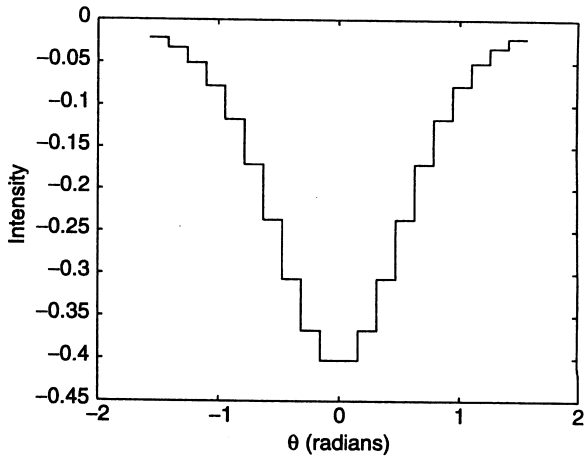

Figure 4.20 $\mathbf{V}_{\cdot,1}$.

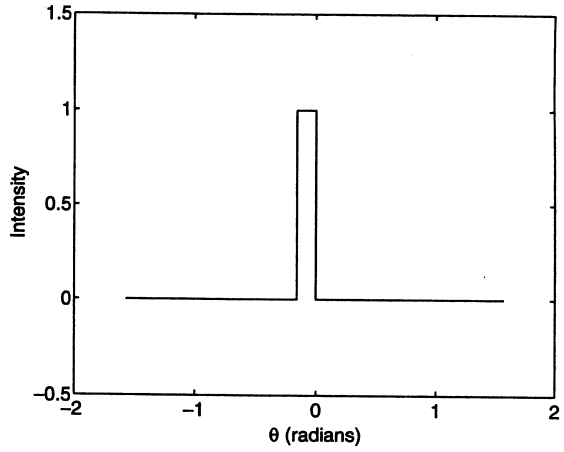

Figure 4.21 The spike model.

However, if we add a very small amount of noise to the data in Figure 4.22, things change dramatically. Adding $N(0,\ (10^{-6})^2)$ noise to the data of Figure 4.22 and computing a generalized inverse solution using $p = 18$ produces the wild solution of Figure 4.24, which bears no resemblance to the true model. Note that the vertical scale in Figure 4.24 is multiplied by 10^6! Furthermore, the solution involves negative intensities, which are not physically possible. This inverse solution is even more sensitive to noise than that of the previous deconvolution example, to the extent that even noise on the order of 1 part in 10^6 will destabilize the solution.

Next, we consider what happens when we use only the 10 largest singular values and their corresponding model space vectors to construct a TSVD solution. Figure 4.25 shows the solution using 10 singular values with the same noise as Figure 4.24. Because we have

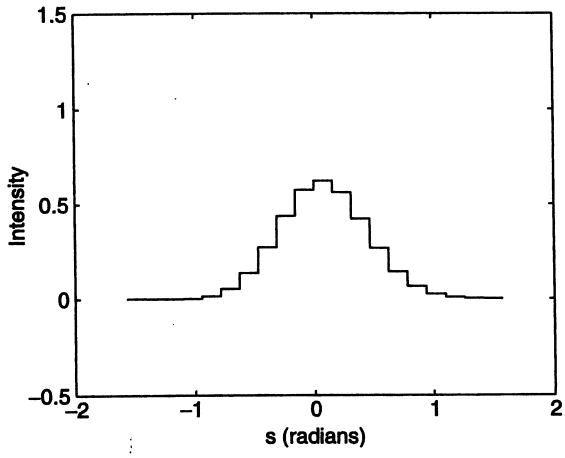

Figure 4.22 Noise-free data predicted for the spike model.

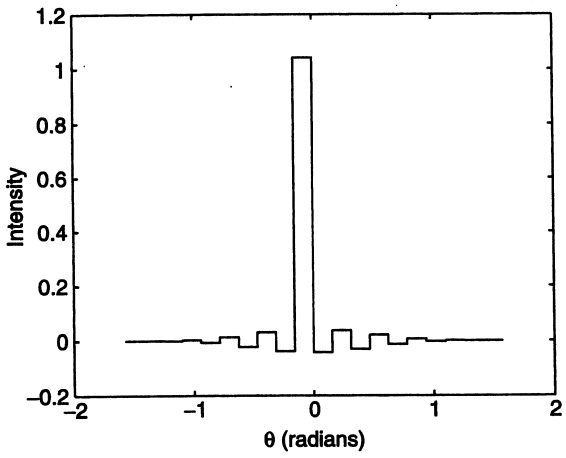

Figure 4.23 The generalized inverse solution for the spike model, no noise.

cut off a number of singular values, we have reduced the model resolution. The inverse solution is smeared out, but it is still possible to conclude that there is some significant spike-like feature near $\theta = 0$. In contrast to the situation that we observed in Figure 4.24, the model recovery is now not visibly affected by the noise. The trade-off is that we must now accept the imperfect resolution of this solution and its attendant bias towards smoother models.

What happens if we discretize the problem with a larger number of intervals? Figure 4.26 shows the singular values for the $\mathbf{G}$ matrix with $n = 100$ intervals. The first 20 or so singular values are apparently nonzero, whereas the last 80 or so singular values are effectively zero.

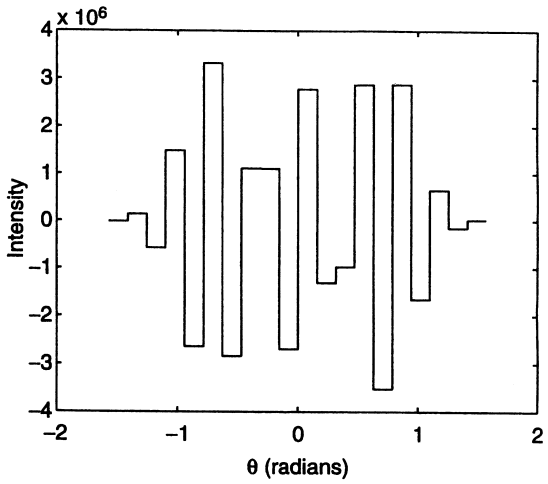

Figure 4.24 Recovery of the spike model with noise ($p = 18$).

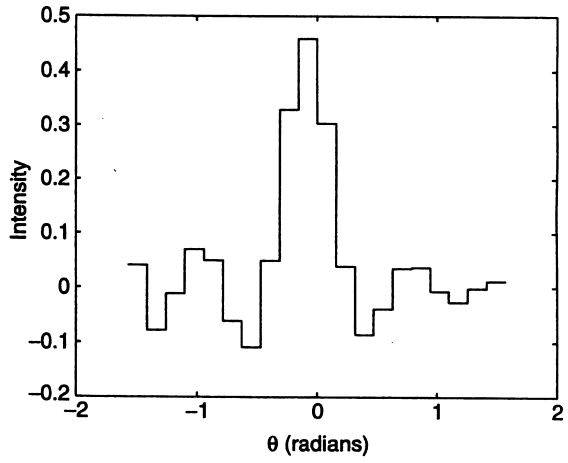

Figure 4.25 Recovery of the spike model with noise ($p = 10$).

Figure 4.27 shows the inverse solution for the spike model with $n = 100$ and $p = 10$. This solution is very similar to the solution shown in Figure 4.25. In general, discretizing over more intervals does not hurt as long as the solution is appropriately regularized and the additional computation time is acceptable.

What about a smaller number of intervals? Figure 4.28 shows the singular values of the $\mathbf{G}$ matrix with $n = 6$. In this case there are no terribly small singular values. However, with only six elements in this coarse model vector, we cannot hope to resolve the details of a source intensity distribution with a complex intensity structure. This is an example of regularization by discretization.

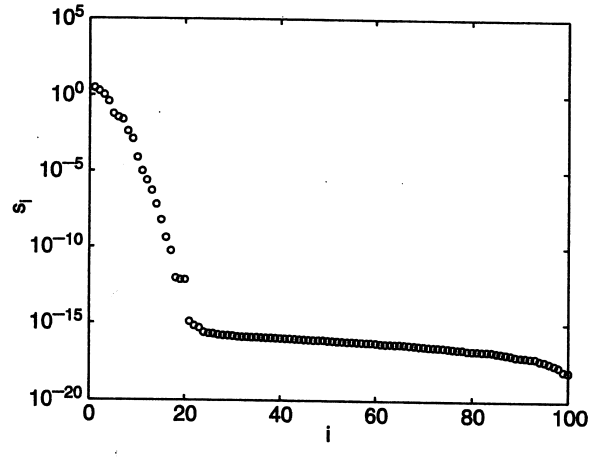

Figure 4.26 Singular values of **G** for the Shaw example ($n = 100$).

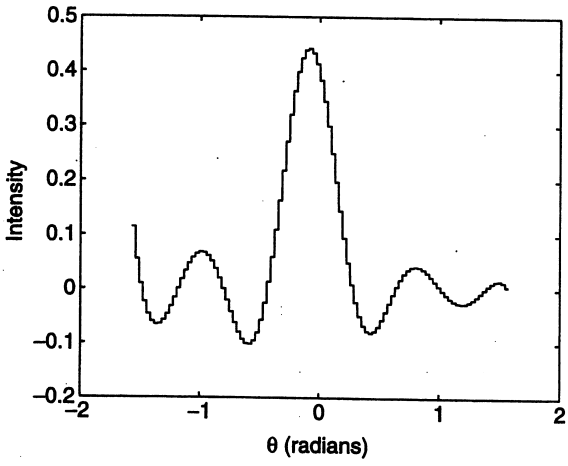

Figure 4.27 Recovery of the spike model with noise ($n = 100$, $p = 10$).

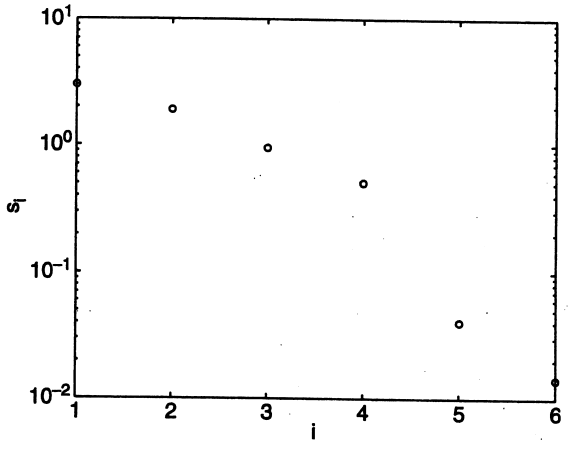

Figure 4.28 Singular values of **G** for the Shaw example ($n = 6$).

This example demonstrates the dilemma posed by small singular values. If we include the small singular values, then our inverse solution becomes unstable in the presence of data noise. If we do not include the smaller singular values, then our solution is not as sensitive to noise in the data, but we lose resolution and introduce bias. ■

4.6 EXERCISES

4.1 The pseudoinverse of a matrix $\mathbf{G}$ was originally defined by Moore and Penrose as the unique matrix $\mathbf{G}^\dagger$ with the properties

(a) $\mathbf{G}\mathbf{G}^\dagger\mathbf{G} = \mathbf{G}$.
(b) $\mathbf{G}^\dagger\mathbf{G}\mathbf{G}^\dagger = \mathbf{G}^\dagger$.
(c) $(\mathbf{G}\mathbf{G}^\dagger)^T = \mathbf{G}\mathbf{G}^\dagger$.
(d) $(\mathbf{G}^\dagger\mathbf{G})^T = \mathbf{G}^\dagger\mathbf{G}$.

Show that $\mathbf{G}^\dagger$ as given by (4.20) satisfies these four properties.

4.2 Another resolution test commonly performed in tomography studies is a **checkerboard test**, which consists of using a test model composed of alternating positive and negative perturbations. Perform a checkerboard test on the tomography problem in Example 4.1. Evaluate the difference between the true model and the recovered model, and interpret the pattern of differences.

4.3 A large north–south by east–west oriented, nearly square plan view, sandstone quarry block (16 m by 16 m) with a bulk P-wave seismic velocity of approximately 3000 m/s is suspected of harboring higher-velocity dinosaur remains. An ultrasonic P-wave travel-time tomography scan is performed in a horizontal plane bisecting the boulder, producing a data set consisting of 16 E→W, 16 N→S, 31 SW→NE, and 31 NW→SE travel times. See Figure 4.29. Each travel-time measurement has statistically independent errors with estimated standard deviations of 15 μs.

The data files that you will need to load from your working directory into your MATLAB program are **rowscan, colscan, diag1scan, diag2scan** containing the travel-time data, and **std** containing the standard deviations of the data measurements. The travel time contribution from a uniform background model (velocity of 3000 m/s) has been subtracted from each travel-time measurement for you, so you will be solving for perturbations from a uniform slowness model of 3000 m/s. The row format of each data file is (x_1, y_1, x_2, y_2, t) where the starting-point coordinate of each shot is (x_1, y_1), the end-point coordinate is (x_2, y_2), and the travel time along a ray path between the start and end points is a path integral (in seconds)

$$t = \int_l s(\mathbf{x})dl, \tag{4.113}$$

where s is the slowness along the path, l, between source and receiving points, and Δl_{block} is the length of the ray in each block.

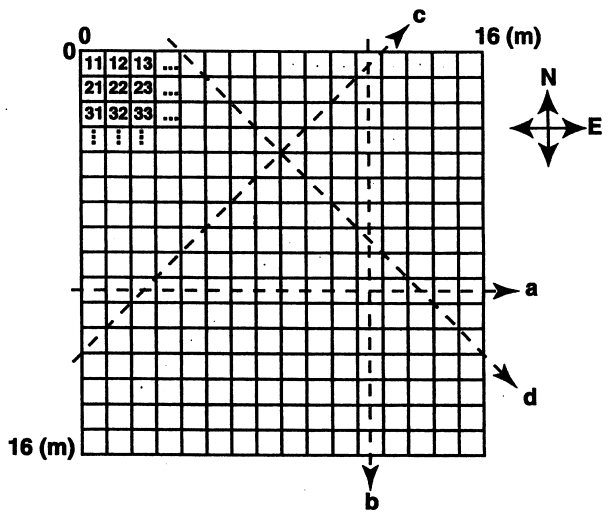

Figure 4.29 Tomography exercise, showing block discretization, block numbering convention, and representative ray paths going east–west (a), north–south (b), southwest–northeast (c), and northwest–southeast (d).

Parameterize the slowness structure in the plane of the survey by dividing the boulder into a 16 by 16 grid of 256 1-m-square, N by E blocks to construct a linear system for the problem. See Figure 4.29. Assume that the ray paths through each homogeneous block can be well approximated by straight lines, so that the travel time expression is

$$t = \int_l s(\mathbf{x})\, dl \tag{4.114}$$

$$= \sum_{\text{blocks}} s_{\text{block}} \cdot \Delta l_{\text{block}} \tag{4.115}$$

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

Use the SVD to find a minimum-length/least-squares solution, $\mathbf{m}_\dagger$, for the 256 block slowness perturbations which fit the data as exactly as possible. Perform two inversions:

(A) Using the row and column scans only, and
(B) Using the complete data set.

For each inversion:

(a) State and discuss the significance of the elements and dimensions of the data and model null spaces.
(b) Note if there any model parameters that have perfect resolution.
(c) Note the condition number of your G matrix relating the data and model.
(d) Note the condition number of your generalized inverse matrix.

(e) Produce a 16 by 16 element contour or other plot of your slowness perturbation model, displaying the maximum and minimum slowness perturbations in the title of each plot. Anything in there? If so, how fast or slow is it (in m/s)?
(f) Show the model resolution by contouring or otherwise displaying the 256 diagonal elements of the model resolution matrix, reshaped into an appropriate 16 by 16 grid.
(g) Construct and contour or otherwise display a nonzero model which fits the trivial data set $\mathbf{d} = \mathbf{0}$ exactly.
(h) Describe how one could use solutions of the type discussed in (g) to demonstrate that very rough models exist which will fit any data set just as well as a generalized inverse model. Show one such wild model.

4.4 Find the singular value decomposition of the $\mathbf{G}$ matrix from Exercise 3.1. Taking into account the fact that the measured data are only accurate to about four digits, use the truncated SVD to compute a solution to this problem.

4.5 Revisiting Example 3.4, apply the generalized inverse to estimate the density of the Earth as a function of radius, using the given values of mass and moment of inertia. Obtain a density model composed of 20 spherical shells of equal thickness, and compare your results to a standard model.

4.7 NOTES AND FURTHER READING

The Moore–Penrose generalized inverse was independently discovered by Moore in 1920 and Penrose in 1955 [102, 125]. Penrose is generally credited with first showing that the SVD can be used to compute the generalized inverse [125]. Books that discuss the linear algebra of the generalized inverse in more detail include [10, 20].

There was significant early work on the SVD in the 19th century by Beltrami, Jordan, Sylvester, Schmidt, and Weyl [154]. However, the singular value decomposition in matrix form is typically credited to Eckart and Young [34]. Some books that discuss the properties of the SVD and prove its existence include [49, 101, 155]. Lanczos presents an alternative derivation of the SVD [89]. Algorithms for the computation of the SVD are discussed in [31, 49, 167]. Books that discuss the use of the SVD and truncated SVD in solving discrete linear inverse problems include [59, 100, 156].

Resolution tests with spike and checkerboard models as in Example 4.1 are very commonly used in practice. However, Leveque, Rivera, and Wittlinger discuss some serious problems with such resolution tests [93].

Matrices like those in Example 4.2 in which the elements along diagonals are constant are called **Toeplitz matrices** [69]. Specialized methods for regularization of problems involving Toeplitz matrices are available [60].

As we have seen, it is possible to effectively regularize the solution to a discretized version of a continuous inverse problem by selecting a coarse discretization. This approach is analyzed in [38]. However, in doing so we lose the ability to analyze the bias introduced by the regularization. In general, we prefer to user a fine discretization and then explicitly regularize the discretized problem.