Computers & Geosciences Vol. 11, No. 2, pp. 111–127, 1985
Printed in the U.S.A.

0098-3004/85 $3.00 + .00

COKRIGING—A COMPUTER PROGRAM

JAMES R. CARR
Department of Geological Engineering, University of Missouri, Rolla, Missouri 65401, U.S.A.

DONALD E. MYERS
Department of Mathematics, University of Arizona,
Tucson, Arizona 85721, U.S.A.

and

CHARLES E. GLASS
Department of Mining and Geological Engineering, University of Arizona,
Tucson, Arizona 85721, U.S.A.

(*Received* 13 *June* 1983; *revised* 4 *January* 1984; *received for publication* 24 *October* 1984)

Abstract—Cokriging is a process wherein several variables can be jointly estimated on the basis of intervariable and spatial structure information. Presented herein is the program COKRIG, for punctual cokriging, a program in a simple form to demonstrate the utility of cokriging. Equation solution follows a modification of a method developed for the solution of large-scale linear systems. Several example problems show that, at least for earthquake data, the inclusion of intervariable information results in a more accurate BLUE (best linear unbiased estimator).

Key Words: ART, COKRIG, Cokriging, Cross-variograms, Earthquake data, Projection method.

INTRODUCTION

Cokriging is the extension of kriging to several variables whereby several variables are estimated jointly utilizing a BLUE (best linear unbiased estimator). The notation and terminology used in this paper is presented in Myers (1982, 1983a–c). The general concept of cokriging is described in Matheron (1971). All of the examples given in Matheron, Journel and Huijbregts (1978), and François-Bongarçon (1981) are concerned with the undersampled problem. Borgman and Frahme (1976) used principal component analysis to approximate the variogram of a linear combination. Matheron (1979) provides an alternative to cokriging for linear combinations. The general formulation of cokriging in matrix form is given in Myers (1981, 1982) and for cokriging of linear combinations in Myers (1983a–c).

The objective of this paper is to present a computer program which will perform punctual cokriging both in the general form and for the "undersampled" form. The solution of the system of matrix equations is obtained by an iterative technique as described in Myers (1983b). This technique, a generalization of the method given by Tanabe (1971) and termed ART by Herman, Lent, and Stuart (1973), provides for the solution with only a few iterations.

NOTATION AND THE COKRIGING EQUATIONS

Let $Z_1(x), \ldots, Z_m(x)$ be second-order stationary random functions, where x is a point in 1, 2, or 3 space. Denote $\bar{Z}(x) = [Z_1(x), \ldots, Z_m(x)]$ and assume $E[\bar{Z}(x)] = [0, \ldots, 0]$, where E denotes expected value. The covariance matrix may be written as

$$\bar{C}(h) = E[\bar{Z}(x + h)^T \bar{Z}(x)]. \tag{1}$$

If $x_1, \ldots, x_n$ are sample locations, then the objective is to estimate $\bar{Z}(x)$ by

$$\bar{Z}^*(x) = \sum_{j=1}^{n} \bar{Z}(x_j)\Gamma_j = [Z_1^*(x), \ldots, Z_m^*(x)], \tag{2}$$

where $\Gamma_1, \ldots, \Gamma_m$ are $m \times m$ matrices.

$$\sum_{j=1}^{n} \Gamma_j = I \tag{3}$$

is a sufficient condition for

$$E[\bar{Z}(x) - \bar{Z}^*(x)] = [0, 0, \ldots, 0]. \tag{4}$$

The Γ_j's are selected to minimize

$$\sum_{j=1}^{m} \text{var}\,(Z_j(x) - Z_j^*(x)). \tag{5}$$

As shown in Myers (1982) the cokriging system of equations may be written as

$$UY = D, \tag{6}$$

where

$$U = \begin{bmatrix} \bar{C}(x_1 - x_1) & \cdots & \bar{C}(x_1 - x_n) & I \\ \vdots & & \vdots & \vdots \\ \bar{C}(x_n - x_1) & \cdots & \bar{C}(x_n - x_n) & I \\ I & \cdots & I & 0 \end{bmatrix}, \quad (7)$$

$$Y = \begin{bmatrix} \Gamma_1 \\ \vdots \\ \Gamma_n \\ \bar{\mu} \end{bmatrix}, \quad D = \begin{bmatrix} \bar{C}(x_1 - x) \\ \vdots \\ \bar{C}(x_n - x) \\ I \end{bmatrix}. \quad (8)$$

The minimal value of (5) is given by

$$\sigma^2_{CK} = \mathrm{Tr}\ \bar{C}(0) - \mathrm{Tr} \sum \bar{C}(x_j - x)\Gamma_j - \mathrm{Tr}\ \bar{\mu} \quad (9)$$

where Tr denotes the trace.

This result is completely analogous to the one variable version. The system (6)–(8) also can be written with variograms (and cross-variograms). To solve the system (6)–(8) requires solving a system of equations in which all entries are $m \times m$ matrices.

ART: The iterative method

As proposed by Tanabe (1971), it might better be termed the projection method for solving a system of linear equations. Each equation in the system can be thought of as the projection of the solution vector onto the hyperplane corresponding to that equation. Herman, Lent, and Stuart (1973) utilized this iterative technique to "algebraically reconstruct" a digital image from rays, for example, projections; hence the acronym ART.

Let X, Y be $p \times 1$ matrices whose elements are $m \times m$ matrices; then let

$$\langle X, Y \rangle = Y^T X \quad (10)$$

and

$$(X, Y) = \mathrm{Tr}\ Y^T X. \quad (11)$$

$\langle X, Y \rangle$ is a matrix-valued inner product on the linear space of $m \times m$ matrices and (X, Y) is a scalar valued inner product. Let

$$\mathcal{A} = \begin{bmatrix} \mathcal{A}_1 \\ \vdots \\ \mathcal{A}_n \end{bmatrix} = \begin{bmatrix} A_{11} & \cdots & A_{1n} \\ \vdots & & \vdots \\ A_{n1} & \cdots & A_{nn} \end{bmatrix} \quad (12)$$

be an $n \times n$ matrix whose entries are $m \times m$ matrices. If

$$\mathcal{X} = \begin{bmatrix} X_1 \\ \vdots \\ X_n \end{bmatrix}, \quad \mathcal{B} = \begin{bmatrix} B_1 \\ \vdots \\ B_n \end{bmatrix}, \quad (13)$$

then

$$\mathcal{A}\mathcal{X} = \mathcal{B} \quad (14)$$

is equivalent to

$$\mathcal{A}_i\mathcal{X} = \mathcal{B}_i; \quad i = 1, \ldots, n. \quad (15)$$

That is, the solution vector $\mathcal{X}$ is the usual projection onto all the hyperplanes determined by the rows of $\mathcal{A}$ and $\mathcal{B}$. The ART routine in the program in this paper requires less core than a matrix inversion routine by only operating on one row at a time.

Let

$$\mathcal{A}^* = \begin{bmatrix} A^*_{11} & \cdots & A^*_{p1} \\ \vdots & & \vdots \\ A^*_{1n} & \cdots & A^*_{np} \end{bmatrix}, \quad (16)$$

where A^*_{ij} is the complex conjugate transpose. Let $\bar{A}_i$ be the ith column in $\mathcal{A}^*$. Assume $(A_i, A_i) > 0$ for all i. Define f_i: $C^n_m \to C^n_m$ by

$$f_i(\mathcal{X}) = \mathcal{X} - \frac{1}{\alpha_i}[\bar{A}_i\langle \mathcal{X}, A_i \rangle - \bar{A}_i B_i], \quad (17)$$

$$\alpha_i = (\bar{A}_i, \bar{A}_i). \quad (18)$$

For any two functions, $f \circ g$ denotes composition. That is, $(f \circ g)(x) = f(g(x))$. Composition is associative but not commutative. Finally let $F(\mathcal{B}, \mathcal{X}) = f_1 \circ \cdots \circ f_p(\mathcal{X})$ and for any initial element $\mathcal{X}_0$, set

$$\mathcal{X}_{i+1} = F(\mathcal{B}, \mathcal{X}_i). \quad (19)$$

$\mathcal{X}_0, \ldots, \mathcal{X}_i, \cdots$ converges to a solution of $\mathcal{A}\mathcal{X} = \mathcal{B}$.

THE UNDERSAMPLED OPTION

One of the more frequently used applications of cokriging is for undersampled problems. That is, one or more variables are not sampled at all locations. The objective is to improve the estimation of the undersampled variables using the correlation with the other sampled variables. This version of cokriging is a special situation, and the corresponding system of equations is obtained by a simple modification of the full-sampled version. This is described in more detail by Myers (1983c). The undersampled option in the program proceeds as follows:

(1) The jth variable is unsampled at location i, and the matrices in the ith column of U [equation (7)] are modified by changing all entries in the jth column of those matrices to zeros.
(2) Likewise, all the matrices in the ith row of U and the matrix in the ith row of D have the entries in the jth row changed to zeros.

This is equivalent to requiring that the jth row of Γ_i be all zeros; for example, the jth variable has zero weight in estimating all variables for location i.

This modification of U, D will result in arbitrary values in the jth row of Γ_i. These arbitrary values are ignored.

Description of the program COKRIG

In an actual sense, a program for cokriging has all the basic elements of a program for ordinary kriging, except that provisions are made to accommodate more than one variable. Equation solution is more difficult, however, because, whereas ordinary kriging considers a single variable, this scalar dimension becomes a multiple dimension in cokriging. Nonetheless, the estimation process is the same for both methods, and cokriging is equivalent to ordinary kriging when only one variable is considered. The major purpose of cokriging, however, is to utilize intervariable information, along with spatial structure, to estimate more than one variable simultaneously.

Our purpose, hereafter, is to present a program for the simplest situation, relying on samples having punctual support; provision for samples having block support is being developed. In this form, the program serves to demonstrate the utility of cokriging.

As presented in Appendix 1, the COKRIG program is written in standard FORTRAN for an IBM 4331 computational system. This program utilizes a temporary, sequentially accessed disk file for equation solution. Because the revised Tanabe method (Myers, 1983b) requires only one row of the intersample covariance matrix in core at any one instant, disk storage allows large linear systems to be solved without using a large amount of in-core storage.

As an alternative, this program also allows in-core equation solution using a modified Gauss elimination scheme. The sequentially accessed disk file used for the Tanabe method is required here for intermediate storage. Use of the Gauss scheme allows more rapid execution time, but the maximum size of the linear systems that can be analyzed is more limited compared with the Tanabe method.

Program COKRIG comprises a main program which reads and prints input data, establishes the cokriging boundary, computes the estimates and estimation variance, and prints the result. In conjunction with the main program, there are nine subroutines and three function subprograms.

Subroutine AFORM forms the intersample covariance–cross-covariance matrix, modifies this matrix to consider undersampled locations if desired, and then writes the result to disk for use in subsequent equation solution. In addition, the measurement vector, the point-sample covariance–cross-covariance vector, also is computed and stored in core. Subroutine ART performs the equation solution relative to Myers (1983b). Subroutine EQSOLV performs the equation solution relative to Gauss elimination. Subroutines MATMUL, ANI, SOLN, RZRO, ARIQ, and SCALAR supplement ART and EQSOLV. Subprogram FUNCTION TRACE computes the trace of a square matrix, FUNCTION COVAR computes the covariance value using either a spherical, Gaussian, or linear model (Journel and Huijbregts, 1978), and FUNCTION CROSS computes the cross-covariance value using the same three covariance models. The actual cross-covariance value used in the program is

$$\text{ACTUAL} = 0.5[\text{CROSS}_{ij} - \text{COVAR}_i - \text{COVAR}_j],$$

with CROSS_{ij} being the value yielded by one of the cross-covariance models.

To modify COKRIG to execute on other computational systems, we need to consider two aspects. It is essential to ensure the ability to use the temporary disk file for equation solution. As presented, COKRIG opens this file implicitly. Other systems may require that the file be opened explicitly. Throughout COKRIG, this temporary logical device is termed IUNIT; the actual integer definition of IUNIT is accomplished using a DATA statement in the main program. To change the unit designation of IUNIT requires changing only this COK 00950 statement. The other aspect to be considered is the handling of the ALAB array, defined using a 15A4 format (COK 01140).

Program COKRIG is dimensioned to accommodate a maximum of 500 samples, of up to five variables each, contained in the array DAT. For larger problems, the dimension of each array in the labeled COMMON/ DAT2/, as well as the dimension of JUNSAM in the labeled COMMON/ AMAT/, must be increased. To accommodate more than five variables, the dimension of each array in the labeled COMMON/ VAR/ and COMMON/ CVAR/ must be increased. The dimension of each array in CVAR must be at least

$$0.5[\text{MVAR}(\text{MVAR} - 1)],$$

where MVAR is the total number of possible variables. Furthermore, the dimension of the arrays DIST, EST, AW, ATEMP, XTEMP, TEMP1, TEMP2, and TEMP3, in the main program XMEAS, in the labeled COMMON/ FORM/, as well as ATEMP and IPOS in AFORM, and the dimension of each array in the ART series and EQSOLV, in which a "5" occurs in the DIMENSION declaration statement, must be increased.

At present, the maximum number of surrounding samples the program can use for estimation is fixed at eight. This is accomplished by ISTOP, defined at statement COK 02250. More samples can be used for estimation by modifying this statement. The maximum value of ISTOP, relative to current program dimensioning, is shown here:

No. of variables	ISTOP
1	90
2	45
3	30
4	22
5	16

Program COKRIG uses three special disk files for an optional check on the sum of the weights (LT), for a check on the estimation variance of each variable separately (LG), and as an optional file to access when contouring (LB), which contains the row and column number of each estimation location, the estimated values at each location, and the estimation variance. For large problems, it may be beneficial to suppress the creation of the LT file by removing statements COK 03450–COK 03590; this will minimize program output. The LT file is useful initially, however, to ensure correct execution of the program. Only the main program of COKRIG accesses these files.

Input to COKRIG is documented in statements COK 00240–COK 00780. For added detail, card 1 defines the options; if ICREAT is 1, files LT, LB, and LG are created. For equation solution, if ISOLV is 0, the Tanabe method is used; if ISOLV is 1, Gauss elimination is used. With card 2, the cokriging boundary is established from row 1, column 1, to row NROW, column NCOL. The y coordinate of row 1 is YMAX-YDIM/2, where YDIM is the y increment between rows. The x coordinate of the first column is XMAX-XDIM* (NCOL + 0.5), where XDIM is the x increment between columns; in this manner, XMAX and XDIM can manipulate the minimum x coordinate. With card 3, alphanumeric labels for each variable are defined; the spacing of these labels on input also will be the spacing on output. For the card 4 series of cards, variogram parameters are entered first, one card per variable, then cross-variogram parameters are entered, one card for each possible pair of variables. (Variograms are modeled on input; the program automatically computes covariance values for use in equation solution.) As an example, given four variables, 1, 2, 3, and 4, four variogram cards would be followed by six cross-variogram cards for the pairs: 1–2, 1–3, 1–4, 2–3, 2–4, 3–4, *in this order.* This series of cards is followed by the card 5 series, a multiple F10.3 format. The columns used for data entry depend on the number of variables being estimated. Values of zero for x and y coordinates signify the end of data entry. Modify COK 01530 if this is not desired.

In modeling cross-variograms, a new variable is created as $\mathrm{NEW}_{ij}(x) = \mathrm{Old}_i(x) + \mathrm{Old}_j(x)$, where i and j are the variables constituting the pair. A variogram (Journel and Huijbregts, 1978) is computed using $\mathrm{New}_{ij}(x)$, to result in γ_{ij}. It is essential, however, when modeling the cross-variogram, that

$$|\gamma_{ij}| \leq \mathrm{SQRT}[(\gamma_{ii})(\gamma_{jj})].$$

These criteria help in defining the input to the card 4 series.

Review of input options

The following gives some additional detail or references in an attempt to make data input more understandable.

Section 1: Option initialization—there are four options to initialize:

(a) IKRIG, which is either zero, if all variables are sampled at *all* locations, or 1, if some variables are not sampled at locations; if IKRIG is 1, a zero data value indicates undersampling (to change this, modify statement COK 01660).

(b) MVAR, which is simply the number of variables to be estimated.

(c) ICREAT. If ICREAT is set equal to 1, files are created to check on the sum of the weights and the individual values of kriging variance, and a file is created containing the row and column numbers, and estimated values; this latter file is designed to be accessed when contouring.

(d) ISOLV, which is zero for Tanabe equation solution or 1 for Gauss elimination. Gauss elimination is more efficient.

Sections 2 and 3: kriging boundary definition and variable names, were discussed adequately and should require no further elaboration.

Section 4: Variogram and cross-variogram data entry. For those familiar with regionalized variables theory, a variogram is a known tool. For those unfamiliar with this theory, Journel and Huijbregts (1978) explain the variogram clearly. A cross-variogram is a variogram computed on two variables added together; hence it is the cross relationship between two variables.

Section 5: Sample data entry. Here, variable values are entered first, then the y coordinate, followed by the x coordinate. For example, if two variables are to be estimated, variable 1 is entered in columns 1–10, variable 2 in columns 11–20, the y coordinate in columns 21–30, and the x coordinate in columns 31–40.

Examples of program execution

With cokriging, intervariable relationships are used, along with spatial relationships to estimate values of several variables at locations in one, two, or three space. It might be difficult to appreciate the utility this method offers without comparing it to ordinary kriging. This latter method relies solely on spatial relationships to estimate the value of a single variable at a location in space.

Appendices 2 and 3 make such a comparison between cokriging and ordinary kriging. In Appendix 2, a single variable, modified Mercalli intensity data for the 9 February 1971 San Fernando, California

(U.S.A.) earthquake (United States Earthquakes, 1971), was estimated. A cross-validation-type procedure was used, whereby an estimate is made at a known data location *without* including data for that location in the estimation process. The purpose is to make a comparison between known data values and estimated values. Appendix 3 uses the same estimation procedure, yet a second variable is introduced—peak velocity as recorded during the 1971 San Fernando earthquake (Trifunac, Brady, and Hudson, 1973).

It is observed in Appendix 3 that inclusion of the intervariable relationship between velocity and intensity improved the estimation of intensity data relative to the single-variable kriging presented in Appendix 2. We conclude from this that, cokriging can be more accurate than ordinary kriging, provided the several variables to be estimated are correlated. This was an implicit assumption with respect to the two aspects of earthquake data estimated herein; both describe the level of ground motion. This implicit assumption was explicitly defined by the cross-variogram. For the examples shown in Appendices 2 and 3, cokriging proved to be superior to ordinary kriging.

Acknowledgment—The authors wish to acknowledge the help given by Dr. Thomas A. Doerr, University of Marquette, Marquette, Wisconsin, U.S.A., in describing the operation and error characteristics of Tanabe's projection method, and for programming suggestions. Development of COKRIG was accomplished using the University of Arizona's Digital Image Analysis Laboratory (DIAL), the University of Arizona Computer Center, and the University of Missouri—Rolla Computer Center.

REFERENCES

Borgman, L. E., and Frahme, R. B., 1976, A case study: multivariate properties of bentonite in northeastern Wyoming *in* Advanced geostatistics in the mining industry: D. Reidel, Dordrecht, Netherlands, 381 p.

François-Bongarçon, D., 1981, Les coregionalisations, le cokrigeage: Centre de Geostatistique, Fontainebleau, C-86, 26 p.

Herman, G. T., Lent, A., and Rowland, S. W., 1973, ART: mathematics and applications: Jour. Theor. Biol. v. 42., no. 1, p. 1–32.

Journel, A., and Huijbregts, Ch., 1978, Mining geostatistics: Academic Press, London, 600 p.

Matheron, G., 1971, The theory of regionalized variables and its applications: Centre de Geostatistique, Fontainebleau, France.

Matheron, G., 1979, Recherche de simplification dans un probleme de cokrigeage: Centre de Geostatistique, Fontainebleau, France, Rep. No. N-698, 19 p.

Myers, D. E., 1981, Joint estimation of random functions: the matrix form: Res. Rep., Dept. of Mathematics, Univ. Arizona, 32 p.

Myers, D. E., 1982, Matrix formulation of co-kriging: Jour. Math Geology, v. 14, no. 3, p. 249–257.

Myers, D. E., 1983a, Estimation of linear combinations and co-kriging: Jour. Math. Geology, v. 15, no. 5, p. 633–637.

Myers, D. E., 1983b, On solving large scale linear systems: submitted for publication in Numer. Math.

Myers, D. E., 1983c, Co-kriging—new developments: Proceedings of GEOSTAT—Tahoe 83, NATO ASI, 6–17 September 1983, D. Reidel, Dordrecht, Netherlands.

Tanabe, K., 1971, Projection method for solving a singular system of equations and its applications: Numer. Math., v. 17, no. 3, p. 203–124.

Trifunac, M. D., Brady, A. G., and Hudson, D. E., 1973, Strong motion earthquake accelergrams, digitized and plotted data, v. II, corrected accelergrams and integrated ground velocity and displacement curves, parts C–S: California Institute of Technology, prepared for the National Science Foundation, distributed by the National Technical Information Service, U.S. Department of Commerce, 2000 p.

United States Earthquakes, published annually by the U.S. Department of Commerce, Coast and Geodetic survey from 1928 through 1968, the NOAA National Ocean Survey in 1969, the NOAA Environmental Data Service from 1970 through 1972, and jointly by the NOAA Environment Data and Information Service and U.S. Geological Survey thereafter.

APPENDIX I. PROGRAM LISTING

```
C       PROGRAM COKRIG                                                  COK00010
C       ***************************************************************COK00020
C                                                                       COK00030
C         PROGRAM COKRIG WAS WRITTEN BY JAMES R. CARR                   COK00040
C         ADDRESS:                       DEPT. OF GEOLOGICAL ENGR.      COK00050
C                                        UNIVERSITY OF MISSOURI         COK00060
C                                        ROLLA, MISSOURI U.S.A. 65401   COK00070
C         TELEPHONE:                     (314) 341-4557/(314) 341-4867  COK00080
C         IBM 4331 FORTRAN IV VERSION:   SEPTEMBER, 1983                COK00090
C                                                                       COK00100
C         THIS PROGRAM PERFORMS THE CO-KRIGING PROCEDURE.               COK00110
C         EQUATION SOLUTION FOLLOWS THE TECHNIQUE DEVELOPED             COK00120
C         BY K. TANABE(1971), NUMER. MATH., VOL. 17, PP 203-            COK00130
C         214.  THE GLOBAL LOGIC OF THIS PROGRAM                        COK00140
C         FOLLOWS MYERS, D.E., 1982, MATHEMATICAL GEOLOGY,              COK00150
C         VOL. 14, NO. 3, PP. 249 - 257.                                COK00160
C                                                                       COK00170
C         THREE COVARIANCE AND CROSS-COVARIANCE MODELS ARE              COK00180
C         AVAILABLE:  SPHERICAL, GAUSSIAN, AND LINEAR. AS               COK00190
C         PRESENTED, PROGRAM COKRIG IS DIMENSIONED FOR 500              COK00200
C         DATA SAMPLES AND LOCATIONS, EACH LOCATION LIMITED TO          COK00210
C         A MAXIMUM OF 5 VARIABLES.                                     COK00220
C                                                                       COK00230
C                   GUIDE TO DATA INPUT                                 COK00240
C                                                                       COK00250
C         CARD 1.  OPTION INITIALIZATION                                COK00260
C                  COL 1   IKRIG   FULLY SAMPLED (0) OR UNDER-          COK00270
C                                  SAMPLED (1)                          COK00280
C                                  FOR THE UNDERSAMPLED CASE,           COK00290
C                                  ZERO DATA VALUES ARE CON-            COK00300
C                                  SIDERED TO INDICATE UNDER-           COK00310
C                                  SAMPLING. SEE STATEMENT 1640         COK00320
C                                  TO CHANGE THIS REQUIREMENT.          COK00330
C                  COL2    MVAR    NUMBER OF VARIABLES TO BE EST. (1-5) COK00340
```

```
C                    COL3   ICREAT OPTION FOR FILE CREATION                COK00350
C                    COL4   ISOLV  OPTION FOR EQUATION SOLUTION            COK00360
C                           0 = ART ITERATIVE SCHEME                       COK00370
C                           1 = GAUSS DECOMPOSITION                        COK00380
C          CARD 2.   ESTABLISH KRIGING BOUNDARY                            COK00390
C                    COL 1-5  NROW  NUMBER OF ROWS TO BE KRIGED            COK00400
C                    COL 6-10 NCOL  NUMBER OF COLS TO BE KRIGED            COK00410
C                    COL11-20 YMAX  MAXIMUM Y-COORDINATE (NORTH)           COK00420
C                    COL21-30 XMAX  MAXIMUM X-COORDINATE (EAST)            COK00430
C                    COL31-40 XDIM  INCREMENT IN X BETWEEN COLUMNS         COK00440
C                    COL41-50 YDIM INCREMENT IN Y BETWEEN ROWS             COK00450
C          CARD 3.   VARIABLE NAMES (15A4)                                 COK00460
C                    COL  1-10 NAME OF VARIABLE 1                          COK00470
C                    COL 11-20 NAME OF VARIABLE 2                          COK00480
C                    COL 21-30 NAME OF VARIABLE 3                          COK00490
C                    COL 31-40 NAME OF VARIABLE 4                          COK00500
C                    COL 41-50 NAME OF VARIABLE 5                          COK00510
C          CARD 4.   VARIOGRAM AND CROSS-VARIOGRAM DATA ENTRY.             COK00520
C                    VARIOGRAM PARAMETERS FIRST-ONE CARD PER VAR.          COK00530
C                    COL    1-5  VARIABLE NUMBER (1-MVAR) - J              COK00540
C                    COL   6-10  MODEL(J) - VARIO. MODEL (1:SPHER;          COK00550
C                                           2:GAUSS; 3:LINEAR)              COK00560
C                    COL 11-20   CO(J) - NUGGET VALUE                       COK00570
C                    COL 21-30   C(J)  - SILL OF VARIOGRAM                  COK00580
C                    COL 31-40   RANGE(J) - RANGE OF VARIOGRAM              COK00590
C                    COL 41-50   ANIS(J) - ANGLE OF ANISOTROPY              COK00600
C                    COL 51-60   RATIO(J)- RATIO OF MAJ AXIS/MIN AXIS       COK00610
C                    COL 61-70   RINFLU(J) - RANGE OF INFLUENCE FOR EST.    COK00620
C                    CROSS-VARIOGRAM ENTRY:  1 CARD PER VARIABLE PAIR.      COK00630
C                    COL    1-5  J - VARIABLE J                             COK00640
C                    COL   6-10  K - VARIABLE K CROSS CORRELATED WITH J     COK00650
C                    COL 11-15   CMODEL - CROSS-VARIOGRAM MODEL(1:S;2:G;3:LCOK00660
C                    COL 16-25   CCO    - NUGGET VALUE                      COK00670
C                    COL 26-35   CC     - SILL VALUE                        COK00680
C                    COL 36-45   CRANGE - RANGE OF CROSS-VARIOGRAM          COK00690
C                    COL 46-55   CANIS  - ANGLE OF ANISOTROPY (DEGREES)     COK00700
C                    COL 56-65   CRATIO - RATIO OF MAJ AXIS/MIN AXIS        COK00710
C                    COL 66-70   CRNFLU - RANGE OF INFLUENCE FOR EST.       COK00720
C          CARD 5.   DATA ENTRY.                                           COK00730
C                    COL 1-?     DATA VALUES, FORMAT = MVAR:F10.3           COK00740
C                    COL ?-?     DEPENDING ON NUMBER OF VARIABLES, COLUMN   COK00750
C                                LOCATION WILL VARY.  AFTER DATA VALUES:    COK00760
C                    COL ?-?     Y-COORDINATE OF DATA LOCATION              COK00770
C                    COL ?-?     X-COORDINATE OF DATA LOCATION              COK00780
C                                                                           COK00790
C          ************************************************************     COK00800
C                                                                           COK00810
      COMMON /DAT2/ X(500),Y(500),DAT(500,5)                                COK00820
      COMMON /PARM/ MROW,MCOL,MVAR,MTOT                                     COK00830
      COMMON /FILES/ IUNIT                                                  COK00840
      COMMON /VAR/ CO(5),C(5),RANGE(5),MODEL(5),ANIS(5),RATIO(5),           COK00850
     1             RINFLU(5)                                                COK00860
      COMMON /CVAR/ CCO(10),CC(10),CRANGE(10),CMODEL(10),CANIS(10),         COK00870
     1              CRNFLU(10),CRATIO(10)                                   COK00880
      COMMON /FORM/ XMEAS(100,5)                                            COK00890
      COMMON /AMAT/ IKRIG,JUNSAM(500),KCOUNT                                COK00900
      DIMENSION DIST(5),TEMP(15),NHOLE(15),EST(5),AW(5,5),ATEMP(5,100)COK00910
      DIMENSION XTEMP(100,5),TEMP1(5,5),TEMP2(5,5),TEMP3(5,5)               COK00920
      DIMENSION ALAB(15)                                                    COK00930
      INTEGER CMODEL                                                        COK00940
      IUNIT = 1                                                             COK00950
      LT    = 2                                                             COK00960
      LG    = 3                                                             COK00970
      LB    = 4                                                             COK00980
C...                                                                        COK00990
C.......SET INITIAL DATA                                                    COK01000
C...                                                                        COK01010
      READ(5,5) IKRIG,MVAR,ICREAT,ISOLV                                     COK01020
5     FORMAT(4I1)                                                           COK01030
      READ(5,6) NROW,NCOL,YMAX,XMAX,XDIM,YDIM                               COK01040
6     FORMAT(2I5,4F10.3)                                                    COK01050
      READ(5,8) ALAB                                                        COK01060
C...                                                                        COK01070
C.......VARIOGRAM AND CROSS-VARIOGRAM DATA ENTRY                            COK01080
C...                                                                        COK01090
      DO 10 I = 1,MVAR                                                      COK01100
      READ(5,7) J,MODEL(J),CO(J),C(J),RANGE(J),ANIS(J),RATIO(J),            COK01110
     1          RINFLU(J)                                                   COK01120
7     FORMAT(2I5,6F10.3)                                                    COK01130
8     FORMAT(15A4)                                                          COK01140
      IF(MODEL(J).EQ.3) RINFLU(J) = 0.45 * RANGE(J)                         COK01150
10    CONTINUE                                                              COK01160
      IF(MVAR.EQ.1) GO TO 25                                                COK01170
C.......CROSS-VARIOGRAM DATA ENTRY                                          COK01180
      M2          =  (MVAR * (MVAR - 1)) / 2                                COK01190
      DO 20 I = 1,M2                                                        COK01200
      READ(5,12) J,K,IMOD,ACCO,ACC,DRANGE,AAN,AR,AFLU                       COK01210
12    FORMAT(3I5,5F10.3,1F5.0)                                              COK01220
      IF(J.EQ.1) GO TO 16                                                   COK01230
      KM          =  J - 1                                                  COK01240
      KP          =  0                                                      COK01250
      DO 15 KN = 1,KM                                                       COK01260
      KP          =  KP + MVAR - KN                                         COK01270
15    CONTINUE                                                              COK01280
      KP          =  KP + K - J                                             COK01290
      GO TO 17                                                              COK01300
16    CONTINUE                                                              COK01310
      KP          =  K - J                                                  COK01320
17    CONTINUE                                                              COK01330
      CCO(KP)     =  ACCO                                                   COK01340
      CC(KP)      =  ACC                                                    COK01350
      CRANGE(KP)  =  DRANGE                                                 COK01360
      CMODEL(KP)  =  IMOD                                                   COK01370
      CANIS(KP)   =  AAN                                                    COK01380
      CRATIO(KP)  =  AR                                                     COK01390
```

```
      CRNFLU(KP) =  AFLU                                                COK01400
20    CONTINUE                                                          COK01410
25    CONTINUE                                                          COK01420
C...                                                                    COK01430
C.......DATA ENTRY                                                      COK01440
C...                                                                    COK01450
      ICOUNT     =  0                                                   COK01460
40    CONTINUE                                                          COK01470
      ICOUNT     =  ICOUNT + 1                                          COK01480
      READ(5,45) (DAT(ICOUNT,JK),JK=1,MVAR),Y(ICOUNT),X(ICOUNT)         COK01490
45    FORMAT(7F10.3)                                                    COK01500
      CHK        =  Y(ICOUNT)                                           COK01510
      CHK1       =  X(ICOUNT)                                           COK01520
      IF(CHK.EQ.0..AND.CHK1.EQ.0.) GO TO 50                             COK01530
      GO TO 40                                                          COK01540
50    CONTINUE                                                          COK01550
      ICOUNT     =  ICOUNT - 1                                          COK01560
C...                                                                    COK01570
C.......CHECK FOR UNDERSAMPLED LOCATIONS -- IF IKRIG = 1,               COK01580
C.......A ZERO DATA VALUE INDICATES UNDERSAMPLING.                      COK01590
C...                                                                    COK01600
      IF(IKRIG.NE.1) GO TO 52                                           COK01610
      KCOUNT     =  0                                                   COK01620
      DO 51 IJ = 1,ICOUNT                                               COK01630
      ITEMP      =  0                                                   COK01640
      DO 51 IK = 1,MVAR                                                 COK01650
      IF(DAT(IJ,IK).NE.0.0)  GO TO 51                                   COK01660
      IF(ITEMP.EQ.1)  GO TO 51                                          COK01670
      KCOUNT     =  KCOUNT + 1                                          COK01680
      ITEMP      =  1                                                   COK01690
      JUNSAM(KCOUNT) = IJ                                               COK01700
51    CONTINUE                                                          COK01710
52    CONTINUE                                                          COK01720
C...                                                                    COK01730
C.......ECHO INPUT DATA TO OUTPUT FILE                                  COK01740
C...                                                                    COK01750
      WRITE(6,53)                                                       COK01760
53    FORMAT(1H1,' ******************',T30,' CO-KRIGING PROGRAM',      COK01770
     1 T60,' **************',//)                                        COK01780
      IF(IKRIG.EQ.1) WRITE(6,60)                                        COK01790
      WRITE(6,55) NROW,NCOL,YMAX,XMAX,XDIM,YDIM                         COK01800
55    FORMAT(T25,' NO OF ROWS IN KRIGED ARRAY = ',T65,I5,/,             COK01810
     1 T25,' NO OF COLS IN KRIGED ARRAY = ',T65,I5,/,                   COK01820
     2 T25,' MAXIMUM Y COORDINATE       = ',T60,F10.3,/,                COK01830
     3 T25,' MAXIMUM X COORDINATE       = ',T60,F10.3,/,                COK01840
     4 T25,' INCREMENT ON X             = ',T60,F10.3,/,                COK01850
     5 T25,' INCREMENT ON Y             = ',T60,F10.3,//)               COK01860
60    FORMAT(T25,' UNDERSAMPLED CASE',//)                               COK01870
      WRITE(6,65) MVAR                                                  COK01880
65    FORMAT(T25,' A TOTAL OF ',I2,' VARIABLE(S) WILL BE ESTIMATED',    COK01890
     1 //)                                                              COK01900
      WRITE(6,70)                                                       COK01910
70    FORMAT(T10,'********* VARIOGRAM AND CROSS-VARIOGRAM PARAMETERS'COK01920
     1 T63,'*******',//,T20,'SINGLE VARIABLE (VARIOGRAM) PARAMETERS',   COK01930
     2 /,' VARIABLE',T10,' NUGGET',T20,'SILL',T30,'RANGE',T40,'ANGLE',  COK01940
     3 T50,' RATIO',T60,' INFLUENCE',T70,'   MODEL',/)                  COK01950
      DO 100 I = 1,MVAR                                                 COK01960
      WRITE(6,85) I,CO(I),C(I),RANGE(I),ANIS(I),RATIO(I),RINFLU(I),     COK01970
     1 MODEL(I)                                                         COK01980
85    FORMAT(I5,6F10.3,I10)                                             COK01990
100   CONTINUE                                                          COK02000
      IF(MVAR.EQ.1) GO TO 120                                           COK02010
      WRITE(6,105)                                                      COK02020
105   FORMAT(//,T20,' INTER-VARIABLE (CROSS-VARIOGRAM) PARAMETERS',/,   COK02030
     1 ' VARIABLE',T10,' NUGGET',T20,'SILL',T30,'RANGE',T40,'ANGLE',    COK02040
     2 T50,'RATIO',T60,' INFLUENCE',T70,'   MODEL',/)                   COK02050
      DO 110 I = 1,M2                                                   COK02060
      WRITE(6,85) I,CCO(I),CC(I),CRANGE(I),CANIS(I),CRATIO(I),          COK02070
     1 CRNFLU(I),CMODEL(I)                                              COK02080
110   CONTINUE                                                          COK02090
120   CONTINUE                                                          COK02100
      WRITE(6,130)                                                      COK02110
130   FORMAT(//,T30,' ***  INPUT DATA  ***',/,                          COK02120
     1 T10,' X-COORD',T20,' Y-COORD',T50,' DATA VALUES',//)             COK02130
      WRITE(6,131) ALAB                                                 COK02140
131   FORMAT(T25,15A4,/)                                                COK02150
      DO 140 I = 1,ICOUNT                                               COK02160
      WRITE(6,135) X(I),Y(I),(DAT(I,JK),JK=1,MVAR)                      COK02170
135   FORMAT(T5,7F10.3)                                                 COK02180
140   CONTINUE                                                          COK02190
C...                                                                    COK02200
C.......START KRIGING:  INCREMENT COLUMNS, THEN ROWS                    COK02210
C...                                                                    COK02220
      XBEGIN     =  XMAX - XDIM * (FLOAT(NCOL) + 0.5)                   COK02230
      YBEGIN     =  YMAX + (YDIM / 2.0)                                 COK02240
      ISTOP      =  8                                                   COK02250
      DO 170 II  =  1,MVAR                                              COK02260
      ANIS(II)   =  ANIS(II) * 0.01745329                               COK02270
      CANIS(II)  =  CANIS(II)* 0.01745329                               COK02280
170   CONTINUE                                                          COK02290
      WRITE(6,180)                                                      COK02300
180   FORMAT(//,T20,' **** ** ***   KRIGING RESULTS   *** ** ****',    COK02310
     1 /,2X,'ROW',1X,'COL',4X,'NORTH',5X,'WEST',25X,'DATA ESTIMATES',   COK02320
     2 T72,' VARIANCE',/)                                               COK02330
      WRITE(6,185) ALAB                                                 COK02340
185   FORMAT(T27,15A4,/)                                                COK02350
      DO 2000 JROW = 1,NROW                                             COK02360
      YCORD      =  YBEGIN - FLOAT(JROW) * YDIM                         COK02370
      DO 2000 JCOL = 1,NCOL                                             COK02380
      XCORD      =  XBEGIN + FLOAT(JCOL) * XDIM                         COK02390
      INIT       =  0                                                   COK02400
      DO 500 ISERCH = 1,ICOUNT                                          COK02410
      DIFX       =  XCORD - X(ISERCH)                                   COK02420
      DIFY       =  YCORD - Y(ISERCH)                                   COK02430
C...                                                                    COK02440
```

```
C.......COMPUTE DISTANCE BETWEEN SAMPLE AND ESTIMATION POINT
C...
      DO 200 IANG=  1,MVAR
      DIST(IANG) =  (DIFX*COS(ANIS(IANG))+DIFY*SIN(ANIS(IANG)))**2
     1              + (RATIO(IANG)*(DIFY*COS(ANIS(IANG))-DIFX
     2              * SIN(ANIS(IANG))))**2
      DIST(IANG) =  SQRT(DIST(IANG))
      IF(DIST(IANG).GT.RINFLU(IANG)) GO TO 500
200   CONTINUE
      IF(INIT.GE.ISTOP) GO TO 400
      INIT       =  INIT + 1
C..
C.......FIND CLOSEST SAMPLES ON THE BASIS OF FIRST VARIABLE.  IT IS
C.......ALREADY GUARANTEED BY THE 200 LOOP THAT ALL DISTANCES ARE
C.......WITHIN THE RANGE OF INFLUENCE OF EACH VARIABLE.
C...
      TEMP(INIT) =  DIST(1)
      NHOLE(INIT)=  ISERCH
      GO TO 500
400   CONTINUE
      X2         =  0.0
      KOUNT      =  0
      DO 450 III =  1,INIT
      IF(TEMP(III).LT.X2) GO TO 450
      KOUNT      =  III
      X2         =  TEMP(III)
450   CONTINUE
      IF(DIST(1).GT.X2) GO TO 500
      TEMP(KOUNT)=  DIST(1)
      NHOLE(KOUNT) = ISERCH
500   CONTINUE
      IF(INIT.LE.2) GO TO 2000
      MROW       =  INIT + 1
      MCOL       =  INIT + 1
      MTOT       =  MVAR * MCOL
      CALL AFORM(YCORD,XCORD,NHOLE)
      IF(ISOLV.EQ.0) CALL ART(XMEAS,XTEMP,IIT)
      IF(ISOLV.EQ.1) CALL EQSOLV(XMEAS,XTEMP)
C...
C.......COMPUTE CO-KRIGING ESTIMATES
C...
      DO 550 II = 1,MVAR
      EST(II)    =  0.0
550   CONTINUE
      M9         =  MCOL - 1
      DO 600 II  =  1,M9
      N1         =  NHOLE(II)
      DO 600 JJ  =  1,MVAR
      DO 600 KK  =  1,MVAR
      N2         =  (II - 1) * MVAR + KK
      EST(JJ)    =  EST(JJ) + DAT(N1,KK) * XTEMP(N2,JJ)
600   CONTINUE
C...
C.......COMPUTE CO-KRIGING VARIANCE AS:
C.......VAR = TR(C1) - TR(XMEAS*XTEMP) - TR(LAGRANGE)
C...
C.......TR(C1) = SUM OF INDIVIDUAL SILL VALUES
C...
      DUM1       =  0.0
      DO 700 II  =  1,MVAR
      DUM1       =  DUM1 + C(II)
700   CONTINUE
C.......COMPUTE TRACE(XMEAS*XTEMP)
      CALL RZRO(TEMP3,MVAR,MVAR)
      DO 750 II = 1,M9
      DO 710 JJ = 1,MVAR
      J9         =  (II - 1) * MVAR + JJ
      DO 710 KK = 1,MVAR
      TEMP1(JJ,KK) = XMEAS(J9,KK)
      TEMP2(JJ,KK) = XTEMP(J9,KK)
710   CONTINUE
      DO 720 JJ = 1,MVAR
      DO 720 KK = 1,MVAR
      DO 720 KJ = 1,MVAR
      TEMP3(JJ,KK) = TEMP3(JJ,KK) + TEMP1(JJ,KJ) * TEMP2(KJ,KK)
720   CONTINUE
750   CONTINUE
      DUM2       =  TRACE(TEMP3)
C.......COMPUTE TRACE(LAGRANGE)
      M8         =  M9 * MVAR
      DO 760 II  =  1,MVAR
      M7         =  M8 + II
      DO 760 JJ  =  1,MVAR
      TEMP2(II,JJ) = XTEMP(M7,JJ)
760   CONTINUE
      DUM3       =  TRACE(TEMP2)
      VAR        =  DUM1 - DUM2 - DUM3
      DO 761 IC  =  1,MVAR
      DO 761 ID  =  1,MVAR
      AW(IC,ID)  =  0.0
761   CONTINUE
      DO 765 IA  =  1,M9
      DO 765 IB  =  1,MVAR
      IC         =  (IA - 1) * MVAR + IB
      DO 765 ID  =  1,MVAR
      AW(IB,ID)  =  AW(IB,ID) + XTEMP(IC,ID)
765   CONTINUE
      WRITE(6,770) JROW,JCOL,YCORD,XCORD,(EST(JK),JK=1,MVAR),VAR
770   FORMAT(1X,2I4,8F9.3)
      IF(ICREAT.NE.1) GO TO 2000
      DO 780 IC = 1,MVAR
      WRITE(LT,790) (AW(IC,ID), ID=1,MVAR)
780   CONTINUE
      WRITE(LB,795) JROW,JCOL,(EST(JK),JK=1,MVAR),VAR
      WRITE(LT,920) IIT
790   FORMAT(T20.5F10.3)
```

```
795     FORMAT(2I5,6F10.3)                                              COK03510
800     CONTINUE                                                        COK03520
        DO 900 IA = 1,MVAR                                              COK03530
        DUM5        =  C(IA) - TEMP3(IA,IA) - TEMP2(IA,IA)              COK03540
        WRITE(LG,910) JROW,JCOL,IA,DUM5                                 COK03550
900     CONTINUE                                                        COK03560
910     FORMAT(T10,' ROW = ',I5,' COL = ',I5,' VARIABLE = ',I5,         COK03570
     1  T60,' VARIANCE = ',F10.3)                                       COK03580
920     FORMAT(/,T20,' NO. OF ITERATIONS OF ART = ',I5,//)              COK03590
2000    CONTINUE                                                        COK03600
        STOP                                                            COK03610
        END                                                             COK03620
        SUBROUTINE AFORM(YCORD,XCORD,KHOLE)                             COK03630
        COMMON /DAT2/ X(500),Y(500),DAT(500,5)                          COK03640
        COMMON /PARM/ MROW,MCOL,MVAR,MTOT                               COK03650
        COMMON /AMAT/ IKRIG,JUNSAM(500),KCOUNT                          COK03660
        COMMON /VAR/ CO(5),C(5),RANGE(5),MODEL(5),ANIS(5),RATIO(5),     COK03670
     1               RINFLU(5)                                          COK03680
        COMMON /CVAR/ CCO(10),CC(10),CRANGE(10),CMODEL(10),CANIS(10),   COK03690
     1               CRNFLU(10),CRATIO(10)                              COK03700
        COMMON /FORM/ XMEAS(100,5)                                      COK03710
        COMMON /FILES/ IUNIT                                            COK03720
        DIMENSION ATEMP(5,100),KHOLE(15),IPOS(5)                        COK03730
        INTEGER CMODEL                                                  COK03740
C...                                                                    COK03750
C.......THIS SUBROUTINE CREATES A SEQUENTIALLY ACCESSED DISK FILE       COK03760
C.......OF INTERSAMPLE COVARIANCES COMPATIBLE WITH SUBROUTINE ART.      COK03770
C.......FOR THE UNDERSAMPLED CASE, THE A-MATRIX IS MODIFIED ACCORDING   COK03780
C.......TO THE PROCEDURE OUTLINED IN:  MYERS, D.E., "CO-KRIGING -       COK03790
C.......THE MATRIX FORM,"  GEOSTAT TAHOE-83, NATO ASI, 6-17 SEPT.,1983. COK03800
C...                                                                    COK03810
        M1          =  MROW - 1                                         COK03820
        M2          =  MCOL - 1                                         COK03830
        REWIND IUNIT                                                    COK03840
        DO 1000 II  =  1,M1                                             COK03850
        NI          =  KHOLE(II)                                        COK03860
        DO 750  JJ  =  1,M2                                             COK03870
        KPOS        =  0                                                COK03880
        NK          =  KHOLE(JJ)                                        COK03890
        DO 500  KK  =  1,MVAR                                           COK03900
        KTOT        =  (JJ - 1) * MVAR + KK                             COK03910
        DO 250  LL  =  1,MVAR                                           COK03920
        NTOT        =  (JJ - 1) * MVAR + LL                             COK03930
        DIFX        =  X(NI) - X(NK)                                    COK03940
        DIFY        =  Y(NI) - Y(NK)                                    COK03950
        IF(LL.NE.KK)  GO TO 100                                         COK03960
        DISTAN      =  SQRT((DIFX*COS(ANIS(LL)) + DIFY*SIN(ANIS(LL)))   COK03970
     1                 **2 + (RATIO(LL)*(DIFY*COS(ANIS(LL)) - DIFX *    COK03980
     2                 SIN(ANIS(LL))))**2)                              COK03990
        I9          =  LL                                               COK04000
        ATEMP(KK,NTOT) = COVAR(DISTAN,I9)                               COK04010
        GO TO 250                                                       COK04020
100     CONTINUE                                                        COK04030
        IF(LL.LT.KK)  GO TO 250                                         COK04040
        KPOS        =  KPOS + 1                                         COK04050
        DISTAN      =  SQRT((DIFX*COS(CANIS(KPOS)) + DIFY*SIN(CANIS(KPOS COK04060
     1                 ))**2 + (CRATIO(KPOS)*(DIFY*COS(CANIS(KPOS)) -   COK04070
     2                 DIFX*SIN(CANIS(KPOS))))**2)                      COK04080
        I8          =  KK                                               COK04090
        I9          =  LL                                               COK04100
        ATEMP(KK,NTOT) = CROSS(DISTAN,KPOS,I8,I9)                       COK04110
        ATEMP(LL,KTOT) = ATEMP(KK,NTOT)                                 COK04120
250     CONTINUE                                                        COK04130
500     CONTINUE                                                        COK04140
C.......MODIFY ATEMP COLUMNS FOR UNDERSAMPLED CASE                      COK04150
        IF(IKRIG.NE.1) GO TO 750                                        COK04160
        IMOD        =  0                                                COK04170
        DO 600 IAB  =  1,KCOUNT                                         COK04180
        NA          =  JUNSAM(IAB)                                      COK04190
        IF(NK.EQ.NA)  IMOD = 1                                          COK04200
        IF(NK.EQ.NA)  GO TO 610                                         COK04210
600     CONTINUE                                                        COK04220
610     CONTINUE                                                        COK04230
        IF(IMOD.EQ.0)  GO TO 750                                        COK04240
        JOUNT = 0                                                       COK04250
        DO 650 IAC  =  1,MVAR                                           COK04260
        IF(DAT(NK,IAC).NE.0.0) GO TO 650                                COK04270
        JOUNT       =  JOUNT + 1                                        COK04280
        IPOS(JOUNT)=  IAC                                               COK04290
650     CONTINUE                                                        COK04300
        DO 670 IAD  =  1,JOUNT                                          COK04310
        KZ1         =  IPOS(IAD)                                        COK04320
        KZ2         =  (JJ - 1) * MVAR + KZ1                            COK04330
        DO 670 JAD  =  1,MVAR                                           COK04340
        ATEMP(JAD,KZ2) = 0.0                                            COK04350
670     CONTINUE                                                        COK04360
750     CONTINUE                                                        COK04370
C.......LAST MATRIX ENTRY IN ATEMP ROW IS AN IDENTITY MATRIX            COK04380
        DO 900  MM  =  1,MVAR                                           COK04390
        DO 900  NM  =  1,MVAR                                           COK04400
        KM          =  M2 * MVAR + NM                                   COK04410
        IF(MM.EQ.NM)  ATEMP(MM,KM) = 1.0                                COK04420
        IF(MM.NE.NM)  ATEMP(MM,KM) = 0.0                                COK04430
900     CONTINUE                                                        COK04440
C.......MODIFY ATEMP ROW FOR UNDERSAMPLED CASE                          COK04450
        IF(IKRIG.NE.1) GO TO 930                                        COK04460
        IMOD        =  0                                                COK04470
        DO 905 IAB  =  1,KCOUNT                                         COK04480
        NA          =  JUNSAM(IAB)                                      COK04490
        IF(NI.EQ.NA)  IMOD = 1                                          COK04500
        IF(NI.EQ.NA)  GO TO 906                                         COK04510
905     CONTINUE                                                        COK04520
906     CONTINUE                                                        COK04530
        IF(IMOD.EQ.0) GO TO 930                                         COK04540
        JOUNT       =  0                                                COK04550
        DO 910 IAC  =  1,MVAR                                           COK04560
```

```
      IF(DAT(NI,IAC).NE.0.0) GO TO 910                                  COK04570
      JOUNT       =  JOUNT + 1                                          COK04580
      IPOS(JOUNT)=  IAC                                                 COK04590
910   CONTINUE                                                          COK04600
      DO 915 IAD =  1,JOUNT                                             COK04610
      KZ1         =  IPOS(IAD)                                          COK04620
      DO 915 JAD =  1,MTOT                                              COK04630
      ATEMP(KZ1,JAD) = 0.0                                              COK04640
915   CONTINUE                                                          COK04650
930   CONTINUE                                                          COK04660
C.......WRITE RESULT (1 ROW OF MATRICES) TO DISK FILE                   COK04670
      DO 950 MM  =  1,MVAR                                              COK04680
      WRITE(IUNIT) (ATEMP(MM,JZ), JZ=1,MTOT)                            COK04690
950   CONTINUE                                                          COK04700
1000  CONTINUE                                                          COK04710
C...                                                                    COK04720
C.......LAST ROW OF A-MATRIX IS A ROW OF IDENTITY MATRICES              COK04730
C...                                                                    COK04740
      DO 1200 LM =  1,M2                                                COK04750
      DO 1200 MM =  1,MVAR                                              COK04760
      DO 1200 NM =  1,MVAR                                              COK04770
      NTOT        =  (LM - 1) * MVAR + NM                               COK04780
      IF(MM.EQ.NM)  ATEMP(MM,NTOT) = 1.0                                COK04790
      IF(MM.NE.NM)  ATEMP(MM,NTOT) = 0.0                                COK04800
1200  CONTINUE                                                          COK04810
C.......LAST MATRIX IN THIS ROW IS ZERO                                 COK04820
      N1          =  M2 * MVAR + 1                                      COK04830
      DO 1400 NM =  1,MVAR                                              COK04840
      DO 1400 MM = N1,MTOT                                              COK04850
      ATEMP(NM,MM) = 0.0                                                COK04860
1400  CONTINUE                                                          COK04870
C.......MODIFY THIS ROW FOR UNDERSAMPLED CASE                           COK04880
      IF(IKRIG.NE.1) GO TO 1490                                         COK04890
      DO 1480 IAB=  1,M2                                                COK04900
      NK          =  KHOLE(IAB)                                         COK04910
      IMOD        =  0                                                  COK04920
      DO 1420 IAC=  1,KCOUNT                                            COK04930
      NA          =  JUNSAM(IAC)                                        COK04940
      IF(NK.EQ.NA)  IMOD = 1                                            COK04950
      IF(NK.EQ.NA)  GO TO 1430                                          COK04960
1420  CONTINUE                                                          COK04970
1430  CONTINUE                                                          COK04980
      IF(IMOD.EQ.0) GO TO 1480                                          COK04990
      JOUNT       =  0                                                  COK05000
      DO 1440 IAD=  1,MVAR                                              COK05010
      IF(DAT(NK,IAD).NE.0.0) GO TO 1440                                 COK05020
      JOUNT       =  JOUNT + 1                                          COK05030
      IPOS(JOUNT)=  IAD                                                 COK05040
1440  CONTINUE                                                          COK05050
      DO 1450 IAE=  1,JOUNT                                             COK05060
      KZ1         =  IPOS(IAE)                                          COK05070
      KZ2         =  (IAB - 1) * MVAR + KZ1                             COK05080
      DO 1450 IAF=  1,MVAR                                              COK05090
      ATEMP(IAF,KZ2) = 0.0                                              COK05100
1450  CONTINUE                                                          COK05110
1480  CONTINUE                                                          COK05120
1490  CONTINUE                                                          COK05130
      DO 1500 MM =  1,MVAR                                              COK05140
      WRITE(IUNIT) (ATEMP(MM,JZ),JZ=1,MTOT)                             COK05150
1500  CONTINUE                                                          COK05160
C...                                                                    COK05170
C.......FORM THE MEASUREMENT VECTOR:  POINT-SAMPLE COVARIANCES          COK05180
C...                                                                    COK05190
      DO 2000 II =  1,M1                                                COK05200
      K7          =  KHOLE(II)                                          COK05210
      DIFX        =  XCORD - X(K7)                                      COK05220
      DIFY        =  YCORD - Y(K7)                                      COK05230
      KPOS        =  0                                                  COK05240
      DO 2000 JJ =  1,MVAR                                              COK05250
      KZ          =  (II - 1) * MVAR + JJ                               COK05260
      DO 2000 KK =  1,MVAR                                              COK05270
      KW          =  (II - 1) * MVAR + KK                               COK05280
      IF(KK.NE.JJ)  GO TO 1600                                          COK05290
      DISTAN      =  SQRT((DIFX*COS(ANIS(KK)) + DIFY*SIN(ANIS(KK)))     COK05300
     1               **2 + (RATIO(KK)*(DIFY*COS(ANIS(KK)) - DIFX        COK05310
     2               * SIN(ANIS(KK))))**2)                              COK05320
      I9          =  KK                                                 COK05330
      XMEAS(KZ,KK) = COVAR(DISTAN,I9)                                   COK05340
      GO TO 2000                                                        COK05350
1600  CONTINUE                                                          COK05360
      IF(KK.LT.JJ)   GO TO 2000                                         COK05370
      KPOS        =  KPOS + 1                                           COK05380
      DISTAN      =  SQRT((DIFX*COS(CANIS(KPOS)) + DIFY*SIN(CANIS(      COK05390
     1               KPOS)))**2 + (CRATIO(KPOS)*(DIFY*COS(CANIS(KPOS))  COK05400
     2               - DIFX*SIN(CANIS(KPOS))))**2)                      COK05410
      I8          =  JJ                                                 COK05420
      I9          =  KK                                                 COK05430
      XMEAS(KZ,KK) = CROSS(DISTAN,KPOS,I8,I9)                           COK05440
      XMEAS(KW,JJ) = XMEAS(KZ,KK)                                       COK05450
2000  CONTINUE                                                          COK05460
C.......MODIFY MEASUREMENT VECTOR FOR UNDERSAMPLED CASE                 COK05470
      IF(IKRIG.NE.1) GO TO 2400                                         COK05480
      DO 2350 II =  1,M1                                                COK05490
      IMOD        =  0                                                  COK05500
      K7          =  KHOLE(II)                                          COK05510
      DO 2200 JJ =  1,KCOUNT                                            COK05520
      NA          =  JUNSAM(JJ)                                         COK05530
      IF(K7.EQ.NA)  IMOD = 1                                            COK05540
      IF(K7.EQ.NA)  GO TO 2210                                          COK05550
2200  CONTINUE                                                          COK05560
2210  CONTINUE                                                          COK05570
      IF(IMOD.EQ.0) GO TO 2350                                          COK05580
      JOUNT       =  0                                                  COK05590
      DO 2250 KK =  1,MVAR                                              COK05600
      IF(DAT(K7,KK).NE.0.0) GO TO 2250                                  COK05610
      JOUNT       =  JOUNT + 1                                          COK05620
```

```
      IPOS(JOUNT)=  KK                                                  COK05630
2250  CONTINUE                                                          COK05640
      DO 2260 LL =  1,JOUNT                                             COK05650
      K8         =  (II - 1) * MVAR + IPOS(LL)                          COK05660
      DO 2260 MM =  1,MVAR                                              COK05670
      XMEAS(K8,MM)  =  0.0                                              COK05680
2260  CONTINUE                                                          COK05690
2350  CONTINUE                                                          COK05700
2400  CONTINUE                                                          COK05710
C.......LAST ENTRY IN MEASUREMENT VECTOR IS AN IDENTITY MATRIX          COK05720
      N1         =  M1 * MVAR                                           COK05730
      DO 2500 II =  1,MVAR                                              COK05740
      N3         =  N1 + II                                             COK05750
      DO 2500 JJ =  1,MVAR                                              COK05760
      IF(JJ.EQ.II)  XMEAS(N3,JJ) = 1.0                                  COK05770
      IF(JJ.NE.II)  XMEAS(N3,JJ) = 0.0                                  COK05780
2500  CONTINUE                                                          COK05790
      RETURN                                                            COK05800
      END                                                               COK05810
      FUNCTION COVAR(DIST,K)                                            COK05820
      COMMON /VAR/ CO(5),C(5),RANGE(5),MODEL(5),ANIS(5),RATIO(5),       COK05830
     1             RINFLU(5)                                            COK05840
C...                                                                    COK05850
C.......THIS FUNCTION EVALUATES THE MODEL COVARIANCE ASSOCIATED         COK05860
C.......WITH THE SEPARATION DISTANCE,DIST.                              COK05870
C...                                                                    COK05880
      II         =  MODEL(K)                                            COK05890
      GO TO (100,200,300) II                                            COK05900
100   CONTINUE                                                          COK05910
C...                                                                    COK05920
C.......SPHERICAL COVARIANCE                                            COK05930
C...                                                                    COK05940
      IF(DIST.GE.RANGE(K)) GO TO 120                                    COK05950
      B          =  C(K) - CO(K)                                        COK05960
      DUM1       =  CO(K) + B*(1.5*DIST/RANGE(K) - 0.5*(DIST/           COK05970
     1             RANGE(K))**3)                                        COK05980
      IF(DIST.EQ.0.0) DUM1 = 0.0                                        COK05990
      COVAR      =  C(K) - DUM1                                         COK06000
      RETURN                                                            COK06010
120   CONTINUE                                                          COK06020
      COVAR      =  0.0                                                 COK06030
      RETURN                                                            COK06040
200   CONTINUE                                                          COK06050
C...                                                                    COK06060
C.......GAUSSIAN COVARIANCE                                             COK06070
C...                                                                    COK06080
      B          =  C(K) - CO(K)                                        COK06090
      R3         =  1.732050808                                         COK06100
      RAN2       =  RANGE(K)/R3                                         COK06110
      DR         =  (DIST/RAN2) ** 2                                    COK06120
      DR         =  -DR                                                 COK06130
      DUM1       =  CO(K) + B*(1.0 - EXP(DR))                           COK06140
      IF(DIST.EQ.0.0) DUM1 = 0.0                                        COK06150
      COVAR      =  C(K) - DUM1                                         COK06160
      RETURN                                                            COK06170
300   CONTINUE                                                          COK06180
C...                                                                    COK06190
C.......LINEAR COVARIANCE                                               COK06200
C...                                                                    COK06210
      B          =  C(K) - CO(K)                                        COK06220
      SLOPE      =  B/RANGE(K)                                          COK06230
      DUM1       =  CO(K) + SLOPE*DIST                                  COK06240
      IF(DIST.EQ.0.0) DUM1 = 0.0                                        COK06250
      COVAR      =  C(K) - DUM1                                         COK06260
      RETURN                                                            COK06270
      END                                                               COK06280
      FUNCTION CROSS(DIST,K,IA,IB)                                      COK06290
      COMMON /CVAR/ CCO(10),CC(10),CRANGE(10),CMODEL(10),CANIS(10),     COK06300
     1              CRNFLU(10),CRATIO(10)                               COK06310
      INTEGER CMODEL                                                    COK06320
C...                                                                    COK06330
C.......THIS FUNCTION EVALUATES THE CROSS-COVARIANCE VALUE              COK06340
C.......CORRESPONDING TO THE DISTANCE, DIST.                            COK06350
C...                                                                    COK06360
      II         =  CMODEL(K)                                           COK06370
      GO TO (100,200,300)  II                                           COK06380
100   CONTINUE                                                          COK06390
C...                                                                    COK06400
C.......SPHERICAL CROSS-COVARIANCE                                      COK06410
C...                                                                    COK06420
      IF(DIST.GE.CRANGE(K)) GO TO 120                                   COK06430
      B          =  CC(K) - CCO(K)                                      COK06440
      DUM1       =  CCO(K) + B*(1.5*DIST/CRANGE(K)                      COK06450
     1             - 0.5*(DIST/CRANGE(K))**3)                           COK06460
      IF(DIST.EQ.0.0) DUM1 = 0.0                                        COK06470
      DUM2       =  CC(K) - DUM1                                        COK06480
      GO TO 400                                                         COK06490
120   CONTINUE                                                          COK06500
      DUM2       =  0.0                                                 COK06510
      GO TO 400                                                         COK06520
200   CONTINUE                                                          COK06530
C...                                                                    COK06540
C.......GAUSSIAN CROSS-COVARIANCE                                       COK06550
C...                                                                    COK06560
      B          =  CC(K) - CCO(K)                                      COK06570
      R3         =  1.732050808                                         COK06580
      RAN2       =  CRANGE(K) / R3                                      COK06590
      DR         =  (DIST/RAN2) ** 2                                    COK06600
      DR         =  -DR                                                 COK06610
      DUM1       =  CCO(K) + B*(1.0 - EXP(DR))                          COK06620
      IF(DIST.EQ.0.0) DUM1 = 0.0                                        COK06630
      DUM2       =  CC(K) - DUM1                                        COK06640
      GO TO 400                                                         COK06650
300   CONTINUE                                                          COK06660
C...                                                                    COK06670
C.......LINEAR CROSS-COVARIANCE                                         COK06680
```

```
C...
      B            =  CC(K) - CCO(K)
      SLOPE        =  B/CRANGE(K)
      DUM1         =  CCO(K) + SLOPE*DIST
      IF(DIST.EQ.0.0) DUM1 = 0.0
      DUM2         =  CC(K) - DUM1
400   CONTINUE
      DUM3         =  COVAR(DIST,IA)
      DUM4         =  COVAR(DIST,IB)
      CROSS        =  0.5 * (DUM2 - DUM3 - DUM4)
      RETURN
      END
      SUBROUTINE ART(XMEAS,XTEMP,IIT)
C...
C.......SUBROUTINE ART SOLVES THE SYSTEM OF LINEAR EQUATIONS,
C.......AX=B, USING AN ITERATIVE TECHNIQUE DEVELOPED BY
C.......K. TANABE.
C...
C.......REFERENCES:1)TANABE, K. (1971)., NUMER. MATH.,17,203-214.
C.......            2)HERMAN,G.T.,LENT,A., AND ROWLAND,S.,
C.......              J. THEOR. BIOLOGY (1973), 42, 1-32.
C.......            PROCEDURES DESCRIBED ABOVE WERE MODIFIED BY
C.......            CARR,J.R., FOR SUB-MATRIX OPERATION.
C...
      COMMON /FILES/ IUNIT
      COMMON /PARM/  MROW,MCOL,MVAR,MTOT
      DIMENSION ATEMP(5,100),RIQ(5,5),XLAST(100,5)
      DIMENSION XMEAS(100,5),XTEMP(100,5)
      DIMENSION TEMP(5,5),TEMP1(5,5),TEMP2(5,5),TEMP3(5,5)
      REAL NI(5,5)
C...
C.......INITIALIZE UNITS AND VARIABLES
C...
      REWIND IUNIT
      CHECK        =  0.0
C...
C.......IUNIT       =  DISK DEVICE CONTAINING THE ARRAY, A
C.......MVAR        =  NUMBER OF VARIABLES TO BE ESTIMATED
C.......MROW        =  NUMBER OF SUB-MATRIX ROWS OF A
C.......MCOL        =  NUMBER OF SUB-MATRIX COLUMNS OF A
C.......MTOT        =  MCOL * MVAR
C...
C.......FIRST GUESS AT SOLUTION VECTOR, X, IS ZERO.
C.......(HERMAN, ET.AL., P.7, EQ. 3).
C...
      DO 20 J = 1,MCOL
      DO 20 K = 1,MVAR
      XROW         =  (J - 1) * MVAR + K
      DO 10 L = 1,MVAR
      XTEMP(XROW,L) = 0.0
      XLAST(XROW,L) = 0.0
10    CONTINUE
20    CONTINUE
C...
C.......BEGIN ART ITERATION
C...
      DO 1000 II = 1,200
      IIT          =  II
C.......II CONTROLS ITERATIONS -- MAXIMUM NUMBER IS 5.
      DO 500  KK =  1,MROW
C...
C.......READ ONE ROW OF SUB-MATRICES OF ARRAY, A, INTO
C.......CORE, THEN COMPUTE N-SUB-I (HERMAN,ET.AL.,P.7, EQ.1).
C...
      DO 100 IA  =  1,MVAR
      READ(IUNIT) (ATEMP(IA,IB),IB = 1,MTOT)
100   CONTINUE
      CALL RZRO(NI,MVAR,MVAR)
      CALL ANI(NI,ATEMP)
C...
C.......COMPUTE RIQ(HERMAN,ET.AL., P.7, EQ. 2).
C...
      CALL RZRO(RIQ,MVAR,MVAR)
      CALL ARIQ(ATEMP,XTEMP,RIQ)
C...
C.......UPDATE THE SOLUTION VECTOR, X. (HERMAN, ET.AL.,P.7, EQ. 3).
C...
      I7           =  KK
      CALL SOLN(XTEMP,ATEMP,XMEAS,RIQ,I7,NI)
C.......INCREMENT KK -- ROW COUNTER.
500   CONTINUE
      REWIND IUNIT
C...
C.......CHECK FOR CONVERGENCE (SQUARE-ERROR DIFFERENCE CHECK).
C...
      CALL RZRO(TEMP3,MVAR,MVAR)
      DO 600 I = 1,MCOL
      DO 550 J = 1,MVAR
      JB           =  (I - 1) * MVAR + J
      DO 550 K = 1,MVAR
      TEMP(J,K)    =  XTEMP(JB,K) - XLAST(JB,K)
      TEMP1(K,J)   =  TEMP(J,K)
550   CONTINUE
      CALL MATMUL(TEMP1,TEMP,TEMP2,MVAR,MVAR,MVAR)
      DO 590 J = 1,MVAR
      DO 590 K = 1,MVAR
      TEMP3(J,K)   =  TEMP3(J,K) + TEMP2(J,K)
590   CONTINUE
600   CONTINUE
      DIFF         =  TRACE(TEMP3)
      A2           =  0.01 * CHECK
C...
C.......UPDATE XLAST ARRAY
C...
      DO 800 I = 1,MCOL
      DO 800 J = 1,MVAR
```

```
      JB          =  (I - 1) * MVAR + J                                COK07750
      DO 800 K = 1,MVAR                                                COK07760
      XLAST(JB,K)=  XTEMP(JB,K)                                        COK07770
800   CONTINUE                                                         COK07780
C     IF(DIFF.LE.A2) RETURN                                            COK07790
      CHECK       =  DIFF                                              COK07800
1000  CONTINUE                                                         COK07810
      RETURN                                                           COK07820
      END                                                              COK07830
      SUBROUTINE MATMUL(A,B,C,J,K,L)                                   COK07840
      DIMENSION A(5,5),B(5,5),C(5,5)                                   COK07850
C...                                                                   COK07860
C.......SUBROUTINE MATMUL PERFORMS A GENERAL MATRIX                    COK07870
C.......MULTIPLICATION OF THE FORM:  C = A X B.                        COK07880
C...                                                                   COK07890
      JJ          =  J                                                 COK07900
      JL          =  L                                                 COK07910
      CALL RZRO(C,JJ,JL)                                               COK07920
      DO 100 I    =  1,J                                               COK07930
      DO 100 M    =  1,L                                               COK07940
      DO 100 N    =  1,K                                               COK07950
      DUM1        =  ABS(0.0 - A(I,N))                                 COK07960
      DUM2        =  ABS(0.0 - B(N,M))                                 COK07970
      IF(DUM1.LT.0.000000001) A(I,N) = 0.0                             COK07980
      IF(DUM2.LT.0.000000001) B(N,M) = 0.0                             COK07990
      C(I,M)      =  C(I,M) + A(I,N) * B(N,M)                          COK08000
100   CONTINUE                                                         COK08010
      RETURN                                                           COK08020
      END                                                              COK08030
      SUBROUTINE ANI(NI,ATEMP)                                         COK08040
      COMMON /PARM/ MROW,MCOL,MVAR,MTOT                                COK08050
      REAL NI(5,5),ATEMP(5,100)                                        COK08060
      DIMENSION TEMP1(5,5),TEMP2(5,5),TEMP3(5,5)                       COK08070
C...                                                                   COK08080
C.......THIS SUBROUTINE COMPUTES EQ.1 OF HERMAN, ET.AL., P.7,          COK08090
C.......PROCEDURE MODIFIED FOR MATRIX OPERATIONS.                      COK08100
C...                                                                   COK08110
      DO 100 I = 1,MCOL                                                COK08120
      DO 20  J = 1,MVAR                                                COK08130
      DO 20  K = 1,MVAR                                                COK08140
      KB          =  (I - 1) * MVAR + K                                COK08150
      TEMP1(J,K)  =  ATEMP(J,KB)                                       COK08160
      TEMP3(K,J)  =  TEMP1(J,K)                                        COK08170
20    CONTINUE                                                         COK08180
      CALL MATMUL(TEMP3,TEMP1,TEMP2,MVAR,MVAR,MVAR)                    COK08190
      DO 50  J = 1,MVAR                                                COK08200
      DO 50  K = 1,MVAR                                                COK08210
      NI(J,K)     =  NI(J,K) + TEMP2(J,K)                              COK08220
50    CONTINUE                                                         COK08230
100   CONTINUE                                                         COK08240
      RETURN                                                           COK08250
      END                                                              COK08260
      SUBROUTINE RZRO(A,I,J)                                           COK08270
C.......THIS SUBROUTINE INITIALIZES REAL ARRAYS                        COK08280
      REAL A(5,5)                                                      COK08290
      DO 10 K = 1,I                                                    COK08300
      DO 10 L = 1,J                                                    COK08310
      A(K,L)  = 0.0                                                    COK08320
10    CONTINUE                                                         COK08330
      RETURN                                                           COK08340
      END                                                              COK08350
      FUNCTION TRACE(A)                                                COK08360
      COMMON /PARM/ MROW,MCOL,MVAR,MTOT                                COK08370
      REAL A(5,5)                                                      COK08380
C...                                                                   COK08390
C.......THIS FUNCTION COMPUTES THE TRACE OF THE SQUARE MATRIX, A.      COK08400
C...                                                                   COK08410
      TRACE       =  0.0                                               COK08420
      DO 100  I   =  1,MVAR                                            COK08430
      TRACE       =  TRACE + A(I,I)                                    COK08440
100   CONTINUE                                                         COK08450
      RETURN                                                           COK08460
      END                                                              COK08470
      SUBROUTINE SOLN(XTEMP,ATEMP,XMEAS,RIQ,KK,NI)                     COK08480
      COMMON /PARM/ MROW,MCOL,MVAR,MTOT                                COK08490
      DIMENSION XTEMP(100,5),ATEMP(5,100),XMEAS(100,5)                 COK08500
      DIMENSION RIQ(5,5),TEMP(5,5),TEMP1(5,5)                          COK08510
      DIMENSION TEMP2(5,5),TEMP3(5,5)                                  COK08520
      REAL NI(5,5)                                                     COK08530
C...                                                                   COK08540
C.......THIS SUBROUTINE PERFORMS EQ. 3, P.7, OF HERMAN, ET.AL.,        COK08550
C.......PROCEDURE MODIFIED FOR MATRIX OPERATIONS.                      COK08560
C...                                                                   COK08570
      DO 200  I   =  1,MCOL                                            COK08580
C.......FORM THE DIFFERENCE, RI - RIQ, AND NORMALIZE BY NI.            COK08590
      CALL RZRO(TEMP,MVAR,MVAR)                                        COK08600
      DO 100  J   =  1,MVAR                                            COK08610
      JB          =  (KK - 1) * MVAR + J                               COK08620
      DO 100  K   =  1,MVAR                                            COK08630
      TEMP(J,K)   =  (XMEAS(JB,K) - RIQ(J,K))                          COK08640
100   CONTINUE                                                         COK08650
      CALL SCALAR(NI,TEMP,TEMP3)                                       COK08660
C.......FORM THE PRODUCT: PIJ * (RI - RIQ)/NI.                         COK08670
      DO 150  J   =  1,MVAR                                            COK08680
      DO 150  K   =  1,MVAR                                            COK08690
      KB          =  (I - 1) * MVAR + K                                COK08700
      TEMP1(J,K)  =  ATEMP(J,KB)                                       COK08710
150   CONTINUE                                                         COK08720
      CALL MATMUL(TEMP1,TEMP3,TEMP2,MVAR,MVAR,MVAR)                    COK08730
      CALL RZRO(TEMP,MVAR,MVAR)                                        COK08740
      DO 175  J   =  1,MVAR                                            COK08750
      JB          =  (I - 1) * MVAR + J                                COK08760
      DO 175  K   =  1,MVAR                                            COK08770
      TEMP(J,K)   =  XTEMP(JB,K) + TEMP2(J,K)                          COK08780
      XTEMP(JB,K)=  TEMP(J,K)                                          COK08790
175   CONTINUE                                                         COK08800
```

```
200     CONTINUE                                                        COK08810
        RETURN                                                          COK08820
        END                                                             COK08830
        SUBROUTINE ARIQ(ATEMP,XTEMP,RIQ)                                COK08840
        COMMON /PARM/ MROW,MCOL,MVAR,MTOT                               COK08850
        DIMENSION ATEMP(5,100),XTEMP(100,5),RIQ(5,5)                    COK08860
        DIMENSION TEMP1(5,5),TEMP2(5,5),TEMP3(5,5)                      COK08870
C...                                                                    COK08880
C.......THIS SUBROUTINE PERFORMS EQ. 2, P.7, OF HERMAN, ET.AL.,         COK08890
C.......PROCEDURE MODIFIED FOR MATRIX OPERATIONS.                       COK08900
C...                                                                    COK08910
        DO 100  I  =  1,MCOL                                            COK08920
        DO 20   J  =  1,MVAR                                            COK08930
        JB         =  (I - 1) * MVAR + J                                COK08940
        DO 20   K  =  1,MVAR                                            COK08950
        KB         =  (I - 1) * MVAR + K                                COK08960
        TEMP1(J,K) =  ATEMP(J,KB)                                       COK08970
        TEMP2(J,K) =  XTEMP(JB,K)                                       COK08980
20      CONTINUE                                                        COK08990
        CALL MATMUL(TEMP1,TEMP2,TEMP3,MVAR,MVAR,MVAR)                   COK09000
        DO 50   J  =  1,MVAR                                            COK09010
        DO 50   K  =  1,MVAR                                            COK09020
        RIQ(J,K)   =  RIQ(J,K) + TEMP3(J,K)                             COK09030
50      CONTINUE                                                        COK09040
100     CONTINUE                                                        COK09050
        RETURN                                                          COK09060
        END                                                             COK09070
        SUBROUTINE SCALAR(A,B,X)                                        COK09080
        COMMON /PARM/ MROW,MCOL,MVAR,MTOT                               COK09090
        DIMENSION A(5,5),B(5,5),X(5,5)                                  COK09100
        REAL NI                                                         COK09110
C...                                                                    COK09120
C.......THIS SUBROUTINE PERFORMS THE OPERATION: (RI - RIQ)/(NI)         COK09130
C.......WHERE (RI - RIQ) AND (NI) ARE SQUARE MATRICES.                  COK09140
C...                                                                    COK09150
C.......FIRST GUESS AT SOLN IS ZERO; HERMAN, EQ. 3.                     COK09160
        CALL RZRO(X,MVAR,MVAR)                                          COK09170
C.......COMPUTE NI; HERMAN, EQ.1.                                       COK09180
        NITN       = MVAR * 100                                         COK09190
        DO 100 II  = 1,NITN                                             COK09200
        DO 50  KK  = 1,MVAR                                             COK09210
        NI         = 0.0                                                COK09220
        DO 20  J   = 1,MVAR                                             COK09230
        NI         = NI + A(KK,J) ** 2                                  COK09240
20      CONTINUE                                                        COK09250
        IF(NI.EQ.0.0) GO TO 50                                          COK09260
C.......ADD LL LOOP BECAUSE SOLN IS A  2-D ARRAY, NOT A VECTOR.         COK09270
        DO 45  LL  = 1,MVAR                                             COK09280
C.......COMPUTE RIQ (HERMAN, EQ.2).                                     COK09290
        RIQ        = 0.0                                                COK09300
        DO 30  J   = 1,MVAR                                             COK09310
        RIQ        = RIQ + A(KK,J) * X(J,LL)                            COK09320
30      CONTINUE                                                        COK09330
C.......COMPUTE NEW SOLUTION; HERMAN, EQ. 3.                            COK09340
        DO 35  J   = 1,MVAR                                             COK09350
        X(J,LL)    = X(J,LL) + (A(KK,J) * (B(KK,LL) - RIQ)/NI)          COK09360
35      CONTINUE                                                        COK09370
45      CONTINUE                                                        COK09380
50      CONTINUE                                                        COK09390
100     CONTINUE                                                        COK09400
        RETURN                                                          COK09410
        END                                                             COK09420
        SUBROUTINE EQSOLV(XMEAS,XTEMP)                                  COK09430
        COMMON/FILES/ IUNIT                                             COK09440
        COMMON/PARM/ MROW,MCOL,MVAR,MTOT                                COK09450
        DIMENSION ATEMP(100,100),XTEMP(100,5),XMEAS(100,5)              COK09460
        DIMENSION TEMP(5,5),TEMP1(5,5),TEMP2(5,5),TEMP3(5,5)            COK09470
C...                                                                    COK09480
C.......SUBROUTINE EQSOLV IS A SPECIALIZED VERSION OF THE               COK09490
C.......GAUSS DECOMPOSITION FOR SUB-MATRIX SOLUTION.                    COK09500
C.......FIRST STEP:  FILL ATEMP ARRAY FOR IN-CORE SOLUTION.             COK09510
C...                                                                    COK09520
        REWIND IUNIT                                                    COK09530
        DO 10 I=1,MROW                                                  COK09540
        DO 10 J=1,MVAR                                                  COK09550
        JB         =  (I-1)*MVAR + J                                    COK09560
        READ(IUNIT) (ATEMP(JB,JK),JK=1,MTOT)                            COK09570
10      CONTINUE                                                        COK09580
        DO 15 I=1,MTOT                                                  COK09590
        DO 15 J=1,MVAR                                                  COK09600
        N          =  MTOT + J                                          COK09610
        ATEMP(I,N) =  XMEAS(I,J)                                        COK09620
15      CONTINUE                                                        COK09630
C...                                                                    COK09640
C.......BEGIN GAUSS DECOMPOSITION                                       COK09650
C...                                                                    COK09660
        MN         =  MROW + 1                                          COK09670
        DO 100 I = 1,MROW                                               COK09680
        IP         =  I + 1                                             COK09690
        DO 20  KJ = 1,MVAR                                              COK09700
        KB         =  (I-1)*MVAR + KJ                                   COK09710
        DO 20  KK = 1,MVAR                                              COK09720
        KC         =  (I-1)*MVAR + KK                                   COK09730
        TEMP(KJ,KK)=  ATEMP(KB,KC)                                      COK09740
20      CONTINUE                                                        COK09750
        DUM1       =  TRACE(TEMP)                                       COK09760
        DO 100 J = 1,MROW                                               COK09770
        IF(I-J) 40,100,40                                               COK09780
40      CONTINUE                                                        COK09790
        DO 50 KJ = 1,MVAR                                               COK09800
        KB         =  (J-1)*MVAR + KJ                                   COK09810
        DO 50 KK = 1,MVAR                                               COK09820
        KC         =  (I-1)*MVAR + KK                                   COK09830
        TEMP1(KJ,KK) =-ATEMP(KB,KC)                                     COK09840
50      CONTINUE                                                        COK09850
        IF(MVAR.NE.1) GO TO 71                                          COK09860
```

```
      DO 70 KJ = 1,MVAR                                                 COK09870
      DO 70 KK = 1,MVAR                                                 COK09880
      TEMP1(KJ,KK) =  TEMP1(KJ,KK)/DUM1                                 COK09890
70    CONTINUE                                                          COK09900
      GO TO 79                                                          COK09910
71    CONTINUE                                                          COK09920
      CALL SCALG(TEMP,TEMP1,TEMP3)                                      COK09930
      DO 72 KJ = 1,MVAR                                                 COK09940
      DO 72 KK = 1,MVAR                                                 COK09950
      TEMP1(KJ,KK) = TEMP3(KJ,KK)                                       COK09960
72    CONTINUE                                                          COK09970
79    CONTINUE                                                          COK09980
      DO 90 K = IP,MN                                                   COK09990
      DO 80 KA = 1,MVAR                                                 COK10000
      JA          =  (I-1)*MVAR + KA                                    COK10010
      DO 80 KB = 1,MVAR                                                 COK10020
      JB          =  (K-1)*MVAR + KB                                    COK10030
      TEMP2(KA,KB) =  ATEMP(JA,JB)                                      COK10040
80    CONTINUE                                                          COK10050
      CALL MATMUL(TEMP1,TEMP2,TEMP3,MVAR,MVAR,MVAR)                     COK10060
      DO 90 KA = 1,MVAR                                                 COK10070
      JA          =  (J-1)*MVAR + KA                                    COK10080
      DO 90 KB = 1,MVAR                                                 COK10090
      JB          =  (K-1)*MVAR + KB                                    COK10100
      ATEMP(JA,JB) = ATEMP(JA,JB) + TEMP3(KA,KB)                        COK10110
90    CONTINUE                                                          COK10120
100   CONTINUE                                                          COK10130
      DO 200 I = 1,MROW                                                 COK10140
      DO 150 KJ = 1,MVAR                                                COK10150
      KB          =  (I-1)*MVAR + KJ                                    COK10160
      DO 150 KK = 1,MVAR                                                COK10170
      KC          =  (I-1)*MVAR + KK                                    COK10180
      TEMP(KJ,KK)=  ATEMP(KB,KC)                                        COK10190
150   CONTINUE                                                          COK10200
      DUM2        =  TRACE(TEMP)                                        COK10210
      DO 160 J = 1,MVAR                                                 COK10220
      JA          =  (I-1)*MVAR + J                                     COK10230
      DO 160 K = 1,MVAR                                                 COK10240
      JB          =  MTOT + K                                           COK10250
      TEMP2(J,K) =  ATEMP(JA,JB)                                        COK10260
160   CONTINUE                                                          COK10270
      IF(MVAR.NE.1) GO TO 180                                           COK10280
      DO 170 J = 1,MVAR                                                 COK10290
      JA          =  (I-1)*MVAR + J                                     COK10300
      DO 170 K = 1,MVAR                                                 COK10310
      JB          =  MTOT + K                                           COK10320
      XTEMP(JA,K)=  ATEMP(JA,JB)/DUM2                                   COK10330
170   CONTINUE                                                          COK10340
      GO TO 190                                                         COK10350
180   CONTINUE                                                          COK10360
      CALL SCALG(TEMP,TEMP2,TEMP3)                                      COK10370
      DO 185 J = 1,MVAR                                                 COK10380
      JA          =  (I-1)*MVAR + J                                     COK10390
      DO 185 K = 1,MVAR                                                 COK10400
185   XTEMP(JA,K)=TEMP3(J,K)                                            COK10410
190   CONTINUE                                                          COK10420
200   CONTINUE                                                          COK10430
      RETURN                                                            COK10440
      END                                                               COK10450
C                                                                       COK10460
C                                                                       COK10470
      SUBROUTINE SCALG(A,B,X)                                           COK10480
      COMMON /PARM/ MROW,MCOL,MVAR,MTOT                                 COK10490
      DIMENSION A(5,5),B(5,5),X(5,5),TEMP(5,10)                         COK10500
C...                                                                    COK10510
C.......THIS SUBROUTINE PERFORMS THE NORMALIZATION BY THE               COK10520
C.......DIAGONAL TERM FOR GAUSS ELIMINATION FOR MATRIX                  COK10530
C.......OPERATIONS.                                                     COK10540
C...                                                                    COK10550
      MVAR2       =  MVAR * 2                                           COK10560
      DO 10   I   =  1,MVAR                                             COK10570
      DO 10   J   =  1,MVAR                                             COK10580
      TEMP(I,J)   =  A(I,J)                                             COK10590
10    CONTINUE                                                          COK10600
      DO 20   I   =  1,MVAR                                             COK10610
      DO 20   J   =  1,MVAR                                             COK10620
      K           =  MVAR + J                                           COK10630
      TEMP(I,K)   =  B(I,J)                                             COK10640
20    CONTINUE                                                          COK10650
      DO 30   I   =  1,MVAR                                             COK10660
      IP          =  I + 1                                              COK10670
      DO 30   K   =  1,MVAR                                             COK10680
      IF(TEMP(I,I).EQ.0.0) GO TO 30                                     COK10690
      IF(I-K) 26,30,26                                                  COK10700
26    F           =  (-TEMP(K,I)) / TEMP(I,I)                           COK10710
      DO 27   L   =  IP,MVAR2                                           COK10720
27    TEMP(K,L)   =  TEMP(K,L) + F*TEMP(I,L)                            COK10730
30    CONTINUE                                                          COK10740
      DO 40   I   =  1,MVAR                                             COK10750
      DO 40   J   =  1,MVAR                                             COK10760
      K           =  MVAR + J                                           COK10770
      IF(TEMP(I,I).EQ.0.0) GO TO 40                                     COK10780
      X(I,J)      =  TEMP(I,K) / TEMP(I,I)                              COK10790
40    CONTINUE                                                          COK10800
      RETURN                                                            COK10810
      END                                                               COK10820
```

APPENDIX II. KRIGING EXAMPLE

```
******************          COKRIGING PROGRAM          ***************

                        NO OF ROWS IN KRIGED ARRAY =              1
                        NO OF COLS IN KRIGED ARRAY =              1
                        MAXIMUM Y COORDINATE       =        250.000
                        MAXIMUM X COORDINATE       =        250.000
                        INCREMENT ON X             =          4.550
                        INCREMENT ON Y             =          4.550

                        A TOTAL OF  1 VARIABLE(S) WILL BE ESTIMATED

        ********** VARIOGRAM AND CROSS-VARIOGRAM PARAMETERS  *******
                   SINGLE VARIABLE (VARIOGRAM) PARAMETERS
VARIABLE NUGGET    SILL      RANGE      ANGLE      RATIO      INFLUENCE  MODEL

   1      0.500     1.800    30.000     0.000      1.000    100.000        1

                                  ***  INPUT DATA  ***
          X-COORD    Y-COORD                          DATA VALUES

                                 INTENSITY
       132.360     91.170        7.000
       133.210    102.280        7.000
        71.850    182.890        5.000
        76.490    173.440        5.000
       141.490     94.500        7.000
       167.240     71.710        6.000
       119.210     92.611        7.000
       108.810    163.430        6.000
       169.670     58.920        5.000
       189.820    130.080        5.000
       132.550     63.370        5.000
       220.260     93.390        5.000
         0.000    135.640        5.000
        97.860    141.200        6.000
       143.470    152.310        6.000
        72.370     44.470        6.000
       248.490     57.810        5.000
        44.410     98.950        6.000

                  **** ** ***   KRIGING RESULTS   *** ** ****
ROW COL     NORTH      WEST                       DATA ESTIMATES      VARIANCE
                                 INTENSITY VARIANCE
  1   1    91.170   132.360       6.663     1.370
  2   1   102.280   133.210       6.463     1.520
  3   1   182.890    71.850       5.426     1.695
  4   1   173.440    76.490       5.640     1.664
  5   1    94.500   141.490       6.413     1.486
  6   1    71.710   167.240       5.530     1.747
  7   1    92.611   119.210       6.177     1.721
  8   1   163.430   108.810       5.978     2.016
  9   1    58.920   169.670       5.868     1.757
 10   1   130.080   189.820       5.849     1.984
 11   1    63.370   132.550       5.953     1.982
 12   1    93.390   220.260       5.724     2.046
 13   1   135.640     0.000       5.576     2.318
 14   1   141.200    97.860       5.804     1.955
 15   1   152.310   143.470       5.712     1.975
 16   1    44.470    72.370       6.088     2.107
 17   1    57.810   248.490       5.219     2.306
 18   1    98.950    44.410       5.865     2.007
```

APPENDIX III. CO-KRIGING EXAMPLE

```
*******************          COKRIGING PROGRAM          ***************

                    NO OF ROWS IN KRIGED ARRAY =              1
                    NO OF COLS IN KRIGED ARRAY =              1
                    MAXIMUM Y COORDINATE       =        250.000
                    MAXIMUM X COORDINATE       =        250.000
                    INCREMENT ON X             =          4.550
                    INCREMENT ON Y             =          4.550

                    A TOTAL OF  2 VARIABLE(S) WILL BE ESTIMATED

        ********** VARIOGRAM AND CROSS-VARIOGRAM PARAMETERS  *******
                    SINGLE VARIABLE (VARIOGRAM) PARAMETERS
VARIABLE NUGGET     SILL        RANGE       ANGLE       RATIO       INFLUENCE   MODEL

   1     1.500      12.000      30.000      0.000       1.000       100.000       1
   2     0.500       1.800      30.000      0.000       1.000       100.000       1

                    INTER-VARIABLE (CROSS-VARIOGRAM) PARAMETERS
VARIABLE NUGGET     SILL        RANGE       ANGLE       RATIO       INFLUENCE   MODEL

   1     0.050       2.000      30.000      0.000       1.000       100.000       1

                                *** INPUT DATA ***
          X-COORD   Y-COORD                          DATA VALUES

                                VELOCITY    INTENSITY
       132.360     91.170       10.200      7.000
       133.210    102.280       15.600      7.000
        71.850    182.890        1.000      5.000
        76.490    173.440        3.800      5.000
       141.490     94.500        8.200      7.000
       167.240     71.710        2.300      6.000
       119.210     92.611        5.100      7.000
       108.810    163.430       11.700      6.000
       169.670     58.920        3.900      5.000
       189.820    130.080        2.000      5.000
       132.550     63.370        6.100      5.000
       220.260     93.390        1.500      5.000
         0.000    135.640        1.700      5.000
        97.860    141.200        6.200      6.000
       143.470    152.310        7.600      6.000
        72.370     44.470        3.500      6.000
       248.490     57.810        2.300      5.000
        44.410     98.950        3.200      6.000

                    **** ** ***   KRIGING RESULTS   *** ** ****
ROW COL     NORTH       WEST                        DATA ESTIMATES        VARIANCE
                                VELOCITY INTENSITY  VARIANCE
  1   1    91.170     132.360     8.191     7.010     9.327
  2   1   102.280     133.210     6.962     6.750    10.843
  3   1   182.890      71.850     5.005     5.256    12.109
  4   1   173.440      76.490     4.446     5.232    11.927
  5   1    94.500     141.490     9.647     6.878    10.376
  6   1    71.710     167.240     4.885     5.348    12.877
  7   1    92.611     119.210     8.069     6.427    12.666
  8   1   163.430     108.810     5.602     5.911    15.470
  9   1    58.920     169.670     4.133     5.738    12.943
 10   1   130.080     189.820     6.181     5.782    15.234
 11   1    63.370     132.550     5.051     5.879    15.217
 12   1    93.390     220.260     5.847     5.652    15.726
 13   1   135.640       0.000     3.701     5.625    17.832
 14   1   141.200      97.860     5.673     5.776    14.990
 15   1   152.310     143.470     5.474     5.654    15.176
 16   1    44.470      72.370     5.985     5.982    16.234
 17   1    57.810     248.490     2.334     5.205    17.722
 18   1    98.950      44.410     6.092     5.801    15.420
```