RI 9361

REPORT OF INVESTIGATIONS/1991

Direct Tomographic Reconstruction and Applications to Mining

By W. P. Stroud and R. S. Dennen

UNITED STATES DEPARTMENT OF THE INTERIOR

BUREAU OF MINES

Mission: As the Nation's principal conservation agency, the Department of the Interior has responsibility for most of our nationally-owned public lands and natural and cultural resources. This includes fostering wise use of our land and water resources, protecting our fish and wildlife, preserving the environmental and cultural values of our national parks and historical places, and providing for the enjoyment of life through outdoor recreation. The Department assesses our energy and mineral resources and works to assure that their development is in the best interests of all our people. The Department also promotes the goals of the Take Pride in America campaign by encouraging stewardship and citizen responsibility for the public lands and promoting citizen participation in their care. The Department also has a major responsibility for American Indian reservation communities and for people who live in Island Territories under U.S. Administration.

Report of Investigations 9361

Direct Tomographic Reconstruction and Applications to Mining

By W. P. Stroud and R. S. Dennen

UNITED STATES DEPARTMENT OF THE INTERIOR
Manuel Lujan, Jr., Secretary

BUREAU OF MINES
T S Ary, Director

Library of Congress Cataloging in Publication Data:

Stroud, W. P.
Direct tomographic reconstruction and applications to mining / by W. P. Stroud and R. S. Dennen.

p. cm. — (Report of investigations; 9361)

Includes bibliographical references (p. 11).

Supt. of Docs. no.: I 28.23:9361.

1. Mine safety. 2. Tomography. I. Dennen, R. S. II. Title. III. Series: Report of investigations (United States. Bureau of Mines); 9361.

TN23.U43 [TN295] 622 s—dc20 [622'.8] 90-22804 CIP

CONTENTS

ILLUSTRATIONS

TABLES

UNIT OF MEASURE ABBREVIATIONS USED IN THIS REPORT

m	meter	s	second
ns	nanosecond		

DIRECT TOMOGRAPHIC RECONSTRUCTION AND APPLICATIONS TO MINING

By W. P. Stroud[1] and R. S. Dennen[2]

ABSTRACT

In this U.S. Bureau of Mines report, the conditions required for the direct reconstruction of a tomogram from observed line integrals are examined. Because of the limited access to the working faces in coal mines, the special case of access to only two opposite faces is emphasized. Classical matrix methods were used for small dimensional (2 by 2, 3 by 3, and 4 by 4) applications to determine the limitations and the requirements for a solution. The results for large (n by n) grids were then inferred and generalized from those of the smaller grids and matrices. Limited access tomography is found to be appropriate for in-seam hazard detection from seismic and microwave data, but it is shown that additional data must be added to obtain a direct solution.

[1]Physical scientist.
[2]Geophysicist.
Denver Research Center, U.S. Bureau of Mines, Denver, CO.

INTRODUCTION

The U.S. Bureau of Mines has been developing a ground-penetrating radar system for hazard detection in coal mines as part of its research program to increase the safety of mining operations. The application of tomography to mine hazard detection offers the possibility of seeing in-seam anomalies such as cracks, voids, and inclusions from data taken only from the outer surface(s). Because of specialized mining operations and techniques, internal structure observations are often limited to access to only one or two surfaces. For example, in longwall mining, only two surfaces might be available to any external instrumentation, i.e., the head and tailgate entries. If the working face is also made available, the analytical problem is simplified, but such access is often not available during longwall production. It is also possible that only one face might be available, either the working face itself or one of the side entries. Because of practical considerations, limited penetration with a tolerable signal-to-noise ratio could also limit access to only one face. Single-face access leads to reflection tomography, which is not considered here. In the case of room-and-pillar mining, access to all four faces may be available. These cases are depicted in figure 1. Here the case in figure 1*A* is emphasized.

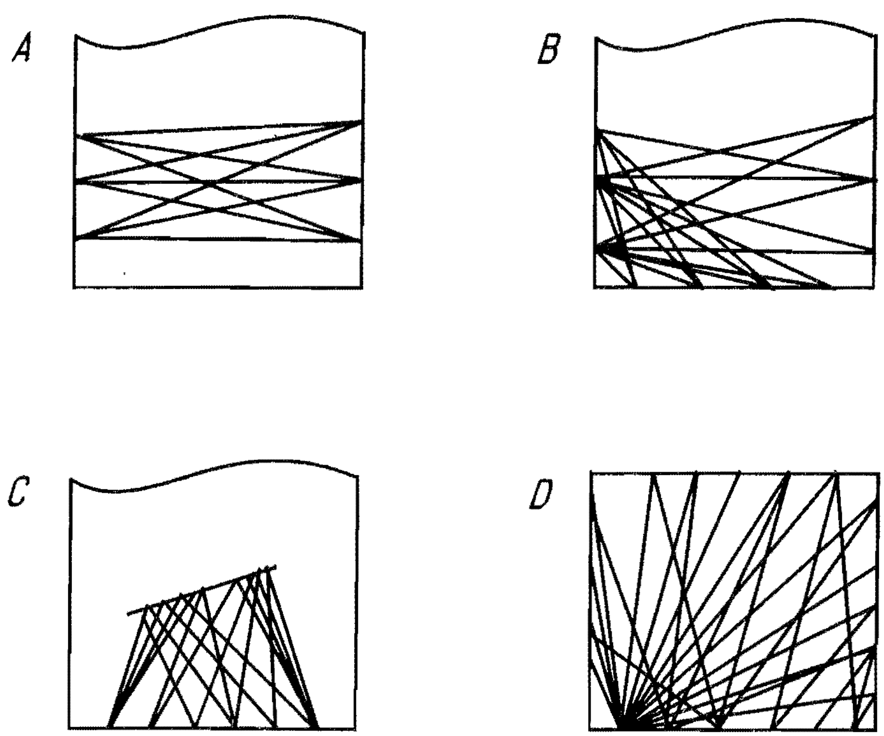

Figure 1.—Schematic of surface access in mining operations. ***A*****, Longwall (working face) 2-face access;** ***B*****, longwall (working face) 3-face access;** ***C*****, longwall (working face) 1-face access reflection tomography;** ***D*****, pillar 4-face access.**

BACKGROUND

A detailed view of the tomographic problem is depicted in figure 2, where a 2 by 2 grid is considered to superpose a pillar. Each cell or resolution element is labeled in matrix notation c_{ij}, where the "i" and "j" refer to the row and column of the i,jth cell whose properties are considered uniform, and a solution consists of determining these properties for each cell. In any probing system, the material is penetrated by some energy source, for example, microwave or radar energy. What is to be learned about the interior of the structure must be inferred from the signals originating at the transmitter T_k, entering the receiver R_ℓ, over the path $P_{k\ell}$, where k and ℓ are generalized subscripts. The information received is in the form of a line integral because each received signal contains information from each of the cells along the line traversed by that path. Then, for example, if the cellular velocities are of interest, the time of traverse of each of the k,ℓ paths is measured, and the line integral represents the time ($T_{k\ell}$) for the k,ℓth path and is

$$T_{k\ell} = \int_{t_k}^{t_\ell} dt. \tag{1}$$

Because the cells or resolution elements are considered finite, equation 1 is more conveniently represented as a summation along the k,ℓth path or a line summation. Then,

$$T_{k\ell} = \sum_{ij} \Delta t_{ij}^{k\ell}, \tag{2}$$

where the summation is taken over each of the ij cells traversed by the k,ℓth path. To simplify the matrix notation, both the paths k,ℓ and the cells i,j can be degenerated into the single subscript system m,n denoting the mth path and nth cell. An m by n matrix may then be considered for n cells traversed by m paths. Then equation 2 becomes

$$T_m = \sum_n \Delta t_{mn} \tag{3}$$

or

$$T_m = \sum_n t_{mn}. \tag{4}$$

Each component, t_{mn}, represents the time taken to traverse the mth path through the nth cell, and the summation, T_m, is the total traverse time for the mth path.

Since each time increment, t_{mn} is related to the velocity in the nth cell, v_n,

$$t_{mn} = d_{mn}/v_n, \tag{5}$$

where d_{mn} represents the distance traversed by the mth path through the nth cell.

In matrix form, equation 5 becomes

$$[T] = [D][\frac{1}{v}], \tag{6}$$

where T and 1/v are, respectively, m × 1 and n × 1 column vectors. The vector T represents the observables (i.e., the line summations), and [1/v] the material properties, in this case inverse velocities or slowness factors in the n cells. Each element of the slowness vector, [1/v], is the inverse or reciprocal of the cellular velocity, $1/v_n$, and because of this relationship, is sometimes called the velocity vector.

In matrix form, the solution to equation 6 is

$$[\frac{1}{v}] = D^{-1}T, \tag{7}$$

for a square matrix, i.e., n = m, where D^{-1} is the inverse of the distance matrix whose elements contain the d_{mn} elements.

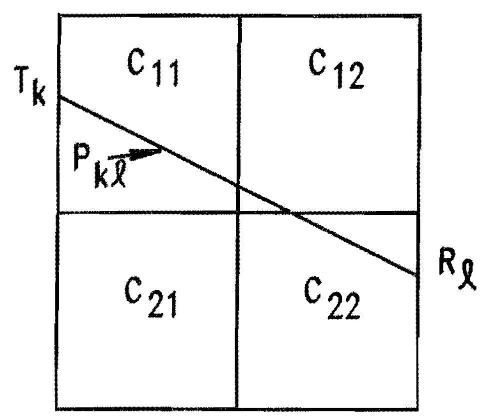

Figure 2.—Pillar superposed by 2 by 2 grid. Path ($P_{k\ell}$) crosses cell elements C_{11}, C_{12}, C_{21}, and C_{22} from transmitter (T_k) to receiver (R_ℓ).

The simplified solution in equation 7 applies only to a square matrix. In the general case for $n \neq m$, it is possible to reformulate equation 6 when D^{-1} does not exist. Then from equation 6,

$$D^T T = D^T D[\frac{1}{V}],$$

$$[D^T D]^{-1} D^T T = [\frac{1}{V}], \qquad (8)$$

where D^T is the transpose of the nonsquare D matrix. It can be shown that equation 8 represents a least squares solution to the problem originally posed in equation 6 (*5-6*)[3]. While equation 7 represents a simplified analytical method of solving for the n cell velocities, it contains several potential pitfalls. First, what are the conditions for which D^{-1} exists? Several cases of practical interest were examined, and these are discussed in the next section. Second, how are these conditions of existence of D^{-1} affected by the practical considerations of making the necessary measurements of the related T_m components of the T vector?

The direct solution, equation 7 or equation 8 is of practical importance because of the possibility of having a real-time solution available without the need for an initial estimate of the cellular velocities.

DIRECT SOLUTION FOR MATERIAL PROPERTIES

To determine the D^{-1} matrix for the direct solution of equation 6 requires that D be square (n = m), i.e., that the number of line integral equations (m paths) be equal to the number of cells, n, of unknown properties V_n, and that the determinant of D exists, i.e., $|D| \neq 0$.

Since there are n cells, n independent equations of the form of equation 1 are required to uniquely determine the n independent variables, V_{ij} (or V_n) (*1*). Accordingly, a total of n transmitter-receiver paths are needed for a solution.

In practice, however, equation 7 frequently fails to yield unique, positive solutions. Investigation showed that the indeterminate solutions were associated with a singular [D] matrix. The cause of this singularity is linear dependence of the row vectors constituting the [D] matrix (*2*). Since the elements of [D], d_{ij}, represent the lengths of each transmitter-receiver path through the individual cells, the ultimate cause for the failure of the matrix inversion is found to be the scanning geometry itself.

Here the definition of linear dependence is taken as that given by Bellman (*3*). If X_1, X_2----X_k represent a set of k N-dimensional vectors, and if a set of scalars, C_1, C_2----$,C_k$, at least one of which is nonzero, exists with the property that

$$C_1X_1 + C_2X_2 ---- + C_kX_k = 0, \qquad (9)$$

then the vectors are linearly dependent.

Recalling that n independent equations are required for an exact solution, it is evident that if any of the n equations in the set are linearly dependent on the others, sufficient information will not be available to uniquely determine values of the independent variable V_{ij}. The solution will be indeterminate, and $|D| = 0$. Furthermore, if $|D| = 0$, it will be impossible to determine $[D]^{-1}$, since

$$[D]^{-1} = \frac{\overset{\Delta}{[D]}}{|D|}, \qquad (10)$$

where $\overset{\Delta}{[D]}$ is the conjugate of [D].

An effective tool used to determine linear dependence is the row-echelon reduction method. This method reduces a set of vectors to only those which are linearly independent (*4*).

The number of nonzero row vector entries into the D matrix produced by a given ray path is equal to the number of cells traversed by that path. For an n by n grid then, the number of nonzero entries may take values from n to 2n-1. For example (fig. 3), a ray path across opposite, parallel sides of a 3 by 3 grid may traverse three, four, or a maximum of five pixels, as shown in figures 3*A*, 3*B*, and 3*C*, respectively. Table 1 summarizes the number of distinct families of paths possible from opposite sides of grids up to dimension 4 by 4, but these paths are not necessarily linearly independent.

Table 1.—Summary of possible paths

Grid size	Cells cut	Paths: Each cut	Paths: Total
2 by 2 . .	2	4	8
	3	4	
3 by 3 . .	3	13	35
	4	16	
	5	6	
4 by 4 . .	4	28	112
	5	36	
	6	20	
	7	28	

[3]Italic numbers in parentheses refer to items in the list of references preceding the appendix at the end of this report.

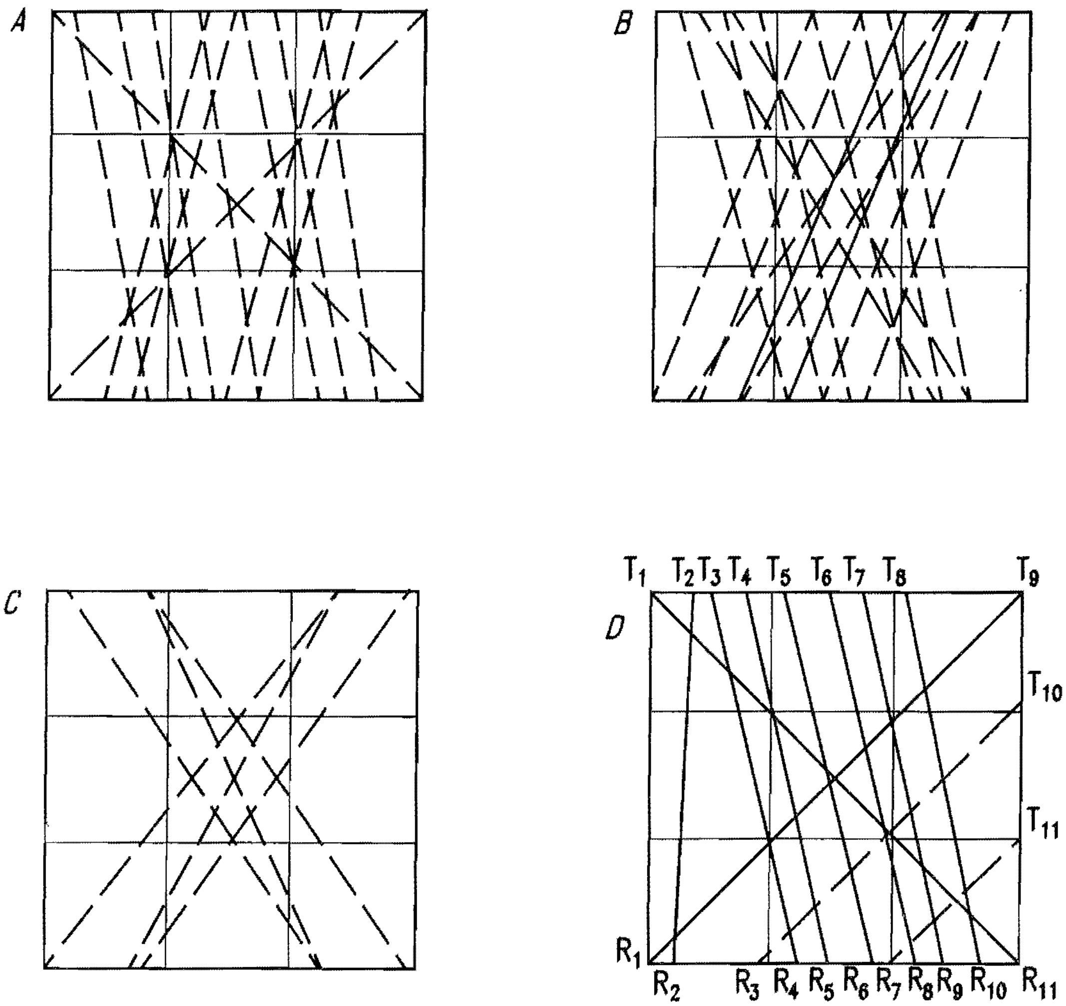

Figure 3.—Paths used in direct tomography calculation on 3 by 3 grid. ***A***, **Paths from opposite faces crossing only 3 cells;** ***B***, **paths from opposite faces crossing 4 cells;** ***C***, **paths from opposite faces crossing 5 cells;** ***D***, **paths used in calculation; original (solid), added (dashed).**

With a knowledge of the total number of distinct paths, the maximum number of independent paths that could be drawn from opposite, parallel sides of an n by n grid was determined for n = 2 to 6. For each n by n grid, the entire set of all possible distinct paths was analyzed for linear dependence using the row-echelon reduction method. Table 2 shows the maximum number of independent paths found in each case. An expression for the maximum number of linearly independent paths

$$N = n^2 - (n - 1), \tag{11}$$

was inferred from the data in table 2. For the general case of an n by n grid, the maximum number of independent paths that can be drawn from opposite, parallel sides is n^2 - (n-1), a deficit of (n-1) from the n^2 independent paths necessary for a unique, determinate solution.

Table 2.—Summary of results

Grid size	Max independent paths, N	n^2	Deficit
2 by 2 . .	3	4	1
3 by 3 . .	7	9	2
4 by 4 . .	13	16	3
5 by 5 . .	21	25	4
6 by 6 . .	31	36	5
n by n . .	n^2 - (n - 1)	n^2	n - 1

NOTE.—N is n^2 - (n - 1), the number of independent paths, n^2 is the number of cells in an n by n grid, and deficit is n - 1.

As an example of the application of the row-echelon method, consider the 3 by 3 grid system shown in figure 3. There are nine distinct paths (shown as solid lines in figure 3*D*) between transmitters and receivers. The D-matrix for these paths is shown in table 3. The order for each path (row) refers to the number of zeros to the left of the first entry in the matrix and is shown to the right of table 3. If these paths were linearly independent, equation 7 would yield the nine values of $\frac{1}{V_{k\ell}}$, and, therefore, $V_{k\ell}$, for the nine-grid cell shown in figure 3. If the paths are linearly dependent, $|D|^{-1}$ does not exist, i.e., $|D| = 0$. While manipulation of the nine rows (paths) of the D-matrix by linear operations to test for dependence could be attempted, the row-echelon method not only directly tests for dependence, but also gives an indication of the path modifications or additions necessary to obtain an independent set.

Results of the method are shown in table 4. The operations carried out in obtaining each row in the D-matrix of table 4 are shown on the left. Thus, for example, row 4 of table 4 was obtained by subtracting row 4 from row 3 of the original D-matrix. Each entry in table 4 is unity to indicate the order (preceding zeros). In practice, these would be replaced by constants which are not necessary to show the order.

If all the orders 0 to 8 were represented, the D-matrix would be in a triangular form and would be nonsingular; but as may be seen from table 4, orders 5 and 8 are missing. Further reduction would lead to a null row and a singular determinant. Thus, for example, if the two rows of order 4 are added, a second row of the D-matrix of order 6 is obtained. Repeating the process for order 6 and order 7 leads to order 9, which in this case, is a row of all zeroes and a value of $|D| = 0$.

The missing orders as seen from table 4 are orders 5 and 8. To obtain these orders, two new paths are required. These are shown in figure 3*D* as paths 10 and 11. Path 10 leads to a D-matrix row of order 5, since the path does not traverse any of the first five cells, i.e., cells C_{11}, C_{12}, C_{13}, C_{21}, and C_{22}. Similarly, path 11 leads to a row of order 8. When these paths are added to the paths represented by the D-matrix, the new D-matrix can be arranged in the triangular form shown in table 5. The determinant is nonsingular and can be used in equation 3 to obtain the value of $V_{k\ell}$.

As is suggested by table 2, the only way to obtain a unique solution, $|D| \neq 0$, is if the paths chosen give a complete set of orders from 0 to n-1. This condition requires, as shown in the 3 by 3 grid example, that paths be chosen in addition to those passing across opposite faces only (i.e., paths 10 and 11, shown as dashed lines in figure 3*D*). When only opposite faces are available, either an iterative solution is required, or additional information must be provided. If, for example, the structure is known or assumed to be known well enough to assume values for several of the cells, $V_{k\ell}$, transit times may be calculated for these $C_{k\ell}$ cells, and the corresponding paths may be used to supply the missing orders so that $|D| \neq 0$.

Table 3.—Original D-matrix for paths and cells in figure 3.

Path	Cell								
	C_{11}	C_{12}	C_{13}	C_{21}	C_{22}	C_{23}	C_{31}	C_{32}	C_{33}
1	X				X				X
2	X			X			X		
3	X			X				X	
4	X				X			X	
5		X			X			X	
6		X			X				X
7		X				X			X
8			X			X			X
9			X		X		X		

Table 4.—Rearranged D-matrix showing orders and missing orders

Operation	Path	Cell									
		C_{11}	C_{12}	C_{13}	C_{21}	C_{22}	C_{23}	C_{31}	C_{32}	C_{33}	Order
[1]1	1	1	0	0	0	1	0	0	0	1	0
[2]5	2	0	1	0	0	1	0	0	1	0	1
8	3	0	0	1	0	0	1	0	0	1	2
[3](3 minus 4)	4	0	0	0	1	-1	0	0	0	0	3
(6 minus 7)	5	0	0	0	0	1	-1	0	0	0	4
-(1 minus 2)	6	0	0	0	-1	1	0	-1	0	1	[4]3
(2 minus 3)	7	0	0	0	0	0	0	1	-1	0	6
(5 minus 6)	8	0	0	0	0	0	0	0	1	-1	7
(8 minus 9)	9	0	0	0	0	-1	1	-1	0	1	4
(6 minus 7) plus (8 minus 9) ..		0	0	0	0	0	0	-1	0	1	6

[1]Original path 1 from figure 3 multiplied by constant.
[2]Original path 5 from figure 3 multiplied by constant.
[3]Original path 3 minus path 4 multiplied by constant.
[4]Regardless of linear operations, orders 5 and 8 cannot be constructed from paths shown in figure 3.

Table 5.—D-matrix using nine independent paths

Operation	Path	Cell									
		C_{11}	C_{12}	C_{13}	C_{21}	C_{22}	C_{23}	C_{31}	C_{32}	C_{33}	Order
1	1	1	0	0	0	1	0	0	0	1	0
5	2	0	1	0	0	1	0	0	1	0	1
8	3	0	0	1	0	0	1	0	0	1	2
(3 minus 4) ..	4	0	0	0	1	-1	0	0	0	0	3
(6 minus 7) ..	5	0	0	0	0	1	-1	0	0	0	4
10	6	0	0	0	0	0	1	0	1	0	5
(2 minus 3) ..	7	0	0	0	0	0	0	1	-1	0	6
(5 minus 6) ..	8	0	0	0	0	0	0	0	1	0	7
11	9	0	0	0	0	0	0	0	0	1	8

APPLICATION OF DIRECT SOLUTION

A program was written to carry out the operations of equations 7 and 8 using D-matrices having the characteristics shown in tables 3 and 4. The following steps describe the method used:

1) A set of cell velocities was assumed.
2) A set of transmitter and receiver locations was chosen as those shown in figure 3*D*.
3) The D-matrix elements were calculated from equation 2 using straight line paths from the transmitters to each receiver.
4) From equations 4 and 5, the transit times were calculated.

Equation 7, for the case when $|D|$ is nonsingular, yields the inverse cell velocities. In table 6, the results are shown for the original D-matrix shown in table 3. In table 6, the path numbers and coordinates are given for a set of transmitters and receivers. In table 7, the assumed cell velocities are given, and in table 8, the calculated travel times are shown using the D-matrix elements shown in table 9. In this example, only the first nine paths (fig. 3) are used. Each of these cross from opposite faces. As indicated by equation 11, a deficit of two independent paths is expected. If the paths were independent, equation 7 could be used to solve for the velocities using the inverse D-matrix. The inverse of the D-matrix shown in table 9 is indeterminate since $|D| = 0$, and the product, $D^{-1}T = \frac{1}{v}$, cannot be calculated. This is brought about by a singular D-matrix ($|D| = 0$) as demonstrated by the row-echelon method in table 4, where two missing orders are indicated.

When the two orders are restored by adding the two paths 10 and 11 (fig. 3*D*), the problem is corrected. Here a least square calculation, represented by equation 8, is made. Table 10 shows the calculated travel times for the 11 paths, 9 of which are now linearly independent. The travel times are calculated from the new 11 by 9 D-matrix (table 11), which now includes two added paths, paths 10 and 11 (fig. 3*D*). These paths are seen to cut across the corners of the grid and supply the missing orders, orders 5 and 8, as seen in table 5. The calculations of equation 8 are shown in tables 12 through 17. Table 12 shows the transpose D-matrix, D^T; table 13 shows the product D^TD; and table 14 shows the inverse of $(D^TD)^{-1}$. The transpose distance matrix (D^T) multiplied by the time vector (T), is shown as follows:

$$
\begin{array}{c}
[D^T T \text{ - Vector}] \\
0.7942 \times 10^{-7} \\
0.5026 \times 10^{-7} \\
0.4744 \times 10^{-7} \\
0.3135 \times 10^{-7} \\
1.1309 \times 10^{-7} \\
0.5270 \times 10^{-7} \\
0.4411 \times 10^{-7} \\
0.7026 \times 10^{-7} \\
0.9077 \times 10^{-7}
\end{array}
$$

The final product of equation 8, for the slowness elements, is shown in table 15, and the calculated cell velocities are shown in table 16. These are seen to agree with the assumed cell velocities in table 7 showing the self-consistency of the method. However, to use the direct method, it was necessary to supply the missing orders, which was not possible using only paths across opposite faces.

Table 6.—Path numbers for various transmitters to receivers

Path	Transmitter[1]			Receiver[1]		
	No.	Coordinate		No.	Coordinate	
		X	Y		X	Y
1	T_1	0.000	0.000	R_{11}	3.000	3.000
2	T_2	.300	.000	R_2	.300	3.000
3	T_3	.400	.000	R_4	1.300	3.000
4	T_4	.700	.000	R_5	1.600	3.000
5	T_5	1.000	.000	R_6	1.900	3.000
6	T_6	1.400	.000	R_8	2.300	3.000
7	T_7	1.700	.000	R_9	2.600	3.000
8	T_8	2.000	.000	R_{10}	2.900	3.000
9	T_9	3.000	.000	R_1	.000	3.000
10 ...	T_{10}	3.000	1.000	R_3	1.000	3.000
11 ...	T_{11}	3.000	2.000	R_7	2.000	3.000

[1]See figure 3.

Table 7.—Assumed cell velocities, meters per second

Cell[1]	Velocity	Cell[1]	Velocity	Cell[1]	Velocity
C_{11}	2.0×10^8	C_{12}	2.0×10^8	C_{13}	2.0×10^8
C_{21}	2.0	C_{22}	1.8	C_{23}	2.0
C_{31}	2.0	C_{32}	2.0	C_{33}	2.0

[1]The cells are in the same configuration as in figure 3.

Table 8.—Calculated travel times from transmitter (T) to receiver (R), nine paths, nanoseconds

	R_1	R_2	R_3	R_4	R_5	R_6	R_7	R_8	R_9	R_{10}	R_{11}
T_1											22.0
T_2		15.0									
T_3				15.66							
T_4					16.24						
T_5						16.24					
T_6								16.24			
T_7									15.66		
T_8										15.66	
T_9	22.0										

Table 9.—D-matrix elements, nine paths, meters

Path	Cell								
	1	2	3	4	5	6	7	8	9
1	1.414	0.000	0.000	0.000	1.414	0.000	0.000	0.000	1.414
2	1.000	.000	.000	1.000	.000	.000	1.000	.000	.000
3	1.044	.000	.000	1.044	.000	.000	.000	1.044	.000
4	1.044	.000	.000	.000	1.044	.000	.000	1.044	.000
5	.000	1.044	.000	.000	1.044	.000	.000	1.044	.000
6	.000	1.044	.000	.000	1.044	.000	.000	.000	1.044
7	.000	1.044	.000	.000	.000	1.044	.000	.000	1.044
8	.000	.000	1.044	.000	.000	1.044	.000	.000	1.044
9	.000	.000	1.414	.000	1.414	.000	1.414	.000	.000

Table 10.—Calculated travel times from transmitter (T) to receiver (R), 11 paths, nanoseconds

	R_1	R_2	R_3	R_4	R_5	R_6	R_7	R_8	R_9	R_{10}	R_{11}
T_1											22.0
T_2		15.0									
T_3				15.66							
T_4					16.24						
T_5						16.24					
T_6								16.24			
T_7									15.66		
T_8										15.66	
T_9	22.0										
T_{10}			14.14								
T_{11}							7.07				

Table 11.—D-matrix elements, 11 paths, meters

Path	Cell								
	1	2	3	4	5	6	7	8	9
1	1.414	0.000	0.000	0.000	1.414	0.000	0.000	0.000	1.414
2	1.000	.000	.000	1.000	.000	.000	1.000	.000	.000
3	1.044	.000	.000	1.044	.000	.000	.000	1.044	.000
4	1.044	.000	.000	.000	1.044	.000	.000	1.044	.000
5	.000	1.044	.000	.000	1.044	.000	.000	1.044	.000
6	.000	1.044	.000	.000	1.044	.000	.000	.000	1.044
7	.000	1.044	.000	.000	.000	1.044	.000	.000	1.044
8	.000	.000	1.044	.000	.000	1.044	.000	.000	1.044
9	.000	.000	1.414	.000	1.414	.000	1.414	.000	.000
10	.000	.000	.000	.000	.000	1.414	.000	1.414	.000
11	.000	.000	.000	.000	.000	.000	.000	.000	1.414

Table 12.—Transpose D-matrix (D^T)

Row	Column										
	1	2	3	4	5	6	7	8	9	10	11
1	1.414	1.000	1.044	1.044	0.000	0.000	0.000	0.000	0.000	0.000	0.000
2	.000	.000	.000	.000	1.044	1.044	1.044	.000	.000	.000	.000
3	.000	.000	.000	.000	.000	.000	.000	1.044	1.414	.000	.000
4	.000	1.000	1.044	.000	.000	.000	.000	.000	.000	.000	.000
5	1.414	.000	.000	1.044	1.044	1.044	.000	.000	1.414	.000	.000
6	.000	.000	.000	.000	.000	.000	1.044	1.044	.000	1.414	.000
7	.000	1.000	.000	.000	.000	.000	.000	.000	1.414	.000	.000
8	.000	.000	1.044	1.044	1.044	.000	.000	.000	.000	1.414	.000
9	1.414	.000	.000	.000	.000	1.044	1.044	1.044	.000	.000	1.414

Table 13.—D transpose times D-matrix (D^TD)

Row	Column								
	1	2	3	4	5	6	7	8	9
1	5.100	0.000	0.000	2.090	3.090	0.000	1.000	2.100	2.000
2	.000	3.270	.000	.000	2.100	1.090	.000	1.090	2.100
3	.000	.000	3.090	.000	2.000	1.090	2.000	.000	1.090
4	2.090	.000	.000	2.090	.000	.000	1.000	1.090	.000
5	3.090	2.180	2.000	.000	7.270	.000	2.000	2.100	3.098
6	.000	1.090	1.090	.000	.000	4.188	.000	2.000	2.100
7	1.000	.000	2.000	1.000	2.000	.000	3.000	.000	.000
8	2.100	1.090	.000	1.090	2.100	2.000	.000	5.278	.000
9	2.000	2.100	1.090	.000	3.098	2.100	.000	.000	7.270

Table 14.—Inverse of D^TD matrix $(D^TD)^{-1}$

Row	Column								
	1	2	3	4	5	6	7	8	9
1	3.081	2.238	1.664	-4.244	-3.107	-1.492	1.350	0.992	0.000
2	2.238	2.163	1.439	-3.177	-2.535	-1.342	1.044	.841	.000
3	1.663	1.439	1.864	-2.110	-1.908	-1.192	.178	.692	.000
4	-4.244	-3.177	-2.110	7.272	5.015	2.685	-2.946	-2.185	-.500
5	-3.107	-2.535	-1.908	5.015	4.053	2.263	-2.066	-1.763	-.500
6	-1.492	-1.342	-1.192	2.685	2.263	1.767	-1.111	-1.267	-.500
7	1.350	1.044	.178	-2.946	-2.066	-1.111	2.124	1.111	.500
8	.992	.841	.692	-2.185	-1.763	-1.267	1.111	1.267	.500
9	.000	.000	.000	-.500	-.500	-.500	.500	.500	.500

Table 15.—Slowness elements $S = \frac{1}{V} = [D^TD]^{-1}[D^TT] \times 10^{-8}$

Cell	Slowness	Cell	Slowness	Cell	Slowness
C_{11}	0.5000	C_{12}	0.4999	C_{13}	0.4999
C_{21}	.5000	C_{22}	.5555	C_{23}	.5000
C_{31}	.4999	C_{32}	.5000	C_{33}	.4999

Table 16.—Calculated cell velocities $V = \frac{1}{S} \times 10^8$, meters per second

Cell	Velocity	Cell	Velocity	Cell	Velocity
C_{11}	1.9999	C_{12}	2.0000	C_{13}	2.0000
C_{21}	1.9999	C_{22}	1.8000	C_{23}	1.9999
C_{31}	2.000	C_{32}	1.9999	C_{33}	2.0000

CONCLUSIONS

The direct method of tomographic reconstruction offers potential benefits in field operations. These include the possibility of using simplified microprocessors or array processors to locate in-seam hazards. The development of such methods and equipment offers the possible operational advantage of on-site hazard detection and location in near real time. Nevertheless, as discussed here, such direct methods require that the locations of the probing transmitters and receivers be properly chosen so that the resulting D-matrix is nonsingular. It is not possible to obtain a nonsingular D-matrix with probing paths across opposite faces only because of imbedded linear dependencies. It is possible, however, to use the row-echelon method to determine for any configuration of transmitters and receivers, the missing orders and, therefore, the required grid cells to be cut by additional paths. The number of additional paths depends on the grid size and, therefore, on the ultimate resolution. As the cell size is reduced, the resolution is improved, but the grid dimension (n) is increased, and the number of additional paths, as represented by the deficit column in table 2, is also increased.

To overcome this difficulty, it is possible to assume cell values of the properties to be determined, in this case cell velocity. Then, synthesized path integrals can be used to supplement the raw data and to obtain a nonsingular D-matrix and solve the problem. Naturally, the a priori assumption of cell values must be based on experience and on the knowledge that anomalous values (hazards) are not located in those cells.

The calculational procedure shown here indicates that the direct method can be used successfully when sufficient a priori knowledge is available. Furthermore, in mining operations, it is not likely that the working area would be so saturated by anomalous hazards that the required assumptions cannot be made. Nevertheless, if that is not the case, iterative methods and other special methods for underdetermined solutions, where m paths are fewer than n grid cells, are available.

REFERENCES

1. Fox, L. An Introduction to Numerical Linear Algebra. Oxford Univ. Press (New York), 1965, pp. 18-20.

2. Forsythe, G., and C. Moler. Computer Solution of Linear Algebraic Systems. Prentice-Hall, 1967, p. 56.

3. Bellman, R. Introduction to Matrix Analysis. McGraw-Hill, 1970, p. 44.

4. Samelson, H. An Introduction to Linear Algebra. Wiley, 1974, pp. 27-35.

5. Menke, W. Geophysical Data Analysis: Discrete Inverse Theory Academic, 1984, pp. 40-48.

6. Eykhoff, P. System Identification, Parameter, and State Estimation. Wiley, 1974, pp. 39, 187.

APPENDIX.—ROWREDUCE[1]

This program[2] reduces the row vectors of a matrix to a linearly independent set and identifies the missing row-echelon orders. The current version has been modified for use on the U.S. Bureau of Mines Prime Computer System and is compiled using the standard Pascal compile command.

INPUT:

Input Matrix - Matrix to be reduced, standard ASCII form.
In the context of this report, this will be the D matrix of the direct solution equation.

Number of Rows of the Input matrix- Integer

Number of Columns of the Input Matrix- Integer

Cutoff Zero Value- Integer
Vectors whose residuals after reduction differ by less than this amount will be regarded as linearly dependent.

File Name for Reduced Orders- Text
Reduced row echelon orders will be written to this file as a column vector.

OUTPUT:

Reduced Row Echelon Orders

```
****************************************************************************************************
PROGRAM Matrix(INPUT, OUTPUT);
CONST printwidth  = 10;
      nr          = 200;
TYPE  Shortstring = String[24];
      Arraytype   = Array[0..nr] of Real;
      Aptrtype    = ^Arraytype;
VAR   numrow, numcol,jst,jen,err,k,j,x,y:Integer;
      hour1,hour2,min1,min2,sec1,sec2,frac: Integer;
      d : Array[1..nr] of Aptrtype;
      id : Array[1..nr] of Aptrtype;
      n  : Array[1..nr] of Integer;
      fname,resp : Shortstring;
      ch : Char;
      fv : Text;
      cutoff : Real;

PROCEDURE DiskRead;
VAR j,k:Integer;
BEGIN
```

[1]P.E. Wilson, Physicist, Denver Research Center, provided the original matrix reduction program.
[2]The U.S. Bureau of Mines expressly declares that there are no warranties express or implied which apply to the software contained herein. By acceptance and use of said software, which is conveyed to the user without consideration by the U.S. Bureau of Mines, the user hereof expressly waives any and all claims for damage and/or suits for or by reason of personal injury, or property damage, including, special, consequential, or other similar damages arising out of or in any way connected with the use of the software contained herein.

```
 Writeln;
 Write('Input Matrix File Name? ');
 Readln(fname);
 RESET(fv,fname);
 Reset(fv);
 For j:=1 to numrow do
  begin
  For k:=1 to (numcol) do Read(fv,d[j]^[k]);
  end;
 Close(fv);
 END; {DiskRead}

FUNCTION CheckCol(row,col:Integer):Boolean;
BEGIN
 If (row<=numrow) and (row>0) then CheckCol:=d[row]^[col]<>0
  else CheckCol:=false;
END; {CheckCol}

FUNCTION SearchCol(row:Integer):Integer;
VAR k:Integer;
BEGIN
 k:=1;
 While (not CheckCol(row,k)) and (k<=numcol) do k:=k+1;
 SearchCol:=k;
END; {SearchCol}

PROCEDURE Subrow(row1,row2,col:Integer);
VAR k:Integer;
  x:Real;
BEGIN
 x:=d[row1]^[col];
 d[row1]^[col]:=0.0;
 For k:=col+1 to numcol do
  begin
  d[row1]^[k]:=d[row1]^[k]-x*d[row2]^[k];
  If Abs(d[row1]^[k]) < cutoff then d[row1]^[k]:=0.0;
  end; {for}
 For k:=1 to numrow do
  begin
  id[row1]^[k]:=id[row1]^[k]-x*id[row2]^[k];
  If Abs(id[row1]^[k]) < cutoff then id[row1]^[k]:=0.0;
  end; {for}
 END; {Subrow}

PROCEDURE Sort;
VAR col,lastrow,firstrow,startrow,row,k:Integer;

 FUNCTION MaxCol(startrow,endrow,col:Integer):Integer;
 VAR i,j : Integer;
     max : Real;
 BEGIN
  max:=d[startrow]^[col];
  j:=startrow;
```

```
 For i:=startrow to endrow do
     If d[i]^[col] > max then j:=i;
 MaxCol := j;
END; {Function MaxCol}

PROCEDURE Swaprow(row1,row2:Integer);
VAR k:Integer;
     x:Real;
BEGIN
 For k:=1 to numcol do  {Swap main matrix rows}
   begin
   x:=d[row1]^[k];
   d[row1]^[k]:=d[row2]^[k];
   d[row2]^[k]:=x;
   end; {for swapping main matrix}
 For k:=1 to numrow do {Swap identity matrix rows}
   begin
   x:=id[row1]^[k];
   id[row1]^[k]:=id[row2]^[k];
   id[row2]^[k]:=x;
   end; {for swapping identity matrix}
 k:=n[row1];
 n[row1]:=n[row2];
 n[row2]:=k;
END; {Swaprow}

BEGIN {Sort}
 col:=1;
 lastrow:=1;
 Repeat
  firstrow:=lastrow;
  While CheckCol(lastrow,col) do lastrow:=lastrow+1;
  startrow:=lastrow+1;
  If lastrow<numrow then
    For row:=startrow to numrow do
      If CheckCol(row,col) then
      begin
      Swaprow(row,lastrow);
      lastrow:=lastrow+1;
      end; {if,for,if}
  If lastrow<= numrow then
    Swaprow(firstrow,MaxCol(firstrow,lastrow-1,col));
  col:=col+1;
 Until (lastrow>=numrow) or (col>numcol);
END; {Sort}

PROCEDURE Maketriang;
VAR row,col,k:Integer;

 PROCEDURE Normrow(row,col:Integer);
 LABEL 1;
 VAR k:Integer;
      x:Real;
```

```
 BEGIN
  If Abs(1.0 - d[row]^[col]) < cutoff then
  GOTO 1;
  x:=1.0/d[row]^[col];
  d[row]^[col]:=1.0;
  If col < numcol then
  For k:=col+1 to numcol do d[row]^[k]:=x*d[row]^[k];
  For k:=1 to numrow do id[row]^[k]:=id[row]^[k]*x;
  1:
 END; {Normrow}

BEGIN {Maketriang}
 row:=0;
 Repeat
  Sort;
  row:=row+1;
  col:=SearchCol(row);
  If col<=numcol then
    begin
    Normrow(row,col);
    k:=1;
    While (CheckCol(row+k,col)) and (row+k<=numrow) do
      begin
      Subrow(row+k,row,col);
      k:=k+1;
      end; {While}
    end; {if}
 Until (row>numrow) or (col>numcol);
END; {Maketraing}

PROCEDURE Finishid;
VAR k,row,col,lowerrow:Integer;
BEGIN
 row:=numrow+1;
 While row>2 do
  begin
  Repeat
    row:=row-1;
    col:=SearchCol(row);
  Until (col<=numcol) or (row<2);
  lowerrow:=row;
  row:=row-1;
    If row>0 then
    For k:=row downto 1 do
     If CheckCol(k,col) then
       Subrow(k,lowerrow,col);
  row:=lowerrow;
 end; {while}
END; {Finishid}

PROCEDURE Sortout;
VAR i,j,k:integer;
BEGIN
```

```
 Writeln;
 Writeln('File Name for Reduced Row Echelon Orders?');
 Readln(fname);
 Rewrite(fv,fname);
 Writeln(fv,fname:15,'      Reduced Row Echelon Orders');
 writeln(fv);
 For i:=1 to numrow do
  begin
  k:=SearchCol(i);
  Write(fv,'[',k:3,']');
  Writeln(fv);
  end;
 Close(fv);
 END; {Sortout}

PROCEDURE Setup;
 VAR J,K:INTEGER;
BEGIN
 For k:=1 to nr do new(d[k]);
 For k:=1 to nr do new(id[k]);
 For k:=1 to nr do for j:=1 to nr do d[k]^[j]:=0.0;
 For k:=1 to nr do for j:=1 to nr do id[k]^[j]:=0.0;
 Writeln;
 Writeln;
 Writeln('MATRIX REDUCTION PROGRAM');
 Writeln('Originally Written in TURBO PASCAL by P. E. Wilson');
 Writeln('(Edited in PRIME PASCAL by W.P. Stroud)');
 Writeln;
 Writeln('This program reduces a set of row vectors to a linearly');
 Writeln('independent set and identifies the missing row echelon orders.');
 numrow:= 10; numcol:= 10; fname:='';
 Writeln;
 Writeln('For the Input Matrix:');
 Repeat
  Write('     Enter The Number of Rows '); Readln(numrow);
  Writeln;
 Until ((numrow > 0) and (numrow < nr+1));
 Repeat
  Write('     Enter The Number of Columns '); Readln(numcol);
  Writeln;
 Until ((numcol > 0) and (numcol < nr+1));
 For k:=1 to numrow do
  begin
  n[k]:=k;
  id[k]^[k]:=1.0;
  end;
END; { Procedure Setup }

BEGIN { Main Program }
Setup;
DiskRead;
cutoff:=5.0e-04;
Writeln;Writeln('The Default Zero Cutoff Value Is:',cutoff:7);
```

```
Writeln;
Writeln('WARNING! Setting too small a cutoff value may result ');
Writeln('in failure to identify some linear dependencies.');
Writeln;
Writeln('Input a new cutoff value?');
Writeln('Answer y or n');
Readln(ch);
If ((ch) ='Y') or ((ch) ='y') then
 begin
 Write('Enter The New Zero Cutoff Value? '); Readln(cutoff);
 Writeln('The New Zero Cutoff Value Is: ',cutoff:7);
 end;
Writeln;
Maketriang;
Writeln('Triangalization Complete');
FinishId;
Writeln;
Writeln('Identity Matrix Complete');
Sortout;
Writeln;
Writeln('Row Reduction Complete');
END.
```