Computers & Geosciences Vol. 17, No. 9, pp. 1265–1280, 1991
Printed in Great Britain. All rights reserved

0098-3004/91 $3.00 + 0.00

COKRIGING WITH MATLAB

DENIS MARCOTTE
Département de Génie Minéral, Ecole Polytechnique, C.P. 6079, Succ. A, Montréal, Québec, Canada H3C 3A7

(Received 2 April 1991; accepted 21 May 1991)

Abstract—A program termed COKRI is presented, which will perform point or block kriging or cokriging in any number of dimensions, with any number of variables and basic structures. Different forms of cokriging are offered by COKRI: simple, ordinary with one nonbias condition or with the usual p (number of variables) nonbias conditions, and universal cokriging with drift of order 1 or 2. Factorial kriging or cokriging also can be performed. All the basic structures can have different geometric anisotropies thus allowing great modeling flexibility. The addition of a new basic model to the five currently offered by COKRI requires only one line of code. The program operates within the user-friendly Matlab environment.

Key Words: Block cokriging, Factorial cokriging, Universal cokriging, Matlab, User-friendly, Personal computer.

INTRODUCTION

Cokriging is the multivariable extension of kriging. The theory was presented first for the estimation of one variable from several variables by Matheron (1970), and then by numerous other authors (e.g. Journel and Huijbregts, 1978; François-Bongarçon, 1981; Myers, 1982, 1983, 1984, 1988; Dong, 1988; Isaaks and Srivastava, 1989). Meyers (1982, 1988) first presented a general formulation in matrix form for the simultaneous estimation of many variables. Many cokriging applications in different fields of interest are given in the literature: McBratney and Webster (1983), Vauclin and others (1983), Yates and Warrick (1987), and Leenaers, Okx, and Burrough (1989) in the soil sciences domain; Aboufirassi and Marino (1984), Hoeksama and Kitanidis (1985), Ahmed, de Marsily, and Talbot (1988), Ahmed and de Marsily (1987, 1989), Ribeiro and Muge (1989), Muge and Cabecadas (1989), Hoeksama and others (1989), and Rouhani and Myers (1990) in the hydrological domain; Smyth and Istok (1989) in the study of water contamination; and Carr and others (1987) and Creutin, Andrieu, and Delrieu (1989) in teledetection. A special situation of cokriging aimed at filtering the components of a signal, factorial kriging (Matheron, 1982, Sandjivy, 1984; Galli, Gerdil-Neuillet, and Dadou, 1984; Chiles and Guillen, 1984; Daly, Jeulin, and Lajaunie, 1989; Wackernagel, Petitgas, and Touffait, 1989), also have received much attention. Finally, a FORTRAN computer program for performing point cokriging (in 2-D) with undersampled data was presented in Carr, Myers, and Glass (1985).

The object of this paper is to present a computer program that can perform point or block, simple, ordinary, or universal (with drift of order 1 or 2) kriging or cokriging for a field of any dimension in the fully sampled or undersampled case. The program can perform cross-validation and factorial kriging or cokriging as well. It is written in Matlab, a widespread software package for matrix operations which shares many of the characteristics of a high-level programming language. It operates interactively on a personal computer (IBM, Macintosh, Sun, etc.) in the user-friendly Matlab environment and, as such, may be modified easily to meet particular user requirements. In particular, results of cokriging can be visualized either as a contour map or a 3-D mesh representation. Drivers for different hard-copy devices are available within Matlab. A review of Matlab is given in Oman (1990). Applications are given in, among others, Zygmont (1984, 1986), Barraud, Laporte, and Gentil (1986), Tetewsky (1987), Moler (1988), and Boyle, Ford, and Maciejowski (1988). Possibilities offered by Matlab as a teaching tool are contained in O'Haver (1989), DeMoyer and Mitchell (1990), Haupt (1990), de Boor (1990), and Alvarado and others (1991).

EQUATIONS AND NOTATIONS

For simplicity, the notation assumes a fully sampled case. The definitions are as follows:

d = dimensionality of the field. Coordinate vector is $x = (x_1, x_2 \ldots x_d)$
n = number of data points for cokriging

p = number of variables
r = number of elementary variogram models
$\mathbf{A}$ = a $p \times r$ matrix of coefficients in the linear coregionalization model
$\mathbf{I}$ = the $p \times p$ identity matrix
$\mathbf{II}$ = the $np \times p$ matrix constituted of $n\mathbf{I}$ submatrices
$\mathbf{K}$ = the $np \times np$ matrix of point–point covariances between data points; this matrix is formed of $n \times n$ submatrices $\mathbf{K}_{ij}$, each of size $p \times p$; $\mathbf{K}_{ij}$ gives the covariances between points i and j for the p variables
$\mathbf{K}_0$ = the $np \times p$ matrix of point–block covariances between data points and block or point to estimate
$\mathbf{X}$ = the $np \times t$ matrix with coordinate monomials; t depends on the dimensionality d and on the order of the polynomial drift
$\mathbf{Z}$ = the $n \times p$ data matrix
$\mathbf{z}$ = the $np \times 1$ data vector. The first p values of $\mathbf{z}$ corresponds to the first row of $\mathbf{Z}$, the next p values to the second row of $\mathbf{Z}$ and so on
$\mathbf{z}_\mu$ = the vector $\mathbf{z}$ centered around the mean vector μ ($\mathbf{z}_\mu = \mathbf{z} - \mathbf{II}\mu$)
$\mathbf{z}_0$ = the $p \times 1$ vector of unknown values to estimate at point x_0
$\boldsymbol{\Gamma}$ = the $np \times p$ matrix of cokriging lambda weights
$\boldsymbol{\sigma}^2$ = the $p \times 1$ vector of point variances for the p variables
$\boldsymbol{\sigma}_v^2$ = the $p \times 1$ vector of block variances for the p variables
$\boldsymbol{\mu}$ = the $p \times 1$ vector of global means
$\mathbf{0}$ = a vector or matrix of zeros
$\mathbf{1}$ = a vector or matrix of ones

Simple cokriging

Under second-order stationarity and if the global means are known beforehand or a reliable estimate is available, the simple cokriging estimate can be formed with:

$$\mathbf{z}_0^* = \boldsymbol{\Gamma}^T \mathbf{z}_\mu + \boldsymbol{\mu} \tag{1}$$

where $\mathbf{z}_0^*$ is the estimated vector of the p variables at point x_0 of coordinates $[x_1, x_2, \ldots, x_d]$, and $\boldsymbol{\Gamma}$ is the solution matrix ($np \times p$) of the simple kriging system:

$$\mathbf{K}\boldsymbol{\Gamma} = \mathbf{K}_0 \tag{2}$$

where $\mathbf{K}$ is the ($np \times np$) matrix of point–point covariances for the p variables, $\mathbf{K}$ is formed of $n \times n$ submatrices $\mathbf{K}_{ij}$ of size $p \times p$ giving the covariances between point i and point j for the p variables, $\mathbf{K}_0$ is the $np \times p$ matrix of point–(point or block to estimate) covariances, $\mathbf{K}_0$ is formed of n matrices $\mathbf{K}_{0i}$ of size $p \times p$ giving the covariances between point i and point or block to estimate for the p variables.

The simple cokriging variances of individual variable are given by:

$$\boldsymbol{\sigma}_k^2 = \boldsymbol{\sigma}^2 - \mathrm{diag}(\mathbf{K}_0^T \boldsymbol{\Gamma}) \tag{3}$$

where $\mathrm{diag}(\mathbf{K}_0^T\boldsymbol{\Gamma})$ stands for the $\mathbf{p} \times 1$ vector formed of the diagonal elements of $\mathbf{K}_0^T\boldsymbol{\Gamma}$.

Ordinary cokriging

When the means of the p variables are unknown, constraints are added to the simple cokriging system in order to ensure the absence of bias:

$$\mathbf{z}_0^{b*} = \boldsymbol{\Gamma}^{b^T} \mathbf{z}^b \tag{4}$$

where

$$\mathbf{z}^b = \begin{pmatrix} \mathbf{z} \\ 0 \end{pmatrix} \tag{5}$$

where $\boldsymbol{\Gamma}^b$ satisfies the equation:

$$\mathbf{K}^b \boldsymbol{\Gamma}^b = \mathbf{K}_0^b \tag{6}$$

with:

$$\mathbf{K}^b = \begin{pmatrix} \mathbf{K} & \mathbf{II} \\ \mathbf{II}^T & \mathbf{0} \end{pmatrix} \tag{7}$$

where $\mathbf{II}$ is the $np \times p$ matrix formed of n identity matrices of size p, $\mathbf{0}$ is a $p \times p$ matrix of zeros, and

$$\mathbf{K}_0^b = \begin{pmatrix} \mathbf{K}_0 \\ \mathbf{I} \end{pmatrix}. \tag{8}$$

The ordinary cokriging variances of each individual variable are given by:

$$\boldsymbol{\sigma}_k^{2b} = \boldsymbol{\sigma}^2 - \mathrm{diag}(\mathbf{K}_0^{b^T} \boldsymbol{\Gamma}^b). \tag{9}$$

Isaaks and Srivastava (1989) proposed modifying the ordinary cokriging estimate by putting only one, instead of p, nonbias constraint in the ordinary cokriging system. The constraint requires that the sum of all the cokriging weights be equal to 1. This estimate requires knowledge of the global means of the p variables (or the assumption that the variables present the same global means). Here, local stationarity of the covariances is required as well as global stationarity of the means. In comparison, ordinary cokriging requires the local stationarity of both, whereas simple cokriging requires global stationarity.

Isaaks and Srivastava (1989) proposed the following cokriging estimator:

$$\mathbf{z}_0^{a*} = \boldsymbol{\Gamma}^{a^T} \mathbf{z}_\mu^a + \boldsymbol{\mu} \tag{10}$$

where $\boldsymbol{\Gamma}^a$ satisfies the equation:

$$\mathbf{K}^a \boldsymbol{\Gamma}^a = \mathbf{K}_0^a \tag{11}$$

with:

$$\mathbf{K}^a = \begin{pmatrix} \mathbf{K} & \mathbf{1} \\ \mathbf{1}^T & \mathbf{0} \end{pmatrix}. \tag{12}$$

$$\mathbf{K}_0^a = \begin{pmatrix} \mathbf{K}_0 \\ \mathbf{1}^T \end{pmatrix} \tag{13}$$

The cokriging variances now are given by:

$$\boldsymbol{\sigma}_k^2 = \boldsymbol{\sigma}^2 - \mathrm{diag}(\mathbf{K}_0^{a^T} \boldsymbol{\Gamma}^a). \tag{14}$$

In a case study, Isaaks and Srivastava (1989, p. 412) reported better results with this estimator than with ordinary kriging. As one referee pointed out, in some circumstances, it also might be interesting to retain the nonbias constraints for the auxiliary variables and to remove the nonbias constraint for the main variable to estimate.

Universal cokriging

Just as for kriging, the cokriging system can be modified to filter out an unknown drift by adding additional constraints to the cokriging system. It is assumed here that the drift, for each variable, can be represented adequately, locally, by a low-order polynomial (in practice, of order 1 or 2).

If

$$Z(x) = m(x) + Y(x) \tag{15}$$

with

$$m(x) = \Sigma c_l f_l(x)$$

$f_l(x)$: a monomial in one of the coordinates. Then, the universal cokriging estimate is:

$$\mathbf{z}_0^{c*} = \boldsymbol{\Gamma}^{c^T}\mathbf{z}^c \tag{16}$$

with

$$\mathbf{z}^c = \begin{pmatrix} \mathbf{z} \\ \mathbf{0} \end{pmatrix} \tag{17}$$

The universal cokriging system is:

$$\mathbf{K}^c\boldsymbol{\Gamma}^c = \mathbf{K}_0^c \tag{18}$$

with:

$$\mathbf{K}^c = \begin{pmatrix} \mathbf{K} & \mathbf{II} & \mathbf{X} \\ \mathbf{II}^T & \mathbf{0} & \mathbf{0} \\ \mathbf{X}^T & \mathbf{0} & \mathbf{0} \end{pmatrix} \tag{19}$$

$$\mathbf{K}_0^c = \begin{pmatrix} \mathbf{K}_0 \\ \mathbf{I} \\ \mathbf{X}_0 \end{pmatrix} \tag{20}$$

where **X** is the $np \times t$ matrix of monomials in the coordinates, **X** is composed of n submatrices of size $p \times t$, each submatrix is constituted of p times the vector of monomials. As an example, in 3-D, with 2 variables, 4 observations and an order 2 drift, **X** is given by:

$$\mathbf{X}^T = \begin{bmatrix} x_1 & x_1 & \dots & x_4 & x_4 \\ y_1 & y_1 & \dots & y_4 & y_4 \\ z_1 & z_1 & \dots & z_4 & z_4 \\ x_1^2 & x_1^2 & \dots & x_1^4 & x_1^4 \\ x_1y_1 & x_1y_1 & \dots & x_4y_4 & x_4y_4 \\ . & . & \dots & . & . \\ y_1z_1 & y_1z_1 & \dots & y_4z_4 & y_4z_4 \\ z_1^2 & z_1^2 & \dots & z_4^2 & z_4^2 \end{bmatrix}. \tag{21}$$

The universal cokriging variances are given by:

$$\boldsymbol{\sigma}_k^2 = \boldsymbol{\sigma}^2 - \operatorname{diag}(\boldsymbol{\Gamma}^{c^T}\mathbf{K}_0^c). \tag{22}$$

The problem of inferencing the covariance function in the presence of a drift is a difficult one. Different approaches are possible. The easiest one consists in calculating, when possible, the variograms and cross-variograms along a direction where the drift is constant and thus has no influence on the structure (Chiles and Gable, 1984). Another approach is first to estimate the drift (e.g with a polynomial function estimated by least squares) and then calculate the variograms of the residuals. This method has two major drawbacks: first the resulting variograms will be biased (Matheron, 1973; Sabourin, 1976); second, even if the bias can be neglected, the drift will have to be estimated globally in order to avoid sharp boundary discontinuities between local polynomial surfaces. A third approach generalizes the FAI-k theory (Matheron, 1973) to the multivariate situation. Although this does not pose major theoretical problems, the estimation of the generalized covariance function is difficult (Dowd, 1989) in the multivariate situation.

Note the obvious hierarchy of all these cokrigings:

$$\boldsymbol{\sigma}_k \leqslant \boldsymbol{\sigma}_k^a \leqslant \boldsymbol{\sigma}_k^b \leqslant \boldsymbol{\sigma}_k^c. \tag{23}$$

Block cokriging

The same equations hold for the various forms of block cokriging except that σ_v^2 (the block variances) replace σ^2 in the expressions for the cokriging variances and the right member represent now averaged values of K_0 for points descretizing the block centered on point x_0. Details are given in Myers (1984).

The undersampled case

When variables have not been observed at some sample locations, the only modification needed for the preceding cokriging systems consists in removing the rows in both members, and the columns in the left member corresponding to the missing values. Myers (1984) demonstrated that this is the equivalent of imposing a zero value to all the weights corresponding to missing values.

Factorial cokriging

Factorial cokriging (Matheron, 1982; Sandjivy, 1984; Chiles and Guillen, 1984; Wackernagel, Petitgas, and Touffait, 1989; Bourgault and Marcotte, 1991) is the decomposition of p observed variables in r underlying (unobserved) variables. It usually is assumed that these underlying variables are orthonormal, although this is not necessary. Factorial cokriging can be seen as an extreme example of undersampling where the underlying variables are not observed at all sample points. The model is:

$$\mathbf{Z}(x) = \mathbf{A}\mathbf{Y}(x) + \boldsymbol{\mu} \tag{24}$$

where $\mathbf{Y}(x)$ is the $r \times 1$ vector of underlying zero expectation orthogonal random variables (not directly observable) at location x, $\mathbf{Z}(x)$ is the $p \times 1$ vector of original random variables at location x, **A** is the $p \times r$ matrix of coefficients relating underlying $\mathbf{Y}(x)$ variables to observed $\mathbf{Z}(x)$ variables. The coefficients normally would be estimated after fitting all variograms and cross-variograms of the observed variables.

The preceeding model (24) provides the information necessary to compute all the covariances required for cokriging the $\mathbf{Y}(x)$ variables from the observed $\mathbf{Z}(x)$ variables. For example, assuming that the $\mathbf{Y}(x)$ are orthonormal, the covariances between **Z** variables at

Table 1. Description of different functions in COKRI

Function name	Description	Calling program
COKRI	Main program Echo input options search of nearest neighbors Computation of variance of points (or blocks) in universe Call repetitively COKRI2 to perform cokriging	Matlab
COKRI2	Calculate distances and evaluate variograms and cross-variograms Construct left and right member of cokriging equations Add required constraints following cokriging type Solve cokriging system and compute cokriging estimate and cokriging variance	COKRI
TRANS	Rotation and reduction of coordinates following specification described in matrix 'model'	COKRI2
MEANS	Computation of means of matrix or of vector	COKRI COKRI2

location x and $\mathbf{Z}$ variables at location $x + h$ will be given by:

$$\mathbf{C}_{zz}(h) = \mathbf{A}\mathbf{D}_r(h)\mathbf{A}^T \tag{25}$$

where $\mathbf{D}_r(h)$ is a diagonal matrix of size r with ith element equal to the covariance between $Y_i(x)$ and $Y_i(x + h)$.

Similarly, the covariance between $\mathbf{Z}(x)$ and $\mathbf{Y}(x + h)$ are obtained from:

$$\mathbf{C}_{ZY}(h) = \mathbf{A}\mathbf{D}_r(h). \tag{26}$$

The following relation hold true when $r \geqslant p$ between estimated Z and Y variables:

$$\mathbf{Z}(x)^* = \mathbf{A}\mathbf{Y}(x)^* + \mu. \tag{27}$$

When $r < p$, the cokriging matrix is singular. Among the infinite number of solutions possible, the solution obtained with the Moore–Penrose inverse of $\mathbf{K}$ presents good properties. For further details see Bourgault and Marcotte (1991).

DESCRIPTION OF THE PROGRAM COKRI

The program runs only in the user-friendly interactive Matlab environment. It is composed of four functions. Table 1 describes the roles of each function and the relations between these functions. The program can perform simple cokriging, two different ordinary cokrigings, universal cokriging with drift of order 1 or 2 and factorial cokriging. It allows for point or block cokriging, in the fully sampled or undersampled case. It also permits cross-validation either by removing, at an observation point, one variable at a time or all variables simultaneously. When the program is called with only one variable, the results will be those of a kriging of the corresponding type (simple, ordinary, universal, point, or block).

Program limitations

The limitations of COKRI are those of the Matlab package and depend on the computer system used to run Matlab. On an IBM PC or AT, the maximum size of any matrix is limited to 8188 elements. Table 2 gives

Table 2. Maximum size of main matrices of COKRI

Function name	Main matrices	Maximum size*
COKRI	x	n•(d + p)
	x0	m•d
	x0s	m•(d + p)
	s	m•(d + p)
COKRI2	h	(nk + (ntok•ng))•(nk + (ntok•ng))
	k (line 42)	(nk p)•((nk + (ntok•ng))•p)
	k	((nk + nc)•p)•((nk + nc)•p)
	k0	((nk + nc)•p)•(ntok•ng•p)
	l	((nk + nc)•p)•(ntok•p)

*Description of variables: n-total number of samples; d-dimension of the field; p-number of variables; nk-number of neighbors used in the cokriging; ntok-number of points or blocks to cokrige simultaneously (ntok may always be set to 1); nb-number of points for the discretization of blocks; nc-number of constraints added to the simple cokriging system.

the size of the main matrices used in the program as a function of the different options. As an example, in 2–D, with three variables, the data matrix is limited to 8188/5 = 1637 samples; the size of the left member in simple cokriging will be $nk^2 * 3^2$ where nk is the number of samples used for kriging, thus limiting nk to 30. In addition to this limitation on individual matrices, there also is a limitation related to the total space available on the computer. On 386 systems, however, virtual memory can be used and there is no limitation on the space occupied by any single matrix, or on the total space, other than the amount of space available on the hard disk.

Input–output description

Once in Matlab, COKRI is called with the following statement:

[x0s,s,sv,id,b] =

cokri(x,x0,model,c,itype,avg,block,nd,ival,nk,rad,ntok)

The left part of the assignment is the output, whereas the elements in parentheses on the right part are input values that must be specified prior to the call.

Output elements:

x0s = the $m \times (p + d)$ matrix of grid coordinates and estimated cokriged values.

s = the $m \times (p + d)$ matrix of grid coordinates and cokriging variances.

sv = the $1 \times p$ vector of point or block variances (or pseudovariances) when the model does not have a sill) for the p variables.

The next two output items refer only to the last cokriging system solved:

id = is a matrix of identifiers of the kriging weights. The first column gives the sample number and second column the variable number.

b = the matrix of kriging weights and Lagrange multipliers.

Input elements:

x = The $n \times (p + d)$ data matrix. This data matrix can be imported from an existing ASCII file. Missing values are coded 'nan' (not-a-number).

x0 = The $m \times d$ matrix of coordinates of points to estimate. A regular grid matrix is easy to create in Matlab. An irregular grid can be imported from an ASCII file.

model = Each row of this matrix describes a different elementary structure. The first column is a code for the model type, the d following columns give the ranges along the different coordinates and the subsequent columns give rotation angles (a maximum of three).

The codes for the current models are: (1) nugget effect, (2) exponential model, (3) Gaussian model, (4) spherical model, (5) linear model.

Examples ['any' stands for 'any positive (excluding zero) value']:

1-D: $\text{model} = \begin{matrix} 1 & \text{any} \\ 4 & 30 \end{matrix}$

This specifies a nugget component plus a spherical model of range 30.

2-D: $\text{model} = \begin{matrix} 1 & \text{any} \\ 4 & 30 \end{matrix}$

This specifies a nugget plus an isotropic spherical model of range 30.

$\text{model} = \begin{matrix} 1 & \text{any} & \text{any} & \text{any} \\ 4 & 30 & 10 & 20 \end{matrix}$

This specifies a nugget plus an anisotropic spherical model of range 30 in the first rotated coordinate direction and range 10 in the second rotated coordinate direction. The rotation angle is 20° counterclockwise.

3-D: $\text{model} = \begin{matrix} 1 & \text{any} \\ 4 & 30 \end{matrix}$

This specifies a nugget plus an anisotropic spherical model of range 30.

$\text{model} = \begin{matrix} 1 & \text{any} & \text{any} & \text{any} & \text{any} & \text{any} & \text{any} \\ 4 & 30 & 10 & 50 & 20 & 45 & 30 \end{matrix}$

This specifies a nugget plus an anisotropic model of range 30, 10, and 50 along the rotated axes. These axes are obtained after rotation of 20, 45, and 30° around the first coordinate, the second rotated coordinate and the third rotated coordinate, in that order. These rotations are done counterclockwise looking in the negative direction of the axes.

D > 3: $\text{model} = \begin{matrix} 1 & \text{any} \\ 4 & 30 \end{matrix}$

This specifies a nugget plus an anisotropic spherical model of range 30.

$\text{model} = \begin{matrix} 1 & \text{any (repeated } (d+3) \text{ times)} \\ 4 & (d \text{ ranges) 3 rotation angles} \end{matrix}$

This specifies a nugget plus a spherical model with d ranges in the rotated coordinate system. The rotations take place only on the first three coordinates as before, leaving the other coordinates unchanged.

Note: a linear model is specified by arbitrary ranges and a sill such that sill/ranges gives the desired slope in the direction considered.

c: The $(rp \times p)$ coefficient matrix of the coregionalization model. Position (i,j) in each submatrix of size $p \times p$ give the sill of the elementary component for each cross-variogram (variogram) between variable i and variable j.

itype: Code to indicate which type of cokriging is to be performed: (1) simple cokriging, (2) ordinary cokriging with one nonbias condition (Isaaks and Srivastava, 1989), (3) ordinary cokriging with p nonbias condition, (4) universal cokriging with drift of order 1, (5) universal cokriging with drift of order 2.

99: cokriging is not performed, only sv is computed.

block: Vector $(1 \times d)$, giving the size of the block to estimate; any values when point cokriging is required.

nd: Vector $(1 \times d)$, giving the discretization grid for block cokriging; put every element equal to 1 for point cokriging.

ival: Code for cross-validation.

0: no-cross-validation
1: cross-validation is performed by removing one variable at a time at a given location.
2: cross-validation is performed by removing all variables at a given location.

nk: Number of nearest neighbors in x matrix to use in the cokriging (this includes locations with missing values even if all variables are missing).

rad: Search radius for neighbors.

ntok: Points in x0 will be kriged by groups of ntok grid points. When ntok > 1, the search will determine the nk nearest samples within distance rad from the current ntok grid points centroid.

Adding new models to COKRI

Adding a new model to the five already included in COKRI is really an easy task. It suffices to add the equation for the new model as a literal in matrix GAM at lines 12–16 of function COKRI2. The model equation is written in terms of covariance (or pseudo-covariance) for the rotated reduced h distance. As an illustration, the following models would be written:

Piecewise linear:	‘1-min(1,h) ’
Sine hole effect:	‘sin(h*2*pi)./max(eps,h*2*pi)’
Cosine hole effect:	‘cos(h*2*pi) ’

For the last two models, the ‘range’ specified in ‘model’ will be the period.

Quadratic model: ‘1-2*min(h,1) + min(h,1).∧2 ’

Note that the program will not, in its present version, check for the admissibility of the basic models. For example, the piecewise linear model is an admissible model only in 1-D. The Gaussian model (Christakos, 1985), the nugget, and the linear models are admissible in any dimension. The program could be modified easily to check for the admissibility of the basic models but remains the question of the validity of the coregionalization model as a whole. It is known from François-Bongarçon (1981) that a necessary but not sufficient (for more than two variables) condition of the admissibility of the coregionalization model is that:

$$|\gamma_{ij}(h)| \leqslant \sqrt{(\gamma_{ii}(h)\gamma_{jj}(h))}. \qquad (28)$$

In the frame of the linear coregionalization model, it is possible to estimate model parameters with procedures that ensure Equation (28) (Bourgault and Marcotte, 1991). When there is more than two variables and the coregionalization model differs from the linear model of coregionalization, respect of Equation (28) is more difficult to check. A function (CHECKMOD) is presented in Appendix 2 that simulates points into a segment (1-D), a circle (2-D), a sphere (3-D), or a hypersphere (D > 3), and evaluates (28) for all possible pairs of locations and variables with the coregionalization model given in input.

EXAMPLES

In this section, some small examples of execution of the program in different situations is presented. These are not aimed at demonstrating the usefulness of cokriging but only to provide numerical examples to check computations and to show input required. The first example uses published data (Carr, Myers, and Glass, 1985), whereas the other examples use small synthetic data sets and models.

Example 1: Carr, Myers, and Glass (1985) data

The first example uses the data and coregionalization model from Carr, Myers, and Glass (1985). Their coregionalization model is not admissible because it does not respect Equation (28). It should be noted that the cross-variogram model given in Carr, Myers, and Glass (1985) represents γ^+_{ij}, with $\gamma^+_{ij} = 2\gamma_{ij} + \gamma_{ii} + \gamma_{jj}$, whereas this program requires γ_{ij} for input. Consequently, the coregionalization model is entered into this program as:

$$\text{model} = \begin{array}{ccl} 1 & 1 & \text{nugget} \\ 4 & 30 & \text{spherical} \end{array}$$

$$\text{c} = \begin{array}{cc|l} 1.5 & -0.975 & \text{nugget} \\ -0.975 & 0.5 & \\ 12.0 & -5.9 & \text{spherical} \\ -5.9 & 1.8 & \end{array}$$

The condition (28) is not satisfied and thus the model is not admissible. Nevertheless, kriging and cokriging are performed as a computational check. As in Carr, Myers, and Glass (1985), cross-validation kriging and cokriging by removing all of the variables at sample points was performed. All of the samples occurring within a search radius of 100 (units are not specified on their paper) were used.

COKRI results for kriging are:

Coordinates		Kriging estimates	
x	y	Intensity	σ_k^2
132.360	91.170	6.663	1.370
133.210	102.280	6.463	1.520
71.850	182.890	5.426	1.695
76.490	173.440	5.640	1.664
141.490	94.500	6.413	1.486
167.490	71.710	5.530	1.747
119.210	92.611	6.177	1.721
108.810	163.430	5.978	2.016
169.670	58.920	5.868	1.757
189.820	130.080	5.849	1.984
132.550	63.370	5.953	1.982
220.260	93.390	5.724	2.046
0.000	135.640	5.576	2.318
97.860	141.200	5.804	1.955
143.470	152.310	5.712	1.975
72.370	44.470	6.088	2.107
248.490	57.810	5.219	2.306
44.410	98.950	5.865	2.007

Whereas the cokriging results are:

Coordinates		Cokriging estimates		Corkiging variances		
x	y	velocity	intensity	velocity	intensity	sum
132.360	91.170	8.203	7.010	7.946	1.382	9.328
133.210	102.280	6.976	6.749	9.318	1.527	10.844
71.850	182.890	5.026	5.242	10.407	1.707	12.114
76.490	173.440	4.467	5.215	10.254	1.677	11.931
41.490	94.500	9.665	6.880	8.876	1.503	10.378
167.240	71.710	4.903	5.336	11.125	1.756	12.881
119.210	92.611	8.092	6.423	10.933	1.736	12.670
108.810	163.430	5.623	5.900	13.461	2.017	15.478
169.670	58.920	4.151	5.728	11.181	1.766	12.947
189.820	130.080	6.207	5.769	13.258	1.984	15.242
132.550	63.370	5.072	5.868	13.242	1.982	15.224
220.260	93.390	5.870	5.640	13.689	2.046	15.735
0.000	135.640	3.717	5.618	15.535	2.319	17.853
97.860	141.200	5.696	5.765	13.040	1.956	14.996
143.470	152.310	5.495	5.644	13.207	1.976	15.183
72.370	44.470	6.006	5.970	14.137	2.109	16.246
248.490	57.810	2.344	5.200	15.437	2.306	17.743
44.410	98.950	6.116	5.789	13.421	2.007	15.428

The kriging results are identical to those of Carr, Myers, and Glass (1985) whereas the cokriging results show some small discrepancies presumably the result of different equation solvers used in the two programs. Matlab uses a Gauss elimination method whereas Carr, Myers, and Glass (1985) used an iterative algorithm which converges to the exact solution after a sufficient number of iterations.

Example 2: block cokriging in 3-D, with three variables and a missing value

Let the data matrix be:

$$x = \begin{matrix} -3 & 6 & 1 & 5 & 0 & 41 \\ -8 & -5 & 0 & 52 & 38 & \text{NaN*} \\ 3 & -3 & 3 & 67 & 6 & 58 \end{matrix}$$

*NaN is recognized by Matlab as missing value.

The first three colums give the coordinates x, y, and z, the last three columns give the values of the variables; the third variable was not observed at the second location ('NaN' stands for 'not a number').

—The two blocks to estimate are located at:

$$x0 = \begin{matrix} 0 & 0 & 0 \\ 10 & 20 & 33 \end{matrix}$$

—The block size is:

block = 5 10 5

—They are discretized by:

nd = 3 3 2 points in direction x, y and z.

—The coregionalization model involves two basic structures:

model =	1	1	1	1	0	0	0	nugget
	4	50	30	10	0	0	30	spherical

The first structure describe in 'model' is a nugget effect, the second is a spherical structure with anisotropic ranges of 50, 30, and 10 along rotated x, y, and z coordinates. A counterclockwise rotation of 30° is performed around coordinate z (looking in the negative direction).

c =	20	10	5	nugget
	10	25	3	
	5	3	12	
	50	−20	15	spherical
	−20	25	−7	
	15	−7	15	

- —Ordinary cokriging is performed, itype = 3.
- —'avg' will not be used, but must be defined: avg = [0 0 0] or avg = [ ].
- —Cross-validation switch is set to 0: ival = 0.
- —All samples will be used nk = 3 and rad = 100, both blocks will be kriged simultaneously, ntok = 2.
- —Cokriging estimates, cokriging variances, and lambda weights are requested as output.

```
[x0s,s,sv,id,b] = cokri(x,x0,model,c,itype,avg,block,nd,ival,nk,rad,ntok)
```

results in:

```
                 Coordinates             Cokriging estimates
           x        y        z        v1       v2       v3
x0s =    0.000    0.000    0.000   39.624   15.926   50.278
        10.000   20.000   33.000   42.143   14.142   51.072

                 Coordinates             Cokriging variances
           x        y        z        v1       v2       v3
s =      0.000    0.000    0.000   15.977   13.021    9.947
        10.000   20.000   33.000   74.675   42.343   26.879

sv =    35.8451  17.9225  10.7535
```

These are the block variances for the three variables.

b = (Lambda weights)

```
            1st block to estimate           2nd block to estimate
id*       v1        v2        v3          v1        v2        v3
11      0.337    -0.000    -0.053       0.329     0.001    -0.047
12     -0.001     0.336     0.002       0.002     0.330     0.002
13      0.001    -0.002     0.528      -0.001     0.001     0.500
31      0.281     0.020    -0.054       0.355    -0.008    -0.040
32      0.024     0.301     0.011      -0.010     0.347    -0.001
33     -0.001     0.002     0.472       0.001    -0.001     0.500
21      0.382    -0.019     0.107       0.316     0.007     0.087
22     -0.023     0.362    -0.012       0.008     0.323    -0.001
       -9.771    -2.076    -2.593     -38.830     9.526   -11.309
       -2.076    -9.872    -0.570       9.526   -24.421     3.493
       -2.617    -0.537    -7.701     -11.309     3.493   -16.126
```

* 'id' is a *nk* × 2 vector of lambda weights identifiers. The first number design the sample and the second number the variable. Note that b_{23} is missing because the value for the third variable was not observed at sample location 2. The observations are displayed in increasing order of distance from the centroid of the points or blocks to estimate. The last rows of b (without identifiers in 'id') correspond to the Lagrange multipliers.

Example 3: factorial kriging (one observed variable) in 2-D

Suppose that $Z(x) = Y_1(x) + Y_2(x)$, where $Y_1(x)$ is a purely random component (nugget effect) and $Y_2(x)$ is a spherical component with range 35, orthogonal to Y_1. The objective is to decompose the observed variable in these two components. Let the data matrix be:

x =	−3	6	5	NaN	NaN
	−8	−5	52	NaN	NaN
	3	−3	67	NaN	NaN

The first two columns of x give the coordinates and the third one the observed $Z(x)$ value. Each column of NaN correspond to a different unobserved component $Y(x)$.

The following points are to estimate:

x0 =	0	0
	−3	6.001
	−3	6
	−8	5
	3	−3

The model is specified as:

model =	1	1	nugget
	4	35	spherical

	Z	Y_1	Y_2	
c =	20	20	0	nugget
	20	20	0	
	0	0	0	
	50	0	50	spherical
	0	0	50	
	50	0	50	

The underlying Y variables corresponding to each basic structure (nugget and spherical) are orthogonal. In the nugget structure, the component Y_2 does not occur, the variance of Y_1 is 20, which also is the sill of Z for this structure and the covariance between Z and Y_1. In the spherical structure with range 35, Y_1 does not occur and the variance of Y_2 is 50, which also is the sill of Z for this structure and the covariance between Z and Y_2. The sills of 20 for the nugget and 50 for the spherical component would be obtained by modeling an experimental variogram.

```
block =   1     1
nd =      1     1      : point cokriging.
itype =   1            : simple cokriging.
avg =  41.3   41.3     : the sample mean is
                         used here as a global
                         mean and is incorpor-
                         ated, too, in the long
                         range structure.
```

Calling COKRI with:

```
[x0s,s] = cokri(x,x0,model,c,itype,avg,block,nd,0,5,100,1)
```

resulting in

```
                                             Underlying  Variables
x0s =     x          y           z             Y1          Y2
        0.000      0.000      43.022         0.000      43.022
       -3.000      6.001      21.778         0.000      21.778
       -3.000      6.000       5.000       -16.778      21.778
       -8.000     -5.000      52.000         4.367      47.633
       -8.000     -3.000      67.000        12.263      54.737

            Coordinate       Variable      Cokriging variances
  s =     x          y           Z             Y1          Y2
        0.000      0.000      37.812        20.000      17.812
       -3.000      6.001      32.809        20.000      12.809
       -3.000      6.000       0.000        12.805      12.805
       -8.000     -5.000       0.000        12.871      12.871
        3.000     -3.000       0.000        12.651      12.651
```

It should be noted that the sum of the Y estimates gives the Z estimate, and the sum of the kriging variances of the Y components gives the Z kriging variance, except at observed points. The estimate for the Y component corresponding to the nugget effect is, as it should be, zero everywhere but at the sample points, and the kriging variance for this component is equal to its sill at unsampled points.

This factorial kriging could be extended easily to situations where more than one $Z(x)$ variables is observed (factorial cokriging).

CONCLUSION

A program to perform different types of kriging and cokriging on points or blocks has been presented. This program works in the user-friendly Matlab environment. It can operate in any dimension, with any number of variables and any number of basic structures. It can perform simple cokriging, cokriging with one or p nonbias conditions and universal cokriging of order 1 or 2. Cross-validation omitting one variable at a time or all variables simultaneously is available. Geometric anisotropies can be included along any direction separately on each basic structure, thus allowing great modeling flexibility. Factorial kriging or cokriging also can be performed. The inclusion of any new basic model in the program does not necessitate any modification other than adding a single line describing the model equation in terms of h in the COKRI2 function.

Acknowledgments—The author acknowledges helpful suggestions from two anonymous referees.

REFERENCES

Aboufirassi, M., and Marino, M., 1984, Cokriging of aquifer transmissivities from field measurements of transmissivity and specific capacity: Jour. Math. Geology, v. 16, no. 1, p. 19–35.

Ahmed, S., de Marsily, G., and Talbot, A., 1988, Combined use of hydraulic and electrical properties of an aquifer in a geostatistical estimation of transmissivity: Groundwater, v. 26, no. 1, p. 78–86.

Ahmed, S., and de Marsily, G., 1989, Co-kriged estimates of transmissivities using jointly water level data, *in* Armstrong, M., ed., Geostatistics, v. 2, p. 615–628.

Ahmed, S., and de Marsily, G., 1987, Comparison of geostatistical methods for estimating transmissivity data on transmissivity and specific capacity: Water Resour. Res., v. 23, no. 9, p. 1717–1737.

Alvarado, F. L., Canizares, C. A., Keyhani, A., and B. Coates, 1991, Instructional use of declarative languages for the study of machine transients: IEEE Transactions on Power Systems, v. 6, no. 2, p. 407–413.

Barraud, A., Laporte, P., and Gentil, S., 1986, A versatile interactive system for simulation, identification and control design, *in* Proc. 3rd IFAC/IFIP Symposium, Computer Aided Design in Control and Engineering Systems: Advanced Tools for Modern Technology, p. 139–144.

de Boor, G., 1990, An empty exercise (empty matrices): SIGNUM newsletter, v. 25, no. 4, p. 2–6.

Bourgault, G., and Marcotte, D., 1991, The multivariable variogram and its application to the linear coregionalization model: Jour. Math. Geology v. 23, no. 10, p. 899–928.

Boyle, J. M., Ford, M. P., and Maciejowski, J. M., 1988, Multivariable toolbox for use with Matlab, *in* Proc. American Control Conference, American Automatic Control Council Publ., p. 707–712.

Carr, J. R., Glass, C. E., Yang, H., and Myers, D. E., 1987, Application of spatial statistics to analyzing multiple remote sensing data sets, *in* Johnson, A. I., and Peterson, C. B., eds., Geotechnical applications of remote sensing and remote data transmission: ASTM STP 967, p. 138–150.

Carr, J. R., Myers, D. E., and Glass, C. E., 1985, Cokriging —a computer program: Computers & Geosciences, v. 11, no. 2, p. 111–127.

Chiles, J. P., and Guillen, A., 1984, Variogrammes et krigeages pour la gravimétrie et le magnétisme: Sciences de la Terre, v. 20, p. 455–468.

Chiles, J. P., and Gable, R., 1984, Three-dimensional modelling of a geothermal field, *in* Verly, G., and others, eds., Geostatistics for natural resources characterization: NATO-ASI, v. 122, no. 2, p. 587–598.

Christakos, G., 1985, On the problem of permissible covariance and variogram models: Water Resour. Res., v. 20, no. 2, p. 251–265.

Creutin, J. D., Andrieu, H., and Delrieu, G., 1989, Une simplification du cokrigeage appliquée à l'étalonnage d'images de télédétection en hydrométéorogies, *in* Armstrong, M., ed., Geostatistics, v. 2, p. 761–772.

Daly, C., Jeulin, D., and Lajaunie, C., 1989, Application of multivariate kriging to the processing of noisy images, *in* Armstrong, M., ed., Geostatistics, v. 2, p. 749–760.

DeMoyer, R., and Mitchell, E. E., Jr., 1990, Applications of Matlab in undergraduate automatic control system courses: CoED, v. 10, no. 4, p. 2–4.

Dong, A., 1989, Kriging variables that satisfy the partial differential equation $\Delta Z = Y$, *in* Armstrong, M., ed., Geostatistics, v. 1, p. 237–248.

Dowd, P. A., 1989, Generalised cross-covariances, *in* Armstrong, M., ed., Geostatistics, v. 1, p. 177–188.

François-Bongarçon, D., 1981, Les co-régionalisation. Le co-krigeage: Note C-86, Centre de Géostatistique, Paris School of Mines, Fontainbleau, 26 p.

Galli, A., Gerdil-Neuillet, F., and Dadou, C., 1984, Factorial kriging analysis a substitute to spectral analysis of magnetic data, *in* Verly G., and others, eds., Geostatistics for natural resources characterization: NATO-ASI, v. 122, no. 1, p. 543–558.

Haupt, R. L., 1990, An antenna array tutorial using Matlab: CoED, v. 10, no. 4, p. 59–61.

Hoeksama, R. J., and Kitanidis, P. K., 1985, Comparison of Gaussian conditional mean and kriging estimation in the geostatistical solution of the inverse problem: Water Resour. Res., v. 21, no. 6, p. 825–836.

Hoeksama, R. J., Clapp, R. B., Thomas, L., Hunley, H. E., Farrow, N. D., and Dearstone, K. C., 1989, Cokriging model for estimation of water table elevation: Water Resour. Res., v. 25, no. 3, p. 429–438.

Isaaks, E. H., and Srivastava, R. M., 1989, An introduction to applied geostatistics: Oxford Univ. Press, Oxford, 561 p.

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

Leenaers, H., Okx, J. P., and Burrough, P. A., 1989, Co-kriging: an accurate and inexpensive means of mapping floodplain soil pollution by using elevation data, *in* Armstrong, M., ed., Geostatistics, v. 1, p. 371–382.

Matheron, G., 1982, Pour une analyse krigeante des données régionalisées: Note N-732, Centre de Géostatistique, Paris School of Mines, Fontainebleau, 22 p.

Matheron, G., 1973, The intrinsic random functions and their applications: Advanced in Applied Probability, v. 5, p. 439–468.

Matheron, G., 1970, La thèorie des variables régionalisées et ses applications: Fascicule 5, Centre de Géostatistique, Paris School of Mines, Fontainebleau, 212 p.

McBratney, A. B., and Webster, R., 1983, Optimal interpolation and isarithmic mapping of flood deposits: Environmental Geology and Water Sciences, v. 11, no. 1, p. 95–106.

Moler, C., 1988, A mathematical visualization laboratory, *in* Digest of papers—Compcon Spring 88: Intellectual Leverage, Thirty-Third IEEE Computer Society Intern. Conf. p. 480–481.

Muge, F. H., and Cabecadas, G., 1989, A geostatistical approach to eutrophication modelling, *in* Armstrong, M., ed., Geostatistics, v. 1, p. 445–458.

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

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

Myers, D. E., 1984, Cokriging: new developments, *in* Verly, G., and others, eds, Geostatistics for natural resources characterization: NATO-ASI, v. 122, no. 1, Dordrecht, p. 295–305.

Myers, D. E., 1988, Multivariable geostatistical analysis of environmental monitoring: Sciences de la Terre, v. 27, p. 411–427.

O'Haver, T. C., 1989, Teaching and learning chemometrics with Matlab: Chemometrics and Intelligent Laboratory Systems, v. 6, no. 2, p. 95–103.

Oman, P., 1990, Software test lab: Interactive mathematics on the Macintosh: IEEE software, IEEE Computer Society Publ., p. 93–98.

Ribeiro, L. T., and Muge, F. H., 1989, A geostatistical approach to the modelling of a piezometric field, *in* Armstrong, M., ed., Geostatistics, v. 2, p. 651–660.

Rouhani, S., and Myers, D. E., 1990, Problems in space–time kriging of hydrogeological data: Jour. Math. Geology, v. 22, no. 5, p. 611–623.

Sabourin, R., 1976, Application of two methods for the interpretation of the underlying variogram, *in* Guarascio, M., eds, Advanced geostatistics in the mining industry: NATO-ASI, v. 24 p. 101–112.

Sandjivy, L., 1984, The factorial kriging analysis of regionalized data. Its application to geochemical prospecting, *in* Verly, G., and others, eds, Geostatistics for natural resources characterization: NATO-ASI, v. 122, no. 1, p. 559–572.

Smyth, J. D., and Istok, J. D., 1989, Multivariate geostatistical analysis of groundwater contamination by pesticide and nitrate: a cast history, *in* Armstrong, M., ed., Geostatistics, v. 2, p. 713–724.

Tetewsky, A., 1987, Array-processing languages now suit personal-computer users: EDN, v. 32, no. 20, p. 167–172.

Vauclin, M., Vieira, S. M., Vachaud, G., and Nielsen, D. R., 1983, The use of cokriging with limited field soil observations: Soil Sci. Soc. America Jour., v. 47, no. 2, p. 175–184.

Wackernagel, H., Petitgas, P., and Touffait, Y., 1989, Overview of methods for co-regionalisation analysis, *in* Armstrong, M., ed., Geostatistics, v. 1, p. 409–420.

Yates, S. R., and Warrick, A. W., 1987, Estimating soil water content using cokriging: Soil Sci. Soc. America Jour., v. 51, no. 1, p. 23–30.

Zygmont, A., 1986, Using Matlab for control system analysis and design: CoED (Computers in Education Division of ASEE), v. 6, no. 1, p. 16–23.

Zygmont, A., 1984, Using Matlab to construct systems algorithms, *in* Proc. 1984 Frontiers in Education Conference, p. 446–451.

APPENDIX 1

Program COKRI

```
function  [x0s,s,sv,id,b]=cokri(x,x0,model,c,itype,avg,block,nd,ival,nk,rad,ntok)
%
% COKRI performs point or block cokriging in D dimensions (any integer)
%       of P variables (any integer) with a combination of R basic models
%       (any integer).
%
% Syntax:
% [x0s,s,sv,id,l]=cokri(x,x0,model,c,itype,avg,block,nd,ival,nk,rad,ntok)
%
% Input description:
%   x: The n x (p+d) data matrix. This data matrix can be imported from an
%      existing ascii file. Missing values are coded 'nan' (not-a-number).
%   x0: The m x d matrix of coordinates of points to estimate.
%   model: Each row of this matrix describes a different elementary structure.
%          The first column is a code for the model type, the d following
%          columns give the ranges along the different coordinates and the
%          subsequent columns give rotation angles (a maximum of three).
%          The codes for the current models are:
%             1: nugget effect
%             2: exponential model
%             3: gaussian model
%             4: spherical model
%             5: linear model
%          Note: a linear model is specified by arbitrary ranges and a sill
%                such that sill/range gives the desired slope in the direction
%                condidered.
%   c: The (rp x p) coefficient matrix of the coregionalization model.
%      Position (i,j) in each submatrix of size p x p give the sill of the
%      elementary component for each cross-variogram (variogram) between
%      variable i and variable j.
%   itype: Code to indicate which type of cokriging is to be performed:
%             1:  simple cokriging
%             2:  ordinary cokriging with one nonbias condition
%                 (Isaaks and Srivastava).
%             3:  ordinary cokriging with p nonbias condition.
%             4:  universal cokriging with drift of order 1.
%             5:  universal cokriging with drift of order 2.
%             99: cokriging is not performed, only sv is computed.
%   block: Vector (1 x d), giving the size of the block to estimate;
%          any values when point cokriging is required.
%   nd: Vector (1 x d), giving the discretization grid for block cokriging;
%       put every element equal to 1 for point cokriging.
%   ival: Code for cross-validation.
%             0:  no cross-validation
%             1:  cross-validation is performed  by removing one variable at a
%                 time at a given location.
%             2:  cross-validation is performed by removing all variables at a
%                 given location.
%   nk: Number of nearest neighbors in x matrix to use in the cokriging
%       (this includes locations with missing values even if all variables
%       are missing).
%   rad: Search radius for neighbors.
%   ntok: Points in x0 will be kriged by groups of ntok grid points.
%         When ntok>1, the search will find the nk nearest samples within
%         distance rad from the current ntok grid points centroid.
%------------------------------------------------------------------------------
% Output description:
%
%   For the usual application, only x0s and s are required and the other
%   output matrices may be omitted.
%
%   x0s: m x (d+p) matrix of the m points (blocks) to estimate by the
%        d coordinates and p cokriged estimates.
%   s: m x (d+p) matrix of the m points (blocks) to estimate by the
%      d coordinates and the p cokriging variances.
%   sv: 1 x p vector of variances of points (blocks) in the universe.
%   id: (nk x p) x 2 matrix giving the identifiers of the lambda weights for
%       the last cokriging system solved.
%   b: ((nk x p) + nc) x (ntok x p) matrix with lambda weights and
%      Lagrange multipliers of the last cokriging system solved.

%
casesen off;

% definition of some constants.

[m,d]=size(x0);

% check for cross-validation

```

```
if ival>=1,
   ntok=1;
   x0=x(:,1:d);
   nd=ones(1,d);
   [m,d]=size(x0);
end
[rp,p]=size(c);
[n,t]=size(x);
nk=min(nk,n);
ntok=min(ntok,m);
idp=[1:p]';
ng=prod(nd);

% echo of data and input options and parameters

echo_input=['dimension ',num2str(d)]
echo_input=['number of variables ',num2str(p)]
['Strike any key when ready'],pause
x
['Strike any key when ready'],pause
x0
['Strike any key when ready'],pause
model
c
['Strike any key when ready'],pause
itype,ival
avg
block,nd
['Strike any key when ready'],pause
nk,rad,ntok

% compute point (ng=1) or block (ng>1) variance

for i=1:d,
   nl=prod(nd(1:i-1));
   nr=prod(nd(i+1:d));
   t=[.5*(1/nd(i)-1):1/nd(i):.5*(1-1/nd(i))]';
   t2=[t2,kron(ones(nl,1),kron(t,ones(nr,1)))];
end
grid=t2.*(ones(ng,1)*block);
t=[grid,zeros(ng,p)];

% for block cokriging, a double grid is created by shifting slightly the
% original grid to avoid the zero distance effect (Journel and Huijbregts, p.96)

if ng>1,
   grid=grid+ones(ng,1)*block/(ng*1e6);
end
[x0s,s,id,l,k0]=cokri2(t,grid,[],model,c,sv,99,avg,ng);

% sv contain the variance of points or blocks in the universe

for i=1:p,
   sv=[sv,means(means(k0(i:p:ng*p,i:p:ng*p))')];
end

% start cokriging

for i=1:ntok:m,
   nnx=min(m-i+1,ntok);
   ['kriging points #',num2str(i),' to ', num2str(i+nnx-1)]

   % sort x samples in increasing distance relatively to centroid of 'ntok'
   % points to krige

   centx0=ones(n,1)*means(x0(i:i+nnx-1,:));
   tx=[x(:,1:d)-centx0].*[x(:,1:d)-centx0]*ones(d,1);
   [tx,j]=sort(tx);

   % keep samples inside search radius; create an identifier of each sample
   % and variable (id)

   t=[];
   id=[];
   ii=1;
   tx=[tx;nan];
   while ii<=nk & tx(ii)<rad*rad,
      t=[t;x(j(ii),:)];
      id=[id;[ones(p,1)*j(ii),idp]];
      ii=ii+1;
   end
   t2=x0(i:i+nnx-1,:);

   %  if block cokriging discretize the block

   t2=kron(t2,ones(ng,1))-kron(ones(nnx,1),grid);
```

```
      % check for cross-validation

      if ival>=1,
         est=zeros(1,p);
         sest=zeros(1,p);

         % each variable is cokriged in its turn

         if ival==1,
            np=1;
         else
            np=p;
         end
         for ip=1:np:p,

            % because of the sort, the closest sample is the sample to
            % cross-validate and its value is in row 1 of t; a temporary vector
            % keeps the original values before performing cokriging.

            vtemp=t(1,d+ip:d+ip+np-1);
            t(1,d+ip:d+ip+np-1)=ones(1,np)*nan;
            [x0ss,ss]=cokri2(t,t2,id,model,c,sv,itype,avg,ng);
            est(ip:ip+np-1)=x0ss(ip:ip+np-1);
            sest(ip:ip+np-1)=ss(ip:ip+np-1);
            t(1,d+ip:d+ip+np-1)=vtemp;
         end
         x0s=[x0s;[t2,est]]
         s=[s;[t2,sest]];
      else
         [x0ss,ss,id,l]=cokri2(t,t2,id,model,c,sv,itype,avg,ng);
         x0s=[x0s;[x0(i:i+nnx-1,:),x0ss]];
         s=[s;[x0(i:i+nnx-1,:),ss]];
      end
   end

```

```
function [x0s,s,id,b,k0]=cokri2(x,x0,id,model,c,sv,itype,avg,ng)
%
% COKRI2 : This function is called from COKRI. The description for input and
%          output is given in COKRI. The only new variables are 'k0' which is
%          the right member matrix of the cokriging system and 'ng' which is
%          the total number of points for block discretization.

% here we define the equations for the various covariograms. Any new model
% can be added here.


Gam=['h==0                                              '; %nugget
    'exp(-h)                                            '; %exponential
    'exp(-(h).^2)                                       '; %gaussian
    '1-(1.5*min(h,1)/1-.5*(min(h,1)/1).^3)              '; %spherical
    '1-h                                                '];%linear

% definition of some constants

[n,t]=size(x);
[rp,p]=size(c);
r=rp/p;
[m,d]=size(x0);
cx=[x(:,1:d);x0];

% calculation of left covariance matrix K and right covariance matrix K0

k=zeros(n*p,(n+m)*p);
for i=1:r,

   % calculation of matrix of reduced rotated distances H

   [t]=trans(cx,model,i);
   t=t*t';
   h=sqrt(-2*t+diag(t)*ones(1,n+m)+ones(n+m,1)*diag(t)');
   h=h(1:n,:);
   ji=(i-1)*p+1; js=i*p ;

   % evaluation of the current basic structure

   g=eval(Gam(model(i,1),:));
   k=k+kron(g,c(ji:js,:));
end
k0=k(:,n*p+1:(n+m)*p); k=k(:,1:n*p);

% constraints are added according to cokriging type

if itype==99,
s and parameters

r(d)]
s ',num2str(p)]
ause

ause

ause

ause

ause

 (ng>1) variance

'(1-1/nd(i))]';
:,ones(nr,1)))];

 grid is created by shifting slightly the
 distance effect (Journel and Huijbregts, p.96).

.g*1e6);

,model,c,sv,99,avg,ng);

nts or blocks in the universe

p,i:p:ng*p))')];

),' to ', num2str(i+nnx-1)]

} distance relatively to centroid of 'ntok'

-nnx-1,:));
l)-centx0]*ones(d,1);

'adius; create an identifier of each sample

 ]];

 ze the block

 es(nnx,1),grid);
```

```
 (Isaaks and Srivastava, 1990, p.410)

);ones(1,m*p)]; nc=1;

ol, 1982)

t constraints

)]];

lrift constraints

;

ca,nc+nca)]];

ary) for block cokriging

ith one non bias condition, the means

l K0 corresponding to missing values
```

```

   % solution of the cokriging system by gauss elimination

   l=k\k0;

   % calculation of cokriging estimates

   t2=l(1:nz,:)'*z(iz);
   t=zeros(p,m);
   t(:)=t2;

   % if simple or cokriging with one constraint, means are added back

   if itype<3,
      t=t'+ones(m,1)*avg;
   else
      t=t';
   end
   x0s=t;

   % calculation of cokriging variances

   s=kron(ones(m,1),sv);
   t=zeros(p,m);
   t(:)=diag(l'*k0);
   s=s-t';
end

```

```
function [cx]=trans(cx,model,im);
%
% TRANS is called from COKRI2. It takes as input original coordinates and
%       return the rotated and reduced coordinates following specifications
%       described in model(im,:)
%

% some constants are defined

[n,d]=size(cx);
[m,p]=size(model);

% check for 1-D or isotropic model

if p-1>d,

   % perform rotation counterclockwise

   if d==2,
      ang=model(im,4); cang=cos(ang/180*pi); sang=sin(ang/180*pi);
      rot=[cang,-sang;sang,cang];
   else

      % rotation matrix in 3-D (the first three among d coordinates) is
      % computed around x, y and z in that order

      rot=eye(3);
      for i=1:3,
         ang=model(im,4+i); cang=cos(ang/180*pi); sang=sin(ang/180*pi);
         rot2=[cang,-sang;sang,cang];
         axe=ones(3,1); axe(i)=0;
         rot(axe,axe)=rot(axe,axe)*rot2;
      end
   end

   % rotation is performed around x, y and z in that order, the other coordinates
are left unchanged.

   dm=min(3,d);
   cx(:,1:dm)=cx(:,1:dm)*rot;
   t=diag([model(im,2:1+dm),ones(d-dm,1)]);
else
   t=eye(d)*model(im,2);
end

% perform contractions or dilatations (reduced h)
  t=max(t,1e-10);
  cx=cx/t;
end

```

```
function y = means(x)
%
% MEANS  Average or mean value. For column vectors MEANS(x) returns the mean
%        value. For matrices or row vector, MEANS(x) is a row vector containing
%        the mean value of each column. The only difference with MATLAB
%        function MEAN is for a row vector, where MEANS returns the row vector
%        instead of the mean value of the elements of the row.
%
[m,n] = size(x);
if m > 1,
   y = sum(x) / m;
else
   y = x;
end
```

APPENDIX 2

```
function checkmod(model,c,d,rad)
%
% CHECKMOD: This function generates 'ntot' points within a D-sphere of radius
%           'rad' (in reduced rotated distances) and evaluates all
%           cross-variograms and variograms as described in 'model' and 'c'.
%           If the necessary condition |γij(h)| ≤ sqrt(γii(h) γjj(h)) is
%           violated for any simulated point, then the message
%           'Warning the model is not admissible' appears.
%
% Input parameters: model: description of models (see COKRI)
%                       c: matrices of sills (see COKRI)
%                       d: dimension (1-D, 2-D,...)

% initialisation

[t,p]=size(c);
[r,t]=size(model);
ntot=3000/max(d,p*p)

% equations for the various variograms, models should be placed in the same
% order as in COKRI2 note that here the variograms are computed instead
% of the covariograms as in COKRI2.

Gam=['h~=0                                               '; %nugget
     '1-exp(-h)                                          '; %exponential
     '1-exp(-(h).^2)                                     '; %gaussian
     '(1.5*min(h,1)/1-.5*(min(h,1)/1).^3)                '; %spherical
     'h                                                  '];%linear

% generate 'ntot' random points inside a D-sphere of radius 'rad'

t=(rand(ntot,d)-0.5*ones(ntot,d))*rad*2;

% evaluate variograms for each simulated point

k=zeros(ntot*p,p);
for i=1:r,
   t=trans(t,model,i);
   h=sqrt(sum([t.*t]')');
   ji=(i-1)*p+1; js=i*p ;
   g=eval(Gam(model(i,1),:));
   k=k+kron(g,c(ji:js,:));
end

% check that |γij| < sqrt(γii * γjj)

for i=1:ntot,
  ii=(i-1)*p+1;
  is=i*p;
  t=sqrt(diag(k(ii:is,:)));
  k(ii:is,:)=abs(k(ii:is,:))-t*t';
end
k=[k>1e-10];
if sum(sum(k)>0), 'Warning model is not admissible'
end
```