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. Vaucelin and others (1983) used ordinary kriging and co-kriging to estimate the spatial distribution of the available water content and sand content of a Tunisian field. They determined that co-kriging 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 \( X_1, X_2, \ldots, X_n \) is the conditional expectation of \( Y \) given \( X_1, X_2, \ldots, X_n \). For the most general situation, computing this conditional expectation requires knowledge of the joint density of \( Y, X_1, X_2, \ldots, X_n \), although this information is difficult to obtain in practice. This nonlinear estimator has the form

\[
Y^* = g(X_1, X_2, \ldots, X_n)
\]

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

\[
Y_{DK}^* = \sum_{i=1}^{n} f_i(X_i)
\]

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

\[
Y_{DK}^* = \sum_{i=1}^{n} \lambda_i X_i
\]

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: $x_1, x_2, \ldots, x_p$ 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

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

(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 $f_a$'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 $Z(x_i)$ through a transform. The transform relationship, $\phi(x)$, also is expressed as a series of Hermite polynomials (Abramowitz and Stegun, 1965)

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

(5)

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

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

(6)

where $J$ is the total number of abscissas and weights, $v_j$ 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)

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

(7)

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

$$H_k^*[Y(x_o)] = \sum_{i=1}^{n} b_a H_k[Y(x_i)]$$

(8)

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

$$\sum_{i=1}^{n} b_a(q_{aj}) = (q_{aj})^k; \quad j = 1, 2, \ldots, n$$

(9)

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

$$\bar{q}_{aj} = \frac{1}{V} \int_{V} q(x - x_j) dx$$

(10)

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

The disjunctive kriging variance on the estimation is

$$\sigma_{DK}^2 = \sum_{k=0}^{K} k! C_k^2 \left[1 - \sum_{i=1}^{n} b_a(q_{aj})^k\right]$$

(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 $z$, (or the associated transformed cutoff value, $y_z$). 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 \( \text{(Journel and Huijbregts, 1978: Yates, Warrick, and Myers, 1985a)} \)

\[
P^* [Y(x)] = 1 - G(y) + g(y) 
\times \sum_{k=1}^{K} H_k \cdot g(y) 
\times H_k [Y(x)] / k! 
\]

(12)

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

The conditional probability density function, \( P_{df^*} (u) \), is determined by taking the derivative of Equation (12) with respect to \( y_i \).

\[
P_{df^*} (u) = \frac{g(u)}{1 + \sum_{k=1}^{K} H_k (u) \frac{H_k [Y(x)]}{k!}} 
\]

(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 \( H_k [Y(x)] \) values for \( k = 1, 2, \ldots, 20 \) and \( i = 1, 2, \ldots, 220 \). 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 \( b_a \)’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, \( \phi(Y) \) 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, \( Y, \) is obtained by inverting the probability function (Function PRBI). The estimate of the probability used in DISCALC.FTN is

\[
P[Z \leq z] \approx (i - 0.5)/p \] (14)

where \( i \) is the total number of \( Z(x) \) less than or equal to \( z \) and \( p \) is the total number of samples.

(b) Once the pairs \([Z(x), Y(x)]\) have been determined, the values of \( \phi(Y) \) 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 CKI. 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 \( \phi(Y(x)) \) relationship which represents an approximation in terms of transforming \( Y(x) \) into \( Z(x) \). Therefore, to make the inversion exact, new values of \( Y(x) \) 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 $k = 0, 1, \ldots, K$ 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 $Y = YMIN$ and incrementing by DX and DY, $(NX) \times (NY)$ 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, $b_{kn}$, in Equation (9), the AA matrix is raised to the 4th 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 Kolomogorov-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

$$g(x_i - x_j) = C(x_i - x_j)/C(0).$$

The coefficients, $C_k$, used to define the transform given in Equation (5) for $k = 0$ to $K$ are:

$$1.709, 0.1832, -0.06761, -0.04437, -0.09837, 0.1218, 0.116, 0.0938, 0.084."

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 x 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).
Disjunctive kriging program for two dimensions

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 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


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.


APPENDIX I

Disjunctive Kriging Programs DISCALC.FTN and DISJUNC.FTN

C*****************************************************************************C
C DISJUNCTIVE KRIGING PROGRAMS FOR TWO-DIMENSIONAL, SPATIALLY-VARIABLE PROPERTIES
C*****************************************************************************C

C Solves the disjunctive kriging equations and calculates an estimate of a random variable, the associated estimation variance and the conditional probability (for up to 18 cutoff levels). In order to use these programs, values of a random variable of interest along with the spatial coordinates must be obtained. Also, the correlation structure written in terms of the variogram must be determined.
C
C The programs use the following devices:
C
C Disk input file -- currently set as unit #1
C Disk output file -- currently set as unit #2
C Disk plot file -- currently set as unit #3
C Terminal -- currently set as unit #5
C Line printer -- currently set as unit #6
C
C These values are declared at the beginning of each program.
C
C To reduce problem execution time and storage requirements, two programs are used for disjunctive kriging. The first program calculates the Hermite coefficients (i.e. CK's) and then converts the data (ZD) and cutoff values (ZCUT) into transformed data (YZ) and transformed cutoff values (YCUT). Approaching the solution in this manner has the advantage that the coefficients and transformed values only need to be calculated once for a given data set. To run the programs, a data file must be...
Disjunctive kriging program for two dimensions

created according to the "Data File Input Instructions" given below.
The "Interactive Data Instructions For Program One" describes the run-time
information the program requires such as instructions for choosing
input/output devices and file names.

The second program uses the output from the first program directly. No
modification on the data file is necessary. The program asks for the
name of the file interactively, therefore there aren't any additional
"Data File Input Instructions" for the second program. To run the
program use the "Interactive Data Instructions For Program Two" given
below.

**********************************************************************
** DATA FILE INPUT INSTRUCTIONS **
**********************************************************************

RECORDS 1-3:  FORMAT (A76/A76/A76)

TITLE(3)  Three lines of title.

RECORD 4:  FORMAT(515,ZFIO.3)

NZ  Maximum number of data to be included in
calculating an estimate using disjunctive
kriging. That is, the number of nearest
neighbors used in estimation process.
Default: (i.e. zero or blank) is NZ = 7.

NHK  Number of terms in the Hermite expansion.
There will be NHK Hermite coefficients
calculated by the program.
Default: NHK = 7.

NPOLY  Number of terms to be used in the least
squares fit to the [ZD,YZ] pairs. The
least squares polynomial is used only for
calculating the abscissa points for Hermite
integration (to determine the CK's).
Default: NPOLY = 7.

NTRM  Number of terms to be used in Hermite
integration. NTRM may be 5 or 10.
Default: NTRM = 5.

NZCUT  Number of conditional probabilities
to be calculated using disjunctive kriging.
For each conditional probability calculated
a ZCUT value must be provided. The ZCUT
values, if any, are input in RECORD 7.
NZCUT must be less than 19.

RMAX  Maximum radial search distance for a data
point to be included in the estimation
process.
Default: RMAX = 10000. If linear model
for the spatial correlation function is
used Default: RMAX = RANGE (see Record 8).
The program DISJUNC.FTN will not allow a
value of RMAX > RANGE for linear model.

CUTOFF  Data values larger than CUTOFF will not
be included in the analysis.
Default: CUTOFF = 10000.
RECORD 5:  FORMAT(315,3F10.3)

IBLK    If: 0 -- Punctual disjunctive kriging
        If: 1 -- Block disjunctive kriging

NXBLK   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.

NYBLK   Number of block subdivisions in the Y
direction. If IBLK equals 1 then
Default: NYBLK = 5.

WIDX    Length of the block in the X direction.
        Total block area is WIDX*WIDY.

WIDY    Length of the block in the Y direction.

BLKCOV  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.

RECORD 6:  FORMAT(215,4F10.3)

NX      Number of X coordinates in the grid system
of estimates. There will be NX*NY estimates

NY      Number of Y coordinates in the grid system
of estimates. There will be NX*NY estimates
total. Default: NY = 10.

XMIN    The minimum value of X on the grid system.
        Default: XMIN = 0.

YMIN    The minimum value of Y on the grid system.
        Default: YMIN = 0.

DX      The distance between estimates in the X
direction on the grid system. The length
of the grid system in the X direction
is (NX-1)*DX.
        Default: DX = (Max. X in data file )/9.

DY      The distance between estimates in the Y
direction on the grid system. The length
of the grid system in the Y direction is
(NY-1)*DY.
        Default: DY = (Max. Y in data file )/9.

ZCUT(I)  The cutoff values (in terms of the original
variable. A conditional probability will
be calculated for each cutoff value. A
maximum of 18 values are allowed.
Disjunctive kriging program for two dimensions

RECORD 8:  FORMAT(15,F10.3)

MODE
The variogram model number where:

-1 -- exponential:
  \[ C(1) + C(2) \times \exp(-R/C(3)) \]

0 -- linear (with a sill):
  \[ C(1) + C(2) \times R/C(3) \]

1 -- spherical:
  \[ C(1) + C(2) \left[ 1.5 \times R/C(3) - \frac{5}{3} \times (R/C(3))^3 \right] \]

The correlation function is calculated by

\[ \text{correlation} = 1 - \frac{\text{variogram}}{\text{sill}} \]

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.

C(I) NUGGET value of the variogram.

C(2) SILL value minus the NUGGET value of the variogram.

C(3) RANGE of the variogram. For the exponential variogram this is the constant that the distance is divided by.

RECORD 9:  FORMAT(315)

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.

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.

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.

RECORD 10:  FORMAT(4X,16)

NAM A name which describes the property to be kriged.

RECORD 11:  FORMAT(A30)

FMT Format specification for the data. This should be of the form:
RECORDS 12-:

**FORMAT IS GIVEN BY** FMT (RECORD 11) **(15,2F10.3,5X,F10.3)**

**DATA VALUES**

The data set should have a location number (or index), X coordinates, Y coordinates and the property to be kriged. Data is read until an end-of-file marker is found. Therefore, there should not be any blank lines in this part of the data file. It is assumed that the location number is in the first column.

**FILE Input file name. File name must contain less than 31 characters. This is the name of the data file created using "DATA FILE INPUT INSTRUCTIONS" given above.**

**FILE Output file name. File name must contain less than 31 characters. This file is the input file for DISJUNC.FTN. Only those data and steering parameters needed by the program DISJUNC.FTN are printed out to this file. This file is used "as is" by the program "DISJUNC.FTN".**

**Output device number where the output/summary will be sent. Two choices are allowed:**

- **P** -- output sent to line printer
- **T** -- output sent to terminal

**FILE Input file name. File name must contain less than 31 characters. This is the output file created using DISCALC.FTN.**

**Output device designation. Choices allowed:**

- **D** - disk file
- **T** - terminal
- **P** - line printer
Disjunctive kriging program for two dimensions

OPTIONAL RECORD 2b: (include only if ANSW.EQ.'O' in RECORD 2a)

FILE
Output file name. File name must contain less than 31 characters.

RECORD 3: FILE
Plot file name. Produces a file that can be used to make contour maps, etc. The file contains x, y, estimates, estimation variance and conditional probabilities. File name must contain less than 31 characters. If a plot file is NOT wanted type a return.

******************************* MISCELLANEOUS INFORMATION ********************

IMPORTANT ARRAYS: Contains the following information:

INK(I) - Index or sample number for the Ith data record.
XD(I) - X coordinate datum for the Ith data record.
YD(I) - Y coordinate datum for the Ith data record.
ZD(I) - Property of interest that is to be estimated.
PR(I) - Probability of the Ith datum.
CK(J) - Coefficients of the Hermite expansion for the transform.
YCUT(K) - Cutoff values in terms of the sampled variable (i.e. ZD).
YZCUT(K) - Cutoff values in terms of the transform var. (i.e. YZ).
PROB(K) - Conditional probability for cutoff level number K.
CS(L) - Coefficients for the least squares relationship.
C(M) - Variogram model coefficients.
AA(N+1,N+1) - Coefficient matrix of the kriging equations.
BB(N+1) - Right hand side of equations is in AA(i,N+1), i=1,...,N+1
ILOC(N) - The location numbers for the nearest neighbors.
PLOC(N) - The radial distance between each nearest neighbor and the estimation site.
GOX(N) - The correlation between each nearest neighbor and the estimation site.

WHERE:
I = 1,2,...,MINO(number of samples, 220)
J = 1,2,...,MINO(number of Hermite polynomials, 25)
K = 1,2,...,MINO(number of cutoff levels, 18)
L = 1,2,...,MINO(number of least squares polynomials, 20)
M = 1,2,3
N = 1,2,...,MINO(number of nearest neighbors, 20)

STEPS NECESSARY TO USE DISJUNCTIVE KRIGING PROGRAMS:
1) Obtain the data set. Must include an X, Y and Z (property of interest)
2) Determine the variogram. (Not included in these programs)
3) Create data file using "Data File Input Instructions"
4) Run DISCALC.FTN
5) Answer questions prompted by program using "Interactive data instructions for program one". Remember that output from this program is input for DISJUNC.FTN
6) Run DISJUNC.FTN
7) Answer questions prompted by program using "Interactive data instructions for program two". Remember to use the output from DISCALC.FTN as input to this program.
PROGRAM ONE: DISCALC.FTN

PURPOSE:
Calculates Hermite coefficients (i.e. Ck's).
Converts the sample data (ZD) into transformed data (YZ).
Converts cutoff values (ZCUT) into transformed cutoff values (YCUT).
Writes out results into a user specified file ready for DISJU~IC.FTN.
Prints out a summary of results to a user specified device.

DEVICES REQUIRED:
Unit #1 -- input disk device (See "data file input instructions")
Unit #2 -- output disk device (See "interactive input instructions")
Unit #5 -- terminal
Unit #6 -- line printer (Required only if user specifies summary
of results to be sent to line printer. This is an interactive input).

SUBROUTINE AND FUNCTION CALLS:
1) MATINV 2) SORT 3) CKI 4) LEAST
5) HYZINV & HYZIM (entry) 6) PHI 7) PRB
8) PRBI 9) HK

WRITTEN BY:
Scott R. Yates, Computer Sciences Corp.

DOUBLE PRECISION AA(21,21)
CHARACTER TITLE(3)*76,FMT*30,RI*10,NAM*f,ANSW*I
REAL PENTER,XD(220),YD(220),ZD(220),YZ(220),PR(220),CK(25)
# ,C(3),ATMP(4),ZCUT(18),YCUT(18),B@201),CS(20)
INTEGER INX(220)
COMMON IO,IT
DATA NA/220/,NB/18/,NC/21/,ND/25/,NE/20/
..... Machine specific device numbers ......
..... IN is disk input, IO is disk output ......
..... IT is the terminal, IL is the printer ......
IN = 1
IO = 2
IT = 5
IL = 6

----- open files on PDP 11/70 -----  
WRITE(IT,701)' GIVE INPUT FILE NAME >>> '
READ(IT,700) FMT
OPEN(UNIT=IN,FILE=FMT,STATUS='OLD',READONLY)
WRITE(IT,701)' GIVE OUTPUT FILE NAME (Input for kriging) >>> '
READ(IT,700) FMT
OPEN(UNIT=IO,FILE=FMT,STATUS='UNKNOWN',CARRIAGECONTROL='LIST')
 1 WRITE(IT,850)
  READ(IT,855) ANSW

IF(ANSW.EQ.'P') THEN
  KOUT=IL
ELSE IF(ANSW.EQ.'T') THEN
  KOUT=IT
ELSE
  WRITE(IT,865)
  GOTO 1
END IF
Disjunctive kriging program for two dimensions

700 FORMAT(A50)
701 FORMAT(/,,A50.,$)

800 FORMAT(A76)
905 FORMAT(4(I4,6I1))
910 FORMAT(5(I6,F10.3))
915 FORMAT(3(I6,F10.3))
820 FORMAT(2(I6,F10.3))
875 FORMAT(15,F10.3)
830 FORMAT(8(F10.6))
935 FORMAT(A30)
86 FORM (IP5E15.7)
845 FORMAT(15,2(F10.3,F10.5,F10.5))

850 FORMAT(/,' GIVE OUTPUT DEVICE (FOR THE OUTPUT/SUMMARY) ',/,'P --- send to line printer','/,'T --- send to terminal','/,'T50
#,' >>> ','$)
855 FORMAT(AI)
860 FORMAT(/,' ERROR: Array overflow. Execution will cease. ','#"," must be less than or equal to ',13,'.',///)
865 FORMAT(/,' ERROR: NOT AN ALLOWED INPUT. (Try again) ',/)
970 FORMAT(F10.4)

C

----- read input parameters ----- READ(IN,800) (TITLE(I),I=1,3) READ(IN,810) NZ,NHK,NPOLY,NTRM,RMAX,CUTOFF READ(IN,815) IBLK,NXRLK,NYRLK,WIDX,WIDY,BLKCOV READ(IN,820) NX,NY,XRN,YMI RN,IDX,OY

----- default parameters (steering) ----- IF(NZ.LE.0) NZ = 7 IF(NHK.LE.0) NHK = 7 IF(NPOLY.LE.0) NPOLY = 7 IF(NTRM.LE.0) NTRM = 5 IF(RMAX.LE.0) RMAX = 10000. IF(CUTOFF.LE.0) CUTOFF = 10000.

C IF(NZCUT.GT.NR.OR.NZCUT.LT.0) GOTO 10 IF(NZCUT.GT.0) READ(IN,930) (NCUT(I),I=1,NZCUT)

C READ(IN,925) MODE,(C(I),I=1,3) READ(IN,915) IXO,IYO,IZD READ(IN,805) NAM READ(IN,835) FMT

C
I=0 N=10 MEAN=0. XMAX=-99999. YMAX=-99999.

5 READ(IN,FMT,END=10) II,(ATMP(M),M=1,3) IF(ATMP(1).GT.CUTOFF) GOTO 5 
II=1 IN(I)= II YO(I) = ATMP(I) YD(I) = ATMP(I) ZD(I) = ATMP(I) MEAN = MEAN + ZD(I) XMAX = MAX(XMAX,YMAX,YD(I)) YMAX = MAX(YMAX,YMAX,YD(I))

C

----- check for array overflow ----- IF(II.LE.NA) GO TO 5 WRITE(IT,960) ' NOB ',NA ?M
!

IF(FLAG=1) IF(NHK.GT.ND) IF(NHK.GT.NP) WRITE(IT,960) ' NHK ',ND IF(NPOLY.GT.NP) IF(FLAG-1)
IF(NPOLY.GT.NE) WRITE(IT,860) 'NPOLY',IFLAG
IF(NZCUT.GT.NB) IFLAG=1
IF(NZCUT.GT.NR) WRITE(IT,460) 'NZCUT',NB
IF(IFLAG.GT.0) STOP

----- calculate the sample mean and variance ----- 
NOB=1
MEAN = MEAN/FLOAT(NOB)
VAR = 0.
DO 13 I=1,NOB
VAR = VAR + (ZD(I)-MEAN)*(ZD(I)-MEAN)
VAR = VAR/FLOAT(NOB)
13

----- determine the Hermite coefficients ----- 
CALL SORT(NA,NOB,INX,XD,YD,ZD,YZ,PR)
CALL LEAST(NOB,NA,NC,NE,ZD,PR,AA,NPOLY,CS)
CALL CK1(NE,NTRM,NHK,CK,NPOLY,CS)

----- invert PHI(Y) relationship for the data ----- 
CALL HYZINV(NOB,INX,ZD,YZ,NHK,CK)

----- invert PHI(Y) relationship for the Ycut values ----- 
IF(NZCUT.GE.0) GOTO 20
DO 15 I=1,NZCUT
YCUT(I) = (ZCUT(I)-MEAN)/VAR
CALL HYZINI(NZCUT,INX,ZCUT,YCUT,NHK,CK)
15

----- calculate the sample variance from the coefficients ----- 
SUM = CK(2)*CK(2)
FACT = 1.
DO 25 I=3,NHK
FACT = FACT*FLOAT(I-I)
SUM = SUM + FACT*CK(1)*CK(1)
25

----- default parameters for matrix ----- 
IF(DX.LE.0) DX = XMAX/9.
IF(DY.LE.0) NY = YMAX/9.
IF(NX.LE.0) NX = 10
IF(NY.LE.0) NY = 10
IF(IBLK.EQ.1.AND.NXBLK.LE.0) NXBLK=5
IF(IBLK.EQ.1.AND.NYBLK.LE.0) NYBLK=5

----- write data file for disjunctive kriging ----- 
WRITE(IO,800) (TITLE(I),I=1,3)
WRITE(IO,815) NZ,NHK,NZCUT,RMAX,CUTOFF
WRITE(IO,815) IBLK,NXBLK,NYBLK,WIDX,WIDY,BLKCOV
WRITE(IO,820) NX,NY,XMIN,YMIN,DX,DY
IF(NZCUT.NE.0) WRITE(IO,830) (YCUT(I),I=1,NZCUT)
IF(NZCUT.NE.0) WRITE(IO,830) (ZCUT(I),I=1,NZCUT)
WRITE(IO,840) (CK(I),I=1,NHK)
WRITE(IO,850) MODE,(C(I),I=1,3)
WRITE(IO,855) NAM

----- print out summary of calculation ----- 
WRITE(KOUT,900)
DO 35 I=1,3
WRITE(KOUT,905) TITLE(I)
35

NLIN = 31
WRITE(KOUT,915) NAM,NOB,NZ,NTRM,RMAX,CUTOFF
IF(IBLK.GT.0) THEN
Disjunctive kriging program for two dimensions

BI = 'CALCULATE'
IF(BLKCV.GT.O.) WRITE(BI,870) BLKCV
WRITE(KOUT,920) BI,NXBLK,NYBLK,WIDX,WIDY
NLIN = NLIN+4
ELSE
ENDIF

C
IF(MODE.LT.O) BI = 'EXPONENT'
IF(MODE.EQ.O) BI = 'LINEAR'
IF(MODE.GT.O) BI = 'SPHERICAL'
WRITE(KOUT,925) BI,C(1),C(2),C(3)
WRITE(KOUT,930) NAM,NAM

..... print data .....
DO 40 I=I,N0~/2+1
NLIN = NLIN + 1
IF(NLIN.GE.58) WRITE(KOUT,BS5) 'I'
IF(NLIN.GE.58) NLIN = 0
II = NOB/2 + I + I
IF(INX(II).EQ.O) INX(II) = II
IF(INX(II).EQ.O) INX(II) = 0
40 WRITE(KOUT,935) INX(I),XD(1),YD(1),ZD(1),YZ(I),INX(II),XD(II),YD(II),ZD(II),YZ(II)

NLIN = NLIN + 7
IF(NLIN.GE.53) WRITE(KOUT,955) 'I'
IF(NLIN.GE.53) NLIN = 0
WRITE(KOUT,940) 'c.AN,CK(1),VAR,SUM

..... orint coefficients .....
[TMP-MAXO(NHK,NPOLY)
NLIN = NLIN + 7
IF(NLIN.GE.53) WRITE(KOUT,955) 'I'
IF(NLIN.GE.53) NLIN = 0
WRITE(KOUT,940) 'c.AN,CK(1),VAR,SUM

..... end of problem .....
900 FORMAT(IHI,IOX,82(1H*)/IIX,IH*,80X,IH*/IIX,IH*,22X,'DISJUNCTIVE KRIGING PROGRAM: ONE ',22X,IH* ,/,IIX,IH*,22X,' FOR DETERMINING THE COEFFICIENTS ',22X,IH*/IIX,1H*,80X,IH*)
905 FORMAT(IIX,1H*,2X,A76,2X,IH*)
910 FORMAT(//IIX,'INPUT PARAMETERS'/11X,16(IH=)/,11X,'MAXIMUM NUMBER OF NEAREST NEIGHBORS .............. ',110,/,11X,'NUMBER OF TERMS USED IN HERMITE INTEGRATION ...... ',110,/,11X,'MAXIMUM RADIUS ................................... ',F10.4,/,11X,'MAXIMUM ALLOWED DATA VALUE ................... ',F10.4,/,11X,'DISPERSION COVARIANCE WITHIN BLOCK ............... ',At0,/,11X,'NUMBER OF ',A6,'"s INPUT ......................... ',110,/,11X,'MAXIMUM NUMBER OF NEAREST NEIGHBORS .............. ',110,/,11X,'NUMBER OF TERMS USED IN HERMITE INTEGRATION ...... ',110,/,11X,'MAXIMUM RADIUS ................................... ',F10.4,/,11X,'MAXIMUM ALLOWED DATA VALUE ................... ',F10.4,/,11X,'DISPERSION COVARIANCE WITHIN BLOCK ............... ',At0,/,11X,'NUMBER OF ',A6,'"s INPUT ......................... ',110,/,211X,'MAXIMUM NUMBER OF NEAREST NEIGHBORS .............. ',110,/,311X,'MAXIMUM NUMBER OF ',A6,'ighbors/CELLS',34(IH.),5X,'NX = ',',110,5X,'NY = ',',110,/,311X,'BLOCK SIZE',5X,34(IH.),5X,'DX = ',',110,5X,'DY = ',',110,/,411X,'MAXIMUM RADIUS ..... F10.4,/,511X,'MAXIMUM ALLOWED DATA VALUE ....... F10.4/,111X,'DISPERSION COVARIANCE WITHIN BLOCK ............... ',At0,/,211X,'NUMBER OF CELLS',34(IH.),5X,'NX = ',',110,5X,'NY = ',',110,/,311X,'BLOCK SIZE',5X,34(IH.),5X,'DX = ',',110,5X,'DY = ',',110,/,411X,'MAXIMUM RADIUS ..... F10.4,/,511X,'MAXIMUM ALLOWED DATA VALUE ....... F10.4/,111X,'DISPERSION COVARIANCE WITHIN BLOCK ............... ',At0,/,211X,'NUMBER OF CELLS',34(IH.),5X,'NX = ',',110,5X,'NY = ',',110,/,311X,'BLOCK SIZE',5X,34(IH.),5X,'DX = ',',110,5X,'DY = ',',110,/,411X,'MAXIMUM RADIUS ..... F10.4,/,511X,'MAXIMUM ALLOWED DATA VALUE ....... F10.4,/)
S. R. YateS, A. W. Warrick, and D. E. Myers

945 FORMAT(/,11X,'COEFFICIENTS:',/,'LEAST SQUARES:',/,'HERMITE POLYNOMIALS:',/,'(cum. dist-1i)')
950 FORMAT(11X,F14.6,11X,13(IH:),//,11X,'LEAST SQUARES:',/,'HERMITE POLYNOMIALS:',/,11X,'CUM. DISTRIBUTION')
955 FORMAT(11X,F14.6)
960 FORMAT(11X,F14.6)

END

*******************************************************************************
*** PURPOSE: ***
** Locates the NZ nearest neighbors to be used in the estimation process. **
** Calculates the spatial correlations and constructs coefficient matrix **
** Solves matrix equations for disjunctive kriging weights. **
** Calculates estimates, error variances and conditional probabilities **
** Writes out results into a user specified device. **
** Creates a plot file if specified by user. **

** DEVICES REQUIRED: **
** Unit #1 -- input file disk device (Output from DISCALC.FTN) **
** Unit #2 -- output file disk device (Used only if disk output specified) **
** Unit #3 -- plot file disk device (Used only if plot file specified) **
** Unit #5 -- terminal **
** Unit #6 -- line printer (Required only if user specifies summary **
** of results to be sent to line printer. This is an interactive input). **

** SUBROUTINE AND FUNCTION CALLS:**
  1) SOLN  2) MATRX  3) ATOK  4) MATINV
  5) AVECGR  6) BLKPTS  7) VARIN  8) H & H1
  9) H & H1 (entry point) 10) HK  11) PRB

** WRITTEN BY:**
  Scott R. Yates, Computer Sciences Corp.

*******************************************************************************
Disjunctive kriging program for two dimensions

OPEN files on PDP 11/70 ------
WRITE(IT,701)' GIVE INPUT FILE NAME
READ(IT,700) FMT
OPEN(UNIT=IN,FILE=FMT,STATUS='OLD',READONLY)
WRITE(IT,703)
READ(IT,702) ANSW

IF(ANSW.EQ.'T' .OR. ANSW.EQ.'t') THEN
  IO=IT
ELSE IF(ANSW.EQ.'P' .OR. ANSW.EQ.'p') THEN
  IO=IT
ELSE IF(ANSW.EQ.'D' .OR. ANSW.EQ.'d') THEN
  WRITE(IT,701)' GIVE OUTPUT FILE NAME
  READ(IT,700) FMT
  OPEN(UNIT=IO,FILE=FMT,STATUS='UNKNOWN')
  IO=ID
ELSE
  WRITE(IT,*)' INCORRECT VALUE GIVEN -- MUST BE [I], [P] OR [D]'
  GOTO 1
END IF

WRITE(IT,701)* GIVE PLOT FILE NAME (return=none)
READ(IT,700) FMT
IF(FMT.EQ.' ') GOTO 5
OPEN(UNIT=IP,FILE=FMT,STATUS='UNKNOWN')

700 FORMAT(A30)
701 FORMAT(//,A50,=)
702 FORMAT(A1)
703 FORMAT(///,' GIVE THE OUTPUT DEVICE: ',/T20,
  # ' D -- create a disk file ',/T20,' P -- send to line printer ' ,/T20,
  # ' T -- send to terminal',/T46,'>>> ')
800 FORMAT(A76)
805 FORMAT(315,F10.3)
810 FORMAT(215,F10.3)
815 FORMAT(8F10.3)
820 FORMAT(IPS15.8)
825 FORMAT(15,F10.3)
830 FORMAT(4X,A6)
835 FORMAT(15,F10.3)
840 FORMAT(///,' ERROR: array overflow -- ',A5,' is greater than ',/T13,'/)
845 FORMAT(F6.3)

----- read steering parameters -----
5 READ(IN,800) (TITLE(I),I=1,3)
READ(IN,805) NZ,NHK,NZCUT,CUTOFF
READ(IN,805) IBLK,NXRLK,NYBLK,WIDX,WIDY,BLKCOV
READ(IN,810) NX,NY,XMIN,YMIN,DX,DY
IF(NZCUT.KF.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,NHK)
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

----- transform variogram into the correlation function -----
CVARK = C(1)+C(2)
C(1)=C(1)/CVARK
C(2)=C(2)/CVARK

----- read data from file -----

10 READ(IN,835,END=25) INX(I),YN(I),ZD(I),YD(I)

----- open files on PDP 11/70 ------
WRITE(IT,701)' GIVE INPUT FILE NAME
READ(IT,700) FMT
OPEN(UNIT=IN,FILE=FMT,STATUS='OLD',READONLY)
WRITE(IT,703)
READ(IT,702) ANSW

C

5 READ(IN,800) (TITLE(I),I=1,3)
C

----- read steering parameters -----

5 READ(IN,800) (TITLE(I),I=1,3)
C

----- transform variogram into the correlation function -----
CVARK = C(1)+C(2)
C(1)=C(1)/CVARK
C(2)=C(2)/CVARK

----- read data from file -----

10 READ(IN,835,END=25) INX(I),YN(I),ZD(I),YD(I)
IF(ZD(1).GT.CUTOFF) GOTO 20
MEAN = MEAN + ZD(1)
GO TO 15
25 NOB=I-1
C
----- check for array overflow ----- 
IFLAG=0
IF(NOB.GT.NA) IFLAG=1
IF(NHK.GT.ND) WRITE(IT,840) ' NOB ',NA
IF(NHK.GT.ND) WRITE(IT,840) ' NHK ',ND
IF(NZ.GT.NE) WRITE(IT,840) ' NZ ',NE
IF(NZCUT.GT.NB) WRITE(IT,B40)'NZCUT',NB
IF(IFLAG.EQ.1) STOP

..... calculate the sample statistics ----- 
MEAN = MEAN/FLOAT(NOB)
VAR = 0.
27 DO 29 I=1,NOB
VAR = VAR + (ZD(I)-MEAN)*(ZD(I)-MEAN)
VAR = VAR/FLOAT(NOB)
SUM = CK(2)*CK(2)
FACT = 1.
30 DO 32 I=3,NHK
FACT = FACT*FLOAT(I-1)
SUM = SUM + FACT*CK(I)*CK(I)

..... calculate inner block covariance. Function SECNDS is ----- 
..... a system routine which returns the number of seconds ----- 
..... from start of day and used to "randomize" the seed ----- 
IF(IBLK.EQ.0) BLKCOV = SUM
IF(IBLK.EQ.0) GOTO 35
ISEED = I-alpha(SECNDS(0,0))
IF(BLKCOV.GT.O.O) GOTO 35
CALL AVECOR(ND,BLKCOV,C,MOOE,NHK,CK)
CONTINUE

..... write out summary of input data ----- 
WRITE(IO,900)
WRITE(IO,905)
WRITE(IO,910)
DO 40 I=1,3
WRITE(IO,915) TITLE(I)
WRITE(IO,920)
NLIN = 31
WRITE(IO,925) NAM,NOB,NZ,RMAX,CUTOFF
WRITE(IO,930) BLKCOV,MRBLK,NYBLK,WIDX,WIDY
WRITE(IO,935) MI(MODE+2),C(1),C(2),C(3)
WRITE(IO,940) NAM,NPM

..... print out data ----- 
DO 45 I=1,NOB/2+1
NLIN = NLIN + 1
IF(NLIN.GE.58) WRITE(IO,702) 'I'
IF(NLIN.GE.58) NLIN = 0
II = NOB/2 + I + 1
INE(I).EQ.0) INX(I) = I
IF(INX(I).EQ.0) INX(I) = I
II = I+1
IF(INX(I).GT.NOB) INX(I) = 1
45 WRITE(IO,945) INX(I),XD(I),YD(I),ZD(I),YZ(I),INX(I),XD(I)
WRITE(IO,950) MEAN,CK(1),VAR,SUM

C
Disjunctive kriging program for two dimensions

----- print out coefficients -----  
NLIN = NLIN + 5  
IF(NLIN.GE.50) WRITE(IO,702) '1'  
IF(NLIN.GE.50) NLIN = 0  
WRITE(IO,955)  
DO 50 I=I,NHK/2+I  
II = NHK/2 + I + 1  
K = I-1  
KK : I I-1  
NLIN = NLIN + 1  
WRITE(IO,960) K,CK(1),KK,CK(II)  
50 CONTINUE  

----- print out coefficients -----  
NLIN = NLIN + 7
IF(NLIN.GE.58) WRITE(IO,702) '1'  
IF(NLIN.GE.58) NLIN = 0  
RI = ' Zcut '  
DO 55 I=I,NZCUT  
WRITE(AZCUT,820) ZCUT(1)  
READ(AZCUT(I:II),845) ZCUT(1)  
AZEXP(1) = AZCUT(12:15)  
CONTINUE  
IV=MINO(g,NZCUT)  
JV=MAXO(NZCUT-IV,1)  
KV:MAXO(I,(IV*8-16)/2)  
LV:MAXO(I,(IV*8-16)/2)  
WRITE(IO,q65) (BI,I:I,MINO(I,NZCUT))  
WRITE(IO,970) (BI,I:I,MINO(I,NZCUT))  
WRITE(IO,980)  
IF(IV.GT.O) WRITE(IO,985)  
C
C******** REGIN DISJUNCTIVE KRIGING -- ON A GRID ********
C
C ..... create matrix of Hermite polynomials values ......

CALL HCALC(NA,NOP,NHK,YZ)  

XO = XMIN  
DO 60 I=1,NX  
YO = YMIN  

DO 65 J=1,NY  
EST = 0.0  
VARK = 0.0  
IFLG= 0  

DO 70 M=1,NZCUT  
PROB(M) = 0.0  
70 CONTINUE  

CALL MATRX(NA,NC,NE,NZ,NI,NOB,XO,YO,XD,YD,ILOC,RLOC,RMAX,GOX, &
NCOL,NROW,MODE,C,IBLK,AA,IFLG)  
IF(IFLG.EQ.I) GOTO 65  

----- sum series on k -----  
DO 85 IK=I,NHK  
K = (IK-1)  
IF(K.EQ.0) GOTO 80  
IF(K.EQ.1) GOTO 75  
CALL ATOK(NC,NE,K,NROW,NCOL,GOX,AA)  

75 CALL MATINV(NROW,NCOL,GOX,AA)  
80 CALL SOLN(NB,NC,ND,NE,K,NI,EST,VARK,LOC,GOX,CK,PROB,YCUT, &
NZONE,BB)  
85 CONTINUE  

----- calculate the conditional probability -----  
----- note: because the Hermite series is truncated -----  
----- neg. or values greater than 1 are possible -----  

S. R. Yates, A. W. Warrick, and D. E. Myers

these values are set to either 0 or 1

\[ \text{IF}(\text{NZCUT} > 0.0) \text{ GOTO 95} \]

\[ \text{DO 90 } M = 1, \text{NZCUT} \]

\[ \text{TP} = \text{YCUT}(M) \]

\[ \text{PRB}(M) = 1.0 - \text{EXP}(-\text{TP}^2/2) \]

\[ \text{IF}(\text{PRB}(M) > 1.0) \text{ PRB}(M) = 1.0 \]

\[ \text{IF}(\text{PRB}(M) < 0.0) \text{ RPROB}(M) = 0.0 \]

\[ \text{VARK} = \text{BKCOV} - \text{VARK} \]

\[ \text{write out results} \]

\[ \text{IF}(\text{NLIN} \geq 58) \text{ WRITE}(\text{IO},702) '1' \]

\[ \text{IF}(\text{NLIN} \geq 58) \text{ NLIN} = 0 \]

\[ \text{NLIN} = \text{NLIN} + 1 \]

\[ \text{IF}(\text{IP} > 3) \text{WRITE}(\text{IP},990) \text{ XO},\text{YO},\text{EST}, \text{VARK},\{\text{PROB}(M), M = 1, \text{NZCUT}\} \]

\[ \text{WRITE}(\text{IO},995) \text{ NL},\text{XO},\text{YO},\text{EST}, \text{VARK},\{\text{PROB}(M), M = 1, \text{NZCUT}\} \]

\[ \text{YO} = \text{YO} + \text{DY} \]

\[ \text{XO} = \text{XO} + \text{DX} \]

\[ \text{STOP} \]

\[ \text{end of problem} \]

\[ \text{END} \]
Disjunctive kriging program for two dimensions

SUBROUTINE SORT: Sorts ZD(I) data into increasing order

SUBROUTINE SORT(NA, NOB, INX, XD, YD, ZD, YZ, PR)

DIMENSION INX(NA), XD(NA), YD(NA), ZD(NA), YZ(NA), PR(NA)

DO 10 I = 1, NOB
  XMIN = 99999999.
  DO 15 J = I, NOB
    IF(XMIN .LT. ZD(J)) GOTO 15
    XMIN = ZD(J)
  15 CONTINUE
  TZ = ZD(I)
  TX = XD(I)
  TY = YD(I)
  LOC = INX(I)
  ZD(I) = ZD(IJ)
  XD(I) = XD(IJ)
  YD(I) = YD(IJ)
  INX(I) = INX(IJ)
  ZD(IJ) = TZ
  XD(IJ) = TX
  YD(IJ) = TY
  INX(IJ) = LOC
  IJ = J
10 CONTINUE

LCNT = 0
DO 20 I = 1, NOB
  IF(ZD(I) .NE. ZD(I-1) .AND. LCNT .EQ. 0) GOTO 35
  IF(ZD(I) .NE. ZD(I-1) .AND. LCNT .NE. 0) GOTO 25
25 TZ = (FLOAT(I) - .5) / FLOAT(NOB)
   LCNT = LCNT + 1
   GOTO 20
20 CONTINUE

LCNT = LCNT + 1
PR(I) = PRBI(1.0 - PR(I))
YZ(I) = PRBI(1.0 - PR(I))

RETURN
END

SUBROUTINE CKI: Calculated the coefficients "CK" by numerical Hermite integration

SUBROUTINE CKI(NE, NTRM, NHK, CK, NPOLY, CS)

DIMENSION W(15), YW(15), CK(NHK), CS(NE)
VALUES FOR INTEGRATION:

YW(1) & W(1); I=1,2,...,10 are for 10 term

YW(1) & W(1); I=11,12,...,15 are for 5 term

DATA

\[ \begin{align*}
N(1)/& 1045340708E+00/,
YW(1)/& 1.47636813E+00/, \\
W(1)/& 0.476366813E+00/, \\
W(2)/& 0.737473729E+01/, \\
W(3)/& 0.376261601E+00/, \\
W(4)/& 0.11153918E-01/, \\
W(5)/& 0.22547400E+01/, \\
W(6)/& 0.412310259E-01/, \\
W(7)/& 0.278880899E+01/, \\
W(8)/& 0.123407622E+01/, \\
W(9)/& 0.411635918E+01/, \\
W(10)/& 0.538476404E+01/, \\
W(11)/& 0.34290133/, \\
W(12)/& 0.0366108/, \\
W(13)/& 0.3487960574/,
\end{align*} \]

DATA

\[ \begin{align*}
SQ2PI/2.50662829/, \\
\text{IF}(\text{NTRM}.EQ.10) N(1) = 10, \\
\text{IF}(\text{NTRM}.EQ.10) NTRM = 0, \\
\text{IF}(\text{NTRM}.EQ.5) NTRM = 10,
\end{align*} \]

FACT = 1.

DO 5 I=1,NHK
K = (I-1)
CK(I) = 0.
IF(K.GE.2) FACT = FACT*FLOAT(K)
DO 10 J=1+NTRM,NJ
YM = -YW(J)
YP = YM(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)
CONTINUE
CK(I) = CK(I)/(FACT*SQ2PI)
CONTINUE
RETURN
END

SUBROUTINE LEAST: CALCULATES THE LEAST SQUARES REGRESSION
LINE OF THE FORM:
\[ Y = C1 + C2*X + C3*X*X + \ldots \]

DOUBLE PRECISION AA, DBLE
DIMENSION ZD(NA),PR(NA),CS(NE),AA(NC,NC)
COMMON 10, IT

NROW = NPOLY
NCOL = NROW + 1
DO 1 L=1,NROW
AA(L,NCOL) = 0.000
1 CONTINUE
DO 2 M=1,NROW
AA(M,L) = 0.000
2 CONTINUE
DO 10 I=1,NOB
10 AA(L,M) = AA(L,M) + DBLE(PR(I)**FLOAT(L-1)*PR(I)**FLOAT(M-1))
AA(M,L) = AA(L,M)
5 CONTINUE
DO 15 I=1,NOB
15 AA(L,NCOL) = AA(L,NCOL) + DBLE(PR(I)**FLOAT(L-1)*ZD(I))
1 CONTINUE
Disjunctive kriging program for two dimensions

--- solve the matrix ---
CALL MATINV(NROW,NCOL,NC,AA,CS)
RETURN
END

********************************************************************************

SUBROUTINE HYZINV: CALCULATES THE NORMALIZED YZ FROM THE Z
DATA BY INVERTING THE EQUATION
(Z = C(0) + C(1)HK(1,Y) + C(2)HK(2,Y) + ..... C(N)HK(N,Y)
(USE WITH PROGRAM ONE)

SUBROUTINE HYZINV( NOB, INX, ZD, YZ, NHK, CK)
DIMENSION CK(NHK), INX(NOB), ZD(NOB), YZ(NOB)
CHARACTER*6 NAME
DATA TOL/I.0E-6/, NIT/ZO/ COMMON IO,IT
NAME = ' DATA '
DO 10 I=1,NOB
YMI = YZ(I) -.005*YZ(I) -.0005
Y = YZ(I)*.005*YZ(I) + .0005
N = 0
C
C begin secant approximation to .......
C Newton's iterative method ......
FM1 = 0.0
F = 0.0
DO 20 J=1,NHK
K = (J-1)
FM1 = FM1 + CK(J)*HK(K,YMI)
20 F = F + CK(J)*HK(K,Y)
FM1 = FM1-ZD(I)
F = F-ZD(I)
SLOPE = (F-FM1)/(Y-YMI)
IF(ABS(SLOPE).LT.1.0E-10) GOTO 25
IF(ABS(SLOPE).GT.1.0E+10) GOTO 25
YPI = Y - F/SLOPE
YMI = Y
Y = YPI
N = N+1
IF(ABS(Y).GT.10.0) GOTO 25
IF(ABS(Y-YMI).LT.1.0E-6) GOTO 25
GOTO 15
C
C solution diverging -- try alternate method ----
IFLAG=0
N=0
Y = YZ(I)
YL = Y -.1*ABS(Y) -.1
YR = Y + .1*ABS(Y) + .1
DO 35 J=1,NHK
K=J-1
35 Z1 = Z1 + CK(J)*HK(K,Y)
IF(ABS(Z1-ZD(I)).LT.1.00001) GOTO 45
IF(N.GT.50) GOTO 40
GOTO 30

C
IF(Z1.GT.ZD(I)) YR=Y
IF(Z1.LT.ZD(I)) YL=Y
N=N+1
Y=(YR+YL)/2.
GOTO 30
C       ----- solution did'nt converge -----  HY065
40  IF(IFLAG.EQ.0) GOTO 50 HY066
     YL = -3. HY067
     YR = 3. HY068
     Y = 0. HY069
     N = 0 HY070
     IFLAG = IFLAG + 1 HY071
     IF(IFLAG.EQ.1) GOTO 30 HY072
     YL = -4. HY073
     YR = 4. HY074
     Y = .51434556 HY075
     GOTO 30 HY076
C       ----- solution converged -----  HY077
45  IF(ARS(Y).GT.10.0) GOTO 25 HY078
     YZ(I) = Y HY079
     GOTO 10 HY080
C       ----- solution did not converge -----  HY081
50  II = INX(I) HY082
     IF(NAME.EQ.0. ' ZCUT ') II = I HY083
     WRITE(IT,800) II,NAME,YZ(I),Y,YL,YR,ZD(1),Z1 HY084
10  CONTINUE HY085
     RETURN HY086
C       ----- entry for inversion of ZCUT -----  HY087
ENTRY HYZINI(NOB,INX,ZD,YZ,NHK,CK) HY088
     NAME = ' ZCUT ' HY089
     GOTO 5 HY090
C     800  FORMAT(' SOLUTION DID NOT CONVERGE FOR ',A6,' ILOC " ',I5,5X HY091
#/, ' YZ ' = 'F12.5,3X,' Y = 'F12.5,3X,' YL' = 'F12.5,3X, HY092
#/' YU ' = 'F12.5,3X,' ZD ' = 'F12.5,3X,' Z' = 'F12.5,3//)) HY093
     END HY094
C       ****************************************************** PH001
C       ********** FUNCTION PHI: RETURNS THE ZD VALUE FROM AN Nth ORDER ********** PH002
C       ********** POLYNOMIAL FIT TO THE NORMALIZED YZ VALUES ********** PH003
C       (USE WITH PROGRAM ONE) ********** PH004
C     REAL FUNCTION PHI(NE,Y,NPOLY,CS) PH005
     DIMENSION CS(NE) PH006
     C     P = PRB(Y) PH007
     SUM = CS(1) PH008
     DO 1 I=2,NPOLY PH009
          SUM = SUM + CS(I)*P**FLOAT(I-1) PH010
1          PHI = SUM PH011
     RETURN PH012
     END PH013
C       ****************************************************** P1001
C       ********** FUNCTION PRBI: CALCULATES THE INVERSE OF THE PROBABILITY ********** P1002
C       ********** FUNCTION GIVES THE X ASSOCIATED WITH p=pD. ********** P1003
C       (USE WITH PROGRAM ONE) ********** P1004
C     REAL FUNCTION PRBI(X) P1005
     T = AMIN(X,1.-X) P1006
     T = SORT(-2.*ALOG(T+1.3E-20)) P1007
     PRBI = T-((.010328*T+.002852)*T+2.515517)/ P1008
&     ((.001308*T+.19269)*T+.752788)*T+1. P1009
     IF(X.LT.0.5) RETURN P1010
     PRBI = -PRBI P1011
     RETURN P1012
     END P1013
Disjunctive kriging program for two dimensions

SUBROUTINE SOLN: CALCULATES THE DISJUNCTIVE KRIGING SOLUTION: EST, VARK AND PROB (USE WITH PROGRAM TWO)

SUBROUTINE SOLN(NB, NC, ND, NE, K, NI, 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

J = K+I
VARI = 0.
HKV = 0.
IF(K.LE.1) FACT = 1.
IF(K.GE.2) FACT = FACT*FLOAT(K)

IF(K.GT.0) GOTO 10

----- for K=0; BB(I) = 1/N1 and GOX(I) = 1.0 ----- DO 5 I=1,NI
VARI = VAR1 + 1.0/FLOAT(N1)
CALL HI(J,ILOC(1))
HKV = HKV + HHA/FLOAT(N1)
GOTO 30

DO 5 I=I,NI

5 VARI = I.O/FLOAT(NI)

----- for K>0; BB(I) from matrix solution ----- DO 10 M=I,NZ
PROB(M) = PROB(M) + HK(K-I,YCUT(M))*HKV/FACT

VARK = VARK + CK(J)*CK(J)*FACT*VARI
30 EST = EST + CK(J)*HKV
RETURN

END

SUBROUTINE MATRX: CREATES THE DISJUNCTIVE KRIGING MATRIX EQUATION (USE WITH PROGRAM TWO)

SUBROUTINE MATRX(NA, NC, NE, NZ, NI, NOB, XO, YO, XD, YD, RLOC, ILOC, RMAX, GOX, MCOL, NROW, MODE, C, IRLK, AA, IFLG)

DOUBLE PRECISION AA(NA, NC), DBLE
REAL XD(NA), YD(NA), RLOC(NE), C(3), GOX(NE)
INTEGER ILOC(NE)

----- locate NZ nearest neighbors ----- DO 1 J=1,NZ
ILOC(J) = 0
1 RLOC(J) = 9.9E5

DO 10 I=1,NOB

DX = XD(I)-XO
DY = YD(I)-YO
R = SQRT(DX*DX+DY*DY)
IF(R.GT.RMAX) GOTO 10

---
C ----- check if R goes in list -----  
ZMX = -9.9E5  
DO 5 J=1,NZ  
IF(ZMX.LT.RLOC(J)) KI = J  
IF(ZMX.LT.RLOC(J)) ZMX = RLOC(J)  
5 CONTINUE  
C  
IF(R.GT.ZMX) GOTO 10  
RLOC(KI) = R  
ILOC(KI) = I  
10 CONTINUE  
C  
N1 = 0  
DO 15 I=1,N  
IF(ILOC(I).LE.0) GOTO 15  
N1 = N1 + 1  
15 CONTINUE  
C  
C ----- fill in main diagonal and rhs of matrix -----  
DO 25 I=1,N1  
AA(I,I) = 1.0  
25 GO TO (AA(I,N1+1) = DBLE(VARIO(R,C,MODE)) )  
C  
C ----- block disjunctive kriging -----  
IF(IBLK.EQ.0) GOTO 20  
IF(IBLK.NE.0) AA(I,N1+1) = DBLE(BLKPTS(XO,YO,XD(ILOC(1)),YD(ILOC(1)),C,HOOE))  
GOTO 25  
C  
20 R = RLOC(1)  
AA(I,N1+1) = DBLE( VARIO(R,C,MODE) )  
25 GOX(I) = AA(I,N1+1)  
C  
C ----- fill in off-diagonals (calculate upper-half -----  
C ----- and assign to lower) -----  
DO 35 I=1,N1-1  
DO 30 J=I+1,N1  
DX = XD(ILOC(J)) - XD(ILOC(I))  
DY = YD(ILOC(J)) - YD(ILOC(I))  
R = SQRT(DX*DX+DY*DY)  
AA(I,J) = DBLE( VARIO(R,C,MODE) )  
30 CONTINUE  
35 CONTINUE  
C  
RETURN  
END  
C***********************************************************************  
C******* SUBROUTINE AVECOR: CALCULATES THE INNER BLOCK COVARIANCE  
C******* (USE WITH PROGRAM TWO)  
***********************************************************************  
C SUBROUTINE AVECOR(ND,BLKCov,C,MOOE,HOOE,CK)  
DIMENSION CK(ND),C(3)  
INTEGER*4 ISEED  
COMMON /BLK/ MX,NY,WIDX,WIDY,ISEED  
SUM=0.0  
DX = WIDX/NX  
DY = WIDY/NY  
XO = (DX-WIDX)/2.  
DO 10 I=1,NX  
XO = XO + DX  
10 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
Disjunctive kriging program for two dimensions

YO = (DY-WIDY)/2.
DO 10 J=1,NY
YO = YO + DY
XI = (DX-WIDX)/2.
DO 10 K=1,NX
XI = XI + DX
YL = (DY-WIDY)/2.
DO 10 L=1,NY
YL = YL + DY

----- move a random deviation from center of sub-block ----- 
----- random number assumed to be [0,1] ----- 
XI1 = XI + DX*(.5-RAN(ISEED))
YL1 = YL + DY*(.5-RAN(ISEED))
R = SQRT((XI-XI1)*(XI-XI1) + (YL-YL1)*(YL-YL1))
CNT = CNT + 1.0
10 CONTINUE
BLKCOV = SUM/CNT
FACT = 1.0
SUM = 0.0
DO 20 I=I,NHK
IF(I.GE.2) FACT = FLOAT(1)*FACT
SUM = SUM + FACT*CK(I+I)*CK(I+I)*BLKCOV**FLOAT(1)
20 CONTINUE
BLKCOV = SUM
RETURN
END

REAL FUNCTION BLKPTS(XB,YB,X,Y,C,MODE)
DIMENSION C(3)
INTEGER*4 ISEED
COMMON /BLK/ NX,NY,WIDX,WIDY,ISEED
SUM=0.0
CNT=0.0
DX = WIDX/NX
DY = WIDY/NY
XO = XB + (DX-WIDX)/2.
DO 5 J=1,NX
YO = YB + (DY-WIDY)/2.
5 CONTINUE
----- move a random deviation from center of sub-block ----- 
----- random number assumed to be [0,1] ----- 
XO0 = XO + DX*(.5-RAN(ISEED))
YO0 = YO + DY*(.5-RAN(ISEED))
R = SQRT((XO-XO0)*(XO-XO0) + (YO-YO0)*(YO-YO0))
SUM = SUM + VARIO(R,C,MODE)
CNT = CNT + 1.0
CONTINUE
BLKCOV = SUM
RETURN
END

C********************************************************************
C******* SUBROUTINE BLKPTS: CALCULATES THE BLOCK VARIANCE BETWEEN
C******* SAMPLE POINTS AND BLOCK
C******* (USE WITH PROGRAM TWO)
C********************************************************************

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********************************************************************
SUBROUTINE ATOK(NC,NE,K,NROW,NCOL,GOX,AA)
REAL GOX(NE)
DOUBLE PRECISION AA(NC,NC),KK,DBLE
KK = DBLE(FLOAT(K)/FLOAT(K-1))
DO 10 I=I,NROW
   DO 20 J=I,NCOL
      IF(AA(I,J).EQ.0.0D+00) GOTO 20
      IF(AA(I,J).LT.0.0D-00) GOTO 20
      AA(I,J) = AA(I,J)**KK
   CONTINUE
GOX(1) = SNGL(AA(I,NCOL))
CONTINUE
RETURN
END

SUBROUTINE HCALC: CALCULATES THE HERMITE POLYNOMIAL FOR
ORDERS 0 TO NHK FOR EACH NORMALIZED DATA POINT. THIS SPEEDS UP COMPUTATIONS.

SUBROUTINE HCALC(MA,NOB,NHK,YZ)
DIMENSION YZ(MA)
COMMON /H/ HHA

REAL FUNCTION VARIO(R,C,MODE)
DIMENSION C(3)
IF(R.GT.0.0) GOTO 1
VARIO = C(1) + C(2)
RETURN
1 GOTO(10,20,30), MODE+2

10 VARIO = C(1) + C(2)*(1.0 - EXP(-R/C(3)))
GOTO 1000

VARIO = C(1) + C(2)
GOTO 1000

VARIO = C(1) + C(2) + (1.5*TMP-.5*(TMP*TMP*TMP))
GOTO 1000

VARIO = C(1) + C(2)
GOTO 1000

VARIO = C(1) + C(2)
GOTO 1000

VARIO = C(1) + C(2)
GOTO 1000

VARIO = C(1) + C(2)
GOTO 1000

VARIO = C(1) + C(2)
GOTO 1000

VARIO = C(1) + C(2)
GOTO 1000

VARIO = C(1) + C(2)
GOTO 1000

VARIO = C(1) + C(2)
GOTO 1000

VARIO = C(1) + C(2)
GOTO 1000

VARIO = C(1) + C(2)
GOTO 1000

VARIO = C(1) + C(2)
GOTO 1000

VARIO = C(1) + C(2)
GOTO 1000

VARIO = C(1) + C(2)
GOTO 1000

VARIO = C(1) + C(2)
GOTO 1000

VARIO = C(1) + C(2)
GOTO 1000

VARIO = C(1) + C(2)
GOTO 1000

VARIO = C(1) + C(2)
GOTO 1000

VARIO = C(1) + C(2)
GOTO 1000

VARIO = C(1) + C(2)
GOTO 1000

VARIO = C(1) + C(2)
GOTO 1000

VARIO = C(1) + C(2)
GOTO 1000

VARIO = C(1) + C(2)
GOTO 1000

VARIO = C(1) + C(2)
GOTO 1000

VARIO = C(1) + C(2)
GOTO 1000

VARIO = C(1) + C(2)
GOTO 1000

VARIO = C(1) + C(2)
GOTO 1000

VARIO = C(1) + C(2)
GOTO 1000

VARIO = C(1) + C(2)
GOTO 1000

VARIO = C(1) + C(2)
GOTO 1000

VARIO = C(1) + C(2)
GOTO 1000

VARIO = C(1) + C(2)
GOTO 1000

VARIO = C(1) + C(2)
GOTO 1000

VARIO = C(1) + C(2)
GOTO 1000

VARIO = C(1) + C(2)
GOTO 1000

VARIO = C(1) + C(2)
GOTO 1000

VARIO = C(1) + C(2)
GOTO 1000

VARIO = C(1) + C(2)
GOTO 1000

VARIO = C(1) + C(2)
GOTO 1000

VARIO = C(1) + C(2)
GOTO 1000

VARIO = C(1) + C(2)
GOTO 1000

VARIO = C(1) + C(2)
GOTO 1000

VARIO = C(1) + C(2)
GOTO 1000

VARIO = C(1) + C(2)
GOTO 1000

VARIO = C(1) + C(2)
GOTO 1000

VARIO = C(1) + C(2)
GOTO 1000

VARIO = C(1) + C(2)
GOTO 1000

VARIO = C(1) + C(2)
GOTO 1000

VARIO = C(1) + C(2)
GOTO 1000

VARIO = C(1) + C(2)
GOTO 1000

VARIO = C(1) + C(2)
GOTO 1000

VARIO = C(1) + C(2)
GOTO 1000

VARIO = C(1) + C(2)
GOTO 1000

VARIO = C(1) + C(2)
GOTO 1000

VARIO = C(1) + C(2)
GOTO 1000

VARIO = C(1) + C(2)
GOTO 1000

VARIO = C(1) + C(2)
GOTO 1000

VARIO = C(1) + C(2)
GOTO 1000

VARIO = C(1) + C(2)
GOTO 1000

VARIO = C(1) + C(2)
GOTO 1000

VARIO = C(1) + C(2)
GOTO 1000

VARIO = C(1) + C(2)
GOTO 1000

VARIO = C(1) + C(2)
GOTO 1000

VARIO = C(1) + C(2)
GOTO 1000

VARIO = C(1) + C(2)
GOTO 1000

VARIO = C(1) + C(2)
GOTO 1000

VARIO = C(1) + C(2)
GOTO 1000

VARIO = C(1) + C(2)
GOTO 1000

VARIO = C(1) + C(2) - VARIO
800 FORMAT(A50)
RETURN
END

SUBROUTINE HCALC(MA,NOB,NHK,YZ)
DIMENSION YZ(MA)
COMMON /H/ HHA

SUBROUTINE HPCALC: CALCULATES THE HERMITE POLYNOMIAL FOR
ORDERS 0 TO NHK FOR EACH NORMALIZED DATA POINT. THIS SPEEDS UP COMPUTATIONS.
Disjunctive kriging program for two dimensions

DO 1 J=1,NHK
K = (J-1)
DO 5 I=1,NOB
IF(K.EQ.O) GOTO 10
IF(K.EQ.I) GOTO 15
FOR K > 1 CALCULATE THE HERMITE POLYNOMIAL
HHA = HK(K,YZ(I))
GOTO 5
C
FOR K=0 HERMITE POLYNOMIAL = 1.0
10 HHA = 1.0
GOTO 5
C
FOR K=1 HERMITE POLYNOMIAL = Y:
15 HHA = YZ(I)
CALL H(J,I)
CONTINUE
1 CONTINUE
RETURN
END
C

C********************** SUBROUTINE H: GIVES THE VALUE OF THE HERMITE POLYNOMIAL ********
C********************** FOR ORDER K AND LOCATION (OR DATA) I. ********
C********************** (USE WITH PROGRAM TWO) ********
C
SUBROUTINE H(K,I)
COMMON /H/ HHA
VIRTUAL HH(20,220)
..... store hermite polynomial in table ......
HH(K,I) = HHA
RETURN
..... look up hermite polynomial in table ......
ENTRY H(K,I)
20 HHA = HH(K,I)
RETURN
END
C

C********************** SUBROUTINE MATINV -- SOLVES THE MATRIX EQUATION ********
C********************** (USE WITH BOTH PROGRAMS) ********
C
SUBROUTINE MATINV(NROW,NCOL,NC,AA,BB)
DOUBLE PRECISION AA(NC,NC),AWRK(21,21),DBLE,SUM
REAL BB(NROW)
COMMON 10,IT
..... put aa into working matrix ......
DO 100 I=1,NROW
DO 101 J=1,NCOL
AWRK(I,J) = AA(I,J)
CONTINUE
100 CONTINUE
..... partial pivoting is not necessary ......
..... since maximum value is in main ......
..... diagonal of kriging matrix ......
DO 1 I=2,NCOL
IF(AWRK(I,I).NE.0.0000) GOTO 1
WRITE(IT,800)
WRITE(IO,800)
STOP
1 AWRK(1,1) = AWRK(1,1)/AWRK(1,1)
C
DO 5 L=2,NROW
DO 10 I=L,NROW
  SUM = 0.0000
C
DO 15 K=I,L-1
  SUM = SUM + AWRK(I,K)*AWRK(K,L)
15
10 AWRK(I,L) = AWRK(I,L)-SUM
C
DO 20 J=L+1,NCOL
  SUM = 0.0000
DO 25 K=I,L-1
  SUM = SUM + AWRK(L,K)*AWRK(K,J)
25
IF(AWRK(L,L).NE.0.0000) GOTO 20
20 AWRK(L,J) = (AWRK(L,J)-SUM)/AWRK(L,L)
5 CONTINUE
C
----- solution -----
BB(NROW) = SNGL(AWRK(NROW,NCOL))
DO 30 M=1,NROW-1
  I = NROW-M
  SUM = 0.0000
DO 35 J=I+1,NROW
  SUM = SUM + AWRK(I,J)*DBLE(BB(J))
35
30 BB(I) = SNGL(AWRK(I,NCOL) - SUM)
C
FORMAT(///' ERROR: Zero value in main diagonal of matrix.','/,
#9x,'Execution will cease.'///)
RETURN
END
C
REAL FUNCTION PRB(X)
  Z = ABS(X)
  IF(Z.LT.S.) GO TO I
  PRB = .5*(I.+Z/X)
  RETURN
1 PRB = .5*(I.+(((.01952/*Z+.000344)*Z+.115194)*Z+.196854)*Z)**(-4)
  IF(X.LT.O.) RETURN
  PRB = 1.-PRB
  RETURN
END
C
REAL FUNCTION HK(K,Y)
  HK = 1.
  IF(K.EQ.0) GO TO 5
  HK1 = HK
  HK = Y
  IF(K.EQ.1) GO TO 5
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.

```
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.

```

```
### 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: (200, 200), (180, 200) and (340, 200) in Figure 2.

#### BARE SOIL TEMPERATURE DATA -- EXAMPLE

<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
<th>TEMP</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>7</td>
<td>23.000</td>
<td>100.000</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>2.000</td>
<td>200.000</td>
</tr>
<tr>
<td>62.500000</td>
<td>64.000000</td>
<td>65.000000</td>
<td>66.000000</td>
</tr>
<tr>
<td>-0.682487</td>
<td>0.178525</td>
<td>0.674979</td>
<td>1.127083</td>
</tr>
<tr>
<td>6.3890934E+01</td>
<td>1.7091980E+00</td>
<td>1.8315849E-01</td>
<td>-3.6764962E-03</td>
</tr>
<tr>
<td>1</td>
<td>0.700</td>
<td>2.400</td>
<td>23.000</td>
</tr>
<tr>
<td>108</td>
<td>17.000</td>
<td>200.000</td>
<td>60.950000</td>
</tr>
<tr>
<td>75</td>
<td>27.000</td>
<td>198.000</td>
<td>61.190000</td>
</tr>
<tr>
<td>95</td>
<td>33.000</td>
<td>181.000</td>
<td>61.350000</td>
</tr>
<tr>
<td>100</td>
<td>34.000</td>
<td>186.000</td>
<td>61.440000</td>
</tr>
<tr>
<td>87</td>
<td>31.000</td>
<td>37.000</td>
<td>61.450000</td>
</tr>
<tr>
<td>99</td>
<td>33.000</td>
<td>147.000</td>
<td>61.650000</td>
</tr>
<tr>
<td>104</td>
<td>17.000</td>
<td>93.000</td>
<td>61.710000</td>
</tr>
<tr>
<td>60</td>
<td>23.000</td>
<td>176.000</td>
<td>61.780000</td>
</tr>
<tr>
<td>96</td>
<td>33.000</td>
<td>94.000</td>
<td>61.810000</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### APPENDIX 2D

Example of output from DISJUNC.FTN
### Disjunctive Kriging Program for Two Dimensions

**INPUT PARAMETERS**

- **Number of Temp's Input**: 13
- **Maximum Number of Nearest Neighbors**: 5
- **Maximum Radius**: 23,000
- **Maximum Allowed Data Value**: 100,000

**Correlation Function Model Type**: Spherical

**Nugget**: 0.2758

**Sill Minus Nugget**: 0.7742

**Range**: 23,000

### Observed Data

<table>
<thead>
<tr>
<th>Z</th>
<th>X</th>
<th>Y</th>
<th>Temp</th>
<th>Tz</th>
<th>Obs. No.</th>
<th>X</th>
<th>Y</th>
<th>Temp</th>
<th>Tz</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>17,000</td>
<td>200,000</td>
<td>60.950</td>
<td>-2.097879</td>
<td>56</td>
<td>21.000</td>
<td>199,000</td>
<td>64.590</td>
<td>0.477008</td>
</tr>
<tr>
<td>7</td>
<td>27.000</td>
<td>100,000</td>
<td>61.190</td>
<td>-2.648041</td>
<td>19</td>
<td>6.000</td>
<td>188,000</td>
<td>65.540</td>
<td>0.974271</td>
</tr>
<tr>
<td>1</td>
<td>37.000</td>
<td>201,000</td>
<td>62.470</td>
<td>-0.729496</td>
<td>5</td>
<td>2.000</td>
<td>198,000</td>
<td>66.470</td>
<td>1.326744</td>
</tr>
<tr>
<td>7</td>
<td>25.000</td>
<td>211,000</td>
<td>67.910</td>
<td>-0.424904</td>
<td>7</td>
<td>3.000</td>
<td>206,000</td>
<td>67.330</td>
<td>1.881627</td>
</tr>
<tr>
<td>5</td>
<td>25.000</td>
<td>101,000</td>
<td>67.190</td>
<td>-0.331349</td>
<td>15</td>
<td>4.000</td>
<td>187,000</td>
<td>67.430</td>
<td>1.729988</td>
</tr>
<tr>
<td>7</td>
<td>26.000</td>
<td>211,000</td>
<td>63.600</td>
<td>-0.033682</td>
<td>3</td>
<td>1.000</td>
<td>190,000</td>
<td>67.830</td>
<td>1.891670</td>
</tr>
<tr>
<td>5</td>
<td>24.000</td>
<td>199,000</td>
<td>63.650</td>
<td>-0.066678</td>
<td>0</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000000</td>
</tr>
</tbody>
</table>

**Data Set: Hermite Polynomials**

- **Mean**: 63.340
- **Variance**: 3.0904

**Hermite Polynomial Coefficients**

<table>
<thead>
<tr>
<th>k</th>
<th>Ck</th>
<th>Ck</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>6.39083 E+01</td>
<td>-4.43677 E-02</td>
</tr>
<tr>
<td>1</td>
<td>1.39074 E+00</td>
<td>-3.61649 E-03</td>
</tr>
<tr>
<td>2</td>
<td>1.81155 E-01</td>
<td>-5.84811 E-03</td>
</tr>
<tr>
<td>3</td>
<td>-6.31786 E-02</td>
<td>0.00000 E-01</td>
</tr>
</tbody>
</table>

### Disjunctive Kriging Estimates

**Probability That the Estimate Is Greater Than Zcut**

<table>
<thead>
<tr>
<th>Value(s) Zcut</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.999</td>
</tr>
</tbody>
</table>

*Values when entire data set used in calculation*