

## CHAPTER 7: THE GENERALIZED INVERSE AND MEASURES OF QUALITY

### 7.1 Introduction

Thus far we have used the shifted eigenvalue problem to do singular-value decomposition for the system of equations  $\mathbf{G}\mathbf{m} = \mathbf{d}$ . That is, we have

$$\mathbf{G} = \mathbf{U} \quad \Lambda \quad \mathbf{V}^T \quad (6.60)$$

$N \times M \quad N \times N \quad N \times M \quad M \times M$

and also

$$\mathbf{G} = \mathbf{U}_P \quad \Lambda_P \quad \mathbf{V}_P^T \quad (6.69)$$

$N \times M \quad N \times P \quad P \times P \quad P \times M$

where  $\mathbf{U}$  is an  $N \times N$  orthogonal matrix, and where the  $i$ th column is given by the  $i$ th eigenvector  $\mathbf{u}_i$  which satisfies

$$\mathbf{G}\mathbf{G}^T\mathbf{u}_i = \eta_i^2\mathbf{u}_i \quad (6.29)$$

$\mathbf{V}$  is an  $M \times M$  orthogonal matrix, where the  $i$ th column is given by the  $i$ th eigenvector  $\mathbf{v}_i$  which satisfies

$$\mathbf{G}^T\mathbf{G}\mathbf{v}_i = \eta_i^2\mathbf{v}_i. \quad (6.21)$$

$\Lambda$  is an  $N \times M$  diagonal matrix with the singular values  $\lambda_i = \sqrt{\eta_i^2}$  along the diagonal.  $\mathbf{U}_P$ ,  $\Lambda_P$ , and  $\mathbf{V}_P$  are the subsets of  $\mathbf{U}$ ,  $\Lambda$ , and  $\mathbf{V}$ , respectively, associated with the  $P$  nonzero singular values,  $P \leq \min(N, M)$ .

We found four classes of problems for  $\mathbf{G}\mathbf{m} = \mathbf{d}$  based on  $P, N, M$ :

**Class I:**  $P = N = M$ ;  $\mathbf{G}^{-1}$  (mathematical) exists.

**Class II:**  $P = M < N$ ; least squares. Recall  $\mathbf{m}_{LS} = [\mathbf{G}^T\mathbf{G}]^{-1}\mathbf{G}^T\mathbf{d}$ .

**Class III:**  $P = N < M$ ; Minimum Length. Recall  $\mathbf{m}_{ML} = \langle \mathbf{m} \rangle + \mathbf{G}^T[\mathbf{G}\mathbf{G}^T]^{-1} \times [\mathbf{d} - \mathbf{G}\langle \mathbf{m} \rangle]$ .

**Class IV:**  $P < \min(N, M)$ ; at present, we have no way of obtaining an  $\mathbf{m}^{\text{est}}$ .

Thus, in this chapter we seek an inverse operator that has the following properties:

1. Reduces to  $\mathbf{G}^{-1}$  when  $P = N = M$ .
2. Reduces to  $[\mathbf{G}^T \mathbf{G}]^{-1} \mathbf{G}^T$  when  $P = M < N$  (least squares).
3. Reduces to  $\mathbf{G}^T [\mathbf{G} \mathbf{G}^T]^{-1}$  when  $P = N < M$  (minimum length).
4. Exists when  $P < \min(N, M)$ .

In the following pages we will consider each of these classes of problems, beginning with  $P = N = M$  (Class I). In this case,

$$\mathbf{G} = \mathbf{U} \quad \Lambda \quad \mathbf{V}^T \quad (6.60)$$

$N \times N \quad N \times N \quad N \times N \quad N \times N$

with

$$\Lambda = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & \lambda_N \end{bmatrix} \quad (7.1)$$

Since  $P = N = M$ , there are no zero singular values and we have

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_N > 0 \quad (7.2)$$

In order to find an inverse operator based on Equation (6.60), we need to find the inverse of a product of matrices. Applying the results from Equation (2.8) to Equation (6.60) above gives

$$\mathbf{G}^{-1} = [\mathbf{V}^T]^{-1} \Lambda^{-1} \mathbf{U}^{-1} \quad (7.3)$$

We know  $\Lambda^{-1}$  exists and is given by

$$\Lambda^{-1} = \begin{bmatrix} 1/\lambda_1 & 0 & \cdots & 0 \\ 0 & 1/\lambda_2 & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & 1/\lambda_N \end{bmatrix} \quad (7.4)$$

$N \times N$

We now make use of the fact that both  $\mathbf{U}$  and  $\mathbf{V}$  are orthogonal matrices. The properties of  $\mathbf{U}$  and  $\mathbf{V}$  that we wish to use are

$$[\mathbf{V}^T]^{-1} = \mathbf{V} \quad (7.5)$$

and

$$\mathbf{U}^{-1} = \mathbf{U}^T \tag{7.6}$$

Therefore

$\mathbf{G}^{-1} = \mathbf{V} \quad \Lambda^{-1} \quad \mathbf{U}^T \quad P = N = M$	(7.7)
$N \times N \quad N \times N \quad N \times N \quad N \times N$	

Equation (7.7) implies that  $\mathbf{G}^{-1}$ , the mathematical inverse of  $\mathbf{G}$ , can be found using singular-value decomposition when  $P = N = M$ .

What we need now is to find an operator for the other three classes of problems that will reduce to the mathematical inverse  $\mathbf{G}^{-1}$  when it exists.

## 7.2 The Generalized Inverse Operator $\mathbf{G}_g^{-1}$

### 7.2.1 Background Information

We start out with three pieces of information:

1.  $\mathbf{G} = \mathbf{U}\Lambda\mathbf{V}^T$  (6.60)

2.  $\mathbf{G}^{-1} = \mathbf{V}\Lambda^{-1}\mathbf{U}^T$  (when  $\mathbf{G}^{-1}$  exists) (7.8)

3.  $\mathbf{G} = \mathbf{U}_P\Lambda_P\mathbf{V}_P^T$  (singular-value decomposition) (6.69)

Then, by analogy with defining the inverse in Equation (7.7) above on the form of Equation (6.60), we introduce the *generalized inverse operator*:

$\mathbf{G}_g^{-1} = \mathbf{V}_p \quad \Lambda_p^{-1} \quad \mathbf{U}_p^T$	(7.8)
$M \times N \quad M \times P \quad P \times P \quad P \times N$	

and find out the consequences for our four cases. Menke points out that there may be many generalized inverses, but Equation (7.8) is by far the most common generalized inverse.

### 7.2.2 Class I: $P = N = M$

In this case, we have

1.  $\mathbf{V}_P = \mathbf{V}$  and  $\mathbf{U}_P = \mathbf{U}$ .
2.  $\mathbf{V}_0$  and  $\mathbf{U}_0$  are empty.

We start with the definition of the generalized inverse operator in Equation (7.8):

$$\mathbf{G}_g^{-1} = \mathbf{V}_P \Lambda_P^{-1} \mathbf{U}_P^T \quad (7.8)$$

But, since  $P = M$  we have

$$\mathbf{V}_P = \mathbf{V} \quad (7.9)$$

Similarly, since  $P = N$  we have

$$\mathbf{U}_P = \mathbf{U} \quad (7.10)$$

Finally, since  $P = N = M$ , we have

$$\Lambda_P^{-1} = \Lambda^{-1} \quad (7.11)$$

Therefore, combining Equations (7.8)–(7.11), we recover Equation (7.7)

$$\mathbf{G}^{-1} = \mathbf{V} \Lambda^{-1} \mathbf{U}^T \quad P = N = M \quad (7.7)$$

the exact mathematical inverse. Thus, we have shown that the generalized inverse operator reduces to the exact mathematical inverse in the case of  $P = N = M$ . Next we consider the case of  $P = M < N$ .

### 7.2.3 Class II: $P = M < N$

This is the least squares environment where we have more observations than unknowns, but where a unique solution exists. In this case, we have:

1.  $\mathbf{V}_P = \mathbf{V}$
2.  $\mathbf{V}_0$  is empty.
3.  $\mathbf{U}_0$  exists.

Ultimately we wish to show that the generalized inverse operator reduces to the least squares operator when  $P = M < N$ .

*The Role of  $\mathbf{G}^T\mathbf{G}$*

Recall that the least squares operator, as defined in Equation (3.27), for example, is given by

$$\mathbf{m}_{LS} = [\mathbf{G}^T\mathbf{G}]^{-1}\mathbf{G}^T\mathbf{d} \quad (3.31)$$

We first consider  $\mathbf{G}^T\mathbf{G}$ , using singular-value decomposition for  $\mathbf{G}$ , obtaining

$$\mathbf{G}^T\mathbf{G} = [\mathbf{U}_P\Lambda_P\mathbf{V}_P^T]^T[\mathbf{U}_P\Lambda_P\mathbf{V}_P^T] \quad (7.12)$$

Recall that the transpose of the product of matrices is given by the product of the transposes in the reverse order:

$$[\mathbf{AB}]^T = \mathbf{B}^T\mathbf{A}^T$$

Therefore

$$\mathbf{G}^T\mathbf{G} = [\mathbf{V}_P^T]^T\Lambda_P^T\mathbf{U}_P^T\mathbf{U}_P\Lambda_P\mathbf{V}_P^T \quad (7.13)$$

or

$$\mathbf{G}^T\mathbf{G} = \mathbf{V}_P\Lambda_P\mathbf{U}_P^T\mathbf{U}_P\Lambda_P\mathbf{V}_P^T \quad (7.14)$$

since

$$\Lambda_P^T = \Lambda_P \quad (7.15)$$

and

$$[\mathbf{V}_P^T]^T = \mathbf{V}_P \quad (7.16)$$

We know, however, that  $\mathbf{U}_P$  is a semiorthogonal matrix. Thus

$$\mathbf{U}_P^T\mathbf{U}_P = \mathbf{I}_P \quad (7.17)$$

Therefore, Equation (7.14) reduces to

$$\mathbf{G}^T\mathbf{G} = \mathbf{V}_P\Lambda_P\Lambda_P\mathbf{V}_P^T \quad (7.18)$$

Now, consider the product of  $\Lambda_P$  with itself

$$\Lambda_P \Lambda_P = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & \lambda_p \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & \lambda_p \end{bmatrix} \quad (7.19)$$

or

$$\Lambda_P \Lambda_P = \begin{bmatrix} \lambda_1^2 & 0 & \cdots & 0 \\ 0 & \lambda_2^2 & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & \lambda_p^2 \end{bmatrix} = \Lambda_P^2 \quad (7.20)$$

where we have introduced the notation  $\Lambda_P^2$  for the product of  $\Lambda_P$  with itself.

Please be aware, as noted before, that the notation  $\mathbf{A}^2$ , when  $\mathbf{A}$  is a matrix, has no universally accepted definition. We will use the definition implied by Equation (7.20). Finally, then, we write Equation (7.18) as

$$\begin{array}{ccccc} \mathbf{G}^T \mathbf{G} & = & \mathbf{V}_P & \Lambda_P^2 & \mathbf{V}_P^T \\ M \times M & & M \times P & P \times P & P \times M \end{array} \quad (7.21)$$

### Finding the Inverse of $\mathbf{G}^T \mathbf{G}$

I claim that  $\mathbf{G}^T \mathbf{G}$  has a mathematical inverse  $[\mathbf{G}^T \mathbf{G}]^{-1}$  when  $P = M$ . The reason that  $[\mathbf{G}^T \mathbf{G}]^{-1}$  exists in this case is that  $\mathbf{G}^T \mathbf{G}$  has the following eigenvalue problem:

$$\mathbf{G}^T \mathbf{G} \mathbf{v}_i = \eta_i^2 \mathbf{v}_i \quad i = 1, \dots, M \quad (6.21)$$

where, because  $P = M$ , we know that all  $M$   $\eta_i^2$  are nonzero. That is,  $\mathbf{G}^T \mathbf{G}$  has no zero eigenvalues. Thus, it has a mathematical inverse.

Using the theorem presented earlier about the inverse of a product of matrices in Equations (2.8), we have

$$[\mathbf{G}^T \mathbf{G}]^{-1} = [\mathbf{V}_P^T]^{-1} [\Lambda_P^2]^{-1} \mathbf{V}_P^{-1} \quad (7.22)$$

The inverse of  $\mathbf{V}_P^T$  is found as follows. First,

$$\mathbf{V}_P^T \mathbf{V}_P = \mathbf{I}_P \quad (7.23)$$

is always true because  $\mathbf{V}_P$  is semiorthogonal. But, because we have that  $P = M$  in this case, we also have

$$\mathbf{V}_P \mathbf{V}_P^T = \mathbf{I}_M \quad (7.24)$$

Thus,  $\mathbf{V}_P$  is itself an orthogonal matrix, and we have

$$[\mathbf{V}_P^T]^{-1} = \mathbf{V}_P \quad (7.25)$$

and

$$\mathbf{V}_P^{-1} = \mathbf{V}_P^T \quad (7.26)$$

Thus, we can write Equation (7.22) as

$$[\mathbf{G}^T \mathbf{G}]^{-1} = \mathbf{V}_P [\Lambda_P^2]^{-1} \mathbf{V}_P^T \quad (7.27)$$

Finally, we note that  $[\Lambda_P^2]^{-1}$  is given by

$$[\Lambda_P^2]^{-1} = \Lambda_P^{-2} = \begin{bmatrix} 1/\lambda_1^2 & 0 & \cdots & 0 \\ 0 & 1/\lambda_2^2 & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & 1/\lambda_p^2 \end{bmatrix} \quad (7.28)$$

Therefore

$$[\mathbf{G}^T \mathbf{G}]^{-1} = \mathbf{V}_P \Lambda_P^{-2} \mathbf{V}_P^T \quad (7.29)$$

where  $\Lambda_P^{-2}$  is as defined in Equation (7.28).

### *Equivalence of $\mathbf{G}_g^{-1}$ and Least Squares When $P = M < N$*

We start with the least squares operator, from Equation (3.28), for example

$$\mathbf{G}_{LS}^{-1} = [\mathbf{G}^T \mathbf{G}]^{-1} \mathbf{G}^T \quad (3.32)$$

We can use Equation (7.29) for  $[\mathbf{G}^T \mathbf{G}]^{-1}$  in Equation (3.32) and singular-value decomposition for  $\mathbf{G}^T$  to obtain

$$[\mathbf{G}^T \mathbf{G}]^{-1} \mathbf{G}^T = \mathbf{V}_P \Lambda_P^{-2} \mathbf{V}_P^T [\mathbf{U}_P \Lambda_P \mathbf{V}_P^T]^T \quad (7.30)$$

$$= \mathbf{V}_P \Lambda_P^{-2} \mathbf{V}_P^T \mathbf{V}_P \Lambda_P \mathbf{U}_P^T \quad (7.31)$$

But, because  $\mathbf{V}_P$  is semiorthogonal, we have from Equation (7.23)

$$\mathbf{V}_P^T \mathbf{V}_P = \mathbf{I}_P \quad (7.23)$$

Thus, Equation (7.31) becomes

$$[\mathbf{G}^T \mathbf{G}]^{-1} \mathbf{G}^T = \mathbf{V}_P \Lambda_P^{-2} \Lambda_P \mathbf{U}_P^T \quad (7.32)$$

Now, considering Equations (7.20) and (7.21), we see that

$$\Lambda_P^{-2} \Lambda_P = \Lambda_P^{-1} \quad (7.33)$$

Finally, then, Equation (7.32) becomes

$$[\mathbf{G}^T \mathbf{G}]^{-1} \mathbf{G}^T = \mathbf{V}_P \Lambda_P^{-1} \mathbf{U}_P^T = \mathbf{G}_g^{-1} \quad (7.34)$$

as required. That is, we have shown that when  $P = M < N$ , the generalized inverse operator is equivalent to the least squares operator.

### *Geometrical Interpretation of $\mathbf{G}_g^{-1}$ When $P = M < N$*

It is possible to gain some insight into the generalized inverse operator by considering a geometrical argument. An arbitrary data vector  $\mathbf{d}$  may have components in both  $\mathbf{U}_P$  and  $\mathbf{U}_0$  spaces. The generalized inverse operator returns a solution  $\mathbf{m}_g$ , for which the predicted data lies completely in  $\mathbf{U}_P$  space and minimizes the misfit to the observed data. The steps necessary to see this follow:

*Step 1.* Let  $\mathbf{m}_g$  be the generalized inverse solution, given by

$$\mathbf{m}_g = \mathbf{G}_g^{-1} \mathbf{d} \quad (7.35)$$

*Step 2.* Let  $\hat{\mathbf{d}}$  be the predicted data, given by

$$\hat{\mathbf{d}} = \mathbf{G} \mathbf{m}_g \quad (7.36)$$

We now introduce the following theorem about the relationship of the predicted data to  $\mathbf{U}_P$  space

**Theorem:**  $\hat{\mathbf{d}}$  lies completely in  $\mathbf{U}_P$ -space (the subset of  $N$ -space spanned by the  $P$  eigenvectors in  $\mathbf{U}_P$ ).

**Proof:** If  $\hat{\mathbf{d}}$  lies in  $\mathbf{U}_P$ -space, it is orthogonal to  $\mathbf{U}_0$ -space. That is,

$$\begin{aligned} \mathbf{U}_0^T \hat{\mathbf{d}} &= \mathbf{U}_0^T \mathbf{G} \mathbf{m}_g = \mathbf{U}_0^T \mathbf{U}_P \mathbf{\Lambda}_P \mathbf{V}_P^T \mathbf{m}_g \\ &= \mathbf{0} \\ &\quad (N - P) \times 1 \end{aligned} \quad (7.37)$$

which follows from

$$\mathbf{U}_0^T \mathbf{U}_P = \mathbf{0} \quad (7.38)$$

That is, every eigenvector in  $\mathbf{U}_0$  is perpendicular to every eigenvector in  $\mathbf{U}_P$ . Another way to see this is that *all* of the eigenvectors in  $\mathbf{U}$  are perpendicular to each other. Thus, *any* subset of  $\mathbf{U}$  is perpendicular to the rest of  $\mathbf{U}$ . Q.E.D.

*Step 3.* Let  $\mathbf{d} - \hat{\mathbf{d}}$  be the residual data vector (i.e., observed – predicted data, also known as the misfit vector), given by

$$\begin{aligned} \mathbf{d} - \hat{\mathbf{d}} &= \mathbf{d} - \mathbf{G} \mathbf{m}_g \\ &= \mathbf{d} - \mathbf{G} [\mathbf{G}_g^{-1} \mathbf{d}] \\ &= \mathbf{d} - \mathbf{G} \mathbf{G}_g^{-1} \mathbf{d} \\ &= \mathbf{d} - [\mathbf{U}_P \mathbf{\Lambda}_P \mathbf{V}_P^T] [\mathbf{V}_P \mathbf{\Lambda}_P^{-1} \mathbf{U}_P^T] \mathbf{d} \\ &\quad \vdots \\ &= \mathbf{d} - \mathbf{U}_P \mathbf{U}_P^T \mathbf{d} \end{aligned} \quad (7.39)$$

We cannot further reduce Equation (7.39) whenever  $P < N$  because in this case

$$\mathbf{U}_P \mathbf{U}_P^T \neq \mathbf{I}_N$$

Next, we introduce a theorem about the relationship between the misfit vector and  $\mathbf{U}_0$  space.

**Theorem:** The misfit vector  $\mathbf{d} - \hat{\mathbf{d}}$  is orthogonal to  $\mathbf{U}_P$ .

**Proof:**

$$\begin{aligned} \mathbf{U}_P^T [\mathbf{d} - \hat{\mathbf{d}}] &= \mathbf{U}_P^T [\mathbf{d} - \mathbf{U}_P \mathbf{U}_P^T \mathbf{d}] \\ &= \mathbf{U}_P^T \mathbf{d} - \mathbf{U}_P^T \mathbf{U}_P \mathbf{U}_P^T \mathbf{d} \\ &= \mathbf{U}_P^T \mathbf{d} - \mathbf{U}_P^T \mathbf{d} \\ &= \mathbf{0} \quad \text{Q.E.D.} \end{aligned} \quad (7.40)$$

$$P \times 1$$

The crucial step in going from the second to third lines being that  $\mathbf{U}_P^T \mathbf{U}_P = \mathbf{I}_P$  since  $\mathbf{U}_P$  is semiorthogonal. This implies that the misfit vector  $\mathbf{d} - \hat{\mathbf{d}}$  lies completely in the space spanned by  $\mathbf{U}_0$ .

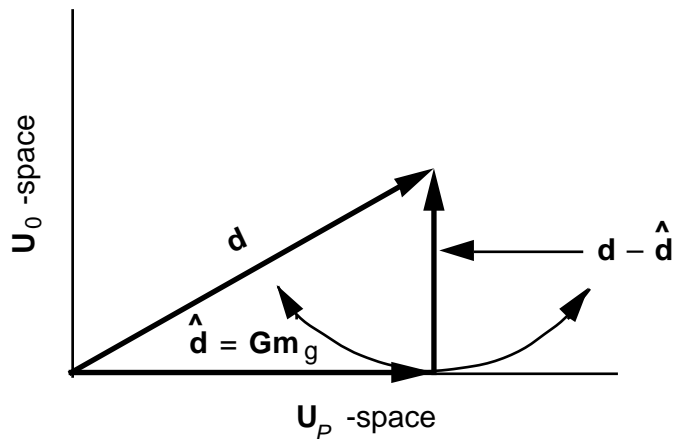
Combining the results from the above two theorems, we introduce the final theorem of this section concerning the relationship between the predicted data and the misfit vector.

**Theorem:** The predicted data vector  $\hat{\mathbf{d}}$  is perpendicular to the misfit vector  $\mathbf{d} - \hat{\mathbf{d}}$ .

**Proof:**

1. The predicted data vector  $\hat{\mathbf{d}}$  lies in  $\mathbf{U}_P$  space.
2. The misfit vector  $\mathbf{d} - \hat{\mathbf{d}}$  lies in  $\mathbf{U}_0$  space.
3. Since the vectors in  $\mathbf{U}_P$  are perpendicular to the vectors in  $\mathbf{U}_0$ ,  $\hat{\mathbf{d}}$  is perpendicular to the misfit vector  $\mathbf{d} - \hat{\mathbf{d}}$ . Q.E.D.

Step 4. Consider the following schematic graph showing the relationship between the various vectors and spaces:



The data vector  $\mathbf{d}$  has components in both  $\mathbf{U}_P$  and  $\mathbf{U}_0$  spaces. Note the following points:

1. The predicted data vector  $\hat{\mathbf{d}} = \mathbf{Gm}_g$  lies completely in  $\mathbf{U}_P$  space.
2. The residual vector  $\mathbf{d} - \hat{\mathbf{d}}$  lies completely in  $\mathbf{U}_0$  space.

3. The shortest distance from the observed data  $\mathbf{d}$  to the  $\mathbf{U}_P$  axis is given by the misfit vector  $\mathbf{d} - \hat{\mathbf{d}}$ .

Thus the generalized inverse  $\mathbf{G}_g^{-1}$  minimizes the distance between the observed data vector  $\mathbf{d}$  and  $\mathbf{U}_P$ , the subset of data space in which all possible predicted data  $\hat{\mathbf{d}}$  must lie.

Recall that the least squares operator given in Equation (3.32)

$$\mathbf{G}_{LS}^{-1} = [\mathbf{G}^T \mathbf{G}]^{-1} \mathbf{G}^T \quad (3.32)$$

minimizes the length of the misfit vector. Thus, the generalized inverse operator is equivalent to the least squares operator when  $P = M < N$ .

- Step 5.* For  $P = M < N$ , it is possible to write the generalized inverse without forming  $\mathbf{U}_P$ . To see this, note that the generalized inverse is equivalent to least squares for  $P = M < N$ . That is,

$$\mathbf{G}_g^{-1} = [\mathbf{G}^T \mathbf{G}]^{-1} \mathbf{G}^T \quad (7.41)$$

But, by Equation (7.28),  $[\mathbf{G}^T \mathbf{G}]^{-1}$  is given by

$$[\mathbf{G}^T \mathbf{G}]^{-1} = \mathbf{V}_P \mathbf{\Lambda}_P^{-2} \mathbf{V}_P^T \quad (7.27)$$

Thus, the generalized inverse in this case is given by

$$\mathbf{G}_g^{-1} = [\mathbf{G}^T \mathbf{G}]^{-1} \mathbf{G}^T = \mathbf{V}_P \mathbf{\Lambda}_P^{-2} \mathbf{V}_P^T \mathbf{G}^T \quad (7.42)$$

Equation (7.42) shows that the generalized inverse can be found without ever forming  $\mathbf{U}_P$  when  $P = M < N$ . In general, this shortcut is not used, even though you can form the inverse, because there is useful information lost about data space.

- Step 6.* Finally, recall the compatibility equations given by

$$\mathbf{U}_0^T \mathbf{d} = \mathbf{0} \quad (6.78)$$

$(N - P) \times 1$

Note that if the observed data  $\mathbf{d}$  has any projection in  $\mathbf{U}_0$  space, is not possible to find a solution  $\mathbf{m}$  that can fit the data exactly. All estimates  $\mathbf{m}$  lead to predicted data  $\mathbf{Gm}$  that lie in  $\mathbf{U}_P$  space. Thus, from the graph above, one sees that if the observed data,  $\mathbf{d}$ , lies completely in  $\mathbf{U}_P$  space, the compatibility equations are automatically satisfied.

### 7.2.4 Class III: $P = N < M$

This is the minimum length environment where we have more model parameters than observations. There are an infinite number of possible solutions that can fit the data exactly. Recall that the minimum length solution is the one that has the shortest length. Ultimately we wish to show that the generalized inverse operator reduces to the minimum length operator when  $P = N < M$ .

For  $P = N < M$  we have

1.  $\mathbf{U}_P = \mathbf{U}$
2.  $\mathbf{U}_0$  is empty.
3.  $\mathbf{V}_0$  is not empty.

#### *The Role of $\mathbf{GG}^T$*

Recall that the minimum length operator, as defined in Equation (3.75), is given by

$$\mathbf{G}_{ML}^{-1} = \mathbf{G}^T[\mathbf{GG}^T]^{-1} \quad (3.75)$$

We seek, thus, to show that  $\mathbf{G}_g^{-1} = \mathbf{G}^T[\mathbf{GG}^T]^{-1}$  in this case. First consider writing  $\mathbf{GG}^T$  using singular-value decomposition:

$$\begin{aligned} \mathbf{GG}^T &= \mathbf{U}_P \mathbf{\Lambda}_P \mathbf{V}_P^T [\mathbf{U}_P \mathbf{\Lambda}_P \mathbf{V}_P^T]^T \\ &= \mathbf{U}_P \mathbf{\Lambda}_P \mathbf{V}_P^T \mathbf{V}_P \mathbf{\Lambda}_P \mathbf{U}_P^T \\ &= \mathbf{U}_P \mathbf{\Lambda}_P^2 \mathbf{U}_P^T \end{aligned} \quad (7.43)$$

#### *Finding the Inverse of $\mathbf{GG}^T$*

Note that  $\mathbf{GG}^T$  is  $N \times N$  and  $P = N$ . This implies that  $[\mathbf{GG}^T]^{-1}$ , the mathematical inverse of  $\mathbf{GG}^T$ , exists. Again using the theorem stated in Equation (2.8) about the inverse of a product of matrices, we have

$$\begin{aligned} [\mathbf{GG}^T]^{-1} &= [\mathbf{U}_P^T]^{-1} [\mathbf{\Lambda}_P^2]^{-1} \mathbf{U}_P^{-1} \\ &= \mathbf{U}_P \mathbf{\Lambda}_P^{-2} \mathbf{U}_P^T \quad P = N \end{aligned} \quad (7.44)$$

Then

$$\begin{aligned}
 \mathbf{G}^T[\mathbf{G}\mathbf{G}^T]^{-1} &= [\mathbf{U}_P\mathbf{\Lambda}_P\mathbf{V}_P^T]^T\mathbf{U}_P\mathbf{\Lambda}_P^{-2}\mathbf{U}_P^T \\
 &= \mathbf{V}_P\mathbf{\Lambda}_P\mathbf{U}_P^T\mathbf{U}_P\mathbf{\Lambda}_P^{-2}\mathbf{U}_P^T \\
 &= \mathbf{V}_P\mathbf{\Lambda}_P\mathbf{\Lambda}_P^{-2}\mathbf{U}_P^T \\
 &= \mathbf{V}_P\mathbf{\Lambda}_P^{-1}\mathbf{U}_P^T \\
 &= \mathbf{G}_g^{-1}
 \end{aligned} \tag{7.45}$$

as required.

*Fitting the Data Exactly When  $P = N < M$*

As before, let the generalized inverse solution  $\mathbf{m}_g$  be given by

$$\mathbf{m}_g = \mathbf{G}_g^{-1}\mathbf{d} \tag{7.35}$$

Then the predicted data  $\hat{\mathbf{d}}$  is given by

$$\begin{aligned}
 \hat{\mathbf{d}} &= \mathbf{G}\mathbf{m}_g \\
 &= \mathbf{G}[\mathbf{G}_g^{-1}\mathbf{d}] \\
 &= \mathbf{G}\mathbf{G}_g^{-1}\mathbf{d} \\
 &= \mathbf{U}_P\mathbf{\Lambda}_P\mathbf{V}_P^T\mathbf{V}_P\mathbf{\Lambda}_P^{-1}\mathbf{U}_P^T\mathbf{d} \\
 &= \mathbf{U}_P\mathbf{U}_P^T\mathbf{d} \\
 &= \mathbf{d}
 \end{aligned} \tag{7.46}$$

since  $\mathbf{U}_P\mathbf{U}_P^T = \mathbf{I}_N$  whenever  $P = N$ .

Thus, one can fit the data exactly whenever  $P = N$ . The reason is that  $\mathbf{U}_0$  is empty when  $P = N$ . That is,  $\mathbf{U}_P$  is equal to  $\mathbf{U}$  space.

*The Generalized Inverse Solution  $\mathbf{m}_g$  Lies in  $\mathbf{V}_P$  Space*

The generalized inverse solution  $\mathbf{m}_g$  is given by

$$\mathbf{m}_g = \mathbf{G}_g^{-1}\mathbf{d} \tag{7.35}$$

and is a vector in model space. It lies completely in  $\mathbf{V}_P$  space. The way to see this is to take the dot product of  $\mathbf{m}_g$  with the eigenvectors in  $\mathbf{V}_0$ . If  $\mathbf{m}_g$  has no projection in  $\mathbf{V}_0$  space, then it lies completely in  $\mathbf{V}_P$  space.

Thus,

$$\begin{aligned}\mathbf{V}_0^T \mathbf{m}_g &= \mathbf{V}_0^T \mathbf{G}^{-1} \mathbf{d} \\ &= \mathbf{V}_0^T \mathbf{V}_P \mathbf{\Lambda}_P^{-1} \mathbf{U}_P^T \mathbf{d} \\ &= \mathbf{0} \\ &\quad (M-P) \times 1\end{aligned}\tag{7.47}$$

since  $\mathbf{V}_0^T \mathbf{V}_P = \mathbf{0}$ .

### *Nonuniqueness of the Solution When $P = N < M$*

The solution to  $\mathbf{Gm} = \mathbf{d}$  is nonunique because  $\mathbf{V}_0$  exists when  $P < M$ . Let the general solution  $\mathbf{m}$  to  $\mathbf{Gm} = \mathbf{d}$  be given by

$$\hat{\mathbf{m}} = \mathbf{m}_g + \sum_{i=P+1}^M \alpha_i \mathbf{v}_i\tag{7.48}$$

That is, the general solution is given by the generalized inverse solution  $\mathbf{m}_g$  plus a linear combination of the eigenvectors in  $\mathbf{V}_0$  space, where  $\alpha_i$  are constants. The predicted data for the general case is given by

$$\begin{aligned}\mathbf{G}\hat{\mathbf{m}} &= \mathbf{G} \left[ \mathbf{m}_g + \sum_{i=P+1}^M \alpha_i \mathbf{v}_i \right] \\ &= \mathbf{G}\mathbf{m}_g + \sum_{i=P+1}^M \alpha_i \mathbf{G}\mathbf{v}_i\end{aligned}\tag{7.49}$$

When  $\mathbf{G}$  operates on a vector in  $\mathbf{V}_0$  space, however, it returns a zero vector. That is,

$$\begin{aligned}\mathbf{G}\mathbf{V}_0 &= \mathbf{U}_P \mathbf{\Lambda}_P \mathbf{V}_P^T \mathbf{V}_0 \\ &= \mathbf{0} \\ &\quad N \times (M-P)\end{aligned}\tag{7.50}$$

which follows from the fact that the eigenvectors in  $\mathbf{V}_P$  are perpendicular to the eigenvectors in  $\mathbf{V}_0$ . Thus,

$$\begin{aligned}\mathbf{G}\hat{\mathbf{m}} &= \mathbf{G}\mathbf{m}_g + \mathbf{0} \\ &= \mathbf{d}\end{aligned}\tag{7.51}$$

Now, consider the length squared of  $\hat{\mathbf{m}}$

$$\|\hat{\mathbf{m}}\|^2 = \|\mathbf{m}_g\|^2 + \sum_{i=P+1}^M \alpha_i^2 \quad (7.52)$$

which follows from the fact that  $[\mathbf{v}_i]^T \mathbf{v}_j = \delta_{ij}$ .

$$\|\hat{\mathbf{m}}\|^2 \geq \|\mathbf{m}_g\|^2 \quad (7.53)$$

That is,  $\mathbf{m}_g$ , the generalized inverse solution, is the smallest of all possible solutions to  $\mathbf{G}\mathbf{m} = \mathbf{d}$ . This is precisely what was stated at the beginning of this section: the generalized inverse solution is equivalent to the minimum length solution when  $P = N < M$ .

*It is Possible to Write  $\mathbf{G}_g^{-1}$  Without  $\mathbf{V}_P$  When  $P = N < M$*

To see this, we write the generalized inverse operator as the minimum length operator and use singular-value decomposition. That is,

$$\begin{aligned} \mathbf{G}_g^{-1} &= \mathbf{G}^T [\mathbf{G}\mathbf{G}^T]^{-1} \\ &\vdots \\ &= \mathbf{G}^T \mathbf{U}_P \mathbf{\Lambda}_P^{-2} \mathbf{U}_P^T \end{aligned} \quad (7.54)$$

Typically, this shortcut is not used because knowledge of  $\mathbf{V}_P$  space is useful in the interpretation of the results.

### 7.2.5 Class IV: $P < \min(N, M)$

This is the class of problems for which neither least squares nor minimum length operators exist. That is, the least squares operator

$$\mathbf{G}_{LS} = [\mathbf{G}^T \mathbf{G}]^{-1} \mathbf{G}^T \quad (3.32)$$

does not exist because  $[\mathbf{G}^T \mathbf{G}]^{-1}$  exists only when  $P = M$ . Similarly, the minimum length operator

$$\mathbf{G}_{ML} = \mathbf{G}^T [\mathbf{G}\mathbf{G}^T]^{-1} \quad (3.75)$$

does not exist because  $[\mathbf{G}\mathbf{G}^T]^{-1}$  exists only when  $P = N$ .

For  $P < \min(N, M)$  we have

1.  $\mathbf{V}_0$  is not empty.

2.  $U_0$  is not empty.

In this environment the solution is both nonunique (because  $V_0$  exists), and it is impossible to fit the data exactly unless the compatibility equations (Equations 6.69) are satisfied. That is, it is impossible to fit the data exactly unless the data have no projection onto  $U_0$  space.

The generalized inverse operator cannot be further reduced and is given by Equation (7.8)

$$\mathbf{G}_g^{-1} = \mathbf{V}_P \Lambda_P^{-1} \mathbf{U}_P^T \quad (7.8)$$

The generalized inverse operator  $\mathbf{G}_g^{-1}$  simultaneously minimizes the misfit vector  $\mathbf{d} - \hat{\mathbf{d}}$  in data space *and* the solution length  $\|\mathbf{m}_g\|^2$  in model space.

In summary, in this section we have shown that the generalized inverse operator  $\mathbf{G}_g^{-1}$  reduces to

1. The exact inverse when  $P = N = M$ .
2. The least squares inverse  $[\mathbf{G}^T \mathbf{G}]^{-1} \mathbf{G}^T$  when  $P = M < N$ .
3. The minimum length inverse  $\mathbf{G}^T [\mathbf{G} \mathbf{G}^T]^{-1}$  when  $P = N < M$ .

Since we have shown that the generalized inverse is equivalent to the exact, least squares, and minimum length operators when they exist, it is worth comparing the way the solution  $\mathbf{m}_g$  is written. In the least squares or unique inverse environment for example, we would then write

$$\mathbf{m}_g = \mathbf{G}_g^{-1} \mathbf{d} \quad (7.55)$$

but in the minimum length environment we would write

$$\mathbf{m}_g = \langle \mathbf{m} \rangle + \mathbf{G}_g^{-1} [\mathbf{d} - \mathbf{G} \langle \mathbf{m} \rangle] \quad (7.56)$$

which explicitly includes a dependence on the prior estimate  $\langle \mathbf{m} \rangle$ . It is somewhat disconcerting to have to carry around two forms of the solution for the generalized inverse. Consider what happens, however if we use Equation (7.56) for the unique or least squares environment. Then

$$\begin{aligned} \mathbf{m}_g &= \langle \mathbf{m} \rangle + \mathbf{G}_g^{-1} [\mathbf{d} - \mathbf{G} \langle \mathbf{m} \rangle] \\ &= \langle \mathbf{m} \rangle + \mathbf{G}_g^{-1} \mathbf{d} - \mathbf{G}_g^{-1} \mathbf{G} \langle \mathbf{m} \rangle \\ &= \mathbf{G}_g^{-1} \mathbf{d} + [\mathbf{I}_M - \mathbf{G}_g^{-1} \mathbf{G}] \langle \mathbf{m} \rangle \end{aligned} \quad (7.57)$$

For the unique inverse environment,  $\mathbf{G}_g^{-1} \mathbf{G} = \mathbf{I}_M$ , and hence Equation (7.57) reduces to Equation (7.55). For the least squares environment, we have

$$\mathbf{G}_g^{-1} \mathbf{G} = [\mathbf{G}^T \mathbf{G}]^{-1} \mathbf{G}^T \mathbf{G} = \mathbf{I}_M \quad (7.58)$$

and hence Equation (7.57) again reduces to Equation (7.55). The unique inverse and least squares environments thus have no dependence on  $\langle \mathbf{m} \rangle$ . Equation (7.56), however, is true for the generalized inverse in all environments and is thus adopted as the general form of the generalized inverse solution  $\mathbf{m}_g$ .

In the next section we will introduce measures of the quality of the generalized inverse operator. These will include the *model resolution matrix*  $\mathbf{R}$ , the *data resolution matrix*  $\mathbf{N}$  (also called the *data information density matrix*), and the *unit (model) covariance matrix*  $[\text{cov}_u \mathbf{m}]$ .

## 7.3 Measures of Quality for the Generalized Inverse

### 7.3.1 Introduction

In this section three measures of quality for the generalized inverse will be considered. They are

1. The  $M \times M$  model resolution matrix  $\mathbf{R}$
2. The  $N \times N$  data resolution matrix  $\mathbf{N}$
3. The  $M \times M$  unit covariance matrix  $[\text{cov}_u \mathbf{m}]$

The model resolution matrix  $\mathbf{R}$  measures the ability of the inverse operator to uniquely determine the estimated model parameters. The data resolution matrix  $\mathbf{N}$  measures the ability of the inverse operator to uniquely determine the data. This is equivalent to describing the importance, or independent information, provided by the data. The two resolution matrices depend upon the partitioning of model and data spaces into  $\mathbf{V}_P$ ,  $\mathbf{V}_0$ , and  $\mathbf{U}_P$ ,  $\mathbf{U}_0$  spaces, respectively. Finally, the unit covariance matrix  $[\text{cov}_u \mathbf{m}]$  is a measure of how uncorrelated noise with unit variance in the data is mapped into uncertainties in the estimated model parameters.

### 7.3.2 The Model Resolution Matrix $\mathbf{R}$

Imagine for the moment that there is some “true” solution  $\mathbf{m}^{\text{true}}$  that exactly satisfies

$$\mathbf{G}\mathbf{m}^{\text{true}} = \mathbf{d} \quad (7.59)$$

In any inversion, we estimate this true solution with  $\mathbf{m}^{\text{est}}$  :

$$\mathbf{m}^{\text{est}} = \mathbf{G}_{\text{est}}^{-1} \mathbf{d} \quad (7.60)$$

where  $\mathbf{G}_{\text{est}}^{-1}$  is some inverse operator. It is then possible to ask how  $\mathbf{m}^{\text{est}}$  compares to  $\mathbf{m}^{\text{true}}$ .

Specifically considering the generalized inverse, we start with Equation (7.61) and replace  $\mathbf{d}$  with  $\mathbf{G}\mathbf{m}^{\text{true}}$ , obtaining

$$\mathbf{m}_g = \mathbf{G}_g^{-1} \mathbf{G} \mathbf{m}^{\text{true}} \quad (7.61)$$

The model resolution matrix  $\mathbf{R}$  is then defined as

$$\mathbf{R} = \mathbf{G}_g^{-1} \mathbf{G} \quad (7.62)$$

where  $\mathbf{R}$  is an  $M \times M$  symmetric matrix.

If  $\mathbf{R} = \mathbf{I}_M$ , then  $\mathbf{m}^{\text{est}} = \mathbf{m}^{\text{true}}$ , and we say that all of the model parameters are perfectly resolved, or equivalently that all of the model parameters are uniquely determined. If  $\mathbf{R} \neq \mathbf{I}_M$ , then  $\mathbf{m}^{\text{est}}$  is some weighted average of  $\mathbf{m}^{\text{true}}$ .

Consider the  $k$ th element of  $\mathbf{m}^{\text{est}}$ , denoted  $m_k^{\text{est}}$ , given by the product of the  $k$ th row of  $\mathbf{R}$  and  $\mathbf{m}^{\text{true}}$ :

$$m_k^{\text{est}} = \left[ \text{---} \right] \begin{bmatrix} m_1 \\ m_2 \\ \vdots \\ m_M \end{bmatrix}^{\text{true}} \quad (7.63)$$

kth row of  $\mathbf{R}$

The rows of  $\mathbf{R}$  can thus be seen as “windows,” or filters, through which the true solution is viewed. For example, suppose that the  $k$ th row of  $\mathbf{R}$  is given by

$$\mathbf{R}_k = [0, 0, \dots, 0, \underset{\substack{\uparrow \\ \text{kth column}}}{1}, 0, \dots, 0, 0] \quad (7.64)$$

We see that

$$m_k^{\text{est}} = 0m_1^{\text{true}} + \dots + 0m_{k-1}^{\text{true}} + 1m_k^{\text{true}} + 0m_{k+1}^{\text{true}} + \dots + 0m_M^{\text{true}} \quad (7.65)$$

or simply

$$m_k^{\text{est}} = m_k^{\text{true}} \quad (7.66)$$

In this case we say that the  $k$ th model parameter is perfectly resolved, or uniquely determined. Suppose, however, that the  $k$ th row of  $\mathbf{R}$  were given by

$$\mathbf{R}_k = [0, \dots, 0, 0.1, 0.3, 0.8, 0.4, 0.2, \dots, 0] \quad (7.67)$$

\uparrow  
kth column

Then the  $k$ th estimated model parameter  $m_k^{\text{est}}$  is given by

$$m_k^{\text{est}} = 0.1m_{k-2}^{\text{true}} + 0.3m_{k-1}^{\text{true}} + 0.8m_k^{\text{true}} + 0.4m_{k+1}^{\text{true}} + 0.2m_{k+2}^{\text{true}} \quad (7.68)$$

Or,  $m_k^{\text{est}}$  is a weighted average of several terms in  $\mathbf{m}^{\text{true}}$ . In the case just considered, it depends most heavily (0.8) on  $m_k^{\text{true}}$ , but it also depends on other components of the true solution. We say, then, that  $m_k^{\text{est}}$  is not perfectly resolved in this case. The closer the row of  $\mathbf{R}$  is to the row of an identity matrix, the better the resolution.

From the above discussion, it is clear that model resolution may be considered element by element. If  $\mathbf{R} = \mathbf{I}_M$ , then all elements are perfectly resolved. If a single row of  $\mathbf{R}$  is equal to the corresponding row of the identity matrix, then the associated model parameter estimate is perfectly resolved.

Finally, we can rewrite Equation (7.57) as

$$\mathbf{m}_g = \mathbf{G}_g^{-1} \mathbf{d} + [\mathbf{I} - \mathbf{R}] \langle \mathbf{m} \rangle \quad (7.69)$$

### Some Properties of $\mathbf{R}$

1.  $\mathbf{R} = \mathbf{V}_P \mathbf{V}_P^T$  (7.70)

Using singular-value decomposition on Equation (7.62),  $\mathbf{R}$  can be written as

$$\begin{aligned} \mathbf{R} &= \{ \mathbf{V}_P \Lambda_P^{-1} \mathbf{U}_P^T \} \{ \mathbf{U}_P \Lambda_P \mathbf{V}_P^T \} \\ &= \mathbf{V}_P \Lambda_P^{-1} \Lambda_P \mathbf{V}_P^T \\ &= \mathbf{V}_P \mathbf{V}_P^T \end{aligned} \quad (7.71)$$

In general,  $\mathbf{V}_P \mathbf{V}_P^T \neq \mathbf{I}$ . However, if  $P = M$ , then  $\mathbf{V}_P = \mathbf{V}$ , and  $\mathbf{V}_0$  is empty. In this case,

$$\mathbf{R} = \mathbf{V}_P \mathbf{V}_P^T = \mathbf{V} \mathbf{V}^T = \mathbf{I}_M \quad (7.72)$$

since  $\mathbf{V}$  is an orthogonal matrix. Thus, the condition for perfect model resolution is that  $\mathbf{V}_0$  be empty, or equivalently that  $P = M$ .

2. Trace ( $\mathbf{R}$ ) =  $\sum_{i=1}^M r_{ii} = P$ , the number of nonzero singular values

**Proof:** If  $\mathbf{R} = \mathbf{I}_M$ , then  $P = M$  and trace ( $\mathbf{R}$ ) =  $M$ .

For the general case, it is possible to write  $\mathbf{R}$  as the product of the following three partitioned matrices:

$$\begin{aligned}
\mathbf{R} &= [\mathbf{V}_P | \mathbf{V}_0] \begin{bmatrix} \mathbf{I}_P & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{V}_P^T \\ \mathbf{V}_0^T \end{bmatrix} \\
&= \mathbf{V} \begin{bmatrix} \mathbf{I}_P & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{V}^T \\
&= \mathbf{V} \mathbf{A} \mathbf{V}^T
\end{aligned} \tag{7.73}$$

where no part of  $\mathbf{V}_0$  actually contributes to  $\mathbf{R}$  because of the extra zeros in  $\mathbf{A}$ .

The trace of  $\mathbf{A}$  is equal to  $P$ . Note, however, that matrix  $\mathbf{A}$  has been obtained from  $\mathbf{R}$  by an orthogonal transformation because  $\mathbf{V}$  is an orthogonal matrix. Thus, by Equation (2.43), which states that the trace of a matrix is unchanged by an orthogonal transformation, we conclude that  $\text{trace } \mathbf{R} = P$ , as required. Q.E.D.

Trace ( $\mathbf{R}$ ) =  $P$  implies that  $\mathbf{G}$  has enough information to uniquely resolve  $P$  aspects of the solution. These aspects are, in fact, the  $P$  directions in model space given by the eigenvectors  $\mathbf{v}_i$  in  $\mathbf{V}_P$ . Whenever a row of  $\mathbf{R}$  is equal to a row of the identity matrix  $\mathbf{I}$ , then no part of the associated model parameter  $\mathbf{m}_i$  falls in  $\mathbf{V}_0$  space (i.e., it all falls in  $\mathbf{V}_P$  space) and that model parameter is perfectly resolved. When  $\mathbf{R}$  is not equal to the identity matrix  $\mathbf{I}$ , some part of the problem is not perfectly resolved. Sometimes this is acceptable and other times it is not, depending on the problem. Forming new model parameters as linear combinations of the old model parameters is one way to reduce the nonuniqueness of the problem. One way to do this is to form new model parameters by using the eigenvectors  $\mathbf{v}_i$  to define the linear combinations. Suppose that  $\mathbf{v}_i$ ,  $i < p$ , is given by

$$\mathbf{v}_i = (1 / \sqrt{M}) [1, 1, 1, \dots, 1]^T \tag{7.74}$$

This tells us that the average of all the model parameters is resolved, even if the individual model parameters may not be. If we defined a new model parameter as the average of all the old model parameters, it would be perfectly resolved.

If, as is often the case,  $\mathbf{G}$  represents some kind of an averaging function, you can attempt to reduce the nonuniqueness of the problem by forming new model parameters that are the sum or average of a subset of the old ones, even without using the full information in  $\mathbf{V}_P$ . If the model parameters are discretized versions of a continuous function, such as velocity or density versus depth, you may be able to improve the resolution by combining layers. A rule of thumb in this case is to sum the entries along the diagonal of the resolution matrix  $\mathbf{R}$  until you get close to one. At this point, your system is able to resolve one aspect of the solution uniquely. You can try forming a new model parameter as the average of the layer velocities or densities up to this point. Depending on the details of  $\mathbf{G}$ , you may have perfect resolution of this average of the old model parameters. Depending on the problem, it may be more useful to uniquely know the average of the

model parameters over some depth range than it is to have nonunique estimates of the values over the same range.

$$3. \quad \sum_{j=1}^M r_{ij}^2 = \sum_{j=1}^M r_{ji}^2 = r_{ii} = \text{“importance” of } i\text{th model parameter}$$

If  $r_{ii} = 1$ , then the  $i$ th model parameter is uniquely resolved, and it is thus said to be very important. If, on the other hand,  $r_{ii}$  is very small, then the parameter is poorly resolved and is said to be not very important.

If we further note that  $\mathbf{R}$  can be written as

$$\mathbf{R} = \begin{bmatrix} \vdots & \vdots & \cdots & \vdots \\ \mathbf{r}_1 & \mathbf{r}_2 & \cdots & \mathbf{r}_M \\ \vdots & \vdots & \cdots & \vdots \end{bmatrix} \quad (7.75)$$

where  $\mathbf{r}_i$  is the  $i$ th column of  $\mathbf{R}$ , then the estimated solution  $\mathbf{m}_g$  from (7.61) can be written as

$$\begin{aligned} \mathbf{m}_g &= \mathbf{R}\mathbf{m}^{\text{true}} \\ &= \mathbf{r}_1 m_1 + \mathbf{r}_2 m_2 + \dots + \mathbf{r}_M m_M \end{aligned} \quad (7.76)$$

where  $m_i$  is the  $i$ th component of  $\mathbf{m}^{\text{true}}$ , which follows from (2.23)–(2.30). That is, the estimated solution vector can also be thought of as the weighted sum of the columns of  $\mathbf{R}$ , with the weighting factors being given by the true solution.

The length squared of each column of  $\mathbf{R}$  can then be thought of as the importance of  $m_i$  in the solution. The length squared of  $\mathbf{r}_i$  is given by

$$\|\mathbf{r}_i\|^2 = \sum_{j=1}^M r_{ji}^2 = r_{ii} \quad (7.77)$$

Thus, the diagonal entries in  $\mathbf{R}$  give the importance of each model parameter for the problem.

We will return to the model resolution matrix  $\mathbf{R}$  later to show how the generalized inverse is the inverse operator that minimizes the difference between  $\mathbf{R} = \mathbf{G}_g^{-1}\mathbf{G}$  and  $\mathbf{I}_M$  in the least squares sense, and when we discuss the trade-off between resolution and variance.

### 7.3.3 The Data Resolution Matrix $\mathbf{N}$

Consider the development of the data resolution matrix  $\mathbf{N}$ , which follows closely that of the model resolution matrix  $\mathbf{R}$ . The estimated solution, for any inverse operator  $\mathbf{G}_{\text{est}}^{-1}$ , is given by

$$\mathbf{m}^{\text{est}} = \mathbf{G}_{\text{est}}^{-1} \mathbf{d} \quad (7.78)$$

The predicted data,  $\hat{\mathbf{d}}$ , for this estimated solution are given by

$$\hat{\mathbf{d}} = \mathbf{G}\mathbf{m}^{\text{est}} \quad (7.79)$$

Replacing  $\mathbf{m}^{\text{est}}$  in (7.79) with (7.78) gives

$$\hat{\mathbf{d}} = \mathbf{G}\mathbf{G}_{\text{est}}^{-1}\mathbf{d} = \mathbf{N}\mathbf{d} \quad (7.80)$$

where  $\mathbf{N}$  is an  $N \times N$  matrix called the *data resolution matrix*.

#### *A Specific Example*

As a specific example, consider the generalized inverse operator  $\mathbf{G}_g^{-1}$ . Then  $\mathbf{N}$  is given by

$$\mathbf{N} = \mathbf{G}\mathbf{G}_g^{-1} \quad (7.81)$$

If  $\mathbf{N} = \mathbf{I}_N$ , then the predicted data  $\hat{\mathbf{d}}$  equal the observed data  $\mathbf{d}$ , and the observed data can be fit exactly. If  $\mathbf{N} \neq \mathbf{I}_N$ , then the predicted data are some weighted average of the observed data  $\mathbf{d}$ .

Consider the  $k$ th element of the predicted data  $\hat{d}_k$

$$\hat{d}_k = \left[ \begin{array}{c} \text{---} \\ \text{kth row of } \mathbf{N} \\ \text{---} \end{array} \right] \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_N \end{bmatrix} \quad (7.82)$$

The rows of  $\mathbf{N}$  are “windows” through which the observed data are viewed. If the  $k$ th row of  $\mathbf{N}$  has a 1 in the  $k$ th column and zeroes elsewhere, the  $k$ th observation is perfectly resolved. We also say that the  $k$ th observation, in this case, provides completely independent information. For this reason,  $\mathbf{N}$  is sometimes also referred to as the *data information density matrix*. Equation

(7.82) shows that the  $k$ th predicted datum is a weighted average of all of the observations, with the weighting given by the entries in the  $k$ th row of  $\mathbf{N}$ . If the  $k$ th row of  $\mathbf{N}$  has many nonzero elements, then the  $k$ th predicted observation depends on the true value of many of the observations, and not just on the  $k$ th. The data resolution of the  $k$ th observation, then, is said to be poor.

*Some Properties of  $\mathbf{N}$  for the Generalized Inverse*

1.  $\mathbf{N} = \mathbf{G}\mathbf{G}_g^{-1} = \mathbf{U}_P\mathbf{U}_P^T$  (7.83)

Using singular-value decomposition, we have that

$$\begin{aligned} \mathbf{N} = \mathbf{G}\mathbf{G}_g^{-1} &= \mathbf{U}_P\Lambda_P\mathbf{V}_P^T \mathbf{V}_P\Lambda_P^{-1}\mathbf{U}_P^T \\ &= \mathbf{U}_P\mathbf{U}_P^T \end{aligned} \quad (7.84)$$

since  $\mathbf{V}_P^T\mathbf{V}_P = \mathbf{I}$  ( $\mathbf{V}_P$  is semiorthogonal) and  $\Lambda_P\Lambda_P^{-1} = \mathbf{I}$ .

In general,  $\mathbf{U}_P\mathbf{U}_P^T \neq \mathbf{I}_N$ . However, if  $P = N$ , then  $\mathbf{U}_P = \mathbf{U}$ , and  $\mathbf{U}_0$  is empty. Then,  $\mathbf{N} = \mathbf{U}_P\mathbf{U}_P^T = \mathbf{U}\mathbf{U}^T = \mathbf{I}_N$ , since  $\mathbf{U}$  is itself an orthogonal matrix. Thus, the condition for perfect data resolution is that  $\mathbf{U}_0$  be empty, or that  $P = N$ .

2.  $\text{trace}(\mathbf{N}) = P$  (7.85)

The proof follows that of  $\text{trace}(\mathbf{R}) = P$ :

$$\mathbf{N} = \mathbf{U}_P\mathbf{U}_P^T = [\mathbf{U}_P | \mathbf{U}_0] \begin{bmatrix} \mathbf{I}_P & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{U}_P^T \\ \mathbf{U}_0^T \end{bmatrix} \quad (7.86)$$

and

$$\text{trace} \begin{bmatrix} \mathbf{I}_P & 0 \\ 0 & 0 \end{bmatrix} = P \quad \text{Q.E.D.} \quad (7.87)$$

If, for example,

$$n_{11} + n_{22} \approx 1 \quad (7.88)$$

one might choose to form a new observation  $d'_1$  as a linear combination of  $d_1$  and  $d_2$ , given in the simplest case by

$$d'_1 = d_1 + d_2 \quad (7.89)$$

The actual linear combination of the two observations that is resolved depends on the eigenvectors in  $\mathbf{U}_P$ , or equivalently upon the structure of the data resolution matrix  $\mathbf{N}$ . In any case, the new observation  $d'$  could provide essentially independent information and could be a way to reduce the computational effort of the inverse problem by reducing  $\mathbf{N}$ . In many cases, however, the benefit of being able to average out data errors over the observations is more important than any computational savings that might come from combining observations.

$$3. \quad \sum_{j=1}^N n_{ij}^2 = \sum_{j=1}^N n_{ji}^2 = n_{ii} = \text{“importance” of the } i\text{th observation} \quad (7.90)$$

That is, the sum of squares of the entries in a row (or column, since  $\mathbf{N}$  is symmetric) of  $\mathbf{N}$  is equal to the diagonal entry in that row. Thus, as the diagonal entry gets large (close to one), the other entries in that row must become small. As the importance of a particular datum becomes large, the dependence of the predicted datum on other observations must become small.

If we further note that we can write  $\mathbf{N}$  as

$$\mathbf{N} = \begin{bmatrix} \vdots & \vdots & & \vdots \\ \mathbf{n}_1 & \mathbf{n}_2 & \cdots & \mathbf{n}_N \\ \vdots & \vdots & & \vdots \end{bmatrix} \quad (7.91)$$

where  $\mathbf{n}_i$  is the  $i$ th column of  $\mathbf{N}$ , then the predicted data  $\hat{\mathbf{d}}$  from (7.80) can be written as

$$\begin{aligned} \hat{\mathbf{d}} &= \mathbf{N}\mathbf{d} \\ &= \mathbf{n}_1 d_1 + \mathbf{n}_2 d_2 + \cdots + \mathbf{n}_N d_N \end{aligned} \quad (7.92)$$

where  $d_i$  is the  $i$ th component of  $\mathbf{d}$ . Equation (7.92) follows from (2.23)–(2.30). That is, the predicted data vector can also be thought of as the weighted sum of the columns of  $\mathbf{N}$ , with the weighting factors being given by the actual observations.

The length squared of each column of  $\mathbf{N}$  can then be thought of as the importance of  $d_i$  in the solution. The length squared of  $\mathbf{n}_i$  is given by

$$\|\mathbf{n}_i\|^2 = \sum_{j=1}^N n_{ji}^2 = n_{ii} \quad (7.93)$$

Thus, the diagonal entries in  $\mathbf{N}$  give the importance of each observation in the solution.

It can also be shown that the generalized inverse operator  $\mathbf{G}_g^{-1} = \mathbf{V}_P \mathbf{\Lambda}_P^{-1} \mathbf{U}_P^T$  minimizes the difference between  $\mathbf{N}$  and  $\mathbf{I}_N$ . Let us now turn our attention to another measure of quality for the generalized inverse, the unit covariance matrix  $[\text{cov}_u \mathbf{m}]$ .

### 7.3.4 The Unit (Model) Covariance Matrix [cov<sub>u</sub> **m**]

Any errors (noise) in the data will be mapped into errors in the estimates of the model parameters. The mapping was first considered in Section 3.7 of Chapter 3. We will now reconsider it from the generalized inverse viewpoint.

Let the error (noise) in the data **d** be  $\Delta\mathbf{d}$ . Then the error in the model parameters due to  $\Delta\mathbf{d}$  is given by

$$\Delta\mathbf{m} = \mathbf{G}_g^{-1} \Delta\mathbf{d} \quad (7.94)$$

*Step 1.* Recall from Equation (2.59) that the  $N \times N$  data covariance matrix [cov **d**] is given by

$$[\text{cov } \mathbf{d}] = \frac{1}{k-1} \sum_{i=1}^k \Delta\mathbf{d}^i [\Delta\mathbf{d}^i]^T \quad (7.95)$$

where  $k$  is the number of experiments, and  $i$  is the experiment number. The diagonal terms are the data variances and the off-diagonal terms are the covariances.

The data covariance is also written as  $\langle \Delta\mathbf{d} \Delta\mathbf{d}^T \rangle$ , where  $\langle \rangle$  denotes averaging.

*Step 2.* We seek, then, a model covariance matrix  $\langle \Delta\mathbf{m} \Delta\mathbf{m}^T \rangle = [\text{cov } \mathbf{m}]$ .

$$\begin{aligned} \langle \Delta\mathbf{m} \Delta\mathbf{m}^T \rangle &= \langle \mathbf{G}_g^{-1} \Delta\mathbf{d} [\mathbf{G}_g^{-1} \Delta\mathbf{d}]^T \rangle \\ &= \langle \mathbf{G}_g^{-1} \Delta\mathbf{d} \Delta\mathbf{d}^T [\mathbf{G}_g^{-1}]^T \rangle \end{aligned} \quad (7.96)$$

$\mathbf{G}_g^{-1}$  is not changing with each experiment, so we can take it outside the averaging, implying

$$\langle \Delta\mathbf{m} \Delta\mathbf{m}^T \rangle = \mathbf{G}_g^{-1} \langle \Delta\mathbf{d} \Delta\mathbf{d}^T \rangle [\mathbf{G}_g^{-1}]^T$$

or

$$[\text{cov } \mathbf{m}] = \mathbf{G}_g^{-1} [\text{cov } \mathbf{d}] [\mathbf{G}_g^{-1}]^T \quad (7.97)$$

The above derivation provides some of the logic behind Equation (2.63), which was introduced in Chapter 2 as magic.

*Step 3.* Finally, define a unit (model) covariance matrix [cov<sub>u</sub> **m**] by assuming that [cov **d**] =  $\mathbf{I}_N$ , that is, by assuming that all the data variances are equal to 1 and the covariances are all 0 (uncorrelated data errors). Then

$$\begin{aligned} [\text{cov}_u \mathbf{m}] &= \mathbf{G}_g^{-1} [\text{cov } \mathbf{d}][\mathbf{G}_g^{-1}]^T \\ &= \mathbf{G}_g^{-1} [\mathbf{G}_g^{-1}]^T \end{aligned} \quad (7.98)$$

*Some Properties of  $[\text{cov}_u \mathbf{m}]$*

- Using singular-value decomposition, we can write the unit model covariance matrix as

$$\begin{aligned} [\text{cov}_u \mathbf{m}] &= \mathbf{G}_g^{-1} [\mathbf{G}_g^{-1}]^T \\ &= \mathbf{V}_P \Lambda_P^{-1} \mathbf{U}_P^T [\mathbf{V}_P \Lambda_P^{-1} \mathbf{U}_P^T]^T \\ &= \mathbf{V}_P \Lambda_P^{-1} \mathbf{U}_P^T \mathbf{U}_P \Lambda_P^{-1} \mathbf{V}_P^T \\ &= \mathbf{V}_P \Lambda_P^{-2} \mathbf{V}_P^T \end{aligned} \quad (7.99)$$

This emphasizes the importance of the size of the singular values  $\lambda_i$  in determining the model parameter variances. As  $\lambda_i$  gets small, the entries in  $[\text{cov}_u \mathbf{m}]$  tend to get big (implying large model parameter estimate variances) due to the terms in  $1/\lambda^2$

Consider the  $k$ th diagonal entry in  $[\text{cov}_u \mathbf{m}]$ ,  $[\text{cov}_u \mathbf{m}]_{kk}$ , where

$$[\text{cov}_u \mathbf{m}] = \begin{bmatrix} \vdots & \vdots & & \vdots \\ \mathbf{v}_1 & \mathbf{v}_2 & \cdots & \mathbf{v}_P \\ \vdots & \vdots & & \vdots \end{bmatrix} \begin{bmatrix} 1/\lambda_1^2 & 0 & \cdots & 0 \\ 0 & 1/\lambda_2^2 & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & 1/\lambda_P^2 \end{bmatrix} \begin{bmatrix} \cdots & \mathbf{v}_1^T & \cdots \\ \cdots & \mathbf{v}_2^T & \cdots \\ \vdots & \vdots & \\ \cdots & \mathbf{v}_P^T & \cdots \end{bmatrix} \quad (7.100)$$

If we multiply out the first two matrices, we can then identify the  $kk$  entry in  $[\text{cov}_u \mathbf{m}]$  as the product of the  $k$ th row times the  $k$ th column of the resulting matrices

$$\begin{aligned} [\text{cov}_u \mathbf{m}]_{kk} &= \begin{bmatrix} v_{11}/\lambda_1^2 & v_{12}/\lambda_2^2 & \cdots & v_{1P}/\lambda_P^2 \\ \vdots & \vdots & & \vdots \\ v_{k1}/\lambda_1^2 & v_{k2}/\lambda_2^2 & \cdots & v_{kP}/\lambda_P^2 \\ \vdots & \vdots & & \vdots \\ v_{M1}/\lambda_1^2 & v_{M2}/\lambda_2^2 & \cdots & v_{MP}/\lambda_P^2 \end{bmatrix} \begin{bmatrix} v_{11} & \cdots & v_{k1} & \cdots & v_{M1} \\ v_{12} & \cdots & v_{k2} & \cdots & v_{M2} \\ \vdots & & \vdots & & \vdots \\ v_{1P} & \cdots & v_{kP} & \cdots & v_{MP} \end{bmatrix} \\ &= \sum_{i=1}^P \frac{v_{ki}^2}{\lambda_i^2} \end{aligned} \quad (7.101)$$

$\uparrow$   
 $k$ th column

Thus, as  $\lambda_i$  gets small,  $[\text{cov}_u \mathbf{m}]_{kk}$  will get large if  $v_{ki}$  is not zero. Recall that  $v_{ki}$  is the  $k$ th component in the  $i$ th eigenvector  $\mathbf{v}_i$  in  $\mathbf{V}_P$ . Thus, it is the combination of  $\lambda_i$  getting small, and  $\mathbf{v}_i$  having a nonzero component in the  $k$ th row, that makes the variance for the  $k$ th model parameter potentially very large.

2. Even if the data covariance is diagonal (i.e., all the observations have errors that are uncorrelated),  $[\text{cov}_u \mathbf{m}]$  need not be diagonal. That is, the model parameter estimates may well have nonzero covariances, even though the data have zero covariances.

For example, from Equation (7.101) above, we can see that

$$[\text{cov}_u \mathbf{m}]_{1k} = \begin{bmatrix} v_{11} / \lambda_1^2 & v_{12} / \lambda_2^2 & \dots & v_{1P} / \lambda_P^2 \end{bmatrix} \begin{bmatrix} v_{k1} \\ v_{k2} \\ \vdots \\ v_{kP} \end{bmatrix} = \sum_{i=1}^P \frac{v_{1i} v_{ki}}{\lambda_i^2} \quad (7.102)$$

Note that the term in the above equation

$$\sum_{i=1}^P \frac{v_{1i} v_{ki}}{\lambda_i^2}$$

is *not* the dot product of two *columns* of  $\mathbf{V}_P$ . In fact, even if the numerator were the dot product between columns (i.e.,  $v_{1i} v_{ik}$ ), the fact that every term is divided by  $\lambda_i^2$  would likely yield something other than 1. The numerator is the dot product of two *rows* of  $\mathbf{V}_P$  and is likely nonzero anyway.

3. Notice that  $[\text{cov}_u \mathbf{m}]$  is a function of the forward problem as expressed in  $\mathbf{G}$ , and *not* a function of the actual data. Thus, it can be useful for experimental design.

### 7.3.5 A Closer Look at Stability

The unit model covariance matrix  $[\text{cov}_u \mathbf{m}]$  is very helpful in experiment design for getting a sense of the basic stability of an inverse problem. And, the general rule of small singular values leading to instability is certainly true. However, oversimplified analysis of  $[\text{cov}_u \mathbf{m}]$  and the size of singular values can be very misleading.

First, what determines the size of singular values  $\lambda_i$ ? There are two main factors at work. The first is simply the size of  $\mathbf{m}$  and  $\mathbf{d}$ . Consider the simple  $1 \times 1$  forward problem:

$$\mathbf{G} \mathbf{m} = \mathbf{d} \quad (1.13)$$

where the expected (or reasonable) value of  $\mathbf{m}$  is 5 widgets/hr and the expected (or reasonable) value of  $\mathbf{d}$  is \$100. Then, clearly, the expected value of  $\mathbf{G}$  is 20. If  $\mathbf{G}$  were indeed 20, then

trivially  $\lambda = 20$ . Another way of looking at this is that, in some sense (and in every sense is this  $1 \times 1$  case):

$$|\mathbf{G}| \cdot |\mathbf{m}| = |\mathbf{d}| \quad (7.103)$$

and thus

$$|\mathbf{G}| = \frac{|\mathbf{d}|}{|\mathbf{m}|} \quad (7.104)$$

Thus one important control on the average size of singular values is simply the relative or average size of the values for data over the relative or average size of the values of the model parameters. Now, consider what happens if we change units for our data to kilodollars. Then, in this case,  $\mathbf{d} = 0.1$ , and the numbers in  $\mathbf{G}$  must change to reflect the new data units. By Equation (7.2a), we would now expect  $|\mathbf{G}| \approx 0.02$  and of course  $\lambda \approx 0.02$ . Is this problem now inherently more unstable? Not really. It would be if the sizes of the expected noise in  $\mathbf{d}$  were unchanged when we changed data units, but that would not make sense.

The second factor that determines the size of singular values is the degree to which columns or rows of  $\mathbf{G}$  are nearly parallel or dependent upon each other. Consider two  $2 \times 2$   $\mathbf{G}$  matrices where the average value in both cases for singular values is about 2:

$$\mathbf{G} = \begin{bmatrix} 2.00 & 2.00 \\ 1.00 & -2.00 \end{bmatrix} \quad \text{and} \quad \mathbf{G} = \begin{bmatrix} 2.00 & 2.00 \\ 2.00 & 2.05 \end{bmatrix} \quad (7.105)$$

For the first case,  $\lambda_1 = 3.00$  and  $\lambda_2 = 2.00$ , while for the second case,  $\lambda_1 = 4.025$  and  $\lambda_2 = 0.025$ . In the first case, the columns or rows of  $\mathbf{G}$  are at high angles to each other, while in the second case, the rows or columns are at low angles to each other, or nearly parallel.

Thus, two factors control the size of singular values: (1) the ratio of the average size of data to model parameters, and (2) the internal structure of  $\mathbf{G}$ , specifically the degree to which columns or rows of  $\mathbf{G}$  are nearly parallel to one another. Or, more properly, the degree to which any column or row of  $\mathbf{G}$  can be nearly written as a linear combination of other columns or rows.

The crux of the stability question is whether small or reasonable or expected noise in the data leads to acceptably small changes in the solution for the model parameters.

A blind application of  $[\text{cov}_u \mathbf{m}]$  can be very misleading. Inherent in  $[\text{cov}_u \mathbf{m}]$  is the assumption that

$$[\text{cov } \mathbf{d}] = \sigma_d^2 \mathbf{I} = \mathbf{I} \quad (7.106)$$

or that data variances all equal 1, and thus data standard deviations  $\sigma_d = 1$ . Is  $\sigma_d = 1$  “small, reasonable, or expected”? In our original example where  $d = 100$  it would represent very good

data, with an expected noise level of 1, or 1%. However, for our second example with the rescaled data of 0.1, it would represent 1000% error!

How do we know what the expected “noise” is in the solution?  $[\text{cov}_u \mathbf{m}]$  gives us this information, and more. On the diagonal of  $[\text{cov}_u \mathbf{m}]$  we have the variances for the model parameters. The  $[1, 1]$  entry in  $[\text{cov}_u \mathbf{m}]$  is  $\sigma_{m_1}^2$ , the variance for  $m_1$ . Correspondingly, the standard deviation of the error for  $m_1$  is  $\sigma_{m_1}$ . Thus, in some inverse problem if we found a solution of 50 for  $m_1$  and  $\sigma_{m_1}^2 = 10$ , then  $\sigma_{m_1} = \sqrt{10} \approx 3.3$ . That would mean our solution for  $m_1$  could be expressed as

$$m_1 = 50 \pm 3.3 \quad (7.107)$$

This may or may not be an acceptable “noise” level in the solution. For example, if you needed to know  $m_1$  to  $\pm 0.1\%$  in order for your satellite to land on Mars rather than become a Mars impactor or flyby, then knowing  $m_1$  to  $\pm 3.3$  would clearly not be acceptable. Like so many things, acceptable noise level is a problem-dependent question.

One way to look at the stability of an inverse problem is to see what a particular percentage noise level in the data does to the expected noise level in the solution. You can look at  $[\text{cov} \mathbf{d}]$  to get the data noise level. For example, the ratio of that data standard deviation to the expected or average data value

$$\frac{\sigma_{d_i}}{|d_i|} \quad (7.108)$$

can be expressed as a percentage. If  $\sigma_{d_i}^2 = 9$ , and  $|d_i| = 20$ , then  $\sigma_{d_i} / |d_i| = 3/20 = 15\%$ .

Then, looking at model parameter stability, you can look at  $[\text{cov} \mathbf{m}]$ . The ratio of the model parameter standard deviation to the expected or derived model parameter value

$$\frac{\sigma_{m_j}}{|m_j|} \quad (7.109)$$

can also be expressed as a percentage. If  $\sigma_{d_i}^2 = 16$ , and  $|m_j| = 50$ , then  $\sigma_{d_i} / |m_j| = 4/50 = 8\%$ . In this example, since a given percentage noise level in the data leads to a lower percentage noise level in this model parameter, then this part of the problem is stable. Of course, you need to look over all  $d_i, i = 1, N$ , and  $m_j, j = 1, M$  to have a full discussion of stability.

What all of this implies is that if you know  $[\text{cov} \mathbf{d}]$ , you should include this information throughout the inverse analysis. If you do not know  $[\text{cov} \mathbf{d}]$ , or just want a quick look at stability, you can assume

$$[\text{cov} \mathbf{d}] = \sigma_d^2 \mathbf{I} = \mathbf{I} \quad (7.106)$$

and use  $[\text{cov}_u \mathbf{m}]$ . You can also scale  $[\text{cov}_u \mathbf{m}]$  for other choices of  $\sigma_d^2$ . For example,  $[\text{cov}_u \mathbf{m}]$  for the second  $\mathbf{G}$  matrix example

$$\mathbf{G} = \begin{bmatrix} 2.00 & 2.00 \\ 2.00 & 2.05 \end{bmatrix} \quad (7.110)$$

is given by

$$[\text{cov}_u \mathbf{m}] \cong \begin{bmatrix} 820 & -810 \\ -810 & 800 \end{bmatrix} \quad (7.111)$$

If  $[\text{cov } \mathbf{d}]$  is given by

$$[\text{cov } \mathbf{d}] = \begin{bmatrix} 0.01 & 0.00 \\ 0.00 & 0.01 \end{bmatrix} \quad (7.112)$$

then

$$[\text{cov } \mathbf{m}] \cong \begin{bmatrix} 8.20 & -8.10 \\ -8.10 & 8.00 \end{bmatrix} \quad (7.113)$$

where we have used  $[\text{cov } \mathbf{m}]$  rather than  $[\text{cov}_u \mathbf{m}]$  because we are no longer assuming that  $[\text{cov } \mathbf{d}] = \mathbf{I}$ .

This example shows that the entries in  $[\text{cov } \mathbf{m}]$  scale with the assumed uniform data variance. You can calculate  $[\text{cov}_u \mathbf{m}]$  and then scale all entries by the assumed scaling factor for the difference between realistic data variances and assumed unit data variances.

Yet another way to look at this is to go back to Equation (7.97) defining the model covariance matrix:

$$[\text{cov } \mathbf{m}] = \mathbf{G}_g^{-1} [\text{cov } \mathbf{d}] [\mathbf{G}_g^{-1}]^T \quad (7.97)$$

If we assume that the data covariance matrix  $[\text{cov } \mathbf{d}]$  is given by

$$[\text{cov } \mathbf{d}] = \sigma_d^2 \mathbf{I} \quad (7.1067.106)$$

then Equation (7.97) becomes

$$[\text{cov } \mathbf{m}] = \sigma_d^2 \mathbf{G}_g^{-1} [\mathbf{G}_g^{-1}]^T \quad (7.114)$$

For  $[\text{cov}_u \mathbf{m}]$  we assumed  $\sigma_d^2 = 1$ , but we can get to  $[\text{cov} \mathbf{m}]$  from  $[\text{cov}_u \mathbf{m}]$  by dividing all entries in  $[\text{cov}_u \mathbf{m}]$  by  $\sigma_d^2$ .

### 7.3.6 Combining $\mathbf{R}$ , $\mathbf{N}$ , $[\text{cov}_u \mathbf{m}]$

Note that, in general,  $\mathbf{G}$ ,  $\mathbf{G}_g^{-1}$ ,  $\mathbf{R}$ ,  $\mathbf{N}$ , and  $[\text{cov}_u \mathbf{m}]$  can be written in terms of singular-value decomposition as

$$1. \quad \mathbf{G} = \mathbf{U}_P \mathbf{\Lambda}_P \mathbf{V}_P^T \quad (7.115)$$

$$2. \quad \mathbf{G}_g^{-1} = \mathbf{V}_P \mathbf{\Lambda}_P^{-1} \mathbf{U}_P^T \quad (7.116)$$

$$3. \quad \mathbf{R} = \mathbf{G}_g^{-1} \mathbf{G} = \mathbf{V}_P \mathbf{V}_P^T \quad (7.117)$$

$$4. \quad \mathbf{N} = \mathbf{G} \mathbf{G}_g^{-1} = \mathbf{U}_P \mathbf{U}_P^T \quad (7.118)$$

$$5. \quad [\text{cov}_u \mathbf{m}] = \mathbf{G}_g^{-1} [\mathbf{G}_g^{-1}]^T \\ = \mathbf{V}_P \mathbf{\Lambda}_P^{-2} \mathbf{V}_P^T \quad (7.119)$$

*Case I:  $P = M = N$*

$$\mathbf{R} = \mathbf{G}_g^{-1} \mathbf{G} = \mathbf{I}_M, \text{ since } \mathbf{G}_g^{-1} = \mathbf{G}^{-1} \quad (7.120)$$

$$\mathbf{N} = \mathbf{G} \mathbf{G}_g^{-1} = \mathbf{G} \mathbf{G}^{-1} = \mathbf{I}_N, \text{ since } \mathbf{G}_g^{-1} = \mathbf{G}^{-1} \quad (7.121)$$

$$[\text{cov}_u \mathbf{m}] = \mathbf{G}_g^{-1} [\mathbf{G}_g^{-1}]^T \\ = \mathbf{V} \mathbf{\Lambda}^{-1} \mathbf{U}^T [\mathbf{V} \mathbf{\Lambda}^{-1} \mathbf{U}^T]^T \\ = \mathbf{V} \mathbf{\Lambda}^{-1} \mathbf{U}^T \mathbf{U} \mathbf{\Lambda}^{-1} \mathbf{V}^T = \mathbf{V} \mathbf{\Lambda}^{-2} \mathbf{V}^T \quad (7.122)$$

*Case II:  $P = M < N$  (Least Squares)*

$$\mathbf{G}_g^{-1} = \mathbf{V}_P \mathbf{\Lambda}_P^{-1} \mathbf{U}_P^T = [\mathbf{G}^T \mathbf{G}]^{-1} \mathbf{G}^T \quad (7.123)$$

$$\mathbf{R} = \mathbf{G}_g^{-1} \mathbf{G} = \mathbf{V}_P \mathbf{V}_P^T = \mathbf{V} \mathbf{V}^T = \mathbf{I}_M \text{ since } P = M \quad (7.124)$$

$$\mathbf{N} = \mathbf{G}\mathbf{G}_g^{-1} = \mathbf{G}\{[\mathbf{G}^T\mathbf{G}]^{-1}\mathbf{G}^T\} = (\text{using SVD } \dots) = \mathbf{U}_P\mathbf{U}_P^T \quad (7.125)$$

$$\begin{aligned} [\text{cov}_u \mathbf{m}] &= \mathbf{G}_g^{-1}[\mathbf{G}_g^{-1}]^T \\ &= [\mathbf{G}^T\mathbf{G}]^{-1}\mathbf{G}^T\{[\mathbf{G}^T\mathbf{G}]^{-1}\mathbf{G}^T\}^T \\ &= (\text{using SVD } \dots) = \mathbf{V}_P\Lambda_P^{-2}\mathbf{V}_P^T \end{aligned} \quad (7.126)$$

*Case III:  $P = N < M$  (Minimum Length)*

$$\mathbf{G}_g^{-1} = \mathbf{V}_P\Lambda_P^{-1}\mathbf{U}_P^T = \mathbf{G}^T[\mathbf{G}\mathbf{G}^T]^{-1} \quad (7.127)$$

$$\mathbf{R} = \mathbf{G}_g^{-1}\mathbf{G} = \mathbf{G}^T[\mathbf{G}\mathbf{G}^T]^{-1}\mathbf{G} = (\text{using SVD } \dots) = \mathbf{V}_P\mathbf{V}_P^T \quad (7.128)$$

$$\mathbf{N} = \mathbf{G}\mathbf{G}_g^{-1} = \mathbf{G}\{\mathbf{G}^T[\mathbf{G}\mathbf{G}^T]^{-1}\} = (\text{using SVD } \dots) = \mathbf{U}_P\mathbf{U}_P^T = \mathbf{U}\mathbf{U}^T = \mathbf{I}_N \quad (7.129)$$

$$\begin{aligned} [\text{cov}_u \mathbf{m}] &= \mathbf{G}_g^{-1}[\mathbf{G}_g^{-1}]^T \\ &= \mathbf{G}^T[\mathbf{G}\mathbf{G}^T]^{-1}\{\mathbf{G}^T[\mathbf{G}\mathbf{G}^T]^{-1}\}^T \\ &= (\text{using SVD } \dots) = \mathbf{V}_P\Lambda_P^{-2}\mathbf{V}_P^T \end{aligned} \quad (7.130)$$

*Case IV:  $P < \min(M, N)$  (General Case)*

This is just the general case.

### 7.3.7 An Illustrative Example

Consider a system of equations  $\mathbf{G}\mathbf{m} = \mathbf{d}$  given by

$$\begin{bmatrix} 1.00 & 1.00 \\ 2.00 & 2.01 \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \end{bmatrix} = \begin{bmatrix} 2.00 \\ 4.10 \end{bmatrix} \quad (7.131)$$

Doing singular-value decomposition, one finds

$$\lambda_1 = 3.169 \quad (7.132)$$

$$\lambda_2 = 0.00316 \quad (7.133)$$

$$\mathbf{U}_P = \mathbf{U} = \begin{bmatrix} 0.446 & -0.895 \\ 0.895 & 0.446 \end{bmatrix} \quad (7.134)$$

$$\mathbf{V}_P = \mathbf{V} = \begin{bmatrix} 0.706 & -0.709 \\ 0.709 & 0.706 \end{bmatrix} \quad (7.135)$$

$$\mathbf{R} = \mathbf{V}_P \mathbf{V}_P^T = \begin{bmatrix} 1.0 & 0.0 \\ 0.0 & 1.0 \end{bmatrix} = \mathbf{I}_2 \quad (7.136)$$

$$\mathbf{N} = \mathbf{U}_P \mathbf{U}_P^T = \begin{bmatrix} 1.0 & 0.0 \\ 0.0 & 1.0 \end{bmatrix} = \mathbf{I}_2 \quad (7.137)$$

$$\mathbf{G}_g^{-1} = \begin{bmatrix} 201 & -100 \\ -200 & 100 \end{bmatrix} \quad (7.138)$$

$$\mathbf{m}_g = \mathbf{G}_g^{-1} \mathbf{d} = \begin{bmatrix} 201 & -100 \\ -200 & 100 \end{bmatrix} \begin{bmatrix} 2.0 \\ 4.1 \end{bmatrix} = \begin{bmatrix} -8.0 \\ 10.0 \end{bmatrix} \quad (7.139)$$

Note that the solution has perfect model resolution ( $\mathbf{R} = \mathbf{I}$ , and hence the solution is unique) and perfect data resolution ( $\mathbf{N} = \mathbf{I}$ , and hence the data can be fit exactly). Note also that  $P = N = M$ , and the generalized inverse is, in fact, the unique mathematical inverse.

This solution is, however, essentially meaningless if the data contain even a small amount of noise. To see this, consider the unit covariance matrix  $[\text{cov}_u \mathbf{m}]$  for this case:

$$\begin{aligned} [\text{cov}_u \mathbf{m}] &= \mathbf{G}_g^{-1} [\mathbf{G}_g^{-1}]^T = \mathbf{V}_P \Lambda_P^{-2} \mathbf{V}_P^T = \mathbf{V}_P \begin{bmatrix} 0.0996 & 0 \\ 0 & 100,300.9 \end{bmatrix} \mathbf{V}_P^T \\ &= \begin{bmatrix} 50,401 & -50,200 \\ -50,200 & 50,000 \end{bmatrix} \end{aligned} \quad (7.140)$$

These are very large covariances for  $m_1$  and  $m_2$ , which indicate that the solution, while unique and fitting the data perfectly, is very unstable, or sensitive to noise in the data. For example, suppose that  $d_2$  is 4.0 instead of 4.1 (2.5% error). Then the generalized inverse solution  $\mathbf{m}_g$  is given by

$$\mathbf{m}_g = \mathbf{G}_g^{-1} \mathbf{d} = \begin{bmatrix} 201 & -100 \\ -200 & 100 \end{bmatrix} \begin{bmatrix} 2.0 \\ 4.0 \end{bmatrix} = \begin{bmatrix} 2.0 \\ 0.0 \end{bmatrix} \quad (7.141)$$

That is, errors of less than a few percent in  $\mathbf{d}$  result in errors on the order of several hundred percent in  $\mathbf{m}_g$ . Whenever small changes in  $\mathbf{d}$  result in large changes in  $\mathbf{m}_g$ , the problem is

considered unstable. In this particular problem, if a solution is desired with a standard deviation of order 0.1, then the data standard deviations must be less than about  $5 \times 10^{-4}$ !

Another way of quantifying the instability of the inversion is with the *condition number*, defined as

$$\text{condition number} = \lambda_{\max} / \lambda_{\min} \quad (7.142)$$

For this particular problem, the condition number is approximately 1000, which indicates considerable instability. The condition number, by itself, can be misleading. If a problem has two singular values  $\lambda_1$  and  $\lambda_2$ , with  $\lambda_1 = 1,000,000$  and  $\lambda_2 = 1000$ , then  $\lambda_1/\lambda_2 = 1000$ . This problem, however, is very stable with changes of length order one in the data (do you see why?). If, however,  $\lambda_1 = 0.001$  and  $\lambda_2 = 0.000001$ , then  $\lambda_1/\lambda_2 = 1000$ , and unit length changes in the data will cause large changes in the solution. In addition to just the condition number, the absolute size of the singular values is important, especially compared to the size of the possible noise in the data.

In order to gain a better understanding of the origin of the instability, one must consider the structure of the  $\mathbf{G}$  matrix itself. For the present example, inspection shows that the columns, or rows, of  $\mathbf{G}$  are very nearly parallel to one another. For example, the angle between the vectors given by the columns is  $0.114^\circ$ , obtained by taking the dot product of the two columns. The two columns of  $\mathbf{G}$  span the two dimensional data space, and hence the data resolution is perfect, but the fact that they are nearly parallel leads to a significant instability.

It is not a coincidence, therefore, that the data eigenvector associated with the larger singular value,  $\mathbf{u}_1 = [0.446, 0.895]^T$ , is essentially parallel to the common direction given by the columns of  $\mathbf{G}$ . Nor is it a coincidence that  $\mathbf{u}_2$ , associated with the smaller singular value, is perpendicular to the almost uniform direction given by the columns of  $\mathbf{G}$ . The eigenvector  $\mathbf{u}_1$  represents a stable direction in data space as far as noise is concerned, while  $\mathbf{u}_2$  represents an unstable direction in data space as far as noise is concerned. Noise in data space parallel to  $\mathbf{u}_1$  will be damped by  $1/\lambda_1$ , while noise parallel to  $\mathbf{u}_2$  will be amplified by  $1/\lambda_2$ .

Similar arguments can be made about the rows of  $\mathbf{G}$ , which lie in model space. That is,  $\mathbf{v}_1$  is essentially parallel to the almost uniform direction given by the rows of  $\mathbf{G}$ , while  $\mathbf{v}_2$  is essentially perpendicular to the direction given by the rows of  $\mathbf{G}$ . Noise parallel to  $\mathbf{u}_1$ , when operated on by the generalized inverse, creates noise in the solution parallel to  $\mathbf{v}_1$ , while noise parallel to  $\mathbf{u}_2$  creates noise parallel to  $\mathbf{v}_2$ . Thus,  $\mathbf{v}_2$  is the unstable direction in model space.

Methods to stabilize the model parameter variances will be considered in a later section, but it will also be shown that any gain in stability is obtained at a cost in resolution. First, however, we will introduce ways to quantify  $\mathbf{R}$ ,  $\mathbf{N}$ , and  $[\text{cov}_u \mathbf{m}]$ . We will return to the above example and show specifically how stability can be enhanced while resolution is lost.

## 7.4 Quantifying the Quality of $\mathbf{R}$ , $\mathbf{N}$ , and $[\text{cov}_u \mathbf{m}]$

### 7.4.1 Introduction

In the proceeding sections we have shown that the model resolution matrix  $\mathbf{R}$ , the data resolution matrix  $\mathbf{N}$ , and the unit model covariance matrix  $[\text{cov}_u \mathbf{m}]$  can be very useful, at least in a qualitative way, in assessing the quality of a particular inversion. In this section, we will quantify these measures of quality, and show that the generalized inverse is the inverse that gives the best possible model and data resolution.

First, consider the following definitions (see Menke, page 68):

$$\text{spread}(\mathbf{R}) = \|\mathbf{R} - \mathbf{I}\|_2^2 = \sum_{i=1}^M \sum_{j=1}^M [r_{ij} - \delta_{ij}]^2 \quad (7.143)$$

$$\text{spread}(\mathbf{N}) = \|\mathbf{N} - \mathbf{I}\|_2^2 = \sum_{i=1}^N \sum_{j=1}^N [n_{ij} - \delta_{ij}]^2 \quad (7.144)$$

and

$$\text{size}([\text{cov}_u \mathbf{m}]) = \sum_{i=1}^M [\text{cov}_u \mathbf{m}]_{ii} \quad (7.145)$$

The spread function measures how different  $\mathbf{R}$  (or  $\mathbf{N}$ ) is from an identity matrix. If  $\mathbf{R}$  (or  $\mathbf{N}$ ) =  $\mathbf{I}$ , then  $\text{spread}(\mathbf{R})$  (or  $\mathbf{N}$ ) = 0. The size function is the trace of the unit model covariance matrix, which gives the sum of the model parameter variances.

We can now look at the spread and size functions for various classes of problems.

### 7.4.2 Classes of Problems

*Class I:  $P = N = M$*

$\text{spread}(\mathbf{R}) = \text{spread}(\mathbf{N}) = 0$ $\text{size}([\text{cov}_u \mathbf{m}])$	perfect model and data resolution depends on the size of the singular values
---	---

*Class II:  $P = M < N$  (Least Squares)*

$\text{spread}(\mathbf{R}) = 0$ $\text{spread}(\mathbf{N}) \neq 0$	perfect model resolution data not all independent
---	--

size ( $[\text{cov}_u \mathbf{m}]$ ) depends on the size of the singular values

*Class III:  $P = N < M$  (Minimum Length)*

spread ( $\mathbf{R}$ ) $\neq 0$	nonunique solution
spread ( $\mathbf{N}$ ) = 0	perfect data resolution
size ( $[\text{cov}_u \mathbf{m}]$ )	depends on the size of the singular values

*Class IV:  $P < \min(N, M)$  (General Case)*

spread ( $\mathbf{R}$ ) $\neq 0$	nonunique solution
spread ( $\mathbf{N}$ ) $\neq 0$	data not all independent
size ( $[\text{cov}_u \mathbf{m}]$ )	depends on the size of the singular values

We also note that the position of an off-diagonal nonzero entry in  $\mathbf{R}$  or  $\mathbf{N}$  does not affect the spread. This is as it should be if the model parameters and data have no physical ordering.

### 7.4.3 Effect of the Generalized Inverse Operator $\mathbf{G}_g^{-1}$

We are now in a position to show that the generalized inverse operator  $\mathbf{G}_g^{-1}$  gives the best possible  $\mathbf{R}$ ,  $\mathbf{N}$  matrices in terms of minimizing the spread functions as defined in (7.143)–(7.144). Menke (pp. 68–70) does this for the  $P = M < N$  case, and less fully for the  $P = N < M$  case. Consider instead, the more general derivation (after Jackson, 1972). For any estimate of the inverse operator  $\mathbf{G}_{\text{est}}^{-1}$ , the model resolution matrix  $\mathbf{R}$  is given by

$$\begin{aligned} \mathbf{R} &= \mathbf{G}_{\text{est}}^{-1} \mathbf{G} \\ &= \mathbf{G}_{\text{est}}^{-1} \mathbf{U}_P \Lambda_P \mathbf{V}_P^T \\ &= \mathbf{B} \mathbf{V}_P^T \end{aligned} \tag{7.146}$$

where  $\mathbf{B} = \mathbf{G}_{\text{est}}^{-1} \mathbf{U}_P \Lambda_P$ . From (2.23)–(2.30), each row of  $\mathbf{R}$  will be a linear combination of the rows of  $\mathbf{V}_P^T$ , or equivalently a linear combination of the columns of  $\mathbf{V}_P$ . The weighting factors are determined by  $\mathbf{B}$ , which depends on the choice of the inverse operator.

The goal, then, is to choose an inverse operator that will make  $\mathbf{R}$  most like the identity matrix  $\mathbf{I}$  in the sense of minimizing spread ( $\mathbf{R}$ ). Define  $\mathbf{b}_k^T$  as the  $k$ th row of  $\mathbf{B}$ , and  $\mathbf{d}_k^T$  as the  $k$ th row of  $\mathbf{I}_M$ . We seek  $\mathbf{b}_k^T$  as the least squares solution to

$$\begin{array}{ccc} \mathbf{b}_k^T & \mathbf{V}_P^T & = \mathbf{d}_k^T \\ 1 \times P & P \times M & 1 \times M \end{array} \tag{7.147}$$

Taking the transposes implies

$$\begin{matrix} \mathbf{V}_P & \mathbf{b}_k & = & \mathbf{d}_k \\ M \times P & P \times 1 & & M \times 1 \end{matrix} \quad (7.148)$$

Equation (7.148) can be solved with the least squares operator [see (3.22)] as

$$\begin{aligned} \mathbf{b}_k &= [\mathbf{V}_P^T \mathbf{V}_P]^{-1} \mathbf{V}_P^T \mathbf{d}_k \\ &= \mathbf{I}^{-1} \mathbf{V}_P^T \mathbf{d}_k \\ &= \mathbf{V}_P^T \mathbf{d}_k \end{aligned} \quad (7.149)$$

Taking the transpose of (7.149) gives

$$\mathbf{b}_k^T = \mathbf{d}_k^T \mathbf{V}_P \quad (7.150)$$

Writing this out specifically, we have

$$[b_{k1} \quad b_{k2} \quad \cdots \quad b_{kP}] = [0 \quad \cdots \quad 0 \quad \underset{\uparrow}{1} \quad 0 \quad \cdots \quad 0] \begin{bmatrix} v_{11} & v_{12} & \cdots & v_{1P} \\ v_{21} & v_{22} & \cdots & v_{2P} \\ \vdots & \vdots & & \vdots \\ v_{M1} & v_{M2} & \cdots & v_{MP} \end{bmatrix} \quad (7.151)$$

Looking at (7.151), we see that the  $i$ th entry in  $\mathbf{b}_k^T$  is given by

$$b_{ki} = v_{ki} \quad (7.152)$$

That is, each element in the  $k$ th row of  $\mathbf{B}$  is the corresponding element in the  $k$ th row of  $\mathbf{V}_P$ . Or, simply put, the  $k$ th row of  $\mathbf{B}$  is given by the  $k$ th row of  $\mathbf{V}_P$ .

Making similar arguments for each row of  $\mathbf{B}$  gives us

$$\mathbf{B} = \mathbf{V}_P \quad (7.153)$$

Substituting  $\mathbf{B}$  back into (7.146) gives

$$\mathbf{R} = \mathbf{V}_P \mathbf{V}_P^T \quad (7.154)$$

This is, however, exactly the model resolution matrix for the generalized inverse, given in (7.65). Thus, we have shown that the generalized inverse is the operator with the best model resolution in the sense that the least squares difference between  $\mathbf{R}$  and  $\mathbf{I}_M$  is minimized. Very similar arguments can be made that show that the generalized inverse is the operator with the best data resolution in the sense that the least squares difference between  $\mathbf{N}$  and  $\mathbf{I}_N$  is minimized.

In cases where the model parameters or data have a natural ordering, such as a discretization of density versus depth (for model parameters) or gravity measurements along a profile (for data), we might want to modify the definition of the spread functions in (7.143)–(7.144). One such modification leads to the Backus–Gilbert inverse. A modified spread function is defined by

$$\text{spread}(\mathbf{R}) = \sum_{i=1}^M \sum_{j=1}^M W(i, j) [r_{ij} - \delta_{ij}]^2 \quad (7.155)$$

where  $W(i, j) = (i - j)^2$ . This gives more weight (penalty) to entries far from the diagonal. It has the effect, however, of canceling out any  $i = j$  contribution to the spread. To handle this, a constraint equation is added and satisfied by the use of Lagrange multipliers. The constraint equation is given by

$$\sum_{j=1}^M r_{ij} = 1 \quad (7.156)$$

This ensures that not all entries in the row of  $\mathbf{R}$  are allowed to go to zero, which would minimize the spread in (7.155). The inverse operator based on (7.155) is called the Backus–Gilbert inverse, first developed for continuous (rather than discrete) problems.

## 7.5 Resolution Versus Stability

### 7.5.1 Introduction

We will see in this section that stability can be improved by removing small singular values from an inversion. We will also see, however, that this reduces the resolution. There is an unavoidable trade-off between solution stability and resolution.

Recall the example from Equation (7.131)

$$\begin{bmatrix} 1.00 & 1.00 \\ 2.00 & 2.01 \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \end{bmatrix} = \begin{bmatrix} 2.00 \\ 4.10 \end{bmatrix} \quad (7.131)$$

The singular values, eigenvector matrices, generalized inverse, and other relevant matrices are given in Equation (7.139).

One option is to arbitrarily set  $\lambda_2 = 0$ . Then  $P$  is reduced from 2 to 1, and

$$\mathbf{U}_P = \begin{bmatrix} 0.446 \\ 0.895 \end{bmatrix} \quad (7.157)$$

$$\mathbf{V}_P = \begin{bmatrix} 0.706 \\ 0.709 \end{bmatrix} \quad (7.158)$$

$$\mathbf{R} = \mathbf{V}_P \mathbf{V}_P^T = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix} \quad (7.159)$$

$$\mathbf{N} = \mathbf{U}_P \mathbf{U}_P^T = \begin{bmatrix} 0.2 & 0.4 \\ 0.4 & 0.8 \end{bmatrix} \quad (7.160)$$

$$\mathbf{G}_g^{-1} = \mathbf{V}_P \mathbf{\Lambda}_P^{-1} \mathbf{U}_P^T = \begin{bmatrix} 0.099 & 0.199 \\ 0.100 & 0.200 \end{bmatrix} \quad (7.161)$$

$$[\text{cov}_u \mathbf{m}] = \mathbf{V}_P \mathbf{\Lambda}_P^{-2} \mathbf{V}_P = \begin{bmatrix} 0.0496 & 0.0498 \\ 0.0498 & 0.0500 \end{bmatrix} \quad (7.162)$$

$$\mathbf{m}_g = \mathbf{G}_g^{-1} \mathbf{d} = \begin{bmatrix} 1.016 \\ 1.020 \end{bmatrix} \quad (7.163)$$

and

$$\hat{\mathbf{d}} = \mathbf{G} \mathbf{m}_g = \begin{bmatrix} 2.04 \\ 4.08 \end{bmatrix} \quad (7.164)$$

First, note that the size of the unit model covariance matrix has been significantly reduced, indicating a dramatic improvement in stability in the solution. The model parameter variances are order 0.05 for data with unit variance.

Second, note that the fit to the data, while not perfect, is fairly close. The misfits for  $d_1$  and  $d_2$  are, at most, 2%.

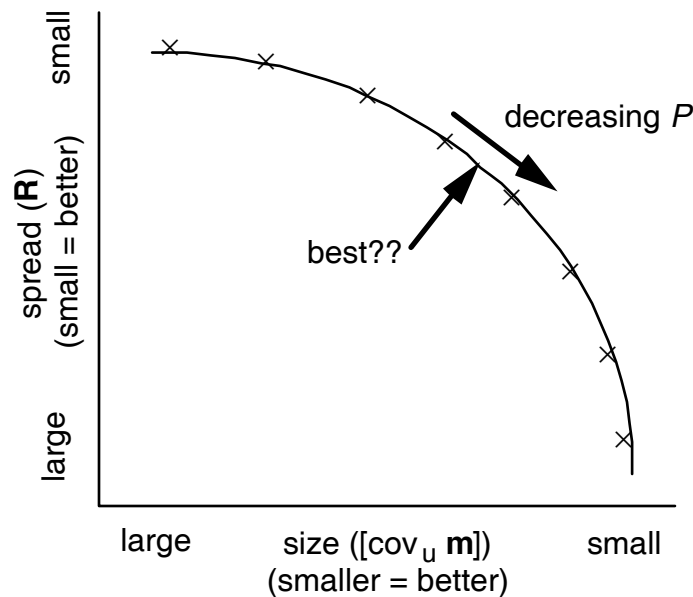
Third, however, note that both model and data resolution have been degraded from perfect resolution when both singular values were retained. In fact,  $\mathbf{R}$  now indicates that the estimates for both  $m_1$  and  $m_2$  are given by the average of the true values of  $m_1$  and  $m_2$ . That is,  $0.706m_1$  plus  $0.709m_2$  is perfectly resolved, but there is no information about the difference. This can also be seen by examining  $\mathbf{V}_P$ , which points in the  $[0.706, 0.709]^T$  direction in model space. This is the only direction in model space that can be resolved.

Recall on page 165 that when  $d_2$  was changed from 4.1 to 4.0, the solution changed from  $[-8, 10]^T$  to  $[2, 0]^T$ . The sum of  $m_1$  and  $m_2$  remained constant, but the difference changed significantly. If  $d_2$  is changed from 4.1 to 4.0 now, the solution is  $[0.998, 1.000]^T$ , very close to the solution with  $d_2 = 4.1$ .

In some cases, knowing the sum of  $m_1$  and  $m_2$  may be useful, such as when  $\mathbf{m}$  gives the velocity of some layered structure. Then knowing the average velocity, even if the individual layer velocities cannot be resolved, may be useful. In any case, we have shown that the original decomposition, with two nonzero singular values, was so unstable that the solution, while unique, was essentially meaningless.

The data resolution matrix  $\mathbf{N}$  indicates that the second observation is more important than the first (0.8 versus 0.2 along the diagonal). This can be seen either from noting that the second row of either column of  $\mathbf{G}$  is larger than the first row, and  $\mathbf{U}_P$  is formed as a linear combination of the columns of  $\mathbf{G}$ , or by looking at  $\mathbf{U}_P$ , which points in the  $[0.446, 0.895]^T$  direction in data space.

Another way to look at the trade-off is by plotting resolution versus stability as shown on the next page:



As  $P$  is decreased, by setting small singular values to zero, the resolution degrades while the stability increases. Sometimes it is possible to pick an optimal cut-off value for small singular values based on this type of graph.

### 7.5.2 $\mathbf{R}$ , $\mathbf{N}$ , and $[\text{cov}_u \mathbf{m}]$ for Nonlinear Problems

The resolution matrices and the unit model covariance matrix are also useful in a nonlinear analysis, although the interpretations are somewhat different than they are for the linear case.

*Model Resolution Matrix  $\mathbf{R}$*

In the linear case the solution is unique whenever  $\mathbf{R} = \mathbf{I}$ . For the nonlinear problem, a unique solution is not guaranteed, even if  $\mathbf{R} = \mathbf{I}$ . In fact, no solution may exist, even when  $\mathbf{R} = \mathbf{I}$ . Consider the following simple nonlinear problem:

$$m^2 = d_1 \tag{7.165}$$

With a single model parameter and a single observation, we have  $P = M = N$ . Thus,  $\mathbf{R} = \mathbf{I}$  at every iteration. If  $d_1 = 4$ , the process will iterate successfully to the solution  $m_1 = 2$  unless, by chance, the iterative process ever gives  $m_1$  exactly equal to zero, in which case the inverse is undefined. However, if  $d_1$  is negative, there is no real solution, and the iterative process will never converge to an answer, even though  $\mathbf{R} = \mathbf{I}$ .

The uniqueness of nonlinear problems also depends on the existence of local minima. It is always a good idea in nonlinear problems to explore solution space to make sure that the solution obtained corresponds to a global minima. Take, for example, the following case with two observations and two model parameters:

$$m_1^4 + m_2^2 = 2 \tag{7.166}$$

$$m_1^2 + m_2^4 = 2 \tag{7.167}$$

This simple set of two nonlinear equations in two unknowns has  $\mathbf{R} = \mathbf{I}$  almost everywhere in solution space. By inspection, however, there are four solutions that fit the data exactly, given by  $[m_1, m_2]^T = [1, 1]^T, [1, -1]^T, [-1, 1]^T$ , and  $[-1, -1]^T$ , respectively.

To see the role of the model resolution matrix for a nonlinear analysis, recall Equations (4.13)–(4.17), where, for example,

$$\Delta \mathbf{c} = \mathbf{G} \Delta \mathbf{m} \tag{4.14}$$

and where  $\Delta \mathbf{c}$  is the misfit to the data, given by the observed minus the predicted data,  $\Delta \mathbf{m}$  are changes to the model at this iteration, and  $\mathbf{G}$  is the matrix of partial derivatives of the forward equations with respect to the model parameters. If  $\mathbf{R} = \mathbf{I}$  at the solution, then the changes  $\Delta \mathbf{m}$  are perfectly resolved in the close vicinity of the solution. If  $\mathbf{R} \neq \mathbf{I}$ , then there will be directions in model space (corresponding to  $\mathbf{V}_0$ ) that do not change the predicted data, and hence the fit to the data. All of this analysis, of course, is based on the linearization of a nonlinear problem in the vicinity of the solution. The analysis of  $\mathbf{R}$  is only as good as the linearization of the nonlinear problem. If the solution is very nonlinear at the solution, the validity of conclusions based on an analysis of  $\mathbf{R}$  may be suspect.

Note also that  $\mathbf{R}$ , which depends on both  $\mathbf{G}$  and the inverse operator, may change during the iterative process. For example, in the Equation (7.167) above, we noted that  $\mathbf{R} = \mathbf{I}$  almost everywhere in solution space. At the point given by  $[m_1, m_2]^T = [0.7071, 0.7071]^T$ , you may

verify for yourselves that all four entries in the  $\mathbf{G}$  matrix of partial derivatives are equal to 1. In this case,  $P$  is reduced from 2 to 1. The next iteration, however, will take the solution to somewhere else where the resolution matrix is again an identity matrix. The analysis of  $\mathbf{R}$  is thus generally reserved for the final iteration at the solution. At intermediate steps,  $\mathbf{R}$  determines whether there is a unique *direction* in which to move toward the solution. Since the path to the solution is less critical than the final solution, little emphasis is generally placed on  $\mathbf{R}$  during the iterative process.

The generalized inverse operator, which finds the minimum length solution for  $\Delta\mathbf{m}$ , finds the smallest possible change to the linearized problem to minimize the misfit to the data. This is a benefit because large changes in  $\Delta\mathbf{m}$  will take the estimated parameter values farther away from the region where the linearization of the problem is valid.

### *Data Resolution Matrix $\mathbf{N}$*

In the linear case,  $\mathbf{N} = \mathbf{I}$  implies perfectly independent (and resolved) data. In the nonlinear case,  $\mathbf{N} = \mathbf{I}$  implies that the misfit  $\Delta\mathbf{c}$ , and not necessarily the data vector  $\mathbf{d}$  itself, is perfectly resolved for the linearized problem. If  $\mathbf{N} \neq \mathbf{I}$ , then any part of the misfit  $\Delta\mathbf{c}$  that lies in  $\mathbf{U}_0$  space will not contribute to changes in the model parameter estimates. In the vicinity of the solution, if  $\mathbf{N} = \mathbf{I}$ , then data space is completely resolved, and the misfit should typically go to zero. If  $\mathbf{N} \neq \mathbf{I}$  at the solution, then there may be a part of the data that cannot be fit. But, even if  $\mathbf{N} = \mathbf{I}$ , there is no guarantee that there is any solution that will fit the data exactly. Recall the example in Equation (7.165) above where  $\mathbf{N} = \mathbf{I}$  everywhere. If  $d_1$  is negative, no real solution can be found that fits the data exactly.

As with the model resolution matrix,  $\mathbf{N}$  is most useful at the solution and less useful during the iterative process. Also, it should always be recalled that the analysis of  $\mathbf{N}$  is only as valid as the linearization of the problem.

### *The Unit Model Covariance Matrix [ $\text{cov}_u \mathbf{m}$ ]*

For a linear analysis, the unit model covariance provides variance estimates for the model parameters assuming unit, uncorrelated data variances. For the nonlinear case, the unit model covariance matrix provides variance estimates for the model parameter changes  $\Delta\mathbf{m}$ . At the solution, these can be interpreted as variances for the model parameters themselves, as long as the problem is not too nonlinear. Along the iterative path, and at the final solution, the unit covariance matrix provides an estimate of the stability of the process. If the variances are large, then there must be small singular values, and the misfit may be mapped into large changes in the model parameters. Analysis of particular model parameter variances is usually reserved for the final iteration. As with both the resolution matrices, the model parameter variance estimates are based on the linearized problem, and are only as good as the linearization itself.

Consider a simple  $N = M = 1$  nonlinear forward problem given by

$$d = m^{1/3} \quad (7.168)$$

The inverse solution (exact, or equivalently the generalized inverse) is of course given by

$$m = d^3 \quad (7.169)$$

These relationships are shown in the figures on the next page.

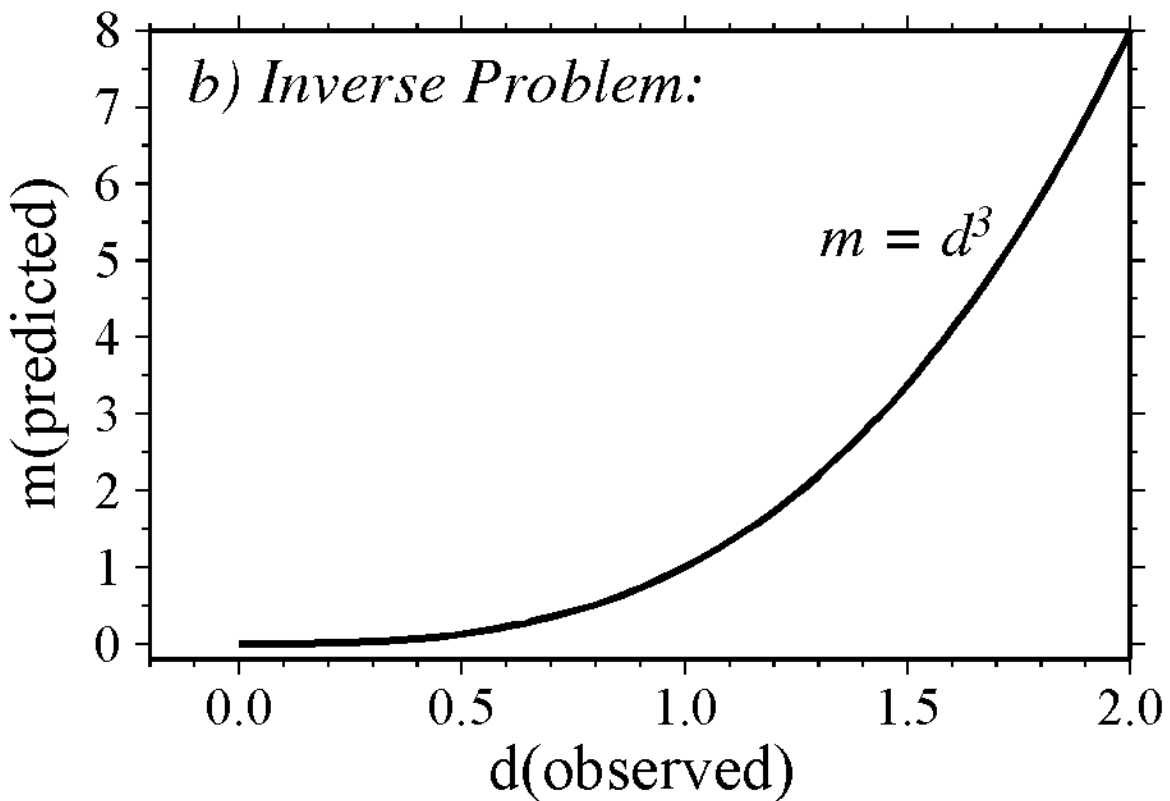
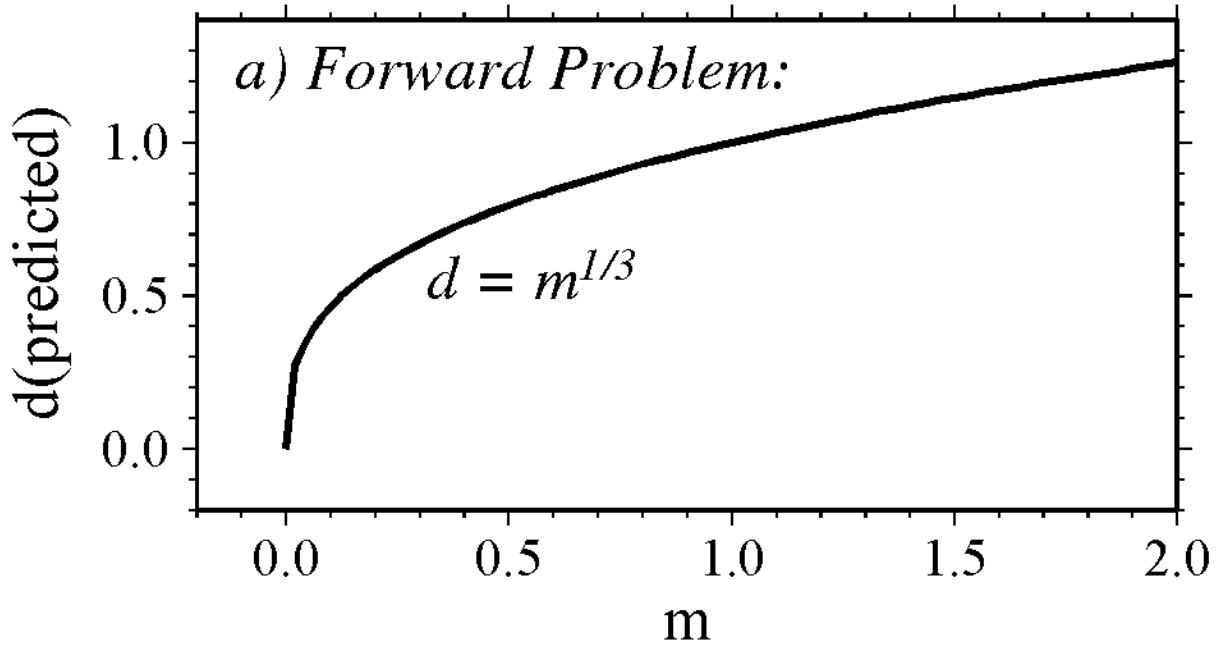
Suppose we consider a case where  $d^{\text{true}} = 1$ . The true solution is  $m = 1$ . A generalized inverse analysis leads to a linearized estimate of the uncertainty in the solution,  $[\text{cov}_u \mathbf{m}]$ , of 9. This analysis assumes uncorrelated Gaussian noise with mean zero and variance 1. If we use Gaussian noise with mean zero and standard deviation  $\sigma = 0.25$  (i.e., variance = 0.0625) then  $[\text{cov} \mathbf{m}] = 0.5625$ . The simple nature of this problem leads to an amplification by a factor of 9 between the data variance and the linearized estimate of the solution variance.

Now consider an experiment in which 50,000 noisy data values are collected. The noise in these data has a mean of 0.0 and a standard deviation of 0.25. For each noisy data value a solution is found from the above equations. This will create 50,000 estimates of the solution. Distributions of both the data noise and the solution are shown in the figures after those for the forward and inverse problems in Equations (7.168)–(7.169).

Note that due to the nonlinear nature of the forward problem, the distribution of solutions is not even Gaussian. The mean value,  $\langle m \rangle$ , is 1.18, greater than the “true” value of 1. The standard deviation is 0.84. Also shown on the figure is the maximum likelihood solution  $\mathbf{m}_{\text{ML}}$ , as determined empirically from the distribution.

The purpose of this example is to show that caution must be applied to the interpretation of all inverse problems, but especially nonlinear ones.

# Forward and Inverse Problems



# Mapping Noisy Data: 50,000 Experiments

