

A DISJUNCTIVE KRIGING PROGRAM FOR TWO DIMENSIONS

S.R. YATES

R. S. Kerr Environmental Research Laboratory, P.O. Box 1198, Ada, OK 74820, U.S.A.

A.W. WARRICK

Department of Soils, Water and Engineering, The University of Arizona, Tucson, AZ 85721, U.S.A.

and

D.E. MYERS

Department of Mathematics, The University of Arizona, Tucson, AZ 85721, U.S.A.

(Received 27 July 1985; revised 20 January 1986)

Abstract—This paper describes the disjunctive kriging (DK) method for two dimensional spatially variable properties. A brief mathematical description is given which includes all pertinent equations to obtain an estimated value, disjunctive kriging variance, and conditional probability that the value of a property at a location is above a known cutoff level. This is followed by a description of the steps which are necessary to implement DK and an example illustrating the method. In order to use DK, a series of complex calculations must be carried out. To facilitate this two FORTRAN programs were developed. The programs, example input and output files as well as information necessary to use the programs is included.

Key Words: Conditional probability, Disjunctive kriging, Joint density, Kriging, Nonlinear estimate, Regionalized random variable, Spatial variability.

INTRODUCTION

Recently, much attention has been given to linear kriging methods in the analysis of spatially dependent phenomena. These include simple, ordinary, and universal kriging either on punctual or block support. Many examples occur in the literature. A few of these include Burgess and Webster (1980a) who used punctual kriging in the analysis of the spatial distribution of the sodium content, cover loam, and stone content. Burgess and Webster (1980b) and Webster and Burgess (1980) also used block and universal kriging in subsequent analyses. Warrick, Myers, and Nielsen (1985) give an example of kriging the electrical conductivity for a southwestern Arizona soil (soil type: Typic Haplargid). Vieria, Nielsen, and Biggar (1981) investigated the correlation between kriged and actual values based on different number of sample values used in the estimation process in an attempt to determine the minimum number of samples necessary to obtain a given level of information. Vauclin and others (1983) used ordinary kriging and cokriging to estimate the spatial distribution of the available water content and sand content of a Tunisian field. They determined that cokriging provided improved estimates of the available water content because of additional information added to the problem by the sand content. Journel and Huijbregts (1978) give many examples of kriging which have mining applications.

These studies represent but a small sample of the work which uses linear kriging estimators in the

analysis of the spatial variability of a physical property. Generally, the linear kriging estimator is not the best possible estimator. The minimum variance unbiased estimator of a random variable Y in terms of random variables X_1, X_2, \dots, X_n is the conditional expectation of Y given X_1, X_2, \dots, X_n . For the most general situation, computing this conditional expectation requires knowledge of the joint density of Y, X_1, X_2, \dots, X_n although this information is difficult to obtain in practice. This nonlinear estimator has the form

$$Y^* = g(X_1, X_2, \dots, X_n) \quad (1)$$

It also is possible to relax the requirement that the joint density of $(n+1)$ variables be known and define another nonlinear estimator

$$Y_{DK}^* = \sum_{i=1}^n f_i(X_i) \quad (2)$$

where each f_i is a nonlinear function of one X variable only. This is the disjunctive kriging (DK) estimator and requires that only the bivariate densities be known.

The linear kriging estimator is a special form of Equation (2) where each f_i is a linear function and therefore only the constants need to be determined

$$Y_k^* = \sum_{i=1}^n \lambda_i X_i. \quad (3)$$

In terms of estimating the value of a random variable at an unsampled location, Equation (2) generally is a

better estimator than Equation (3) in the sense of reduced kriging variance. Moreover, if an estimate of the conditional probability distribution is required a nonlinear estimator is essential.

The purpose of this paper is to present the disjunctive kriging programs and to describe the basic algorithm used to obtain a nonlinear estimate of a regionalized random variable (at an unsampled location), the estimation variance and the conditional probability that the estimate is above a given cutoff level. For brevity, the DK equations will be given without derivation. They are given in full in Yates, Warrick, and Myers (1985a).

THEORY

Consider $Z(x)$ to be an isotropic second-order stationary random function which is sampled on a point support at p locations: x_1, x_2, \dots, x_p , with x a vector in two-dimensional space. For a stationary system, $E[Z(x)]$ is constant, the variance is defined and independent of location, and the spatial covariance function, $C(x_i - x_j)$, is defined and depends solely on the separation distance, $x_i - x_j$. Also, assume that a transform function $\phi[Y(x)]$ exists which will transform $Z(x)$ with an arbitrary frequency distribution into $Y(x)$ which has a standard normal gaussian distribution. The transformation operates such that the uni- and bivariate distributions are normal and jointly normal, respectively.

Disjunctive kriging

Summarizing the results of Yates, Warrick, and Myers (1985a), the disjunctive kriging estimator is given as

$$Z_{DK}^*(x_o) = \sum_{i=1}^n f_i[Y(x_i)] = \sum_{k=0}^{\infty} \sum_{i=1}^n f_{ik} H_k[Y(x_i)] \quad (4)$$

where x is a vector in 2-D space which represents the coordinates of the sample location, n is the number of transformed sample values, $Y(x_i)$, used in the estimation process, f_i is a function to be determined and is expressed on the right hand side of Equation (4) as a series of Hermite polynomials of order k (i.e. $H_k[\arg]$) where the f_{ik} 's are the coefficients of the Hermite expansion.

The disjunctive kriging process is based on a standard normal random function, $Y(x_i)$, which is related to $Z(x)$ through a transform. The transform relationship, $\phi(x)$, also is expressed as a series of Hermite polynomials (Abramowitz and Stegun, 1965)

$$Z(x_i) = \phi[Y(x_i)] = \sum_{k=0}^{\infty} C_k H_k[Y(x_i)]. \quad (5)$$

The coefficients, C_k , are determined by orthogonality and numerical integration as follows

$$C_k = 1/(k! \sqrt{2\pi}) \sum_{j=1}^J \phi(v_j) w_j H_k(v_j) \exp[-v_j^2/2] \quad (6)$$

where J is the total number of abscissas and weights factors, v_j and w_j , respectively, used in the Hermite integration (Abramowitz and Stegun, 1965).

Requiring an unbiased estimator which has minimum variance of errors produces the following disjunctive kriging system of equations (see Journel and Huijbregts, 1978; Rendu, 1980 and Yates, Warrick, and Myers, 1985a for further details)

$$Z_{DK}^*(x_o) = \sum_{k=0}^K C_k H_k[Y(x_o)] \quad (7)$$

where the series has been truncated to K terms and $H_k[Y(x_o)]$ in Equation (7) is estimated by

$$H_k^*[Y(x_o)] = \sum_{i=1}^n b_{ik} H_k[Y(x_i)] \quad (8)$$

where f_{ik} from Equation (4) is written as $C_k b_{ik}$ in Equations (7) and (8). The b_{ik} 's (the weights analogous to simple kriging) are determined from

$$\sum_{i=1}^n b_{ik} (\varrho_{ij})^k = (\varrho_{oj})^k; \quad j = 1, 2, \dots, n \quad (9)$$

where ϱ_{ij} is the spatial correlation function (i.e. autocorrelation) for a separation distance, $x_i - x_j$. The correlation function also can be written as: $\varrho_{ij} = C(x_i - x_j)/C(0)$ or for second-order stationary conditions $\varrho_{ij} = 1 - \gamma(x_i - x_j)/C(0)$, where γ is the variogram. The bar over the b denotes that ϱ_{oj} can be based on either point or block support. For a block estimate, the average correlation

$$(\bar{\varrho}_{oj}) = 1/V \int \varrho(x - x_o) dx \quad (10)$$

should be used on the right hand side of Equation (9).

The disjunctive kriging variance on the estimation is

$$\sigma_{DK}^2 = \sum_{k=1}^K k! C_k^2 \left[1 - \sum_{i=1}^n b_{ik} (\varrho_{oi})^k \right] \quad (11)$$

The conditional probability

Estimating a conditional probability that the value of a random variable is above a specified cutoff value (termed the transfer function by Matheron, 1976; Journel and Huijbregts, 1978; Kim, Myers, and Kundsen, 1977) is possible because the disjunctive kriging estimator is nonlinear. The method used here consists of two steps. The first step is to determine the conditional probability, P^* , (for each cutoff value) that the point value of $Z(x)$ at the location, x_o , inside the block V is above a prescribed cutoff value z_c (or the associated transformed cutoff value, y_c). This is called the point transfer function. Next, to obtain the conditional probability that the block value is above the cutoff value, the probability function then is integrated over the entire block. The resulting conditional

probability is (Journel and Huijbregts, 1978; Yates, Warrick, and Myers, 1985a)

$$\begin{aligned} P^*[Y(x_o)] &= 1 - G(y_c) + g(y_c) \\ &\times \sum_{k=1}^K H_{k-1}(y_c) \\ &\times H_k[Y(x_o)]/k! \end{aligned} \quad (12)$$

where $H_k[Y(x_o)]$ is estimated using Equations (8) and (9). In Equation (12), $g(y_c)$ and $G(y_c)$ are the gaussian density and cumulative distribution functions, respectively.

The conditional probability density function, $Pdf^*(u)$, is determined by taking the derivative of Equation (12) with respect to y_c .

$$Pdf^*(u) = g(u) \left\{ 1 + \sum_{k=1}^K H_k(u) H_k[Y(x_o)]/k! \right\} \quad (13)$$

Two examples using approximately lognormal and normal data and the disjunctive kriging programs described in this paper are given by Yates, Warrick, and Myers (1985b).

THE PROGRAMS: DISCALC.FTN AND DISJUNC.FTN

The programs DISCALC.FTN and DISJUNC.FTN (given in Appendix A) were written in FORTRAN 77 and run on a DEC PDP 11/70. The compiled version of each program requires less than 64K memory for the source code, program constants and many arrays. Program DISJUNC.FTN also requires virtual memory for storage of a large array (20 by 220). For machines which can access more than 64K directly (i.e. do not need or have virtual memory) the "virtual" statements can be replaced with dimension statements.

The program allows up to 220 samples, 25 coefficients (C_k 's), 20 samples to be used in the estimation process (n) and up to a 20th order polynomial fit of the sample distribution [i.e. between $Z(x)$ and $Y(x)$].

In order to improve the program's computational efficiency, an (virtual) array (name HH) 20 by 220 was used to store the $H_k[Y(x_i)]$ values for $k = 1, 2, \dots, 20$ and $i = 1, 2, \dots, 220$. In this way, only one calculation of the Hermite polynomials associated with the samples was necessary. Also, the program was written such that the kriging matrix was formed only once for each estimate (for $k = 1$), was saved and then a working matrix (raised to the appropriate power) was used to determine the b_{ik} 's. Using this strategy requires only one determination of the nearest neighbors per estimate.

Solving the DK equations (program DISJUNC.FTN) requires approximately (No. of C_k 's - 1) times more computer time than ordinary kriging (with a Lagrange multiplier), holding everything else constant. This does not include the

preparatory steps (Steps 1 and 2) outlined next. The extra computer time necessary to solve the DK equations is because of the more complex calculations such as taking powers and calculating Hermite polynomials as well as having to solve the simple kriging equations K times. Although DK requires more computer time, which for a microcomputer can be inconsequential, information about the conditional probability distributions at the estimation site is available.

The following steps give a brief description of the execution of the DK programs. From these programs one can obtain an estimate of a random variable at an unsampled location, the disjunctive kriging variance and the conditional probability for up to 18 cutoff levels using the disjunctive kriging method outlined. Steps 1 and 2 are preliminary to the actual estimation process and the calculations described in Step 1 are made in the program DISCALC.FTN whereas those described in Steps 2 to 7 are made in DISJUNC.FTN.

Step 1. The first step in the DK process is to determine the coefficients, C_k , which define the transform, $\phi(Y_i)$ in Equation (5).

(a) The calculation begins by sorting the data from the smallest to largest value, while keeping track of the associated x 's, (subroutine SORT). Next, an empirical cumulative frequency distribution is generated, from which an estimate of the probability is obtained. The trial transformed value, Y_i , is obtained by inverting the probability function (Function PRBI). The estimate of the probability used in DISCALC.FTN is

$$P[Z \leq z_i] \approx (i - 0.5)/p \quad (14)$$

where i is the total number of $Z(x_i)$ less than or equal to z_i and p is the total number of samples.

(b) Once the pairs $[Z(x_i), Y(x_i)]$ have been determined, the values of $\phi(Y_i)$ in Equation (6) must be determined. The method used by DISCALC.FTN is to fit an n th order polynomial to the data pairs and determine the coefficients of the polynomial via a least-squares fit (subroutine LEAST). Function PHI evaluates the n th order polynomial for each abscissa in Equation (6). Other possibilities exist for determining this relationship, but the n th order polynomial is adequate and better than linear interpolation.

(c) After the relationship is determined, the coefficients, C_k , are calculated by Hermite integration (Abramowitz and Stegun, 1965) in subroutine CK1. From the coefficients the mean and variance of the original data can be determined (exactly as $k \rightarrow \infty$) and are calculated in the MAIN program and printed out for a comparison with the actual values for the data set.

(d) In general, a truncated series is used for the $\phi[Y(x_i)]$ relationship which represents an approximation in terms of transforming $Y(x_i)$ into $Z(x_i)$. Therefore, to make the inversion exact, new values of $Y(x_i)$ are calculated given the C_k 's just determined. The subroutine that inverts Equation (5) is HYZINV,

which uses Newton's iterative method. If the iterative method diverges, an alternate method of bisection is used. The values of $Y(x_i)$ (array YZ in program) are printed out along with other necessary information and saved in a file which is used as an input file for the second program DISJUNC.FTN.

Step 2. Once the transform relationship is defined, the next step is to calculate a table of Hermite polynomials for $k = 0, 1, \dots, K$ and for all sample values $Y(x_i)$. This $K \times p$ matrix stores the values of $H_k[Y(x_i)]$ rather than recalculating them each time a sample is used in the estimation process. The subroutine (in DISJUNC.FTN) that sets up the array is HCALC and the subroutine that looks up the value in the matrix for use in the program is H.

The following steps are required to obtain an estimate, the kriging variance and the conditional probability at an unsampled location. The sequence is repeated for each estimate. The program is written such that estimates are produced on a grid system beginning at $X = XMIN$ and $Y = YMINT$ and incrementing by DX and DY, (NX) · (NY) times.

Step 3. The first step in obtaining an estimate is to generate the DK matrix (subroutine MATRX). This is done only once per estimate for $k = 1$ by sorting through the data and selecting the "n" nearest samples. The spatial correlation between the points is calculated (subroutine VARIO) and stored in array AA.

Step 4. In order to obtain the weights, b_{ik} , in Equation (9), the AA matrix is raised to the k th power by subroutine ATOK and solved by subroutine MATINV. Because MATINV modifies the matrix during the solution, a working matrix AWRK is used by MATINV. For $k = 0$ the weights are equal to $1/n$. Therefore the matrix solving step is skipped and execution transfers to step 5.

Step 5. Various intermediate quantities are calculated in subroutine SOLN. These include $H_k[Y(x_i)]$ in Equation (8) and the right-most series in Equations (11) and (12). After returning to the MAIN program, these intermediate quantities (which are summed on the data used in the estimation or on the number of cutoff values desired) are summed on the current value of k .

Step 6. If $k < K$, then k is incremented and execution is returned to step 4.

Step 7. If $k = K$, the estimated value, the estimation variance and the conditional probability (for each cutoff level) are printed out along with the spatial coordinates.

EXAMPLE

An example using the DK programs is given here. Ninety-two bare soil-surface temperatures were recorded at random locations in a 1 ha field at the University of Arizona's Maricopa Agricultural Center.

The sample mean, variance, skew, and kurtosis for the data were: 63.89, 3.08, 0.61, and 2.64, respectively.

The data also were tested for type of distribution by the Kolmogorov-Smirnov (KS) test (Rao and others, 1979; Rohlf and Sokal, 1981). The KS test statistic calculated from the original and log-transformed data are 0.121 and 0.116, respectively. The KS critical value for 92 points at the 0.1 probability level is 0.084. Based on this test it was concluded that the data set were neither normally nor lognormally distributed.

The sample covariance function (see Journel and Huijbregts, 1978, esp. p. 194) was calculated and then validated using the jackknife procedure (Vauclin, Nielsen and Biggar, 1983). A spherical model was fitted to the sample covariance function with parameters: 0.7 °C for the nugget, 3.1 °C² for the sill and 23.0 m for the range. The autocorrelation function, which is used in Equation (9), was calculated by using $\rho(x_i - x_j) = C(x_i - x_j)/C(0)$.

The coefficients, C_k , used to define the transform given in Equation (5) for $k = 0$ to K are: 63.891, 1.709, 0.1832, -0.06818, -0.04437, -3.676 × 10⁻³ and -5.849 × 10⁻³, respectively. From these C_k 's, estimates of the mean and variance for the sample distribution can be obtained and are 63.89 °C and 3.09 °C² whereas the actual values for the data set are 63.89 °C and 3.08 °C², respectively.

In order to compare the DK method with ordinary kriging (OK) both methods were applied to the data set. From the results, a comparison based on the average kriging variance and the sum of squares deviation (SSQ) between the actual and estimated values can be made.

The kriging variance was calculated by each method at 231 locations located on a 6 × 8-m grid system superimposed over the field. The average kriging variance was determined from these values. From the OK method, the average kriging variance was 2.202 °C² and for DK was 2.098 °C². This represents a 4.7% improvement over OK.

The SSQ between the actual and estimated value was determined by making an estimate at each sample location (92 total) but not using the sample value in the estimation (i.e. jackknifing). From the two values the SSQ can be calculated. The results indicate that there is an overall improvement (for these data) of 2.6% when DK (SSQ of 189.8 °C²) is used instead of OK (SSQ of 194.9 °C²). Improvements of approx. 6% have been demonstrated for another random variable by Yates, Warrick, and Myers (1985b).

Figure 1a gives the estimated value of soil temperature at 9 locations using the DK and OK methods. The number above each location (circle) is the estimated value using DK and the number below is from OK. From this sort of information, the spatial pattern of temperatures in the field can be determined, although one would usually use a finer grid system.

Figure 1b gives the associated estimation variance where the number above and below each location (circles) are for DK and OK, respectively. Using this information, a level of confidence in the estimated

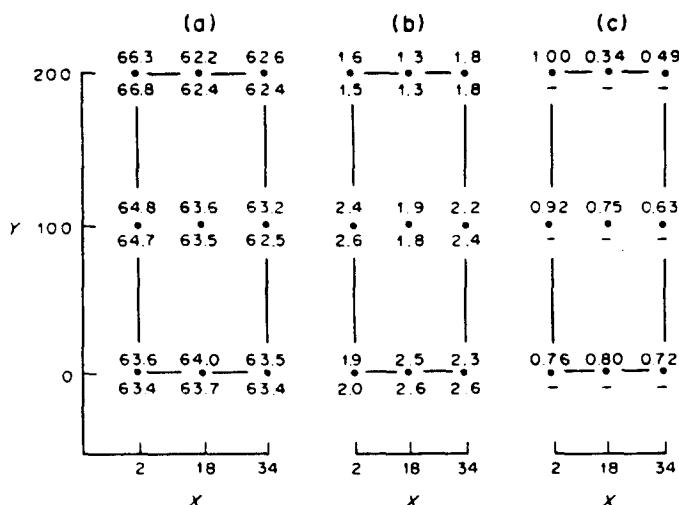


Figure 1. Results of ordinary and disjunctive kriging of soil surface temperature. Numbers above and below locations (circles) indicate estimates by disjunctive and ordinary kriging, respectively. In a, b, and c are temperature estimates, kriging variance, and conditional probability that temperature is above 62.5°C, respectively.

value can be obtained where large variance is associated with small confidence.

Because DK produces a nonlinear estimator, values of the conditional probability that the actual but unknown value of the temperature is above a specified cutoff level can be calculated. Figure 1c gives an example of this where the numbers indicate the probability that the temperature is above 62.5°C. This information is unavailable typically when using linear kriging methods.

Disjunctive kriging also allows one to estimate the conditional probability density function and the cumulative probability distribution. These are shown in Figure 2 for three of the points given in Figure 1. The solid, dashed and dotted lines are for the points: (2, 200), (34, 200), and (18, 0), respectively. In Figure 2a the probability distribution is with respect to the cutoff value whereas in Figure 2b the probability density is plotted as a function of the transformed value Y . From this figure it is possible to obtain an indication of how the samples combine together to form the estimate as well as obtaining the probability level of an occurrence given a specified cutoff value.

CONCLUSIONS

The disjunctive kriging programs, DISCALC.FTN and DISJUNC.FTN, produce a nonlinear estimate of a spatially variable property as well as the conditional probability that the value is above an arbitrary cutoff level. Generally, the DK estimator is better in the sense of reduced kriging variance compared to linear kriging estimators but is not as good as the conditional expectation (unless the random variable is multivariate normal). Disjunctive kriging has the advantage over the conditional expectation in that only the bivariate joint probability distributions need be known.

The calculations required to estimate a random

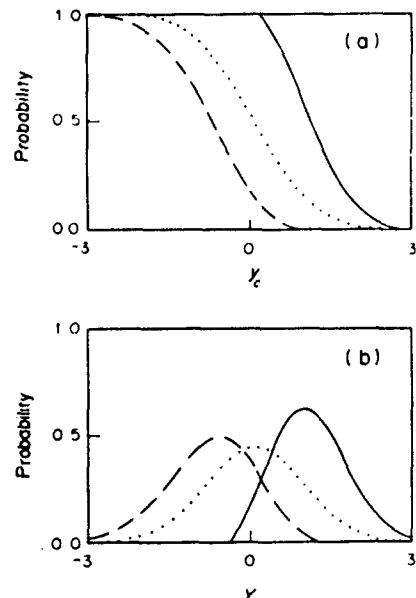


Figure 2. Conditional probability distributions. In a is cumulative probability distribution as function of cutoff value, y_c . In b is probability density with respect to transformed variable, Y .

variable are basically the same as the simple kriging method but must be performed K times per estimate. The other differences include: a transform is necessary and defined by the coefficients, C_k , that are appropriate for the data set and Hermite polynomials up to order K must be calculated for each sample used in the estimation. These steps are done at the beginning of the problem. A third difference is that the interpolation is with respect to $H_k[Y(x)]$ for DK whereas it is based on $Z(x)$ for ordinary (linear) kriging methods. This result also is used in calculation of the conditional probability. For other examples using the DK programs see Yates, Warrick, and Myers (1985b).

Acknowledgments—Although part of the research described in this article has been supported by the United States Environmental Protection Agency through contract No. 68-7176 to Computer Sciences Corporation, it has not been subjected to Agency review and therefore does not necessarily reflect the views of the Agency and no official endorsement should be inferred.

REFERENCES

- Abramowitz, M., and Stegun, A., 1965, Handbook of mathematical functions: Dover Publications Inc., New York, 1046 p.
- Burgess, T. M., and Webster, R., 1980a, Optimal interpolation and isarithmic mapping of soil properties. I. The semivariogram and punctual kriging: *Jour. Soil Sci.*, v. 31, no. 2, p. 315–331.
- Burgess, T. M., and Webster, R., 1980b, Optimal interpolation and isarithmic mapping of soil properties. II. Block kriging: *Jour. Soil Sci.*, v. 31, no. 2, p. 333–341.
- Journal, A. G., and Huijbregts, Ch. J., 1978, Mining geostatistics: Academic Press, New York, 600 p.
- Kim, Y. C., Myers, D. E., and Knudsen, H. P., 1977, Advanced geostatistics in ore reserve estimation and mine planning (practitioner's guide): Report to the U.S. Energy Research and Development Administration, Subcontract No. 76-003-E, Phase II, 154 p.
- Matheron, G., 1976, A simple substitute for conditional expectation: the disjunctive kriging: *Proceedings NATO ASI "Geostat 75"*, D. Reidel Publishing Co., Dordrecht, Netherlands, p. 221–236.
- Rao, P. V., Rao, P. S. C., Davidson, J. M., and Hammond, L. C., 1979, Use of goodness-of-fit tests for characterizing the spatial variability of soil properties: *Soil Sci. Soc. Am. Jour.* v. 43, p. 274–278.
- Rendu, J. M., 1980, Disjunctive kriging: A simplified theory: *Jour. Math. Geology*, v. 12, no. 4, p. 306–321.
- Rohlf, F. J., and Sokal, R. R., 1981, Statistical tables (2nd ed.): W. H. Freeman and Co., New York, 219 p.
- Vauclin, M., Vieira, S. R., Vauchaud, G., and Nielsen, D. R., 1983, The use of cokriging with limited field soil observations: *Soil Sci. Soc. Am. Jour.* v. 47, p. 175–184.
- Vieira, S. R., Nielsen, D. R., and Biggar, J. W., 1981, Spatial variability of field-measured infiltration rate: *Soil Sci. Soc. Am. Jour.* v. 45, p. 1040–1048.
- Warrick, A. W., Myers, D. E., and Nielsen, D. R., 1985, Geostatistical methods applied to soil science: to be in *Methods of Soil Analysis*, ASA.
- Webster, R., and Burgess, T. M., 1980, Optimal interpolation and isarithmic mapping of soil properties. III. Changing drift and universal kriging: *Jour. Soil Sci.*, v. 31, no. 3, p. 505–524.
- Yates, S. R., Warrick, A. W., and Myers, D. E., 1985a, Disjunctive kriging. I. Overview of estimation and conditional probability: *Water Resour. Res.*, v. 22, no. 5, p. 615–621.
- Yates, S. R., Warrick, A. W., and Myers, D. E., 1985b, Disjunctive kriging. II. Examples: *Water Resour. Res.*, v. 22, no. 5, p. 623–630.

APPENDIX 1

Disjunctive Kriging Programs DISCALC.FTN and DISJUNC.FTN

```

C*****DISJUNCTIVE KRIGING PROGRAMS FOR TWO-DIMENSIONAL,
C*****SPATIALLY-VARIABLE PROPERTIES*****D0001
C*****D0002
C*****D0003
C*****D0004
C*****D0005
C*****D0006
C*****D0007
C*****D0008
C*****D0009
C*****D0010
C Solves the disjunctive kriging equations and calculates an estimate D0011
C of a random variable, the associated estimation variance and the D0012
C conditional probability (for up to 18 cutoff levels). In order to use D0013
C these programs, values of a random variable of interest along with the D0014
C spatial coordinates must be obtained. Also, the correlation structure D0015
C written in terms of the variogram must be determined. D0016
C D0017
C D0018
C The programs use the following devices: D0019
C D0020
C Disk input file -- currently set as unit #1 D0021
C Disk output file -- currently set as unit #2 D0022
C Disk plot file -- currently set as unit #3 D0023
C Terminal -- currently set as unit #5 D0024
C Line printer -- currently set as unit #6 D0025
C D0026
C These values are declared at the beginning of each program. D0027
C      ε D0028
C D0029
C To reduce problem execution time and storage requirements, two programs D0030
C are used for disjunctive kriging. The first program calculates the D0031
C Hermite coefficients (i.e. CK's) and then converts the data (ZD) and D0032
C cutoff values (ZCUT) into transformed data (YZ) and transformed cutoff D0033
C values (YCUT). Approaching the solution in this manner has the advantage D0034
C that the coefficients and transformed values only need to be calculated D0035
C once for a given data set. To run the programs, a data file must be D0036

```

C created according to the "Data File Input Instructions" given below. D0037
 C The "Interactive Data Instructions For Program One" describes the run- D0038
 C time information the program requires such as instructions for choosing D0039
 C input/output devices and file names. D0040
 C D0041
 C D0042
 C The second program uses the output from the first program directly. No D0043
 C modification on the data file is necessary. The program asks for the D0044
 C name of the file interactively, therefore there aren't any additional D0045
 C "Data File Input Instructions" for the second program. To run the D0046
 C program use the "Interactive Data Instructions For Program Two" given D0047
 C below. D0048
 C D0049
 C D0050
 C***** D0051
 C***** D0052
 C***** * D0053
 C***** DATA FILE INPUT INSTRUCTIONS D0054
 C***** * D0055
 C***** D0056
 C***** D0057
 C***** D0058
 C RECORDS 1-3: FORMAT (A76/A76/A76) D0059
 C D0060
 C TITLE(3) Three lines of title. D0061
 C D0062
 C D0063
 C RECORD 4: FORMAT(5I5,2F10.3) D0064
 C D0065
 C NZ Maximum number of data to be included in D0066
 C calculating an estimate using disjunctive D0067
 C kriging. That is, the number of nearest D0068
 C neighbors used in estimation process. D0069
 C Default: (i.e. zero or blank) is NZ = 7. D0070
 C D0071
 C NHK Number of terms in the Hermite expansion. D0072
 C There will be NHK Hermite coefficients D0073
 C calculated by the program. D0074
 C Default: NHK = 7. D0075
 C D0076
 C NPOLY Number of terms to be used in the least D0077
 C squares fit to the [ZD,YZ] pairs. The D0078
 C least squares polynomial is used only for D0079
 C calculating the abscissa points for Hermite D0080
 C integration (to determine the CK's). D0081
 C Default: NPOLY = 7. D0082
 C D0083
 C NTRM Number of terms to be used in Hermite D0084
 C integration. NTRM may be 5 or 10. D0085
 C Default: NTRM = 5. D0086
 C D0087
 C NZCUT Number of conditional probabilities D0088
 C to be calculated using disjunctive kriging. D0089
 C For each conditional probability calculated D0090
 C a ZCUT value must be provided. The ZCUT D0091
 C values, if any, are input in RECORD 7. D0092
 C NZCUT must be less than 19. D0093
 C D0094
 C RMAX Maximum radial search distance for a data D0095
 C point to be included in the estimation D0096
 C process. D0097
 C Default: RMAX = 10000. If linear model D0098
 C for the spatial correlation function is D0099
 C used Default: RMAX = RANGE (see Record 8). D0100
 C The program DISJUNC.FTN will not allow a D0101
 C value of RMAX > RANGE for linear model. D0102
 C D0103
 C CUTOFF Data values larger than CUTOFF will not D0104
 C be included in the analysis. D0105
 C Default: CUTOFF = 10000. D0106
 C D0107
 C D0108

| | | | | |
|---|--------|----|---|--|
| C | RECORD | 5: | FORMAT(3I5,3F10.3) | D0109 |
| C | | | IBLK | D0110 |
| C | | | If: 0 -- Punctual disjunctive kriging | D0111 |
| C | | | If: 1 -- Block disjunctive kriging | D0112 |
| C | | | NXBLK | D0113 |
| C | | | Number of block subdivisions in the X direction. NXBLK*NYBLK is the total number of subblocks used in discrete approximations to the integrals of the correlation (or covariance) over the block. If IBLK equals 1 then Default: NXBLK = 5. | D0114 D0115 D0116 D0117 D0118 D0119 D0120 D0121 |
| C | | | NYBLK | D0122 D0123 D0124 |
| C | | | WIDX | D0125 D0126 D0127 |
| C | | | WIDY | D0128 D0129 |
| C | | | BLKCOV | D0130 D0131 D0132 D0133 D0134 D0135 D0136 D0137 |
| C | | | The inner block covariance (i.e. the covariance value resulting from the double integral of the covariance over the block, where each vector independently describes the interior of the block). If a value is given BLKCOV will NOT be calculated. Default: BLKCOV will be calculated. | D0138 D0139 |
| C | RECORD | 6: | FORMAT(2I5,4F10.3) | D0140 |
| C | | | NX | D0141 D0142 |
| C | | | Number of X coordinates in the grid system of estimates. There will be NX*NY estimates total. Default: NX = 10. | D0143 D0144 D0145 |
| C | | | NY | D0146 D0147 D0148 D0149 |
| C | | | XMIN | D0150 D0151 D0152 |
| C | | | YMIN | D0153 D0154 D0155 |
| C | | | DX | D0156 D0157 D0158 D0159 D0160 D0161 |
| C | | | DY | D0162 D0163 D0164 D0165 D0166 D0167 D0168 |
| C | | | OPTIONAL RECORDS 7a,7b: (INCLUDE IF NZCUT > 0) | D0169 D0170 D0171 |
| C | | | ZCUT(I) | D0172 D0173 D0174 D0175 D0176 D0177 D0178 D0179 D0180 D0181 |
| C | | | FORMAT(8F10.3) | |

| | | |
|-------------|--|-------|
| C RECORD 8: | FORMAT(I5,3F10.3) | D0182 |
| C | MODE The variogram model number where: | D0183 |
| C | -1 -- exponential: [C(1) + C(2)*EXP(-R/C(3))] | D0184 |
| C | | D0185 |
| C | 0 -- linear (with a sill): [C(1) + C(2)*R/C(3)] | D0186 |
| C | | D0187 |
| C | 1 -- spherical: [C(1) + C(2){1.5*R/C(3) - .5*(R/C(3))**3}] | D0188 |
| C | | D0189 |
| C | The correlation function is calculated by correlation = 1. - variogram/sill | D0190 |
| C | | D0191 |
| C | Note: In two dimensions a linear model with a ===== sill is an invalid type of variogram (i.e. it is not of the conditionally positive definite type). Therefore, the program will not allow RMAX to be greater than RANGE if a linear model is chosen. | D0192 |
| C | | D0193 |
| C | | D0194 |
| C | | D0195 |
| C | C(1) NUGGET value of the variogram. | D0196 |
| C | | D0197 |
| C | C(2) SILL value minus the NUGGET value of the variogram. | D0198 |
| C | | D0199 |
| C | C(3) RANGE of the variogram. For the exponential variogram this is the constant that the distance is divided by. | D0200 |
| C | | D0201 |
| C | | D0202 |
| C | | D0203 |
| C | | D0204 |
| C | | D0205 |
| C | | D0206 |
| C | RECORD 9: FORMAT(3I5) | D0207 |
| C | | D0208 |
| C | IDX The column in the data file that contains the X coordinate data. Note: an index (or sample number) is always assumed to be in the first column of the data file and the index column is not counted. Therefore, IDX = 1, 2, or 3. | D0209 |
| C | | D0210 |
| C | | D0211 |
| C | IDY The column in the data file that contains the Y coordinate data. Note: an index (or sample number) is always assumed to be in the first column of the data file and the index column is not counted. Therefore, IDY = 1, 2, or 3. | D0212 |
| C | | D0213 |
| C | | D0214 |
| C | | D0215 |
| C | | D0216 |
| C | | D0217 |
| C | | D0218 |
| C | | D0219 |
| C | IDZ The column in the data file that contains the property to be kriged. Note: an index (or sample number) is always assumed to be in the first column of the data file and the index column is not counted. Therefore, IDZ = 1, 2, or 3. | D0220 |
| C | | D0221 |
| C | | D0222 |
| C | | D0223 |
| C | | D0224 |
| C | | D0225 |
| C | | D0226 |
| C | RECORD 10: FORMAT(4X,A6) | D0227 |
| C | | D0228 |
| C | NAM A name which describes the property to be kriged. | D0229 |
| C | | D0230 |
| C | | D0231 |
| C | | D0232 |
| C | | D0233 |
| C | | D0234 |
| C | | D0235 |
| C | | D0236 |
| C | | D0237 |
| C | | D0238 |
| C | | D0239 |
| C | | D0240 |
| C | | D0241 |
| C | | D0242 |
| C | | D0243 |
| C | | D0244 |
| C | | D0245 |
| C | | D0246 |
| C | | D0247 |
| C | | D0248 |
| C | | D0249 |
| C | | D0250 |
| C | RECORD 11: FORMAT(A30) | D0251 |
| C | | D0252 |
| C | FMT Format specification for the data. This should be of the form: | D0253 |
| C | | D0254 |

```

C                               (I5,2F10.3,5X,F10.3)          D0255
C                                         D0256
C                                         D0257
C                                         D0258
C                                         D0259
C RECORDS 12- : FORMAT IS GIVEN BY FMT (RECORD 11)          D0260
C                                         D0261
C DATA VALUES      The data set should have a location number      D0262
C                   (or index), X coordinates, Y coordinates      D0263
C                   and the property to be kriged. Data is      D0264
C                   read until an end-of-file marker is found.      D0265
C                   Therefore, there should not be any blank      D0266
C                   lines in this part of the data file.      D0267
C                   It is assumed that the location number is      D0268
C                   in the first column.      D0269
C                                         D0270
C                                         D0271
C***** D0272
C***** D0273
C*****      ***** D0274
C*****      INTERACTIVE INPUT INSTRUCTIONS: PROGRAM ONE (DISCALC.FTN)  ***** D0275
C*****      ***** D0276
C*****      All input is free format      ***** D0277
C***** D0278
C***** D0279
C                                         D0280
C RECORD 1:          FILE      Input file name. File name must contain      D0281
C                                         less than 31 characters. This is the      D0282
C                                         name of the data file created using      D0283
C                                         "DATA FILE INPUT INSTRUCTIONS" given above. D0284
C                                         D0285
C                                         D0286
C RECORD 2:          FILE      Output file name. File name must contain      D0287
C                                         less than 31 characters. This file is the      D0288
C                                         input file for DISJUNC.FTN. Only those      D0289
C                                         data and steering parameters needed by the      D0290
C                                         program DISJUNC.FTN are printed out to this      D0291
C                                         file. This file is used "as is" by the      D0292
C                                         program "DISJUNC.FTN".      D0293
C                                         D0294
C                                         D0295
C                                         D0296
C                                         D0297
C RECORD 3:          ANSW      Output device number where the output/      D0298
C                                         summary will be sent. Two choices are      D0299
C                                         allowed:      D0300
C                                         D0301
C                                         P -- output sent to line printer      D0302
C                                         T -- output sent to terminal      D0303
C                                         D0304
C                                         D0305
C***** D0306
C***** D0307
C*****      ***** D0308
C*****      INTERACTIVE INPUT INSTRUCTIONS: PROGRAM TWO (DISJUNC.FTN)  ***** D0309
C*****      ***** D0310
C*****      All input is free format      ***** D0311
C***** D0312
C***** D0313
C                                         D0314
C RECORD 1:          FILE      Input file name. File name must contain      D0315
C                                         less than 31 characters. This is the      D0316
C                                         OUTPUT file created using DISCALC.FTN.      D0317
C                                         D0318
C                                         D0319
C                                         D0320
C RECORD 2a:         ANSW      Output device designation. Choices      D0321
C                                         allowed:      D0322
C                                         D0323
C                                         D - disk file      D0324
C                                         T - terminal      D0325
C                                         P - line printer      D0326
C                                         D0327

```

```

C                                         D0328
C                                         D0329
C OPTIONAL RECORD  2b:  (include only if ANSW.EQ.'D' in RECORD 2a) D0330
C                                         D0331
C             FILE          Output file name. File name must contain D0332
C                           less than 31 characters. D0333
C                                         D0334
C                                         D0335
C                                         D0336
C RECORD    3:   FILE          Plot file name. Produces a file that D0337
C                           can be used to make contour maps, etc. D0338
C                           The file contains x, y, estimates, D0339
C                           estimation variance and conditional D0340
C                           probabilities. File name must contain D0341
C                           less than 31 characters. If a plot file D0342
C                           is NOT wanted type a return. D0343
C                                         D0344
C                                         D0345
C***** D0346
C***** D0347
C***** D0348
C***** MISCELLANEOUS INFORMATION D0349
C***** D0350
C***** D0351
C***** D0352
C                                         D0353
C IMPORTANT ARRAYS: Contains the following information: D0354
C                                         D0355
C     INX(I)      - Index or sample number for the Ith data record. D0356
C     XD(I)       - X coordinate datum for the Ith data record. D0357
C     YD(I)       - Y coordinate datum for the Ith data record. D0358
C     ZD(I)       - Property of interest that is to be estimated. D0359
C     YZ(I)       - Transform of ZD used in disjunctive kriging equations. D0360
C     PR(I)       - Probability of the Ith datum. D0361
C     CK(J)       - Coefficients of the Hermite expansion for the transform. D0362
C     ZCUT(K)     - Cutoff values in terms of the sampled variable (i.e. ZD). D0363
C     YCUT(K)     - Cutoff values in terms of the transform var. (i.e. YZ). D0364
C     PROB(K)     - Conditional probability for cutoff level number K. D0365
C     CS(L)       - Coefficients for the least squares relationship. D0366
C     C(M)        - Variogram model coefficients. D0367
C     AA(N+1,N+1) - Coefficient matrix of the kriging equations. D0368
C                   Right hand side of equations is in AA(i,N+1), i=1,...,N+1 D0369
C     BB(N+1)     - Solution vector (i.e. the kriging weights). D0370
C     ILOC(N)     - The location numbers for the nearest neighbors. D0371
C     RLOC(N)     - The radial distance between each nearest neighbor and D0372
C                   the estimation site. D0373
C     GOX(N)      - The correlation between each nearest neighbor and D0374
C                   the estimation site. D0375
C WHERE: D0376
C                                         D0377
C     I = 1,2,...,MINO(number of samples, 220) D0378
C     J = 1,2,...,MINO(number of Hermite polynomials, 25) D0379
C     K = 1,2,...,MINO(number of cutoff levels, 18) D0380
C     L = 1,2,...,MINO(number of least squares polynomials, 20) D0381
C     M = 1,2,3 D0382
C     N = 1,2,...,MINO(number of nearest neighbors, 20) D0383
C                                         D0384
C                                         D0385
C                                         D0386
C STEPS NECESSARY TO USE DISJUNCTIVE KRIGING PROGRAMS: D0387
C                                         D0388
C 1) Obtain the data set. Must include an X, Y and Z (property of D0389
C                           interest) D0390
C 2) Determine the variogram. (Not included in these programs). D0391
C 3) Create data file using "Data File Input Instructions" D0392
C 4) Run DISCALC.FTN D0393
C 5) Answer questions prompted by program using D0394
C     "Interactive data instructions for program one". Remember that D0395
C     output from this program is inout for DISJUNC.FTN D0396
C 6) Run DISJUNC.FTN D0397
C 7) Answer questions prompted by program using D0398
C     "Interactive data instructions for program two". Remember to D0399
C     use the output from DISCALC.FTN as input to this program. D0400
C                                         D0401
C                                         D0402

```

```

C                                         MA001
C*****                                         MA002
C*****                                         MA003
C*****                                         MA004
C*****                                         MA005
C*****                                         MA006
C*****                                         MA007
C                                         MA008
C PURPOSE:                                         MA009
C   Calculates Hermite coefficients (i.e. Ck's).      MA010
C   Converts the sample data (ZD) into transformed data (YZ).  MA011
C   Converts cutoff values (ZCUT) into transformed cutoff values (YCUT).  MA012
C   Writes out results into a user specified file ready for DISJUNC.FTN.  MA013
C   Prints out a summary of results to a user specified device.      MA014
C                                         MA015
C                                         MA016
C DEVICES REQUIRED:                               MA017
C   Unit #1 -- input disk device (See "data file input instructions")  MA018
C   Unit #2 -- output disk device (See "interactive input instructions")  MA019
C   Unit #5 -- terminal                         MA020
C   Unit #6 -- line printer          (Required only if user specifies summary  MA021
C                                of results to be sent to line printer.      MA022
C                                This is an interactive input).      MA023
C                                         MA024
C                                         MA025
C SUBROUTINE AND FUNCTION CALLS:                 MA026
C   1) MATINV    2) SORT     3) CK1      4) LEAST      MA027
C   5) HYZINV & HYZIM1 (entry)  6) PHI       7) PRB      MA028
C   8) PRBI      9) HK      MA029
C                                         MA030
C                                         MA031
C WRITTEN BY:                                 MA032
C   Scott R. Yates, Computer Sciences Corp.      MA033
C   R.S. Kerr Environ. Res. Lab., Ada, OK, 74820  MA034
C                                         MA035
C*****                                         MA036
C*****                                         MA037
C
C DOUBLE PRECISION AA(21,21)                      MA038
C CHARACTER TITLE(3)*76,FMT*30,RI*10,NAM*6,ANSW*1      MA039
C REAL MEAN,XD(220),YD(220),ZD(220),YZ(220),PR(220),CK(25)  MA040
C # ,C(3),ATMP(4),ZCUT(18),YCUT(18),BB(21),CS(20)      MA041
C INTEGER INX(220)                                MA042
C COMMON IO,IT                                     MA043
C DATA NA/220/,NB/18/,NC/21/,ND/25/,NE/20/        MA044
C                                         MA045
C                                         MA046
C ----- Machine specific device numbers -----      MA047
C ----- IN is disk input, IO is disk output -----  MA048
C ----- IT is the terminal, IL is the printer -----  MA049
C   IN = 1                                         MA050
C   IO = 2                                         MA051
C   IT = 5                                         MA052
C   IL = 6                                         MA053
C                                         MA054
C                                         MA055
C                                         MA056
C ----- open files on PDP 11/70 -----      MA057
C WRITE(IT,701)' GIVE INPUT FILE NAME           >>> '  MA058
C READ(IT,700) FMT                                MA059
C OPEN(UNIT=IN,FILE=FMT,STATUS='OLD',READONLY)      MA060
C WRITE(IT,701)' GIVE OUTPUT FILE NAME (input for kriging) >>> '  MA061
C READ(IT,700) FMT                                MA062
C OPEN(UNIT=IO,FILE=FMT,STATUS='UNKNOWN',CARRIAGECONTROL='LIST')  MA063
C 1 WRITE(IT,850)                                  MA064
C READ(IT,855) ANSW                                MA065
C                                         MA066
C IF(ANSW.EQ.'P') THEN
C   KOUT=IL                                         MA067
C ELSE IF(ANSW.EQ.'T') THEN
C   KOUT=IT                                         MA068
C ELSE
C   WRITE(IT,865)                                    MA069
C GOTO 1                                         MA070
C END IF                                         MA071
C                                         MA072
C                                         MA073
C                                         MA074

```

```

C          700 FORMAT(A50)                               MA075
C          701 FORMAT(///,A50,$)                         MA076
C
C          800 FORMAT(A76)                               MA077
C          805 FORMAT(4(4X,A6))                          MA078
C          810 FORMAT(5I5,6F10.3)                        MA079
C          815 FORMAT(3I5,6F10.3)                        MA080
C          820 FORMAT(2I5,7F10.3)                        MA081
C          825 FORMAT(I5,7F10.3)                         MA082
C          830 FORMAT(8F10.6)                           MA083
C          835 FORMAT(A30)                             MA084
C          840 FORMAT(1P5E15.7)                          MA085
C          845 FORMAT(I5,2F10.3,F10.5,F12.8)           MA086
C          850 FORMAT(////' GIVE OUTPUT DEVICE (FOR THE OUTPUT/SUMMARY) ',/,T20
C             #,'P --- send to line printer',/,T20,'T --- send to terminal',T50
C             #,'>> ',$)                            MA087
C          855 FORMAT(A1)                                MA088
C          860 FORMAT(///// ERROR: Array overflow. Execution will cease.',/
C             #,A5,' must be less than or equal to ',I3,'.',//)      MA089
C          865 FORMAT(///// ERROR: NOT AN ALLOWED INPUT. (Try again)',//)  MA090
C          870 FORMAT(F10.4)                            MA091
C
C          ----- read input parameters -----
C          READ(IN,800)  (TITLE(I),I=1,3)              MA092
C          READ(IN,810)  NZ,NHK,NPOLY,NTRM,NZCUT,RMAX,CUTOFF   MA093
C          READ(IN,815)  IBLK,NXBLK,NYRLK,WIDX,WIDY,BLKCOV    MA094
C          READ(IN,820)  NX,NY,XMIN,YMIN,DX,DY            MA095
C
C          ----- default parameters (steering) -----
C          IF(NZ.LE.0) NZ = 7                         MA096
C          IF(NHK.LE.0) NHK = 7                         MA097
C          IF(NPOLY.LE.0) NPOLY = 7                    MA098
C          IF(NTRM.LE.0) NTRM = 5                     MA099
C          IF(RMAX.LE.0.) RMAX = 10000.                MA100
C          IF(CUTOFF.LE.0.) CUTOFF = 10000.            MA101
C
C          IF(NZCUT.GT.NB.OR.NZCUT.LT.0) GOTO 10       MA102
C          IF(NZCUT.GT.0) READ(IN,830) (ZCUT(I),I=1,NZCUT)  MA103
C
C          READ(IN,825) MODE,(C(I),I=1,3)              MA104
C          READ(IN,815) IXD,IYD,IZD                   MA105
C          READ(IN,805) NAM                           MA106
C          READ(IN,835) FMT                           MA107
C
C          I=0
C          NPTS=0
C          MEAN=0.
C          XMAX=-99999.
C          YMAX=-99999.
C
C          5 READ(IN,FMT,END=10) II,(ATMP(M),M=1,3)      MA108
C          IF(ATMP(IZD).GT.CUTOFF) GOTO 5             MA109
C          I=I+1
C          INX(I)= II
C          XD(I) = ATMP(IXD)
C          YD(I) = ATMP(IYD)
C          ZD(I) = ATMP(IZD)
C          MEAN = MEAN + ZD(I)
C          XMAX = AMAX1(XMAX,XD(I))
C          YMAX = AMAX1(YMAX,YD(I))                  MA110
C
C          ----- check for array overflow -----
C          IF(I.LE.NA) GO TO 5                         MA111
C          WRITE(IT,360) ' NOB ',NA
C          STOP
C          10 IFLAG=0
C          IF(NHK.GT.ND) IFLAG=1
C          IF(NHK.GT.ND) WRITE(IT,260) ' NHK ',ND
C          IF(NPOLY.GT.NE) IFLAG=NE

```

```

IF(NPOLY.GT.NE) WRITE(IT,860) 'NPOLY',IFLAG          MA148
IF(NZCUT.GT.NB) IFLAG=1                           MA149
IF(NZCUT.GT.NB) WRITE(IT,860) 'NZCUT',NB           MA150
IF(IFLAG.GT.0) STOP                                MA151
C
C
C      ----- calculate the sample mean and variance -----
NOB=I
MEAN = MEAN/FLOAT(NOB)                            MA152
VAR  = 0.                                         MA153
DO 13 I=1,NOB                                     MA154
13 VAR = VAR + (ZD(I)-MEAN)*(ZD(I)-MEAN)        MA155
VAR = VAR/FLOAT(NOB)                               MA156
C
C
C      ----- determine the Hermite coefficients -----
CALL SORT(NA,NOB,INX,XD,YD,ZD,YZ,PR)            MA157
CALL LEAST(NOB,NA,NC,NE,ZD,PR,AA,NPOLY,CS)       MA158
CALL CK1(NE,NTRM,NHK,CK,NPOLY,CS)                MA159
C
C
C      ----- invert PHI(Y) relationship for the data -----
CALL HYZINV(NOB,INX,ZD,YZ,NHK,CK)                MA160
C
C
C      ----- invert PHI(Y) relationship for the Ycut values -----
IF(NZCUT.EQ.0) GOTO 20                           MA161
DO 15 I=1,NZCUT                                    MA162
15 YCUT(I) = (ZCUT(I)-MEAN)/VAR                  MA163
CALL HYZINV1(NZCUT,INX,ZCUT,YCUT,NHK,CK)         MA164
C
C
C      ----- calculate the sample variance from the coefficients -----
20 SUM = CK(2)*CK(2)                             MA165
FACT = 1.                                         MA166
DO 25 I=3,NHK                                     MA167
FACT = FACT*FLOAT(I-1)                           MA168
25 SUM = SUM + FACT*CK(I)*CK(I)                  MA169
C
C
C      ----- default parameters for matrix -----
IF(DX.LE.0) DX = XMAX/9.                         MA170
IF(DY.LE.0) DY = YMAX/9.                         MA171
IF(NX.LE.0) NX = 10.                            MA172
IF(NY.LE.0) NY = 10.                            MA173
IF(IBLK.EQ.1.AND.NXBLK.LE.0) NXBLK=5          MA174
IF(IRLK.EQ.1.AND.NYBLK.LE.0) NYBLK=5          MA175
C
C
C      ----- write data file for disjunctive kriging -----
WRITE(IO,800) (TITLE(I),I=1,3)                   MA176
WRITE(IO,815) NZ,NHK,NZCUT,RMAX,CUTOFF          MA177
WRITE(IO,815) IBLK,NXBLK,NYBLK,WIDX,WIDY,BLKCOV MA178
WRITE(IO,820) NX,NY,XMIN,YMIN,DY               MA179
IF(NZCUT.NE.0)WRITE(IO,830) (ZCUT(I),I=1,NZCUT) MA180
IF(NZCUT.NE.0)WRITE(IO,830) (YCUT(I),I=1,NZCUT) MA181
WRITE(IO,840) (CK(I),I=1,NHK)                   MA182
WRITE(IO,825) MODE,(C(I),I=1,3)                 MA183
WRITE(IO,805) NAM                                MA184
C
C
C      DO 30 I=1,NOB
30 WRITE(IO,845) INX(I),XD(I),YD(I),ZD(I),YZ(I) MA185
C
C
C      ----- print out summary of calculation -----
WRITE(KOUT,900)
DO 35 I=1,3
35 WRITE(KOIJT,905) TITLE(I)
WRITE(KOIJT,910)
C
NLIN = 31
WRITE(KOUT,915) NAM,NOB,NZ,NTRM,RMAX,CUTOFF
IF(IBLK.GT.0) THEN

```

```

BI = ' CALCULATE'
IF(BLKCOV.GT.0.) WRITE(BI,870) BLKCOV
WRITE(KOUT,920) BI,NXBLK,NYRLK,WIDX,WIDY
NLIN = NLIN+4
ELSE
ENDIF
C
IF(MODE.LT.0) BI = ' EXPONENT'
IF(MODE.EQ.0) BI = ' LINEAR'
IF(MODE.GT.0) BI = ' SPHERICAL'
WRITE(KOUT,925) BI,C(1),C(2),C(3)
WRITE(KOUT,930) NAM,NAM
C
C ----- print data -----
DO 40 I=1,N08/2+1
NLIN = NLIN + 1
IF(NLIN.GE.58) WRITE(KOUT,955) '1'
IF(NLIN.GE.58) NLIN = 0
II = NOB/2 + I + 1
IF(INX(I).EQ.0) INX(I) = I
IF(INX(II).EQ.0) INX(II) = II
IF(II.GT.NOB) INX(II) = 0
40 WRITE(KOUT,935) INX(I),XD(I),YD(I),ZD(I),YZ(I),INX(II),XD(II),
# ,YD(II),ZD(II),YZ(II)
C
NLIN = NLIN + 7
IF(NLIN.GE.50) WRITE(KOUT,955) '1'
IF(NLIN.GE.50) NLIN = 0
WRITE(KOUT,940) MEAN,CK(1),VAR,SUM
C
C ----- print coefficients -----
ITMP=MAX0(NHK,NPOLY)
MLIN = NLIN + 7
IF(NLIN.GE.58) WRITE(KOUT,955) '1'
IF(NLIN.GE.58) NLIN = 0
WRITE(KOUT,945)
DO 45 MN=1,ITMP
IF(MN.LE.NPOLY.AND.MN.LE.NHK) WRITE(KOUT,950) CS(MN),CK(MN)
IF(MN.LE.NPOLY.AND.MN.GT.NHK) WRITE(KOUT,955) CS(MN)
IF(MN.GT.NPOLY.AND.MN.LE.NHK) WRITE(KOUT,960) CK(MN)
NLIN = NLIN + 1
45 CONTINUE
C
STOP
C
----- end of problem -----
900 FORMAT(1H1,10X,82(1H*)/11X,1H*,80X,1H*/11X,1H*,22X,' DISJUNCTIVE
# KRIGING PROGRAM: ONE ',22X,1H*,',/11X,1H*,22X,' FOR DETERMIN
# ING THE COEFFICIENTS ',22X,1H*/11X,1H*,80X,1H*)
MA269
MA270
MA271
MA272
MA273
MA274
MA275
MA276
MA277
MA278
MA279
MA280
MA281
MA282
MA283
MA284
MA285
MA286
MA287
MA288
MA289
MA290
MA291
MA292
MA293
905 FORMAT(11X,1H*,2X,A76,2X,1H*)
910 FORMAT(11X,1H*,80X,1H*,/,11X,82(1H*))
915 FORMAT(//11X,'INPUT PARAMETERS'/11X,16(1H=/
11X,'NUMBER OF ',A6,'''s INPUT.....',I10,/,
211X,'MAXIMUM NUMBER OF NEAREST NEIGHBORS.....',I10,/,
311X,'NUMBER OF TERMS USED IN HERMITE INTEGRATION.....',I10,/,
411X,'MAXIMUM RADIUS.....',F10.4,/,
511X,'MAXIMUM ALLOWED DATA VALUE.....',F10.4,/)

920 FORMAT(
111X,'DISPERSION COVARIANCE WITHIN BLOCK.....',A10,/,
211X,'NUMBER OF CELLS',34(1H.),5X,' NX = ',I10,5X,' NY = ',I10,/,
311X,'BLOCK SIZE',5X,34(1H.),5X,' DX = ',F10.3,5X,'DY = ',F10.3,/)

925 FORMAT(11X,'VARIogram MODEL TYPE',29(1H.),A10,
1./,11X,'NUGGET.....',F10.4,/,
2,11X,'SILL MINUS NUGGET.....',F10.4,/,
3,11X,'RANGE.....',F10.4)
930 FORMAT(//11X,'OBSERVED DATA',/11X,13(1H=)/,7X,2('OBS. NO.',7X,1HX,
111X,1HY,6X,A6,8X,2HYz,15X))
935 FORMAT(2(1I2,2F12.3,F12.4,F12.5,5X))
940 FORMAT(//,11X,'DATA SET:',23X,'HERMITE POLYNOMIALS: ',/,15X,
1'MEAN      = ',F12.5,9X,'MEAN      = ',F12.5,/,15X,
2'VARIANCE = ',F12.5,9X,'VARIANCE = ',F12.5,/)

```

```

945 FORMAT(/,11X,'COEFFICIENTS:',/11X,13(1H=),//,11X,'LEAST SQUARES:',      MA294
     &18X,'HERMITE POLYNOMIALS:',/11X,'(cum. distrib)',/)                         MA295
950 FORMAT(11X,F14.6,22X,F14.6)                                              MA296
955 FORMAT(11X,F14.6)                                                       MA297
960 FORMAT(47X,F14.6)                                                       MA298
C
C          END
C
C*****PROGRAM TWO:  DISJUNC.FTN***** MB002
C*****PURPOSE:***** MB003
C*****LOCATES THE NZ NEAREST NEIGHBORS TO BE USED IN THE ESTIMATION PROCESS. MB010
C*****CALCULATES THE SPATIAL CORRELATIONS AND CONSTRUCTS COEFFICIENT MATRIX MB011
C*****SOLVES MATRIX EQUATIONS FOR DISJUNCTIVE KRIGING WEIGHTS. MB012
C*****CALCULATES ESTIMATES, ERROR VARIANCES AND CONDITIONAL PROBABILITIES MB013
C*****WRITES OUT RESULTS INTO A USER SPECIFIED DEVICE. MB014
C*****CREATES A PLOT FILE IF SPECIFIED BY USER. MB015
C
C*****DEVICES REQUIRED:***** MB017
C      UNIT #1 -- INPUT FILE DISK DEVICE (OUTPUT FROM DISCALC.FTN) MB019
C      UNIT #2 -- OUTPUT FILE DISK DEVICE (USED ONLY IF DISK OUTPUT SPECIFIED) MB020
C      UNIT #3 -- PLOT FILE DISK DEVICE (USED ONLY IF PLOT FILE SPECIFIED) MB021
C      UNIT #5 -- TERMINAL MB022
C      UNIT #6 -- LINE PRINTER      (REQUIRED ONLY IF USER SPECIFIES SUMMARY MB023
C                               OF RESULTS TO BE SENT TO LINE PRINTER. MB024
C                               THIS IS AN INTERACTIVE INPUT). MB025
C
C*****SUBROUTINE AND FUNCTION CALLS:***** MB027
C      1) SOLN      2) MATRX      3) ATOK      4) MATINV      MB029
C      5) AVECCR    6) BLKPTS     7) VARIO      8) HCALC      MB030
C      9) H & H1 (ENTRY POINT)   10) HK        11) PRB       MB031
C
C*****WRITTEN BY:***** MB032
C      SCOTT R. YATES, COMPUTER SCIENCES CORP. MB035
C      R.S. KERR ENVIRON. RES. LAB., ADA, OK, 74820 MB036
C
C*****CHARACTER TITLE(3)*76,FMT*30,MI(3)*11,BI*6,NAM*6,ANSW*I,AZCUT*15, MB038
C      # AZEXP(18)*4 MB042
C      DOUBLE PRECISION AA(21,21) MB043
C      REAL XD(220),YD(220),ZD(220),YZ(220),C(3),CK(25),PROB(18), MB044
C      #ZCUT(18),YCUT(18),RLOC(20),GOX(20),BB(21),MEAN MB045
C      INTEGER INX(220),ILOC(20) MB046
C      INTFGER*4 ISEED MB047
C      COMMON IO,IT MB048
C      COMMON /BLK/ NXRLK,NYBLK,WIDX,WIDY,ISEED MB049
C
C      DATA PI2/2.50662829/,MI(1)/*EXPONENTIAL*/,MI(2)/* LINEAR*/ MB051
C      #,MI(3)/* SPHERICAL*/
C      DATA NA/220/,NB/18/,NC/21/,ND/25/,NE/20/ MB052
C
C----- Machine specific device numbers ----- MB056
C----- IN is disk input, ID is disk output ----- MB057
C----- IP is plot output, IT is the terminal ----- MB058
C----- IL is the printer ----- MB059
C      IN = 1 MB061
C      ID = 2 MB062
C      IP = 3 MB063
C      IT = 5 MB064
C      IL = 6 MB065

```

```

C
C
C      ----- open files on PDP 11/70 -----
C      WRITE(IT,701)' GIVE INPUT FILE NAME          >>> '
C      READ(IT,700) FMT
C      OPEN(UNIT=IN,FILE=FMT,STATUS='OLD',READONLY)
1  WRITE(IT,703)
C      READ(IT,702) ANSW
C
C      IF(ANSW.EQ.'T' .OR. ANSW.EQ.'t') THEN
C          IO=IT
C      ELSE IF(ANSW.EQ.'P' .OR. ANSW.EQ.'p') THEN
C          IO=IL
C      ELSE IF(ANSW.EQ.'D' .OR. ANSW.EQ.'d') THEN
C          WRITE(IT,701)' GIVE OUTPUT FILE NAME        >>> '
C          READ(IT,700) FMT
C          OPEN(UNIT=ID,FILE=FMT,STATUS='UNKNOWN')
C          IO=ID
C      ELSE
C          WRITE(IT,'*)' INCORRECT VALUE GIVEN -- MUST BE [T], [P] OR [D]*
C          GOTO 1
C      END IF
C
C      WRITE(IT,701)' GIVE PLOT FILE NAME (return=none)    >>> '
C      READ(IT,700) FMT
C      IF(FMT.EQ.' ') GOTO 5
C      OPEN(UNIT=IP,FILE=FMT,STATUS='UNKNOWN')
C
C      700 FORMAT(A30)
C      701 FORMAT(////,A50,$)
C      702 FORMAT(A1)
C      703 FORMAT(////,' GIVE THE OUTPUT DEVICE: ',/,T20,
C                 #' D -- create a disk file ',/,T20,' P -- send to line printer '
C                 #./,T20,' T -- send to terminal',T46,'>>> '$)
C
C      800 FORMAT(A76)
C      805 FORMAT(3I5,7F10.3)
C      810 FORMAT(2I5,7F10.3)
C      815 FORMAT(8F10.3)
C      820 FORMAT(1P5E15.8)
C      825 FORMAT(I5,7F10.3)
C      830 FORMAT(4X,A6)
C      835 FORMAT(I5,3F10.3,F12.8)
C      840 FORMAT(////,' ERROR: array overflow -- ',A5,' is greater than '
C                 # ,I3,/)
C      845 FORMAT(F6.3)
C
C
C      ----- read steering parameters -----
5  READ(IN,800) (TITLE(I),I=1,3)
READ(IN,805) NZ,NHK,NZCUT,RMAX,CUTOFF
READ(IN,805) IBLK,NXBLK,NYBLK,WIDX,WIDY,BLKCOV
READ(IN,810) NX,NY,XMIN,YMIN,DX,DY
IF(NZCUT.NE.0) READ(IN,815) (ZCUT(I),I=1,NZCUT)
IF(NZCUT.NE.0) READ(IN,815) (YCUT(I),I=1,NZCUT)
READ(IN,820) (CK(I),I=1,MHK)
READ(IN,825) MODE,(C(I),I=1,3)
IF(MODE.EQ.0.AND.RMAX.GT.C(3)) RMAX = C(3)
READ(IN,830) NAM
C
C      ----- transform variogram into the correlation function -----
CVARK = C(1)+C(2)
C(1)=C(1)/CVARK
C(2)=C(2)/CVARK
C
C      ----- read data from file -----
I=0
MEAN = 0.
15 I=I+1
20 READ(IN,835,END=25) INX(I),XD(I),YD(I),ZD(I),YZ(I)

```

```

IF(ZD(I).GT.CUTOFF) GOTO 20 MB138
MEAN = MEAN + ZD(I)
GO TO 15 MB139
25 NOB=I-1 MB140
C MB141
C MB142
C MB143
C ----- check for array overflow -----
IFLAG=0 MB144
IF(NOB.GT.NA) IFLAG=1 MB145
IF(NOB.GT.NA) WRITE(IT,840)' NOB ',NA MB146
IF(NHK.GT.ND) IFLAG=1 MB147
IF(NHK.GT.ND) WRITE(IT,840)' NHK ',ND MB148
IF(NZ.GT.NE) IFLAG=1 MB149
IF(NZ.GT.NE) WRITE(IT,840)' NZ ',NE MB150
IF(NZCUT.GT.NB)IFLAG=1 MB151
IF(NZCUT.GT.NB)WRITE(IT,840)'NZCUT',NB MB152
IF(IFLAG.EQ.1) STOP MB153
C MB154
C MB155
C ----- calculate the sample statistics -----
MEAN = MEAN/FLOAT(NOB) MB156
VAR = 0. MB157
DO 27 I=1,NOB MB158
27 VAR = VAR + (ZD(I)-MEAN)*(ZD(I)-MEAN) MB159
VAR = VAR/FLOAT(NOB) MB160
C MB161
C MB162
SUM = CK(2)*CK(2) MB163
FACT = 1. MB164
DO 30 I=3,NHK MB165
FACT = FACT*FLOAT(I-1) MB166
30 SUM = SUM + FACT*CK(I)*CK(I) MB167
C MB168
C ----- calculate inner block covariance. Function SECNDS is -----
C ----- a system routine which returns the number of seconds -----
C ----- from start of day and used to "randomize" the seed -----
IF(IBLK.EQ.0) BLKCOV = SUM MB169
IF(IBLK.EQ.0) GOTO 35 MB170
ISEED = IFIX(SECNDS(0.0)) MB171
IF(BLKCOV.GT.0.0) GOTO 35 MB172
CALL AVECOR(ND,BLKCOV,C,MODE,NHK,CK) MB173
35 CONTINUE MB174
C MB175
C MB176
C ----- write out summary of input data -----
WRITE(IO,900) MB177
IF(IBLK.EQ.0) WRITE(IO,905) MB178
IF(IBLK.NE.0) WRITE(IO,910) MB179
DO 40 I=1,3 MB180
40 WRITE(IO,915) TITLE(I) MB181
WRITE(IO,920) MB182
C MB183
NLIN = 31 MB184
WRITE(IO,925) NAM,NOB,NZ,RMAX,CUTOFF MB185
IF(IBLK.GT.0) WRITE(IO,930) BLKCOV,NXBLK,NYBLK,WIDX,WIDY MB186
IF(IBLK.GT.0) NLIN = NLIN+4 MB187
WRITE(IO,935) MI(MODE+2),C(1),C(2),C(3) MB188
WRITE(IO,940) NAM,NAM MB189
C MB190
C ----- print out data -----
DO 45 I=1,NOB/2+1 MB191
NLIN = NLIN + 1 MB192
IF(NLIN.GE.58) WRITE(IO,702) '1' MB193
IF(NLIN.GE.58) NLIN = 0 MB194
II = NOB/2 + I + 1 MB195
IF(INX(I).EQ.0) INX(I) = I MB196
IF(INX(II).EQ.0)INX(II) = II MB197
IF(II.GT.NOB) INX(II)=0 MB198
45 WRITE(IO,945) INX(I),XD(I),YD(I),ZD(I),YZ(I),INX(II),XD(II) MB199
#,YD(II),ZD(II),YZ(II) MB200
C MB201
NLIN = NLIN + 7 MB202
IF(NLIN.GE.50) WRITE(IO,702) '1' MB203
IF(NLIN.GE.50) NLIN = 0 MB204
WRITE(IO,950) MEAN,CK(1),VAR,SUM MB205
MB206
MB207
MB208
MB209
MB210
MB211

```

```

C
C      ----- print out coefficients -----
NLIN = NLIN + 5                                MB212
IF(NLIN.GE.50) WRITE(10,702) '1'                MB213
IF(NLIN.GE.50) NLIN = 0                          MB214
WRITE(10,955)                                     MB215
DO 50 I=1,NHK/2+1                               MB216
II = NHK/2 + I + 1                            MB217
K = I-1                                         MB218
KK = II-1                                       MB219
NLIN = NLIN + 1                                 MB220
50 WRITE(10,960) K,CK(I),KK,CK(II)             MB221
C
NLIN = NLIN + 7                                 MB222
IF(NLIN.GE.58) WRITE(10,702) '1'                MB223
IF(NLIN.GE.58) NLIN = 0                          MB224
BI = ' Zcut '                                    MB225
MB226
DO 55 I=1,NZCUT                                 MB227
WRITE(AZCUT,920) ZCUT(I)                         MB228
READ(AZCUT(1:11),845) ZCUT(I)                   MB229
AZEXP(I) = AZCUT(12:15)                          MB230
55 CONTINUE                                      MB231
C
IV=MINO(9,NZCUT)                                MB232
JV=MAX0(NZCUT-IV,1)                            MB233
KV=MAX0(1,(IV*8-50)/2)                          MB234
LV=MAX0(1,(IV*8-16)/2)                          MB235
C
WRITE(10,965) (BI,I=1,MINO(1,NZCUT))           MB236
WRITE(10,970) (BI,I=1,MINO(1,NZCUT))           MB237
WRITE(10,975) (ZCUT(I),I=1,IV),(AZEXP(I),I=1,IV),
#          (ZCUT(I),I=10,NZCUT),(AZEXP(I),I=10,NZCUT) MB238
WRITE(10,980)                                     MB239
IF(IV.GT.0) WRITE(10,985)                         MB240
C
C*****                                              MB241
C*****      BEGIN DISJUNCTIVE KRIGING -- ON A GRID   MB242
C*****                                              MB243
C*****                                              MB244
C*****                                              MB245
C*****                                              MB246
C*****                                              MB247
C*****                                              MB248
C*****                                              MB249
C*****                                              MB250
C*****                                              MB251
C
C      ----- create matrix of Hermite polynomials values -----
CALL HCALC(NA,NOR,NHK,YZ)                      MB252
C
X0 = XMIN                                       MB253
DO 60 I=1,NX                                     MB254
Y0 = YMIN                                       MB255
C
DO 65 J=1,NY                                     MB256
EST = 0.0                                         MB257
VARK = 0.0                                         MB258
IFLG = 0                                           MB259
C
DO 70 M=1,NZCUT                                  MB260
PROB(M) = 0.0                                     MB261
70
C
CALL MATRX(NA,NC,NE,NZ,N1,NOB,X0,Y0,XD,YD,ILOC,RLOC,RMAX,GOX,
&          NCOL,NROW,MODE,C,IBLK,AA,IFLG)          MB262
IF(IFLG.EQ.1) GOTO 65                           MB263
C
C      ----- sum series on k -----
DO 85 IK=1,NHK                                    MB264
K = (IK-1)                                       MB265
IF(K.EQ.0) GOTO 80                               MB266
IF(K.EQ.1) GOTO 75                               MB267
CALL ATOK(NC,NE,K,NROW,NCOL,GOX,AA)            MB268
75  CALL MATINV(NROW,NCOL,NC,AA,BB)              MB269
80  CALL SOLN(NB,NC,ND,NE,K,N1,EST,VARK,ILOC,GOX,CK,PROB,YCUT
#,NZCUT,BB)                                     MB270
85 CONTINUE                                      MB271
C
C      ----- calculate the conditional probability -----
C      ----- note: because the Hermite series is truncated -----
C      ----- neg. or values greater than 1 are possible -----

```

```

C      ----- these values are set to either 0 or 1 ----- MB285
      IF(NZCUT.EQ.0) GOTO 95
      DO 90 M=1,NZCUT
      TMP = YCUT(M)
      PROB(M) = 1.0 - PRB(TMP) + EXP(-TMP*TMP/2.)*PROB(M)/PI2
      IF(PROB(M).GT.1.0) PROB(M) = 1.0
      90   IF(PROB(M).LT.0.0) PROB(M) = 0.0

C      95 VARK = BLKCOV-VARK
C
C      ----- print out results -----
      IF(NLIN.GE.58) WRITE(IO,702) '1'
      IF(NLIN.GE.58) NLIN = 0
      NLIN = NLIN + 1
      IF(IP.EQ.3)WRITE(IP,990) X0,Y0,EST,VARK,(PROB(M),M=1,NZCUT)
      WRITE(IO,995) N1,X0,Y0,EST,VARK,(PROB(M),M=1,NZCUT)

      65 Y0 = Y0 + DY
      60 X0 = X0 + DX
      STOP

C      ----- end of problem -----
      900 FORMAT(1H1,10X,82(1H*)/11X,1H*,80X,1H*/11X,1H*,22X,'DISJUNCTIVE KR
      #IGING ON A GRID MATRIX',22X,1H*/.,11X,1H*,18X,'FOR ESTIMATING SPAT
      #IALLY DEPENDENT PROPERTIES',16X,1H*/11X,1H*,80X,1H*)
      905 FORMAT(11X,1H*,32X,'POINT ESTIMATION',32X,1H*/.,11X,1H*,80X,1H*)
      910 FORMAT(11X,1H*,32X,'BLOCK ESTIMATION',32X,1H*/.,11X,1H*,80X,1H*)
      915 FORMAT(11X,1H*,2X,A76,2X,1H*)
      920 FORMAT(11X,1H*,2X,76X,2X,1H*/.,11X,82(1H*))
      925 FORMAT(//11X,'INPUT PARAMETERS'/11X,16(1H=)/
      #11X,'NUMBER OF ',A6,'"''S INPUT.....',110./,
      #11X,'MAXIMUM NUMBER OF NEAREST NEIGHBORS.....',110./,
      #11X,'MAXIMUM RADIUS.....',F10.4./,
      #11X,'MAXIMUM ALLOWED DATA VALUE.....',F10.4./)
      930 FORMAT(
      #11X,'DISPERSION COVARIANCE WITHIN BLOCK.....',F10.4./
      #11X,'NUMBER OF CELLS',34(1H.),5X,' NX = ',110,5X,'NY = ',110,/
      #11X,'BLOCK SIZE',5X,34(1H.),5X,' DX = ',F10.3,5X,'DY = ',F10.3,/)
      935 FORMAT(11X,'CORRELATION FUNCTION MODEL TYPE.....',A11
      #./,11X,'NUGGET.....',F10.4./
      #,11X,'SILL MINUS NUGGET.....',F10.4./
      #,11X,'RANGE.....',F10.4)
      940 FORMAT(//11X,'OBSERVED DATA',/11X,13(1H=)/,7X,2('OBS. NO.',7X,1HX,
      #11X,1HY,6X,A6,8X,2HYz,15X))
      945 FORMAT(2(I12.2F12.3,F12.3,F12.6,5X))
      950 FORMAT(//,11X,'DATA SET:',23X,'HERMITE POLYNOMIALS: ',./,15X,
      #'MEAN      = ',F12.5,9X,'MEAN      = ',F12.5./,15X,
      #'VARIANCE = ',F12.5,9X,'VARIANCE = ',F12.5,/)
      955 FORMAT(//,11X,'HERMITE POLYNOMIAL COEFFICIENTS:',/11X,32(1H=),//,
      #15X,'K',7X,'CK',16X,'K',7X,'CK')
      960 FORMAT(11X,I5,1PE15.6,6X,I5,1PE15.6)
      965 FORMAT(///,11X,'DISJUNCTIVE KRIGING ESTIMATES',19X,::,<KV>(' ')
      #'PROBABILITY THAT THE ESTIMATE IS GREATER THAN',A6)
      970 FORMAT(11X,29(1H=),19X,::,<KV>(' '),50(1H=),//,59X,<LV>(' '),
      #'Value(s) of',A6)
      975 FORMAT(/,59X,::,<IV>(:,'(',F6.3,')'),/,59X,<IV>(:,'(',A4,')'),:
      #,/59X,<JV>(:,'(',F6.3,')'),/59X,<JV>(:,'(',A4,')'))
      980 FORMAT(10X,'I',5X,'X',7X,'Y',6X,'EST',6X,'VARK',5)
      985 FORMAT('+' ,15X,<IV>('+'-----+'))
      990 FORMAT(2F8.2,2F9.3,4X,2(9(1X,F6.3,1X),/38X))
      995 FORMAT(9X,I2,2F8.2,2F9.3,14X,2(9(1X,F6.3,1X),/59X))

C      END
C
C***** SUBROUTINES AND FUNCTIONS FOR DISJUNCTIVE KRIGING PROGRAMS *****
C***** ***** S0004
C***** ***** S0005
C***** ***** S0006
C***** ***** S0007
C***** ***** S0008
C***** ***** S0009
C***** ***** S0010

```

```

C***** SUBROUTINE SORT: SORTS ZD(I) DATA INTO INCREASING ORDER *****
C***** (USE WITH PROGRAM ONE) *****
C***** SUBROUTINE SORT(NA,NOB,INX,XD,YD,ZD,YZ,PR)
C***** DIMENSION INX(NA),XD(NA),YD(NA),ZD(NA),YZ(NA),PR(NA)

C DO 10 I=1,NOB
C XMIN = 99999999.
C DO 15 J=I,NOB
C IF(XMIN.LT.ZD(J)) GOTO 15
C XMIN = ZD(J)
C IJ = J
C 15 CONTINUE
C TZ=ZD(I)
C TX=XD(I)
C TY=YD(I)
C LOC=INX(I)
C ZD(IJ)=ZD(IJ)
C XD(I)=XD(IJ)
C YD(I)=YD(IJ)
C INX(I)=INX(IJ)
C ZD(IJ)=TZ
C XD(IJ)=TX
C YD(IJ)=TY
C INX(IJ)=LOC
C 10 CONTINUE
C LCNT = 0
C DO 20 I=1,NOB
C IF(ZD(I).NE.ZD(I+1).AND.LCNT.EQ.0) GOTO 35
C IF(ZD(I).NE.ZD(I+1).AND.LCNT.NE.0) GOTO 25
C ----- duplicate ZD value -----
C LCNT = LCNT + 1
C GOTO 20
C ----- no more duplicates -- add and calculate results -----
C 25 TZ = (FLOAT(I)-.5)/FLOAT(NOB)
C LCNT = LCNT + 1
C TMP = PRBI(1.0-TZ)
C DO 30 KJ=1,LCNT
C PR(I-KJ+1) = TZ
C YZ(I-KJ+1) = TMP
C 30 CONTINUE
C LCNT = 0
C GOTO 20
C ----- no duplicates (LCNT = 0) -----
C 35 CONTINUE
C PR(I) = (FLOAT(I)-.5)/FLOAT(NOB)
C YZ(I) = PRBI(1.0-PR(I))
C 20 CONTINUE
C RETURN
C END

C***** SUBROUTINE CK1: CALCULATED THE COEFFICIENTS "CK" BY *****
C***** NUMERICAL HERMITE INTEGRATION *****
C***** W's = W(i)*EXP(YW(i)*YW(i)/2.) *****
C***** NHK = Maximum order for the Hermite poly *****
C***** (USE WITH PROGRAM ONE) *****
C***** SUBROUTINE CK1(NE,NTRM,NHK,CK,NPOLY,CS)
C***** DIMENSION W(15),YW(15),CK(NHK),CS(NE)

```

```

C----- VALUES FOR INTEGRATION: -----
C----- YW(I) & W(I); I=1,2,...,10 are for 10 term -----
C----- YW(I) & W(I); I=11,12,...,15 are for 5 term -----
C DATA NJ/15.,YW(1)/0.245340708E+00.,W(1)/0.476366813E+00/
&, YW(2)/0.737473729E+00.,W(2)/0.376261601E+00/ CK018
&, YW(3)/0.123407622E+01.,W(3)/0.233452289E+00/ CK019
&, YW(4)/0.173853771E+01.,W(4)/0.112451776E+00/ CK020
&, YW(5)/0.225497400E+01.,W(5)/0.412310259E-01/ CK021
&, YW(6)/0.278880606E+01.,W(6)/0.111539518E-01/ CK022
&, YW(7)/0.334785457E+01.,W(7)/0.211861101E-02/ CK023
&, YW(8)/0.394476404E+01.,W(8)/0.259968734E-03/ CK024
&, YW(9)/0.460368245E+01.,W(9)/0.176028288E-04/ CK025
&, YW(10)/0.538748089E+01.,W(10)/0.447583714E-06/ CK026
&, YW(11)/0.34290133/,W(11)/.647852317/, CK027
&YW(12)/1.0366108/,W(12)/.410960574/,YW(13)/1.75668365/ CK028
&W(13)/.158479939/,YW(14)/2.53273167/,W(14)/.033206690/, CK029
&YW(15)/ 3.43615912/,W(15)/.00279908848/ CK030
DATA SQ2PI/2.50662829/ CK031
IF(NTRM.EQ.10) NJ=10 CK032
IF(NTRM.EQ.10) NTRM=0 CK033
IF(NTRM.EQ.5) NTRM=10 CK034
C FACT = 1.
DO 5 I=1,NHK CK035
K = (I-1)
CK(I) = 0. CK036
IF(K.GE.2) FACT = FACT*FLOAT(K)
DO 10 J=1+NTRM,NJ CK037
YM = -YW(J)
YP = YW(J)
PHIM = PHI(NE,YM,NPOLY,CS)
PHIP = PHI(NE,YP,NPOLY,CS)
CK(I) = CK(I) + W(J)*PHIM*HK(K,YM)
CK(I) = CK(I) + W(J)*PHIP*HK(K,YP)
10 CONTINUE CK038
C CK(I) = CK(I)/(FACT*SQ2PI)
5 CONTINUE CK039
RETURN CK040
END CK041
C*****
C***** SUBROUTINE LEAST: CALCULATES THE LEAST SQUARES REGRESSION *****
C***** LINE OF THE FORM: *****
C***** Y = C1 + C2*X + C3*X*X + .... *****
C***** (USE WITH PROGRAM ONE) *****
C*****
C***** SUBROUTINE LEAST(NOB,NA,NC,NE,ZD,PR,AA,NPOLY,CS) *****
C***** DOUBLE PRECISION AA,DBLE *****
C***** DIMENSION ZD(NA),PR(NA),CS(NE),AA(NC,NC) *****
C***** COMMON IO,IT *****
C
C NROW = NPOLY
C NCOL = NROW + 1
C DO 1 L=1,NROW
C AA(L,NCOL)=0.000
C
C DO 5 M=L,NROW
C AA(L,M)=0.000
C
C DO 10 I=1,NOB
C AA(L,M) = AA(L,M) + DBLE(PR(I)**FLOAT(L-1)*PR(I)**FLOAT(M-1))
C AA(M,L) = AA(L,M)
C 5 CONTINUE
C
C DO 15 I=1,NOB
C AA(L,NCOL) = AA(L,NCOL) + DBLE(PR(I)**FLOAT(L-1)*ZD(I))
C 1 CONTINUE

```

```

C                                         LE033
C                                         LE034
C ----- solve the matrix -----
C                                         CALL MATINV(NROW,NCOL,NC,AA,CS)          LE035
C                                         RETURN          LE036
C                                         END          LE037
C                                         HY001
C*****                                     ***** HY002
C*****                                     ***** HY003
C*****      SUBROUTINE HYZINV: CALCULATES THE NORMALIZED YZ FROM THE Z   *****
C*****          DATA BY INVERTING THE EQUATION                         *****
C*****                                     ***** HY004
C*****                                     ***** HY005
C*****      Z = C(0) + C(1)HK(1,Y) + C(2)HK(2,Y) + ..... C(N)HK(N,Y)    *****
C*****                                     ***** HY006
C*****          (USE WITH PROGRAM ONE)                                *****
C*****                                     ***** HY007
C*****                                     ***** HY008
C*****                                     ***** HY009
C*****                                     ***** HY010
C                                         HY011
C
C SUBROUTINE HYZINV(NOB,INX,ZD,YZ,NHK,CK)          HY012
C DIMENSION CK(NHK),INX(NOB),ZD(NOB),YZ(NOB)        HY013
C CHARACTER*6 NAME          HY014
C DATA TOL/1.0E-6/, NIT/20/          HY015
C COMMON IO,IT          HY016
C                                         HY017
C     NAME = ' DATA '
5 DO 10 I=1,NOB          HY018
  YM1 = YZ(I)-.005*YZ(I) - .0005          HY019
  Y   = YZ(I)+.005*YZ(I) + .0005          HY020
  N   = 0          HY021
C                                         HY022
C----- begin secant approximation to -----
C----- Newton's interative method -----
15 FM1 = 0.0          HY023
  F = 0.0          HY024
  DO 20 J=1,NHK          HY025
    K = (J-1)
    FM1 = FM1 + CK(J)*HK(K,YM1)          HY026
20 F = F + CK(J)*HK(K,Y)          HY027
C                                         HY028
  FM1 = FM1-ZD(I)          HY029
  F = F -ZD(I)          HY030
  SLOPE = (F-FM1)/(Y-YM1)          HY031
  IF(ABS(SLOPE).LT.1.0E-10) GOTO 25          HY032
  IF(ABS(SLOPE).GT.1.0E+10) GOTO 25          HY033
  YP1 = Y - F/SLOPE          HY034
  YM1 = Y          HY035
  Y   = YP1          HY036
  N = N+1          HY037
  IF(ABS(Y).GT.10.0) GOTO 25          HY038
  IF(ABS(Y-YM1).LT.TOL) GOTO 45          HY039
  IF(N.GT.NIT) GOTO 25          HY040
  GOTO 15          HY041
C                                         HY042
C----- solution diverging -- try alternate method -----
25 IFLAG=0          HY043
  N=0          HY044
  Y = YZ(I)          HY045
  YL = Y -.1*ABS(Y) - 1.          HY046
  YR = Y + .1*ABS(Y) + 1.          HY047
30 Z1=0.0          HY048
  DO 35 J=1,NHK          HY049
    K=J-1
    Z1 = Z1 + CK(J)*HK(K,Y)
    IF(ABS(Z1-ZD(I)).LT..00001) GOTO 45
    IF(N.GT.50) GOTO 40
C                                         HY050
  IF(Z1.GT.ZD(I)) YR=Y          HY051
  IF(Z1.LT.ZD(I)) YL=Y          HY052
  N=N+1          HY053
  Y=(YR+YL)/2.          HY054
  GOTO 30          HY055
C                                         HY056
C                                         HY057
C                                         HY058
C                                         HY059
C                                         HY060
C                                         HY061
C                                         HY062
C                                         HY063
C                                         HY064

```



```

C***** SUBROUTINE SOLN: CALCULATES THE DISJUNCTIVE KRIGING ***** SNO01
C***** SOLUTION: EST, VARK AND PROB ***** SNO02
C***** (USE WITH PROGRAM TWO) ***** SNO03
C***** ***** SNO04
C***** ***** SNO05
C***** ***** SNO06
C***** ***** SNO07
C***** ***** SNO08
C***** ***** SNO09
C***** ***** SNO10
C***** ***** SNO11
C***** ***** SNO12
C***** ***** SNO13
C***** ***** SNO14
C***** ***** SNO15
C***** ***** SNO16
C***** ***** SNO17
C***** ***** SNO18
C***** ***** SNO19
C***** ***** SNO20
C***** ***** SNO21
C***** ***** SNO22
C***** ***** SNO23
C***** ***** SNO24
C***** ***** SNO25
C***** ***** SNO26
C***** ***** SNO27
C***** ***** SNO28
C***** ***** SNO29
C***** ***** SNO30
C***** ***** SNO31
C***** ***** SNO32
C***** ***** SNO33
C***** ***** SNO34
C***** ***** SNO35
C***** ***** SNO36
C***** ***** SNO37
C***** ***** SNO38
C***** ***** SNO39
C***** ***** SNO40
C***** ***** SNO41
C***** ***** SNO42
C***** ***** SNO43
C***** ***** SNO44
C***** ***** SNO45
C***** ***** SNO46
C***** ***** SNO47
C***** ***** MX001
C***** ***** MX002
C***** ***** MX003
C***** ***** MX004
C***** ***** MX005
C***** ***** MX006
C***** ***** MX007
C***** ***** MX008
C***** ***** MX009
C***** ***** MX010
C***** ***** MX011
C***** ***** MX012
C***** ***** MX013
C***** ***** MX014
C***** ***** MX015
C***** ***** MX016
C***** ***** MX017
C***** ***** MX018
C***** ***** MX019
C***** ***** MX020
C***** ***** MX021
C***** ***** MX022
C***** ***** MX023
C***** ***** MX024
C***** ***** MX025
C
SUBROUTINE SOLN(NB,NC,ND,NE,K,N1,EST,VARK,ILOC,GOX,CK,PROB,YCUT
# ,NZCUT,BB)
REAL GOX(NE),CK(ND),PROB(NB),BB(NC),YCUT(NB)
INTEGER ILOC(NE)
COMMON /H/ HHA
C
J = K+1
VAR1 = 0.
HKV = 0.
IF(K.LE.1) FACT = 1.
IF(K.GE.2) FACT = FACT*FLOAT(K)
C
IF(K.GT.0) GOTO 10
C
C
----- for K=0; BB(I) = 1/N1 and GOX(I) = 1.0 -----
DO 5 I=1,N1
VAR1 = VAR1 + 1.0/FLOAT(N1)
CALL H1(J,ILOC(I))
5 HKV = HKV + HHA/FLOAT(N1)
GOTO 30
C
C
----- for K>0; BB(I) from matrix solution -----
10 DO 15 I=1,N1
CALL H1(J,ILOC(I))
HKV = HKV + BB(I)*HHA
15 VAR1= VAR1+ BB(I)*GOX(I)
C
IF(NZCUT.EQ.0) GOTO 25
DO 20 M=1,NZCUT
20 PROB(M) = PROB(M) + HK(K-1,YCUT(M))*HKV/FACT
C
25 VARK = VARK + CK(J)*CK(J)*FACT*VAR1
30 EST = EST + CK(J)*HKV
C
RETURN
END
C
C***** SURROUNTING MATRIX: CREATES THE DISJUNCTIVE KRIGING MATRIX ***** MX001
C***** EQUATION. ***** MX002
C***** ***** MX003
C***** ***** MX004
C***** ***** MX005
C***** ***** MX006
C***** ***** MX007
C***** ***** MX008
C***** ***** MX009
C***** ***** MX010
C***** ***** MX011
C***** ***** MX012
C***** ***** MX013
C***** ***** MX014
C***** ***** MX015
C***** ***** MX016
C***** ***** MX017
C***** ***** MX018
C***** ***** MX019
C***** ***** MX020
C***** ***** MX021
C***** ***** MX022
C***** ***** MX023
C***** ***** MX024
C***** ***** MX025
C
SURROUNTING MATRIX(NA,NC,NE,NZ,N1,N0B,X0,Y0,XD,YD,ILOC,RLOC,RMAX
# ,GOX,MCOL,NROW,MODE,C,IBLK,AA,IFLG)
DOUBLE PRECISION AA(NC,NC),DBLE
REAL XD(NA),YD(NA),RLOC(NE),C(3),GOX(NE)
INTEGER ILOC(NE)
C
----- locate NZ nearest neighbors -----
DO 1 J=1,NZ
ILOC(J) = 0
1 RLOC(J) = 9.9E5
C
DO 10 I=1,N0B
DX = XD(I)-X0
DY = YD(I)-Y0
R = SORT(DX*DX+DY*DY)
IF(R.GT.RMAX) GOTO 10

```

```

C
C      ----- check if R goes in list -----
C      ZMX = -9.9E5
C      DO 5 J=1,NZ
C          IF(ZMX.LT.RLOC(J)) K1 = J
C          IF(ZMX.LT.RLOC(J)) ZMX = RLOC(J)
C      5 CONTINUE
C
C          IF(R.GT.ZMX) GOTO 10
C          RLOC(K1) = R
C          ILOC(K1) = I
C      10 CONTINUE
C
C          N1=0
C          DO 15 I=1,MZ
C              IF(ILOC(I).LE.0) GOTO 15
C              N1 = N1 + 1
C      15 CONTINUE
C          IF(N1.EQ.0) IFLG=1
C          IF(IFLG.EQ.1) RETURN
C          NCOL = N1+1
C          NROW = N1
C
C          ----- fill in main diaognal and rhs of matrix -----
C          DO 25 I=1,N1
C              AA(I,I) = 1.0000
C
C          ----- block disjunctive kriging -----
C          IF(IBLK.EQ.0) GOTO 20
C          IF(1BLK.NF.0)AA(I,N1+1) =
C              * DBLE(BLKPTS(X0,Y0,XD(ILOC(I)),YD(ILOC(I)),C,MODE))
C          GOTO 25
C
C          20 R = RLOC(I)
C              AA(I,N1+1) = DBLE( VARIO(R,C,MODE) )
C          25 GOX(I) = AA(I,N1+1)
C
C          ----- fill in off-diagonals (calculate upper-half -----
C          ----- and assign to lower) -----
C          DO 35 I=1,N1-1
C          DO 30 J=I+1,N1
C              DX = XD(ILOC(J)) - XD(ILOC(I))
C              DY = YD(ILOC(J)) - YD(ILOC(I))
C              R = SQRT(DX*DX+DY*DY)
C
C              ----- upper half -----
C              AA(I,J) = DBLE( VARIO(R,C,MODE) )
C
C              ----- lower half -----
C              AA(J,I) = AA(I,J)
C          30 CONTINUE
C          35 CONTINUE
C
C          RETURN
C          END
C
C***** SUBROUTINE AVECOR: CALCULATES THE INNER BLOCK COVARIANCE *****
C***** (USE WITH PROGRAM TWO) *****
C
C      SUBROUTINE AVECOR(ND,BLKCOV,C,MODE,NHK,CK)
C      DIMENSION CK(ND),C(3)
C      INTEGER*4 ISEED
C      COMMON /BLK/ NX,NY,WIDX,WIDY,ISEED
C
C          SUM=0.0
C          DX = WIDX/NX
C          DY = WIDY/NY
C          XO = (DX-WIDX)/2.
C
C          DO 10 I=1,NX
C              XO = XO + DX

```

MX026
MX027
MX028
MX029
MX030
MX031
MX032
MX033
MX034
MX035
MX036
MX037
MX038
MX039
MX040
MX041
MX042
MX043
MX044
MX045
MX046
MX047
MX048
MX049
MX050
MX051
MX052
MX053
MX054
MX055
MX056
MX057
MX058
MX059
MX060
MX061
MX062
MX063
MX064
MX065
MX066
MX067
MX068
MX069
MX070
MX071
MX072
MX073
MX074
MX075
MX076
MX077
MX078
AV001
AV002
AV003
AV004
AV005
AV006
AV007
AV008
AV009
AV010
AV011
AV012
AV013
AV014
AV015
AV016
AV017
AV018
AV019
AV020

```

      Y0 = (DY-WIDY)/2.          AV021
      DO 10 J=1,NY              AV022
      Y0 = Y0 + DY              AV023
      X1 = (DX-WIDX)/2.         AV024
      DO 10 K=1,NX              AV025
      X1 = X1 + DX              AV026
      Y1 = (DY-WIDY)/2.         AV027
      DO 10 L=1,NY              AV028
      Y1 = Y1 + DY              AV029
      C
      C
      C      ----- move a random deviation from center of sub-block -----
      C      ----- random number assumed to be [0,1] -----
      X11 = X1 + DX*(.5-RAN(ISEED))   AV030
      Y11 = Y1 + DY*(.5-RAN(ISEED))   AV031
      R = SQRT((X0-X11)*(X0-X11) + (Y0-Y11)*(Y0-Y11))   AV032
      SUM = SUM + VARIO(R,C,MODE)     AV033
      CNT = CNT + 1.0                AV034
10   CONTINUE                      AV035
      BLKCOV = SUM/CNT               AV036
      C
      FACT = 1.0                     AV037
      SUM = 0.0                       AV038
      DO 20 I=1,NHK                 AV039
      IF(I.GE.2) FACT = FLOAT(I)*FACT   AV040
      SUM = SUM + FACT*CK(I+1)*CK(I+1)*BLKCOV**FLOAT(I)   AV041
20   CONTINUE                      AV042
      BLKCOV = SUM                     AV043
      RETURN                         AV044
      END                           AV045
      C
      C***** SUBROUTINE BLKPTS: CALCULATES THE BLOCK VARIANCE BETWEEN
      C***** SAMPLE POINTS AND BLOCK
      C***** (USE WITH PROGRAM TWO)
      C
      REAL FUNCTION BLKPTS(XB,YB,X,Y,C,MODE)           BL001
      DIMENSION C(3)                                     BL002
      INTEGER*4 ISEED                                    BL003
      COMMON /BLK/ NX,NY,WIDX,WIDY,ISEED               BL004
      C
      SUM=0.0                                         BL005
      CNT=0.0                                         BL006
      DX = WIDX/NX                                    BL007
      DY = WIDY/NY                                    BL008
      C
      XO = XB + (DX-WIDX)/2.                          BL009
      DO 5 I=1,NX                                     BL010
      YO = YB + (DY-WIDY)/2.                          BL011
      DO 10 J=1,NY                                    BL012
      C
      C      ----- move a random deviation from center of sub-block -----
      C      ----- random number assumed to be [0,1] -----
      XOO = XO + DX*(.5-RAN(ISEED))     BL013
      YOO = YO + DY*(.5-RAN(ISEED))     BL014
      R = SQRT((XOO-X)*(XOO-X) + (YOO-Y)*(YOO-Y))   BL015
      SUM = SUM + VARIO(R,C,MODE)       BL016
      CNT = CNT + 1.0                     BL017
10   YO = YO + DY                         BL018
      5 XO = XO + DX                         BL019
      BLKPTS = SUM/CNT                      BL020
      RETURN                                BL021
      END                                    BL022
      C
      C***** SUBROUTINE ATOK: RAISES THE "AA" MATRIX TO THE K/(K-1)
      C***** POWER SO THAT THE "AA" MATRIX NEED NOT
      C***** BE RECOMPUTED FOR EACH K
      C

```

```

***** AT007
C***** (USE WITH PROGRAM TWO) ***** AT008
C***** ***** AT009
C***** ***** AT010
C***** SUBROUTINE ATOK(NC,NE,K,NROW,NCOL,GOX,AA) ***** AT011
C***** REAL GOX(NE) ***** AT012
C***** DOUBLE PRECISION AA(NC,NC),KK,DBLE ***** AT013
C***** ***** AT014
C***** KK = DBLE(FLOAT(K)/FLOAT(K-1)) ***** AT015
C***** DO 10 I=1,NROW ***** AT016
C***** DO 20 J=1,NCOL ***** AT017
C***** IF(AA(I,J).EQ.1.0D+00) GOTO 20 ***** AT018
C***** IF(AA(I,J).LT.1.0D-10) GOTO 20 ***** AT019
C***** AA(I,J) = AA(I,J)**KK ***** AT020
C***** 20 CONTINUE ***** AT021
C***** GOX(I) = SNGL(AA(I,NCOL)) ***** AT022
C***** 10 CONTINUE ***** AT023
C***** RETURN ***** AT024
C***** END ***** AT025
C***** VA001
C***** ***** VA002
C***** ***** VA003
C***** FUNCTION VARIO: RETURNS SILL - VARIOGRAM TO CALLING PROGRAM ***** VA004
C***** ***** VA005
C***** (USE WITH PROGRAM TWO) ***** VA006
C***** ***** VA007
C***** VA008
C***** REAL FUNCTION VARIO(R,C,MODE) ***** VA009
C***** DIMENSION C(3) ***** VA010
C***** ***** VA011
C***** IF(R.GT.0.0) GOTO 1 ***** VA012
C***** VARIO = C(1) + C(2) ***** VA013
C***** RETURN ***** VA014
C***** 1 GOTO(10,20,30), MODE+2 ***** VA015
C***** ***** VA016
C***** ----- exponential variogram model ----- ***** VA017
C***** 10 VARIO = C(1) + C(2)*(1.0 - EXP(-R/C(3))) ***** VA018
C***** GOTO 1000 ***** VA019
C***** ***** VA020
C***** ----- linear variogram model (with sill) ----- ***** VA021
C***** 20 IF(C(3).LE.0.0) GOTO 21 ***** VA022
C***** IF(R.GE.C(3)) GOTO 22 ***** VA023
C***** 21 VARIO = C(1) + C(2)*R/C(3) ***** VA024
C***** GOTO 1000 ***** VA025
C***** 22 VARIO = C(1) + C(2) ***** VA026
C***** GOTO 1000 ***** VA027
C***** ***** VA028
C***** ----- spherical variogram model ----- ***** VA029
C***** 30 IF(R.GE.C(3)) GOTO 31 ***** VA030
C***** TMP = R/C(3) ***** VA031
C***** VARIO = C(1) + C(2)*(1.5*TMP-.5*(TMP*TMP*TMP)) ***** VA032
C***** GOTO 1000 ***** VA033
C***** 31 VARIO = C(1) + C(2) ***** VA034
C***** ***** VA035
C***** ----- calculate the covariogram ----- ***** VA036
C***** 1000 VARIO = C(1) + C(2) - VARIO ***** VA037
C***** 800 FORMAT(A50) ***** VA038
C***** RETURN ***** VA039
C***** END ***** VA040
C***** ***** VA041
C***** HC001
C***** ***** HC002
C***** ***** HC003
C***** ***** SUBROUTINE HCALC: CALCULATES THE HERMITE POLYNOMIAL FOR ***** HC004
C***** ***** ORDERS 0 TO NHK FOR EACH NORMALIZED ***** HC005
C***** ***** DATA POINT. THIS SPEEDS UP COMPUTATIONS. ***** HC006
C***** ***** HC007
C***** (USE WITH PROGRAM TWO) ***** HC008
C***** ***** HC009
C***** ***** HC010
C***** SUBROUTINE HCALC(NA,NOR,NHK,YZ) ***** HC011
C***** DIMENSION YZ(NA) ***** HC012
C***** COMMON /H/ HHA ***** HC013

```

```

C          DO 1 J=1,NHK                         HC014
C          K = (J-1)                           HC015
C          DO 5 I=1,NOB                         HC016
C          IF(K.EQ.0) GOTO 10                   HC017
C          IF(K.EQ.1) GOTO 15                   HC018
C
C          FOR K > 1 CALCULATE THE HERMITE POLYNOMIAL   HC019
C          HHA = HK(K,YZ(I))                     HC020
C          GOTO 5                                HC021
C
C          FOR K=0 HERMITE POLYNOMIAL = 1.0           HC022
C          10 HHA = 1.0                          HC023
C          GOTO 5                                HC024
C
C          FOR K=1 HERMITE POLYNOMIAL = YZ(I)        HC025
C          15 HHA = YZ(I)                         HC026
C          5 CALL H(J,I)                         HC027
C          1 CONTINUE                            HC028
C
C          RETURN
C          END                                     HC029
C
C***** SUBROUTINE H: GIVES THE VALUE OF THE HERMITE POLYNOMIAL      HP001
C***** FOR ORDER K AND LOCATION (OR DATA) I.                         HP002
C***** (USE WITH PROGRAM TWO)                                         HP003
C
C          SUBROUTINE H(K,I)
C          COMMON /H/ HHA
C          VIRTUAL HH(20,220)
C
C          ----- store hermite polynomial in table -----
C          HH(K,I) = HHA
C          RETURN
C
C          ----- look up hermite polynomial in table -----
C          ENTRY H1(K,I)
C          20 HHA = HH(K,I)
C          RETURN
C          END
C
C***** SUBROUTINE MATINV -- SOLVES THE MATRIX EQUATION      MV001
C***** (USE WITH BOTH PROGRAMS)                                 MV002
C
C          SUBROUTINE MATINV(NROW,NCOL,NC,AA,BB)
C          DOUBLE PRECISION AA(NC,NC),AWRK(21,21),DBLE,SUM
C          REAL BB(NROW)
C          COMMON IO,IT
C
C          -----put aa into working matrix -----
C          DO 100 I=1,NROW
C          DO 101 J=1,NCOL
C          101 AWRK(I,J) = AA(I,J)
C          100 CONTINUE
C
C          ----- partial pivoting is not necessary -----
C          ----- since maximum value is in main -----
C          ----- diagonal of kriging matrix -----
C          DO 1 I=2,NCNL
C          IF(AWRK(1,1).NE.0.0000) GOTO 1
C              WRITE(IT,800)
C              WRITE(IO,800)
C              STOP
C
C

```



```

C      ----- arbitrary K -----
DO 1 I=1,K-1
HKP1 = Y*HK - FLOAT(I)*HKM1
HKM1 = HK
HK   = HKP1
1 CONTINUE
5 RETURN
END
C

```

APPENDIX 2A

Example input data for DISCALC.FTN. Data file was created using "The Data File Input Instructions" from program documentation. For brevity, only the first and last ten entries of the I, X, Y, and Z data are reported

| BARE SOIL TEMPERATURE DATA -- EXAMPLE | |
|---------------------------------------|-------|
| 5 7 7 5 5 23.000 100.000 | DA001 |
| 0 0 0 0.000 0.000 0.000 | DA002 |
| 3 3 2.000 0.000 16.000 100.000 | DA003 |
| 62.50 64.00 65.0000 66.00 67.00 | DA004 |
| 1 0.700 2.400 23.000 | DA005 |
| 1 2 3 4 | DA006 |
| TEMP | DA007 |
| (15.2F10.3,F10.5) | DA008 |
| 108 17.000 200.000 60.95000 | DA009 |
| 94 33.000 243.000 61.07000 | DA010 |
| 75 27.000 198.000 61.19000 | DA011 |
| 95 33.000 181.000 61.35000 | DA012 |
| 100 34.000 188.000 61.44000 | DA013 |
| 87 31.000 37.000 61.45000 | DA014 |
| 99 33.000 147.000 61.65000 | DA015 |
| 104 17.000 93.000 61.71000 | DA016 |
| 60 23.000 176.000 61.78000 | DA017 |
| 96 33.000 94.000 61.81000 | DA018 |
| . | DA019 |
| . | DA020 |
| . | DA021 |
| . | DA022 |
| . | DA023 |
| . | DA024 |
| . | DA025 |
| 5 2.000 198.000 66.47000 | DA026 |
| 20 7.000 149.000 66.74000 | DA027 |
| 34 13.000 183.000 67.05000 | DA028 |
| 7 3.000 206.000 67.33000 | DA029 |
| 15 4.000 187.000 67.43000 | DA030 |
| 27 11.000 172.000 67.43000 | DA031 |
| 12 4.000 55.000 67.48000 | DA032 |
| 17 5.000 149.000 67.61000 | DA033 |
| 3 1.000 190.000 67.83000 | DA034 |
| 18 5.000 252.000 68.25000 | DA035 |

APPENDIX 2B

Example of an output file from DISCALC.FTN. This file is used, without modification, as an input file for DISJUNC.FTN. For brevity, only the first and last ten entries of the I, X, Y, Z, and YZ data are reported

| BARE SOIL TEMPERATURE DATA -- EXAMPLE | |
|---|-------|
| 5 7 5 23.000 100.000 | DB001 |
| 0 0 0 0.000 0.000 0.000 | DB002 |
| 3 3 2.000 0.000 16.000 100.000 | DB003 |
| 62.500000 64.000000 65.000000 66.000000 67.000000 | DB004 |
| -0.682487 0.178525 0.674979 1.127083 1.545975 | DB005 |
| 6.3890934E+01 1.7091980E+00 1.8315849E-01 -6.8178676E-02 -4.4367205E-02 | DB006 |
| -3.6764962E-03 -5.8487770E-03 | DB007 |
| 1 0.700 2.400 23.000 | DB008 |
| TEMP | DB009 |
| 108 17.000 200.000 60.95000 -2.89787889 | DB010 |
| 94 33.000 243.000 61.07000 -2.81422019 | DB011 |
| | DB012 |
| | DB013 |
| | DB014 |

| | | | | | |
|-----|--------|---------|----------|-------------|-------|
| 75 | 27.000 | 198.000 | 61.19000 | -2.69484258 | DB015 |
| 95 | 33.000 | 181.000 | 61.35000 | -2.21075654 | DB016 |
| 100 | 34.000 | 188.000 | 61.44000 | -1.76944888 | DB017 |
| 87 | 31.000 | 37.000 | 61.45000 | -1.74350667 | DB018 |
| 99 | 33.000 | 147.000 | 61.65000 | -1.40478790 | DB019 |
| 104 | 17.000 | 93.000 | 61.71000 | -1.33265126 | DB020 |
| 60 | 23.000 | 176.000 | 61.78000 | -1.25641036 | DB021 |
| 96 | 33.000 | 94.000 | 61.81000 | -1.22578073 | DB022 |
| . | . | . | . | . | DB023 |
| . | . | . | . | . | DB024 |
| . | . | . | . | . | DB025 |
| . | . | . | . | . | DB026 |
| 5 | 2.000 | 198.000 | 66.47000 | 1.32675445 | DB027 |
| 20 | 7.000 | 149.000 | 66.74000 | 1.43894422 | DB028 |
| 34 | 13.000 | 183.000 | 67.05000 | 1.56650829 | DB029 |
| 7 | 3.000 | 206.000 | 67.33000 | 1.68162668 | DB030 |
| 27 | 11.000 | 172.000 | 67.43000 | 1.72296810 | DB031 |
| 15 | 4.000 | 187.000 | 67.43000 | 1.72296810 | DB032 |
| 12 | 4.000 | 55.000 | 67.48000 | 1.74372768 | DB033 |
| 17 | 5.000 | 149.000 | 67.61000 | 1.79804289 | DB034 |
| 3 | 1.000 | 190.000 | 67.83000 | 1.89167035 | DB035 |
| 18 | 5.000 | 252.000 | 68.25000 | 2.08196902 | DB036 |

APPENDIX 2C

Example input file for DISJUNC.FTN. This file was created using the data file given in Appendix 2B (note: only a part of the complete file is given in Appendix 2B). The data in this file were used to obtain the estimates at the points: (2, 200), (18, 200) and (34, 200) in Figure 2

| BARE SOIL TEMPERATURE DATA -- EXAMPLE | | | | | |
|---------------------------------------|----------------|---------------|----------------|----------------|-------|
| 5 | 7 | 5 | 23.000 | 100.000 | DC001 |
| 0 | 0 | 0 | 0.000 | 0.000 | DC002 |
| 3 | 1 | 2.000 | 200.000 | 16.000 | DC003 |
| 62.500000 | 64.000000 | 65.000000 | 66.000000 | 67.000000 | DC004 |
| -0.682487 | 0.178525 | 0.674979 | 1.127083 | 1.545975 | DC005 |
| 6.3890934E+01 | 1.7091980E+00 | 1.8315849E-01 | -6.8178676E-02 | -4.4367205E-02 | DC006 |
| -3.6764962E-03 | -5.8487770E-03 | | | | DC007 |
| 1 | 0.700 | 2.400 | 23.000 | | DC008 |
| TEMP | | | | | DC009 |
| 108 | 17.000 | 200.000 | 60.95000 | -2.89787889 | DC010 |
| 75 | 27.000 | 198.000 | 61.19000 | -2.69484258 | DC011 |
| 91 | 32.000 | 201.000 | 62.43000 | -0.72949600 | DC012 |
| 72 | 26.000 | 203.000 | 62.91000 | -0.42498398 | DC013 |
| 54 | 20.000 | 207.000 | 63.10000 | -0.31316853 | DC014 |
| 71 | 26.000 | 201.000 | 63.60000 | -0.03369232 | DC015 |
| 64 | 24.000 | 199.000 | 63.65000 | -0.00667833 | DC016 |
| 56 | 21.000 | 199.000 | 64.59000 | 0.47700810 | DC017 |
| 19 | 6.000 | 188.000 | 65.54000 | 0.92427117 | DC018 |
| 5 | 2.000 | 198.000 | 66.47000 | 1.32675445 | DC019 |
| 7 | 3.000 | 206.000 | 67.33000 | 1.68162668 | DC020 |
| 15 | 4.000 | 187.000 | 67.43000 | 1.72296810 | DC021 |
| 3 | 1.000 | 190.000 | 67.83000 | 1.89167035 | DC022 |
| | | | | | DC023 |
| | | | | | DC024 |
| | | | | | DC025 |

APPENDIX 2D

Example of output from DISJUNC.FTN

```
*****
DISJUNCTIVE KRIGING ON A GRID MATRIX
FOR ESTIMATING SPATIALLY DEPENDENT PROPERTIES
POINT ESTIMATION
BARE SOIL TEMPERATURE DATA -- EXAMPLE
*****
```

| | | | | | |
|---|---|---|---|---|-------|
| • | • | • | • | • | D0001 |
| • | • | • | • | • | D0002 |
| • | • | • | • | • | D0003 |
| • | • | • | • | • | D0004 |
| • | • | • | • | • | D0005 |
| • | • | • | • | • | D0006 |
| • | • | • | • | • | D0007 |
| • | • | • | • | • | D0008 |
| • | • | • | • | • | D0009 |
| • | • | • | • | • | D0010 |
| • | • | • | • | • | D0011 |
| • | • | • | • | • | D0012 |
| • | • | • | • | • | D0013 |
| • | • | • | • | • | D0014 |

INPUT PARAMETERS

NUMBER OF TEMP 'S INPUT..... 13
MAXIMUM NUMBER OF NEAREST NEIGHBORS..... 5
MAXIMUM RADIUS..... 23.0000
MAXIMUM ALLOWED DATA VALUE..... 100.0000
CORRELATION FUNCTION MODEL TYPE..... SPHERICAL
NUGGET..... 0.2258
SILL MINUS NUGGET..... 0.7742
RANGE..... 23.0000

OO015
OO016
OO017
OO018
OO019
OO020
OO021
OO022
OO023
OO024
OO025
OO026
OO027
OO028
OO029
OO030
OO031
OO032
OO033
OO034
OO035
OO036
OO037
OO038
OO039
OO040
OO041
OO042
OO043
OO044
OO045
OO046
OO047
OO048
OO049
OO050
OO051
OO052
OO053
OO054
OO055
OO056
OO057
OO058
OO059
OO060
OO061
OO062
OO063
OO064
OO065
OO066
OO067
OO068
OO069
OO070
OO071
OO072

OBSERVED DATA

80. NO. X Y TEMP YZ OBS. NO. X Y TEMP YZ
102 17.000 200.000 60.950 -2.897879 56 21.000 199.000 64.590 0.477008
75 27.000 198.000 61.190 -2.694843 19 6.000 188.000 65.540 0.924271
31 32.000 201.000 62.430 -0.729496 5 2.000 198.000 66.470 1.326754
72 26.000 203.000 62.910 -0.424984 7 3.000 206.000 67.330 1.681627
54 29.000 207.000 63.100 -0.313169 15 4.000 187.000 67.430 1.722968
71 26.000 201.000 63.600 -0.033692 3 1.000 190.000 67.830 1.891670
61 24.000 199.000 63.650 -0.006678 0 0.000 0.000 0.000 0.000000

DATA SET:
MEAN = 63.89491 MEAN = 63.89093
VARIANCE = 3.07619 VARIANCE = 3.08984

HERMITE POLYNOMIAL COEFFICIENTS:

k Ck k Ck
0 6.399093E+01 4 -4.436721E-02
1 1.709198E+00 5 -3.676496E-03
2 1.831585E-01 6 -5.848777E-03
3 -6.817868E-02 7 0.000000E-01

DISJUNCTIVE KRIGING ESTIMATES PROBABILITY THAT THE ESTIMATE IS GREATER THAN ZCUT

Value(s) Zcut
(6.250)(6.400)(6.500)(6.600)(6.700)
(E+01)(E+01)(E+01)(E+01)(E+01)
I X Y EST VARK +-----+-----+-----+-----+-----+
5 2.00 200.00 66.270 1.574 1.000 0.999 0.777 0.500 0.268
5 18.00 200.00 62.187 1.316 0.335 0.246 0.081 0.001 0.000
5 34.00 200.00 62.615 1.754 0.494 0.122 0.020 0.000 0.000

* Values when entire data set used in calculation