MICHIGAN STATE UNIVERSITY

W. K. KELLOGG BIOLOGICAL STATION
TELEPHONE (616) 671-5117

HICKORY CORNERS • MICHIGAN • 49060-9516
FAX (616) 671-2351;199-2351 (CAMPUS)

ESPS 8733

NOV 8 1993

KBS Geostatistics Programs

KBS no longer distributes on diskette the Geostat programs noted in Robertson (1987) [Robertson, G.P. 1987. Geostatistics in ecology: interpolating with known variance. Ecology 68:744-748]. If you need a copy of the Robertson (1987) programs for documentary purposes you should refer to the microfiche archives of the Ecological Society of America (consult a recent issue of Ecology) or write the author.

Thanks for your interest; we're sorry to be unable to meet your request. You may wish to write to the addresses below for information on other permanently supported programs (we do not endorse these programs, only note their availability):

U.S. EPA (Environmental Protection Agency)

Geo-EAS (Geostatistical Environmental Analysis Software)

U.S. Environmental Protection Agency
Attn: Evan Englund
Environmental Monitoring Systems Laboratory
Las Vegas, NV 89193-3478
USA

Gamma Design Software

GS+: Geostatistics for the Biological, Environmental, and Agronomic Sciences

Gamma Design Software
P.O. Box 201
Plainwell, MI 49080-0201
USA

ESA Supplementary Publication Service

Document No. ESPS 8733

Fortran programs for: Semivariogram analysis
Punctual krig analysis
Block krig analysis

Supplement to

"Geostatistics in ecology: interpolating with known variance"

by

G. Philip Robertson

ECOLOGY 68 (1987)

File Semivar.hlp; to accompany

Program Semivar.for,
Semivariogram Analysis
Version 11.10.86 Copyright (c) 1986 G.P. Robertson

The correct citation for this program is
Robertson, G.P. 1987. Geostatisitics in ecology: interpolating with known variance. Ecology 68:

Later releases of this program available from: Computer Services Laboratory
W.K. Kellogg Biol. Station
Michigan State University
Hickory Corners, MI 49060.

Program parameters:

1. Hardware requirements. This version was written in Microsoft Fortran version 3.3 designed to run under the IBM MS-DOS 2.0+ operating system. Compilation by other compilers compatible with the ANSI 77 standard should be possible.

2. Data Input. Data is read into the program as ASCII characters in free format from a user-specified file. The first 4 lines of the input file are treated as header lines -- they should not contain data. If the first characters of the third line are numeric, they are assumed to represent the number of columns in each data record. A sample header:

```
Line 1: File xxxxx.prn from supporting files xxyyy.ext and xxxyy.ext;
     2: Small grid samples taken 5/19.....
     3: 15 columns: Sample#, Xcor, Ycor, z1,z2,z3,z4,z5,
     4:        z6,z7,z8,z9,z10,z11,z12.
     5: 1 0 15 4.3 2.11 .0003 -99. 2.3 3 5 4 3 2 2 1.9997
```

   If the column no. value on line 3 is not correct or if there are different numbers of columns in any row, the data will not be read correctly and calculations will be incorrect. Missing values are represented by the value -99. Data size limitations are 15 columns x 1000 rows, though redimensioning key variables will allow indefinite input file sizes.

3. Output. Primitive graphics output is written to the screen during program execution and text and graphics output is written to a file of the user's specification. Output data is ASCII format and can be subsequently printed or read into another program.

4. User-specified prompts:
   a) Semivar menu;
      a) 1-dimensional semivariogram uses as input data a single vector; do not use a 2-coordinate dataset with this subprogram -- instead specify the 2-dimensional semivariogram and set the y-range to a single value (e.g. 10 - 10);
      b) 2-dimensional semivariogram uses as input data a grid of points within the dimensions of the overall grid as specified by the user;
      c) read data file; see limitations above;
      d) help; this output;
      e) exit program;
   c) input file name; conform to MSDOS convention;
   d) run title; as desired for labeling output graphs; 20 char. max;
   e) x,z or x,y,z columns: columns in which x coordinates, y coordinates, and active z-value can be found; values for x and y coordinates should not be >5 significant figures each;
   f) z-label: as desired for labeling output, 16 char. max;
   g) use subset of data set; to use partial grid;
   h) transform z-value; if requested, z is logn-transformed with the option of adding a constant to each value before transforming; because of the discontinuous nature of this distribution over the range <-1 to >1, all values <1.0 are ignored for the duration of the analysis; note that the transformation lasts only until the next Semivar menu, and that statistics output may give back-transformed values -- see note below;
   i) test for anisotropy; if yes,
      i) no. of directions (<16);
      ii) direction and tolerance for each; direction = degrees from north (0-360) that any given pair of points must align along in order to be included in the analysis; tolerance = +/- degrees (0-90) away from the given

direction that points may diverge from the direction but still be included in the analysis; isotropic analysis = any direction +/- 90 degrees;

j) specify output file; N = results are not read to output file, appear on screen only;
A = appends results to last output file opened;
f = name of new output file; if file exists, can write over it or append to it as specified;

N.B. Note that statistics on the output file apply only to points used for that specific analysis and not to the entire data set, and that if values were ln-tranformed, means and variance estimates are back-transformed using either standard or Haan (1977) methods, as specified by earlier prompt. Standard mean = exp([mean of logn values]) -A; var = exp[var(logn values)]. Haan (1977) mean = exp([mean of logn values]+0.5*var of logn values)]-A; var= ([mean of untransformed values] ^2)*(exp[var of logn values]-1). Where A = constant added to original values before ln transformation in order to bring original values to >1.0 before ln transformation. (See Haan, C.T. 1977. Chapter 6. Some continuous probability distributions. Statistical Methods in Hydrology. The Iowa State Univ. Press, Ames, Iowa.)

```
$storage:2

      PROGRAM SEMIVAR

C     Version 11.10.86
C     Copyright (c) 1986 G.P. Robertson. All rights reserved.

C     The correct citation for this program is Robertson, G.P. 1987.
C         Geostatistics in ecology: interpolating with known variance.
C         Ecology 68:(in press).
C
C     This program is being maintained at the Computer Services Laboratory
C                                         W.K. Kellogg Biological Station
C                                         Michigan State University
C                                         Hickory Corners, MI 49060.
C
C        Please report significant bugs and well-documented enhancements
C        to this address for inclusion in later versions.

C     Documention for users appears in file Semivar.hlp.

C     Subroutines: SEM1D, SEM2D, SETUP, GETOUT, STATS, DATREAD,
C                  TEXTOUT, PRIMGRPH

C-----Variable list
C
C  NAME        FUNCTION
C  ----        --------
C
C  FILNAM      input data file
C  OUTFIL      file where output data sent
C  PRESOUT     last output file named (used in appending)
C  FTITLE      run title
C  ZLABEL      z-variable title
C  XCOL        column in filnam with x-coordinate values
C  YCOL        column with y-coordinate values
C  ZCOL        column with z-corrdinate values
C  XMAX        greatest x-coordinate
C  XMIN        least x-coordinate
C  YMAX        greatest y-coordinate
C  YMIN        least y-coodinate
C  DIR(15)     directions from north to evaluate anisotropy
C  TOL(15)     tolerances (+/- degrees) for closeness of pair to dir(i)
C  DIST(1000)  average distance for all lags
C  NCPL(1000)  number of pairs
C  GAMA(1000)  gammas
C  DRIFT(1000) drift
C  LOG         log transformation? (logical)
C  HAAN        back-tranform using Haan (1977) (logical)
C  MLAG        greatest possible lag for given input data
C  NLAG        user's choice for lag to be used.
C  NSTP        user's choice for step size
C  NR          number of valid data elements
C  VALUE       If Semivariogram is 1D this is the linear coordinate
C  AMEAN       mean of given input data NB: if LOG, not backtransformed
C  VAR         mariance of given input data
C  STDEV       standard deviation of given input data
C  CVA         C.V. for given input data
C  CSKEW       skewness for given input data
C  CKURT       kurtosis for given input data

      INTEGER      XCOL,YCOL,ZCOL
      LOGICAL      THERE,LOG,HAAN,READIN,CALC,GRAPH
      CHARACTER*1  YORN,BS
      CHARACTER*32 FILNAM,OUTFIL,PRESOUT
      CHARACTER*20 FTITLE,ZLABEL
      CHARACTER*1 HEAD1(79),HEAD2(79),HEAD3(79),HEAD4(79),LINE(79)

      COMMON/PGMPARAM/NDIM,NR,NLAG,MLAG,NSTP,LOG,A,HAAN,
     +                XMAX,XMIN,YMAX,YMIN,XCOL,YCOL,ZCOL,
     +                ZLABEL,GRAPH
      COMMON/FILINFO/FILNAM,OUTFIL,PRESOUT,FTITLE,
     +                NCOL,NROW,HEAD1,HEAD2,HEAD3,HEAD4

C-----Write menu header
      DO 2 I=1,5
         WRITE(*,*)
```

```
 2    CONTINUE
      WRITE(*,*)'Program Semivar'
      WRITE(*,*)'  Semivariogram Analysis'
      WRITE(*,*)'  Version 11.10.86'
      WRITE(*,*)' '
      WRITE(*,*)'See file Semivar.hlp and source code listing'
      WRITE(*,*)'  for documentation and copyright information'
      WRITE(*,'(//1X,A,$)')'Hit <return> to continue.....'
      READ(*,'(A1)') YORN
      WRITE(*,'(/////////////////////////1X)')

C-----Give user menu screen
      NLAG=0
      PRESOUT='NONE'
  5   WRITE(*,*)
      CALC=.FALSE.
      WRITE(*,*) 'Semivar Menu'
      IF (.NOT.READIN) THEN
         WRITE(*,*) 'No active input file'
      ELSE
         WRITE(*,200) FILNAM
         WRITE(*,205) NCOL,NROW
         WRITE(*,*) 'File header:'
         WRITE(*,'(1X,79A1)') (HEAD1(I),I=1,79)
         WRITE(*,'(1X,79A1)') (HEAD2(I),I=1,79)
         WRITE(*,'(1X,79A1)') (HEAD3(I),I=1,79)
         WRITE(*,'(1X,79A1)') (HEAD4(I),I=1,79)
      ENDIF
 200  FORMAT(1X,'Current file: ',A32)
 205  FORMAT(15X,I2,' columns, ',I4,' rows')

      WRITE(*,*)
      WRITE(*,*)'Choose a function:'
      WRITE(*,'(3X,A)')'1. 1-dimensional Semivariogram'
      WRITE(*,'(3X,A)')'2. 2-dimensional Semivariogram'
      WRITE(*,'(3X,A)')'3. Read data file            '
      WRITE(*,'(3X,A)')'4. Help                      '
      WRITE(*,'(3X,A)')'5. End Program               '
      WRITE(*,*)
 50   WRITE(*,'(2X,A\)')' Enter choice: '
      READ(*,'(BN,I1)',ERR=99)NDIM

C-----Act on choice
      IF ((NDIM.EQ.1).OR.(NDIM.EQ.2)) THEN
         WRITE(*,*)
         IF (.NOT.READIN) CALL DATREAD(READIN)
         IF (.NOT.READIN) GOTO 5
         IF (NDIM.EQ.1) CALL SEM1D
         IF (NDIM.EQ.2) CALL SEM2D
         CALC=.TRUE.
      ELSEIF (NDIM.EQ.3) THEN
         WRITE(*,*)
         CALL DATREAD(READIN)
         CALC=.TRUE.
      ELSEIF (NDIM.EQ.4) THEN
C---------SEE IF HELP FILE PRESENT
          INQUIRE(FILE='SEMIVAR.HLP', EXIST=THERE)
          IF (THERE) THEN
             DO 12 I=1,10
               WRITE(*,*)
 12          CONTINUE
             OPEN (7,FILE='SEMIVAR.HLP',STATUS='OLD')
 10          DO 11 J=1,24
               READ (7,'(79A1)',END=20) (LINE(I),I=1,79)
               WRITE(*,'(1X,79A1)') (LINE(I),I=1,79)
 11          CONTINUE
             WRITE(*,'(20X,A,$)')'(.....hit <return> to continue.....)'
             READ(*,'(A1)') YORN
             GOTO 10
 20          CLOSE(7)
             WRITE(*,'(/1X,A,$)')'(.....hit <return> to continue.....)'
             READ(*,'(A1)') YORN
             WRITE(*,'(/////////////////////////1X)')
             CALC=.TRUE.
         ELSE
             WRITE(*,*)'   file Semivar.hlp not in directory...'
         ENDIF
      ELSEIF (NDIM.EQ.5) THEN
         WRITE(*,*)
         WRITE(*,*) 'Normal termination;'
         WRITE(*,*) 'Exit Semivar.'
         GOTO 2000
      ELSE
         WRITE(*,*)'     choose 1-5 only...'
      ENDIF

C-----After act on choice, replay menu
      IF (CALC) THEN
         GOTO 5
      ELSE
         GOTO 50
      ENDIF

C-----Error trapping
 99   WRITE(*,*)'    try again...'
      GOTO 5

 2000 END


C***********************************************************************
C***********************************************************************

      SUBROUTINE SEM1D
C     1-dimensional semivariogram

      LOGICAL      LOG,GRAPH,HAAN,SAS
      INTEGER      SIZE,XCOL,YCOL,ZCOL
      CHARACTER*32 FILNAM,OUTFIL,PRESOUT
      CHARACTER*20 FTITLE,ZLABEL
      CHARACTER*1 HEAD1(79),HEAD2(79),HEAD3(79),HEAD4(79),Q

      COMMON/DATACOL/X(1000),Y(1000),Z(1000)
      COMMON/VARIDAT/NCPL(1000),DIST(1000),GAMA(1000),DRIFT(1000),
     +               DIR(15),TOL(15)
      COMMON/FILINFO/FILNAM,OUTFIL,PRESOUT,FTITLE,
     +               NCOL,NROW,HEAD1,HEAD2,HEAD3,HEAD4
      COMMON/DSTATS/AMEAN,AVAR,NN,STDEV,CVA,CSKEW,CKURT,
     +              SESKEW,SEKURT
      COMMON/PGMPARAM/NDIM,NR,NLAG,MLAG,NSTP,LOG,A,HAAN,
     +               XMAX,XMIN,YMAX,YMIN,XCOL,YCOL,ZCOL,
     +               ZLABEL,GRAPH

C-----Get X,Z Vectors, Log-transform
      CALL SETUP
      IF(NR.LT.3) THEN
         WRITE(*,101) NR
 101     FORMAT(' too few valid points (,I1,') in file...'
         RETURN
      ENDIF

C-----Determine max and mins for coordinates
      XMIN=X(1)
      XMAX=X(1)
      DO 10 I=2,NR
         IF (X(I).LT.XMIN) XMIN=X(I)
         IF (X(I).GT.XMAX) XMAX=X(I)
 10   CONTINUE

      WRITE(*,*)'Transect bounds:'
      WRITE(*,'(4X,A3,F11.3,A3,F11.3)')'X: ',XMIN,' , ',XMAX

C-----See if user wants only a subset of data set
 15   WRITE(*,'(A32,$)')' Use subset of transect? (y/n): '
      READ(*,'(A1)') YORN
      IF ((YORN.EQ.'Y').OR.(YORN.EQ.'y')) THEN
         WRITE(*,'(A26,$)')'Enter min X and max X: '
         READ(*,*)TXMIN,TXMAX
         K=0
         DO 20 L=1,NR
            IF ((X(L).GE.TXMIN).AND.(X(L).LE.TXMAX)) THEN
               K=K+1
               X(K)=X(L)
               Z(K)=Z(L)
            ENDIF
 20      CONTINUE
         IF (K.LT.3) THEN
            WRITE(*,150) K
 150        FORMAT(4X,'too few data (',I1,') within range...')
            WRITE(*,*)
            GOTO 15
         ENDIF
         XMAX=TXMAX
         XMIN=TXMIN
         NR=K
      ELSEIF ((YORN.NE.'N').AND.(YORN.NE.'n')) THEN
         WRITE(*,*) '      answer either y or n...'
         GOTO 15
      ENDIF

C-----Determine the maximum possible lag
      MLAG=NINT(ABS(XMAX-XMIN)+0.5)

C-----See if user wants a different lag
 35   WRITE(*,*)
      WRITE(*,'(A25,I4,)')' Largest lag interval is ',MLAG
      WRITE(*,'(A33,$)')' Enter maximum lag (integer): '
      READ(*,*) NLAG
      IF (NLAG.GT.MLAG) NLAG=MLAG

C-----Ask for step size
 37   WRITE(*,'(A31,$)')' Enter step size (integer): '
      READ(*,*,ERR=600) NSTP
      WRITE(*,*)

C-----See if NLAG>NSTP
      IF(NLAG.LE.NSTP) THEN
         WRITE(*,*)'    step must be less than maximum lag ....'
         GOTO 35
      ENDIF

C-----Initialize main variables
      DO 38 I=1,NLAG
         NCPL(I)=0
         DIST(I)=0.0
         GAMA(I)=0.0
         DRIFT(I)=0.0
 38   CONTINUE

C-----Ask for output file
      CALL GETOUT(SAS)

C-----Semivariogram analysis
      WRITE(*,*)
      WRITE(*,'(A38\)') '...calculating transect point '
      Q=CHAR(8)
      WRITE(*,990)
 990  FORMAT('           '\)
      DO 40 J=1,NR-1
         WRITE(*,991) Q,Q,Q,Q,Q,Q,Q,Q,Q,Q,Q,J,NR-1
 991     FORMAT(11A1,I4,' of',I4\)
         DO 45 I=J+1,NR
            H=ABS(X(J)-X(I))
            IF (H.LT.1.0E-6*NSTP) GOTO 45
            IF (H.GT.NLAG) GOTO 45
            M=(INT((H/NSTP)+0.5))+1
            NCPL(M)=NCPL(M)+1
            DIST(M)=DIST(M)+H
            DZ=Z(I)-Z(J)
            DRIFT(M)=DRIFT(M)+DZ
            GAMA(M)=GAMA(M)+0.5*(DZ**2)
 45     CONTINUE
 40   CONTINUE
      WRITE(*,*)

      DO 50 I=1,NLAG
         CP=FLOAT(NCPL(I))
         IF (CP.EQ.0.0) GOTO 50
         DIST(I)=DIST(I)/CP
         DRIFT(I)=DRIFT(I)/CP
         GAMA(I)=GAMA(I)/CP
 50   CONTINUE

C-----Start output procedures
      CALL STATS(Z,NR)
      CALL PRIMGRPH(0,0)
      IF (OUTFIL.NE.'NONE') CALL TEXTOUT(0,SAS)

      RETURN

C-----Error trapping
 600  WRITE(*,*)'      ...entries must be integers;'
      GOTO 37

      END

C***************************************************************************
C***************************************************************************

      SUBROUTINE SEM2D
C     2-d Semivariogram analysis

      LOGICAL      LOG,INCL(1000),GRAPH,HAAN,SAS
      INTEGER      SEC(8),XCOL,YCOL,ZCOL
      REAL         ZSTAT(1000)
      REAL*8       CDIR,CTOL,SDIR,DX,DY,DHS,H,DTST,PI
      CHARACTER*1  YORN,Q
      CHARACTER*20 ZLABEL,FTITLE
      CHARACTER*32 FILNAM,OUTFIL,PRESOUT
      CHARACTER*1 HEAD1(79),HEAD2(79),HEAD3(79),HEAD4(79)

C-----Common Blocks
      COMMON/DATACOL/X(1000),Y(1000),Z(1000)
      COMMON/VARIDAT/NCPL(1000),DIST(1000),GAMA(1000),DRIFT(1000),
     +               DIR(15),TOL(15)
      COMMON/FILINFO/FILNAM,OUTFIL,PRESOUT,FTITLE,
     +               NCOL,NROW,HEAD1,HEAD2,HEAD3,HEAD4
      COMMON/DSTATS/AMEAN,AVAR,NN,STDEV,CVA,CSKEW,CKURT,
     +              SESKEW,SEKURT
      COMMON/PGMPARAM/NDIM,NR,NLAG,MLAG,NSTP,LOG,A,HAAN,
     +               XMAX,XMIN,YMAX,YMIN,XCOL,YCOL,ZCOL,
     +               ZLABEL,GRAPH

C-----See if run parameters should be reset
      YORN=' '
      WRITE(*,*)
      IF(NLAG.GT.0) THEN
         WRITE(*,705)
 705     FORMAT(' Reset run parameters? (y/n): ',$)
  51     READ(*,'(A1)') YORN
         IF((YORN.EQ.'N').OR.(YORN.EQ.'n')) THEN
C-----------See if user wants a different maximum lag
  50        WRITE(*,*)
            WRITE(*,'(A,I4,)')' Lag interval maximum: ',MLAG
            WRITE(*,'(A,I4,)')'                  last: ',NLAG
            WRITE(*,'(A,$)')  '                   new: '
  55        READ(*,*,ERR=655) NLAG
            IF(NLAG.LE.NSTP) THEN
              WRITE(*,658) NSTP
658           FORMAT(1X,'    .....lag must be >step (',I4,
                         '); respecify: ',$)
              GOTO 55
            ENDIF
            GOTO 61
 655        WRITE(*,656)
 656        FORMAT(1X,          '... entry error; respecify: ',$)
            GOTO 55
         ELSEIF((YORN.NE.'Y').AND.(YORN.NE.'y')) THEN
            WRITE(*,'(1X,A,$)')'... answer either y or n: '
            GOTO 51
         ENDIF
      ENDIF

C--------Get X,Y,Z vectors, log-transform if necessary
         CALL SETUP
         IF(NR.LT.3) THEN
            WRITE(*,101) NR
 101        FORMAT(' too few valid points (,I1,') in file...'
            RETURN
         ENDIF

C-----Determine max and min coordinates for active dataset
      XMIN=X(1)
      XMAX=X(1)
      YMIN=Y(1)
      YMAX=Y(1)
      DO 5 I=2,NR
         IF (X(I).LT.XMIN) XMIN=X(I)
         IF (X(I).GT.XMAX) XMAX=X(I)
         IF (Y(I).LT.YMIN) YMIN=Y(I)
         IF (Y(I).GT.YMAX) YMAX=Y(I)
 5    CONTINUE
      WRITE(*,*)'X-Y Bounds:'
      WRITE(*,'(5X,A3,F11.3,A3,F11.3)')'X: ',XMIN,' , ',XMAX
      WRITE(*,'(5X,A3,F11.3,A3,F11.3)')'Y: ',YMIN,' , ',YMAX

C-----See if user wants only a subset of data set
 10   WRITE(*,'(A31,$)')' Use subset of data set? (y/n): '
      READ(*,'(A1)') YORN
      IF ((YORN.EQ.'Y').OR.(YORN.EQ.'y')) THEN
         WRITE(*,'(A27,$)')'    Enter min X and max X: '
         READ(*,*)TXMIN,TXMAX
         WRITE(*,'(A27,$)')'    Enter min Y and max Y: '
         READ(*,*)TYMIN,TYMAX
         K=0
         DO 15 L=1,NR
            XRANGE=0
            YRANGE=0
            IF ((X(L).GE.TXMIN).AND.(X(L).LE.TXMAX)) XRANGE=1
            IF ((Y(L).GE.TYMIN).AND.(Y(L).LE.TYMAX)) YRANGE=1
            IF ((XRANGE.EQ.1).AND.(YRANGE.EQ.1)) THEN
               K=K+1
               X(K)=X(L)
               Y(K)=Y(L)
               Z(K)=Z(L)
            ENDIF
 15      CONTINUE
         IF (K.LT.3) THEN
            WRITE(*,150) K
 150        FORMAT(4X,'too few data (',I1,') within range...')
            WRITE(*,*)
            GOTO 10
         ENDIF
         XMAX=TXMAX
         XMIN=TXMIN
         YMAX=TYMAX
         YMIN=TYMIN
         NR=K
      ELSEIF ((YORN.NE.'N').AND.(YORN.NE.'n')) THEN
         WRITE(*,*) '   answer either y or n...'
         GOTO 10
      ENDIF
      XORIGIN=XMAX-((XMAX-XMIN)/2.0)
      YORIGIN=YMIN

C-----Determine the maximum possible lag
      XDIF=XMAX-XMIN
      YDIF=YMAX-YMIN
      MLAG=NINT(SQRT((XDIF**2)+(YDIF**2))+0.5)

C-----See if user wants a different maximum lag
 30   WRITE(*,*)
      WRITE(*,'(A25,I4,)')' Largest lag interval is ',MLAG
      WRITE(*,'(A33,$)')' Enter maximum lag (integer): '
      READ(*,*) NLAG
      IF(NLAG.GT.MLAG) NLAG=MLAG

C-----Ask for step size
 31   WRITE(*,'(A31,$)')' Enter step size (integer): '
      READ(*,*,ERR=600) NSTP

C-----See if NLAG>NSTP
      IF(NLAG.LE.NSTP) THEN
         WRITE(*,*)'    step must be less than lag ....'
         GOTO 30
      ENDIF

C-----See if user wants to examine anisotropy
      NDIR=0
 40   WRITE(*,*)
      WRITE(*,'(A28,$)') ' Examine anisotropy? (y/n): '
      READ(*,'(A1)') YORN
 36   IF ((YORN.EQ.'Y').OR.(YORN.EQ.'y')) THEN
 310     WRITE(*,'(A26,$)') 'Number of directions: '
         READ(*,*,ERR=665) NDIR
         IF ((NDIR.LT.0).OR.(NDIR.GT.15)) THEN
            WRITE(*,*) '       value must be between 0-15...'
            GOTO 310
         ENDIF
         WRITE(*,*) '   Degrees from north and tolerance ',
     +   '(+/- degrees) for '
         DO 300 I=1,NDIR
 305        WRITE(*,410) I
 410        FORMAT('         Direction ',I2,': ',$)
            READ(*,*) DIR(I),TOL(I)
            IF ((DIR(I).EQ.0.).OR.(DIR(I).EQ.360.)) DIR(I)=1E-10
            IF ((DIR(I).GT.360.).OR.(DIR(I).LT.0.)) THEN
               WRITE(*,*) '            direction out of bounds...'
               GOTO 305
            ELSEIF ((TOL(I).LT.0.).OR.(TOL(I).GT.180.)) THEN
               WRITE(*,*) '            tolerance out of bounds...'
               GOTO 305
            ENDIF
 300     CONTINUE
      ELSEIF ((YORN.NE.'N').AND.(YORN.NE.'n')) THEN
         WRITE(*,*) '    answer either y or n ...'
         GOTO 40
      ENDIF
      WRITE(*,*)

C-----Get output file name
 61   CALL GETOUT(SAS)

C-----Start semivariogram processing
 60   DO 65 I=0,NDIR
         PI=3.1415926536
         IF(I.EQ.0) THEN
            CDIR=PI*90.0/180.0
            CTOL=DCOS(PI*180./360.)
         ELSE
            CDIR=PI*DIR(I)/180.
            CTOL=DCOS(PI*2.0*TOL(I)/360.)
         ENDIF
         SDIR=DSIN(CDIR)
         CDIR=DCOS(CDIR)
         WRITE(*,*)
         WRITE(*,*)
         WRITE(*,'(A34,I3,A13\)') '...calculating direction',I,
     +                             ', grid point ',
 70      DO 75 J=1,NLAG
            NCPL(J)=0
            DIST(J)=0.0
            GAMA(J)=0.0
            DRIFT(J)=0.0
 75      CONTINUE

         Q=CHAR(8)
         WRITE(*,990)
 990     FORMAT('               '\)
         DO 80 J=1,NR-1
            WRITE(*,991) Q,Q,Q,Q,Q,Q,Q,Q,Q,Q,Q,J,NR-1
 991        FORMAT(11A1,I4, ' of',I4\)
            DO 85 M=J+1,NR
               DX=X(J)-X(M)
               DY=Y(J)-Y(M)
               DHS=(DX**2)+(DY**2)
               IF (DHS.LT.(1.0E-3*NSTP)) GOTO 85
               H=DSQRT(DHS)
               IF(H.GT.DBLE(NLAG)) GOTO 85
               DTST=(DX*SDIR/H)+(DY*CDIR/H)
               IF(DABS(DTST).LT.CTOL) GOTO 85
               M1=IDINT((H/NSTP)+0.5)+1
               NCPL(M1)=NCPL(M1)+1
               DIST(M1)=DIST(M1)+H
               DZ=SNGL(DTST*DBLE(Z(J)-Z(M))/DABS(DTST))
               DRIFT(M1)=DRIFT(M1)+DZ
               GAMA(M1)=GAMA(M1)+0.5*(DZ**2)
               INCL(J)=.TRUE.
               INCL(M)=.TRUE.
 85         CONTINUE
 80      CONTINUE
         WRITE(*,*)

C--------Weight gama, drift, distance for pairs at each distance
         DO 90 J=1,NLAG
            CP=FLOAT(NCPL(J))
            IF (CP.EQ.0.0) GOTO 90
            DIST(J)=DIST(J)/CP
            DRIFT(J)=DRIFT(J)/CP
            GAMA(J)=GAMA(J)/CP
 90      CONTINUE

C--------Calculate stats for z's in this direction only
         JJ=0
         DO 88 J=1,NR
         IF(INCL(J)) THEN
            JJ=JJ+1
            ZSTAT(JJ)=Z(J)
         ENDIF
         INCL(J)=.FALSE.
 88      CONTINUE
         CALL STATS(ZSTAT,JJ)

C--------Call output routines
         CALL PRIMGRPH(I,0)

         IF (OUTFIL.NE.'NONE') CALL TEXTOUT(I,SAS)

 65   CONTINUE
      GOTO 99

C-----Error trapping
 600  WRITE(*,666)
      GOTO 31

 665  WRITE(*,666)
 666  FORMAT('     integer only...')
      GOTO 36

 99   RETURN
      END

C***********************************************************************
C***********************************************************************

      SUBROUTINE SETUP
C     Gets column nos, z-label, fills data vectors x,y,z, transforms z

      LOGICAL      THERE,LOG,GRAPH,HAAN
      INTEGER      XCOL,YCOL,ZCOL,COL1,COL2
      CHARACTER*1  YORN,ANSW
      CHARACTER*32 FILNAM,OUTFIL,PRESOUT
      CHARACTER*20 ZLABEL,FTITLE
      CHARACTER*1 HEAD1(79),HEAD2(79),HEAD3(79),HEAD4(79)

      COMMON/DATAIN/DATA(1000,15)
      COMMON/DATACOL/X(1000),Y(1000),Z(1000)
      COMMON/FILINFO/FILNAM,OUTFIL,PRESOUT,FTITLE,
     +               NCOL,NROW,HEAD1,HEAD2,HEAD3,HEAD4
      COMMON/PGMPARAM/NDIM,NR,NLAG,MLAG,NSTP,LOG,A,HAAN,
     +               XMAX,XMIN,YMAX,YMIN,XCOL,YCOL,ZCOL,
     +               ZLABEL,GRAPH

C-----Ask user for column numbers for X, Y, and Z
 2    IF(NDIM.EQ.1) THEN
         WRITE(*,'(A29,$)') ' Column numbers for X and Z: '
         READ(*,*,ERR=600) XCOL,ZCOL
         YCOL=XCOL
      ELSEIF(NDIM.EQ.2) THEN
```

```
         WRITE(*,'(A32,$)')' Column numbers for X,Y, and Z: '
         READ(*,*,ERR=600) XCOL,YCOL,ZCOL
      ENDIF

C-----See if columns exist
      IF ((XCOL.GT.NCOL).OR.(YCOL.GT.NCOL).OR.(ZCOL.GT.NCOL)) THEN
         WRITE(*,*)'   value out of range...'
         GOTO 2
      ENDIF

C-----Get a label for the Z variable
      WRITE(*,'(A20,$)')' Enter label for Z: '
      READ(*,'(A20)') ZLABEL
      WRITE(*,*)


C-----See if user wants a logarithmic transformation of Z
 20   WRITE(*,105)
      READ(*,'(A1)') YORN
 22   IF ((YORN.EQ.'Y').OR.(YORN.EQ.'y')) THEN
         LOG=.TRUE.
         WRITE(*,106)
         READ(*,*,ERR=605) A
         IF(A.LT.0) THEN
            WRITE(*,*)'       a must be =>0...'
            WRITE(*,*)
            GOTO 22
         ENDIF
 21      WRITE(*,107)
         READ(*,'(A1)') ANSW
         IF ((ANSW.EQ.'H').OR.(ANSW.EQ.'h')) THEN
            HAAN=.TRUE.
         ELSEIF ((ANSW.EQ.'N').OR.(ANSW.EQ.'n')) THEN
            HAAN=.FALSE.
         ELSE
            WRITE(*,*)'       answer H or N only...'
            GOTO 21
         ENDIF
      ELSEIF ((YORN.EQ.'N').OR.(YORN.EQ.'n')) THEN
         LOG=.FALSE.
      ELSE
         WRITE(*,*) '   answer either y or n...'
         GOTO 20
      ENDIF
      WRITE(*,*)
 105  FORMAT(' Transform Z to logn(z)? (y/n): ',$)
 106  FORMAT('    Specify a in the expression Z=ln(z+a):',$)
 107  FORMAT('    Specify if want [H]aan or [N]o '
     +               'backtranformation: ',$)

C-----Put data from proper columns in to data work array
      K=0
      NOUTS=0
      DO 15 I=1,NROW
         IF (DATA(I,XCOL).EQ.-99.0) GOTO 15
         IF (DATA(I,YCOL).EQ.-99.0) GOTO 15
         IF (DATA(I,ZCOL).EQ.-99.0) GOTO 15
         IF(((DATA(I,ZCOL)+A).LT.1.0).AND.(LOG))THEN
            NOUTS=NOUTS+1
            GOTO 15
         ENDIF
         K=K+1
         X(K)=DATA(I,XCOL)
         Y(K)=DATA(I,YCOL)
         IF(LOG) THEN
            Z(K)=ALOG(DATA(I,ZCOL)+A)
         ELSE
            Z(K)=DATA(I,ZCOL)
         ENDIF
 15   CONTINUE
      IF(NOUTS.GT.0) THEN
         WRITE(*,100) NOUTS
         WRITE(*,101)K
 16      WRITE(*,102)
         READ(*,'(A1)') YORN
         IF((YORN.NE.'Y').AND.(YORN.NE.'y')) GOTO 20
         WRITE(*,*)
 100       FORMAT(4X,'N.B. ',I4,' Z-values are out of range (<1.0) and')
 101       FORMAT(4X,'  will be ignored; there remain',I4,' values;')
 102       FORMAT(4X,'  is this ok? (y/n): ',$)
        ENDIF
        NR=K
        RETURN

C-----Error trapping code
 600    WRITE(*,*)'   integers only...'
        GOTO 2
 605    WRITE(*,*)'   numeric input only...'
        GOTO 22

        END

C***************************************************************************
C***************************************************************************

        SUBROUTINE GETOUT(SAS)
C       Gets name of output file, opens

        INTEGER      XCOL,YCOL,ZCOL
        LOGICAL      THERE,OPEND,GRAPH,LOG,HAAN,SAS
        CHARACTER*1  YORN
        CHARACTER*32 FILNAM,OUTFIL,PRESOUT,LASTOUT
        CHARACTER*20 ZLABEL,FTITLE
        CHARACTER*1 HEAD1(79),HEAD2(79),HEAD3(79),HEAD4(79)

        COMMON/FILINFO/FILNAM,OUTFIL,PRESOUT,FTITLE,
     +               NCOL,NROW,HEAD1,HEAD2,HEAD3,HEAD4
        COMMON/PGMPARAM/NDIM,NR,NLAG,MLAG,NSTP,LOG,A,HAAN,
     +               XMAX,XMIN,YMAX,YMIN,XCOL,YCOL,ZCOL,
     +               ZLABEL,GRAPH

C-----Get output filename
 10     WRITE(*,100)
        READ(*,'(A32)') OUTFIL
        IF ((OUTFIL.EQ.'N').OR.(OUTFIL.EQ.'n')) THEN
           OUTFIL='NONE'
        ELSEIF ((OUTFIL.EQ.'A').OR.(OUTFIL.EQ.'a')) THEN
           INQUIRE (6,OPENED=OPEND,NAME=LASTOUT)
           IF (.NOT.OPEND) THEN
            WRITE(*,*) '   there''s no output file to append to...'
            GOTO 10
           ELSE
              OUTFIL=LASTOUT
           ENDIF
        ELSE
           INQUIRE(FILE=OUTFIL,EXIST=THERE)
           INQUIRE(6,OPENED=OPEND)
           IF (THERE) THEN
              WRITE(*,110)
              READ(*,'(A1)') YORN
              IF ((YORN.EQ.'N').OR.(YORN.EQ.'n')) THEN
                 GOTO 10
              ELSEIF ((YORN.NE.'Y').AND.(YORN.NE.'y')) THEN
               WRITE(*,*) '   answer either y or n...'
                 GOTO 10
              ENDIF
           ENDIF
           IF (OPEND) CLOSE(6)
           OPEN(6,FILE=OUTFIL,STATUS='NEW')

C--------Ask if want graphics on output file
 30        WRITE(*,120)
           READ(*,'(A1)') YORN
           IF ((YORN.EQ.'Y').OR.(YORN.EQ.'y')) THEN
              GRAPH=.TRUE.
           ELSEIF ((YORN.EQ.'N').OR.(YORN.EQ.'n')) THEN
              GRAPH=.FALSE.
           ELSE
              WRITE(*,*) '      answer either y or n...'
              GOTO 30
           ENDIF

C--------Ask if want SAS-compatible output
 40        IF(OUTFIL.NE.'NONE') THEN
```

```
            WRITE(*,130)
            READ(*,'(A1)') YORN
            IF ((YORN.EQ.'Y').OR.(YORN.EQ.'y')) THEN
               SAS=.TRUE.
            ELSEIF ((YORN.EQ.'N').OR.(YORN.EQ.'n')) THEN
               SAS=.FALSE.
            ELSE
               WRITE(*,*) '      answer either y or n...'
               GOTO 40
             ENDIF
          ENDIF
       ENDIF

 100   FORMAT(' Output file ([N]one, [A]ppend, or filename): ',$)
 110   FORMAT(' File exists;'
     +       /3X,'Do you want to write over it? (y/n): ',$)
 120   FORMAT('   Should line-graphics be written to it? (y/n): ',$)
 130   FORMAT('   Should output include SAS command file? (y/n): ',$)

 70    RETURN
       END

C***************************************************************************
C***************************************************************************

       SUBROUTINE STATS(X,N)
C      Calculates mean, std dev,... given vector X of length N
C      NB: Means and variances ARE backtransformed if LOG=.TRUE.
C      Returns values in common block: AMEAN,AVAR,NN,STDEV,CVA,CSKEW,
C         CKURT,SESKEW,SEKURT

       INTEGER   XCOL,YCOL,ZCOL
       LOGICAL   LOG,GRAPH,HAAN
       CHARACTER*20 ZLABEL
       REAL X(1000)
       REAL*8 SUMX,SUMX2,SUMX3,SUMX4,AM3,AM4

       COMMON/DSTATS/AMEAN,AVAR,NN,STDEV,CVA,CSKEW,CKURT,
     +              SESKEW,SEKURT
       COMMON/PGMPARAM/NDIM,NR,NLAG,MLAG,NSTP,LOG,A,HAAN,
     +               XMAX,XMIN,YMAX,YMIN,XCOL,YCOL,ZCOL,
     +               ZLABEL,GRAPH

C-----Initialize Variables
       SUMX=0.
       SUMX2=0.
       SUMX3=0.
       SUMX4=0.

C-----Calculate Statistics
       DO 10 I=1,N
          SUMX=SUMX+X(I)
          SUMX2=SUMX2+(X(I)**2)
          SUMX3=SUMX3+(X(I)**3)
          SUMX4=SUMX4+(X(I)**4)
 10    CONTINUE

       AN=FLOAT(N)
       AMEAN=SUMX/AN
       AVAR=SUMX2/AN-AMEAN**2
       AM3=SUMX3/AN-3.*AMEAN*SUMX2/AN+2.*AMEAN**3
       AM4=SUMX4/AN-4.*AMEAN*SUMX3/AN+6.*AMEAN**2*
     +     SUMX2/AN-3.*AMEAN**4
       CSKEW=AM3/(AVAR**1.5)
       CKURT=AM4/(AVAR**2)
       SESKEW=SQRT((6*AN*(AN-1))/((AN-2)*(AN+1)*(AN+3)))
       SEKURT=SQRT((24*AN*(AN-1)**2)/((AN-3)*(AN-2)*(AN+3)*(AN+5)))

C-----Back transform Amean and Avar if (log); if (haan) then  from Haan (1977)
       IF((LOG).AND.(HAAN)) THEN
          AMEAN=(EXP(AMEAN+(0.5*AVAR)))-A
          AVAR=((AMEAN**2)*(EXP(AVAR)-1))
       ENDIF

       STDEV=SQRT(AVAR)
       CVA=STDEV/AMEAN
       NN=N
```

```
      RETURN
      END

C*************************************************************************
C*************************************************************************

      SUBROUTINE DATREAD(READIN)

C     Prompts user for input filename, checks for integrity of name
C     and reads data from the file
C     into a common workspace called DATA(1000,15)
C     which is located in named common DATAIN.
C     The data is read in in a free format style by columns; if
C     the no. of columns is not on 3rd header record user is asked for
C     value.
C     Common area FILEINFO keeps needed information in central memory

C-----Variable list:
C
C   NAME       FUNCTION
C   ----       --------
C     NROW          rows read in from data file
C     NCOL       columns of data in data file
C     READIN     data file read yet? (logical)
C     NUMBER     to find # of cols, if any, in HEAD3
C     PLACE      keeps place in string for # of cols in HEAD3
C     FILNAM     name of input data file
C     OUTFIL     name of present output file
C     PRESOUT    name of past output file (for appending)
C     FTITLE     title of data file
C     THERE      does the user's input filename exist?
C     HEAD1      First header line of filnam (headers contain no data)
C     HEAD2      header 2
C     HEAD3      header 3
C     HEAD4      header 4
C     DATA1      first data line (aid to determine # of cols.)

      LOGICAL   THERE,READIN
      INTEGER ERROR,PLACE
      CHARACTER*32 FILNAM,OUTFIL,PRESOUT
      CHARACTER*20 FTITLE
      CHARACTER*1 HEAD1(79),HEAD2(79),HEAD3(79),HEAD4(79),DATA1(79)

      COMMON/DATAIN/DATA(1000,15)
      COMMON/FILINFO/FILNAM,OUTFIL,PRESOUT,FTITLE,
     +               NCOL,NROW,HEAD1,HEAD2,HEAD3,HEAD4

C-----Get input file
  1   WRITE(*,100)
         FILNAM=' '
         NCOL=0
         READIN=.FALSE.
  5   READ(*,'(A32)') FILNAM
      IF((FILNAM.EQ.'QUIT').OR.(FILNAM.EQ.'Quit').OR.(FILNAM.EQ.'quit'))
     +    RETURN
      INQUIRE(FILE=FILNAM, EXIST=THERE)
      IF (.NOT.THERE) THEN
         WRITE(*,105)
         WRITE(*,106)
         GOTO 5
      ENDIF
 100  FORMAT(1X,'Enter input filename: ',$)
 105  FORMAT(4X,'File does not exist: ',A16)
 106  FORMAT(4X,'try again or enter ''Quit'': ',$)

C-----Get file or run title
      WRITE(*,'(A18,$)') ' Enter run title: '
      READ(*,'(A20)') FTITLE

C-----Read header
      OPEN (5,FILE=FILNAM,STATUS='OLD')
      READ(5,'(79A1)') (HEAD1(I), I=1,79)
      READ(5,'(79A1)') (HEAD2(I), I=1,79)
      READ(5,'(79A1)') (HEAD3(I), I=1,79)
      READ(5,'(79A1)') (HEAD4(I), I=1,79)
      WRITE(*,*)
      WRITE(*,*) 'File header (top 4 records in file):'
      WRITE(*,'(1X,79A1)') (HEAD1(I), I=1,79)
      WRITE(*,'(1X,79A1)') (HEAD2(I), I=1,79)
      WRITE(*,'(1X,79A1)') (HEAD3(I), I=1,79)
      WRITE(*,'(1X,79A1)') (HEAD4(I), I=1,79)
      READ(5,'(79A1)') (DATA1(I), I=1,79)
      WRITE(*,*)
      WRITE(*,*) 'First data record:'
      WRITE(*,'(1X,79A1)') (DATA1(I),I=1,79)
      BACKSPACE 5

C-----Read number of columns in file
      NUM=0
      IPL=0
      DO 10 I=1,79
         IF (HEAD3(I).EQ.' ') THEN
            IF(IPL.GT.0) GOTO 15
            GOTO 10
         ENDIF
         ICH=ICHAR(HEAD3(I))-48
         IF ((ICH.GE.0).AND.(ICH.LE.9)) THEN
            NUM=(NUM*(10**IPL))+ICH
            IPL=IPL+1
            IF (NUM.GT.999) GOTO 15
         ENDIF
 10   CONTINUE

 15   IF ((NUM.LT.2).OR.(NUM.GT.15)) THEN
 17      WRITE(*,'(A33,$)')' Number of columns in data file: '
 16      READ(*,'(I4)',ERR=600) NUM
         IF (NUM.EQ.0) THEN
            CLOSE(5)
            GOTO 1
         ELSEIF ((NUM.LT.2).OR.(NUM.GT.15)) THEN
            WRITE(*,*)'   number of columns out of range......'
            WRITE(*,'(4X,A11,$)') 'try again: '
            GOTO 16
         ENDIF
      ENDIF
      NCOL=NUM

C-----Read data
      WRITE(*,120) NCOL,FILNAM
      J=1
 20   READ(5,*,END=25) (DATA(J,I),I=1,NCOL)
      J=J+1
      GOTO 20
 25   CLOSE(5)
      NROW=J-1
      WRITE(*,122) NROW
 120  FORMAT(/16X,'...reading ',I2,' columns from file ',A32)
 122  FORMAT(16X,'...end of file after ',I4,' values'/)

      READIN=.TRUE.
      RETURN

C-----Error trapping code
 600  WRITE(*,*)'   Value must be an integer;'
      GOTO 15
      END

C*************************************************************************
C*************************************************************************

      SUBROUTINE TEXTOUT(IDIR,SAS)

C     Writes semivariogram output to a user-specified file

      LOGICAL   LOG,TMP,GRAPH,HAAN,SAS
      CHARACTER*1  YORN
      CHARACTER*32 FILNAM,OUTFIL,PRESOUT,Q*19
      CHARACTER*20 ZLABEL,FTITLE
      CHARACTER*1 HEAD1(79),HEAD2(79),HEAD3(79),HEAD4(79)
      INTEGER XCOL,YCOL,ZCOL

      COMMON/DATACOL/X(1000),Y(1000),Z(1000)
      COMMON/VARIDAT/NCPL(1000),DIST(1000),GAMA(1000),DRIFT(1000),
     +                DIR(15),TOL(15)
      COMMON/FILINFO/FILNAM,OUTFIL,PRESOUT,FTITLE,
     +                NCOL,NROW,HEAD1,HEAD2,HEAD3,HEAD4
      COMMON/PGMPARAM/NDIM,NR,NLAG,MLAG,NSTP,LOG,A,HAAN,
     +                XMAX,XMIN,YMAX,YMIN,XCOL,YCOL,ZCOL,
     +                ZLABEL,GRAPH
      COMMON/DSTATS/AMEAN,AVAR,NN,STDEV,CVA,CSKEW,CKURT,
     +               SESKEW,SEKURT

      IF ((NDIM.EQ.1).OR.((NDIM.EQ.2).AND.(IDIR.EQ.0))) THEN
         Q='********************'
         WRITE(*,800) outfil
800      FORMAT(1X,'          ......writing to file ',A32,$)
         WRITE(6,*)
         WRITE(6,*)
         WRITE(6,'(1X,4A19)') Q,Q,Q,Q
         WRITE(6,'(1X,4A19)') Q,Q,Q,Q
         WRITE(6,*)
         IF (NDIM.EQ.1) WRITE(6,*)'Semivariogram Analysis - 1D'
         IF (NDIM.EQ.2) WRITE(6,*)'Semivariogram Analysis - 2D'
         WRITE(6,'(1X,A20)') FTITLE
         WRITE(6,'(1X,A12,A32)')'Input file: ',FILNAM
         IF(NDIM.EQ.1) WRITE(6,600) XCOL,ZCOL
         IF(NDIM.EQ.2) WRITE(6,602) XCOL,YCOL,ZCOL
         WRITE(6,605) ZLABEL
         IF (LOG) WRITE(6,625) 'Ln(z+',A,')'
         IF (.NOT.LOG) WRITE(6,626) 'No'
         WRITE(6,610) NSTP
      ENDIF

      PRESOUT=OUTFIL
      IF(NDIM.EQ.2) THEN
         DO 66 I=1,3
            WRITE(6,*)
 66      CONTINUE
      ENDIF

      IF ((NDIM.EQ.2).AND.(IDIR.EQ.0)) THEN
         WRITE(6,666)
      ELSEIF ((NDIM.EQ.2).AND.(IDIR.GT.0)) THEN
         WRITE(6,667)
         WRITE(6,668) IDIR
         WRITE(6,669) DIR(IDIR), TOL(IDIR)
      ENDIF
      WRITE(6,615) MLAG
      WRITE(6,620) NLAG

      IF(LOG) THEN
         IF(HAAN) WRITE(6,*) ' Statistics (bcktrnsf Haan):'
         IF(.NOT.HAAN) WRITE(6,*) ' Statistics (not bcktrnsf):'
      ELSE
         WRITE(6,*) ' Statistics:'
      ENDIF

      WRITE(6,630)AMEAN
      WRITE(6,635)STDEV
      WRITE(6,640)CVA
      WRITE(6,645)NN
      WRITE(6,650)CSKEW,SESKEW
      WRITE(6,655)CKURT,SEKURT
      WRITE(6,660)XMIN,XMAX
      IF(NDIM.EQ.2) WRITE(6,665)YMIN,YMAX
      WRITE(6,*)

      TMP=.FALSE.
      IF((NDIM.EQ.2).AND.(IDIR.GT.0)) TMP=.TRUE.
      IF(TMP) THEN
        WRITE(6,671)
      ELSE
        WRITE(6,670)
      ENDIF

      J=0
      DO 53 I=1,NLAG
         IF (NCPL(I).EQ.0) GOTO 53
         J=J+1
         IF(TMP) THEN
```

```
            WRITE(6,676)J,DIST(I),GAMA(I),DRIFT(I),NCPL(I),DIR(IDIR)
         ELSE
            WRITE(6,675)J,DIST(I),GAMA(I),DRIFT(I),NCPL(I)
         ENDIF

 53   CONTINUE

      IF(GRAPH) THEN
         WRITE(6,*)
         WRITE(6,*)
         CALL PRIMGRPH(IDIR,1)
      ENDIF

      IF(SAS.AND.(IDIR.EQ.0)) THEN
         WRITE(6,*)
         WRITE(6,*)
         WRITE(6,*) 'SAS Command File for Model Fitting;'
         WRITE(6,*)
         WRITE(6,*)'TITLE1 ''Semivariogram Best Fit Tests'';'
         WRITE(6,*)'TITLE2 ''',FTITLE,': ',ZLABEL,''';'
         WRITE(6,*)'TITLE3 ''Input Data'';'
         WRITE(6,*)'DATA TEMP;'
         IF((NDIM.EQ.1).OR.(IDIR.EQ.0)) THEN
            WRITE(6,*)'   INPUT I H GAMMA DRIFT N;'
         ELSE
            WRITE(6,*)'   INPUT I H GAMMA DRIFT N DIR;'
         ENDIF
         WRITE(6,*)'   CARDS;'
            J=0

         DO 63 I=1,NLAG
            IF (NCPL(I).EQ.0) GOTO 63
            J=J+1
            HMAX=MAX(HMAX,DIST(I))
            GMAX=MAX(GMAX,GAMA(I))
            WRITE(6,675)J,DIST(I),GAMA(I),DRIFT(I),NCPL(I)
 63      CONTINUE

         WRITE(6,*)';'
         WRITE(6,*)
         WRITE(6,*)'PROC PRINT;'
         WRITE(6,*)
         WRITE(6,*)'TITLE3 ''Linear to Range'';'
         WRITE(6,*)'PROC NLIN;'
         WRITE(6,705) GMAX*.01
705      FORMAT(1X,'   PARAMETERS C0=',G10.4)
         WRITE(6,710) GMAX*.7
710      FORMAT(1X,'               C=',G10.4)
         WRITE(6,725) HMAX*.5
725      FORMAT(1X,'               A=',G10.4,';')
         WRITE(6,720)
720      FORMAT(1X,'   BOUNDS  0<C0, 0<C, 0<A;')
         WRITE(6,*)'   _WEIGHT_=N;'
         WRITE(6,*)'   IF H<A THEN DO;'
         WRITE(6,*)'      MODEL GAMMA=H*((C-C0)/A)+C0;'
         WRITE(6,*)'   END;'
         WRITE(6,*)'   ELSE DO;'
         WRITE(6,*)'      MODEL GAMMA=C;'
         WRITE(6,*)'   END;'
         WRITE(6,*)'   OUTPUT PREDICTED=GHAT_L PARMS=C0_L C_L A_L;'
         WRITE(6,*)'PROC PLOT;'
         WRITE(6,*)'   PLOT GAMMA*H=''*'' GHAT_L*H=''L'' ',
     +             '/OVERLAY VZERO HZERO;'
         WRITE(6,*)
         WRITE(6,*)'TITLE3 ''Spherical Model'';'
         WRITE(6,*)'PROC NLIN;'
         WRITE(6,705) GMAX*.01
         WRITE(6,710) GMAX*.7
         WRITE(6,725) HMAX*.5
         WRITE(6,720)
         WRITE(6,*)'   _WEIGHT_=N;'
         WRITE(6,*)'   IF H<A THEN DO;'
         WRITE(6,*)'      MODEL GAMMA=C0+(C-C0)*(1.5*(H/A)',
     +                        '-0.5*((H/A)**3));'
         WRITE(6,*)'   END;'
         WRITE(6,*)'   ELSE DO;'
         WRITE(6,*)'      MODEL GAMMA=C;'
         WRITE(6,*)'   END;'
         WRITE(6,*)'   OUTPUT PREDICTED=GHAT_S PARMS=C0_S C_S A_S;'
         WRITE(6,*)'PROC PLOT;'
         WRITE(6,*)'   PLOT GAMMA*H=''*'' GHAT_S*H=''S'' ',
     +              '/OVERLAY VZERO HZERO;'
         WRITE(6,*)
         WRITE(6,*)'TITLE3 ''Exponential Model'';'
         WRITE(6,*)'PROC NLIN;'
         WRITE(6,705) GMAX*.01
         WRITE(6,710) GMAX*.7
         WRITE(6,725) HMAX/3.
         WRITE(6,720)
         WRITE(6,*)'   _WEIGHT_=N;'
         WRITE(6,*)'   MODEL GAMMA=C0+(C-C0)*(1-EXP(-H/A));'
         WRITE(6,*)'   OUTPUT PREDICTED=GHAT_E PARMS=C0_E C_E A_E;'
         WRITE(6,*)'PROC PLOT;'
         WRITE(6,*)'   PLOT GAMMA*H=''*'' GHAT_E*H=''E'' ',
     +              '/OVERLAY VZERO HZERO;'
         WRITE(6,*)
         WRITE(6,*)'TITLE3 ''Gaussian Model'';'
         WRITE(6,*)'PROC NLIN;'
         WRITE(6,705) GMAX*.01
         WRITE(6,710) GMAX*.7
         WRITE(6,725) HMAX*.5
         WRITE(6,720)
         WRITE(6,*)'   _WEIGHT_=N;'
         WRITE(6,*)'   MODEL GAMMA=C0+(C-C0)*',
     +                    '(1-EXP(-(H**2/(A/3**.5)**2)));'
         WRITE(6,*)'   OUTPUT PREDICTED=GHAT_G PARMS=C0_G C_G A_G;'
         WRITE(6,*)'PROC PLOT;'
         WRITE(6,*)'   PLOT GAMMA*H=''*'' GHAT_G*H=''G'' ',
     +              '/OVERLAY VZERO HZERO;'
         WRITE(6,*)
         WRITE(6,*)'PROC RSQUARE;'
         WRITE(6,*)'   MODEL GAMMA=GHAT_L GHAT_S GHAT_E GHAT_G'
         WRITE(6,*)'                /STOP=1;'
         WRITE(6,*)
         WRITE(6,*)'PROC PRINT;'
         WRITE(6,*)'  VAR  GAMMA H N GHAT_L C0_L C_L A_L'
         WRITE(6,*)'       GHAT_S C0_S C_S A_S'
         WRITE(6,*)'       GHAT_E C0_E C_E A_E'
         WRITE(6,*)'       GHAT_G C0_G C_G A_G;'
         WRITE(6,*)
         WRITE(6,*)'ENDSAS;'
      ENDIF

      WRITE(*,*)
      WRITE(*,*)
      WRITE(*,*)
      WRITE(*,*)
      RETURN

600   FORMAT(1X,'Columns X,Z: ',I3,I3)
602   FORMAT(1X,'Columns X,Y,Z: ',I3,I3,I3)
605   FORMAT(1X,'Z Label: ',A20)
610   FORMAT(1X,'Step Size:   ',I4)
615   FORMAT(1X,'Lag Maximum: ',I4)
620   FORMAT(1X,'       Used: ',I4)
625   FORMAT(1X,'Transform Z: ',A5,F5.1,A1)
626   FORMAT(1X,'Transform Z: ',A2)
630   FORMAT(1X,'   Mean    : ',F8.4)
635   FORMAT(1X,'   Std. Dev: ',F8.4)
640   FORMAT(1X,'   C.V.:     ',F8.4)
645   FORMAT(1X,'   N:        ',I4)
650   FORMAT(1X,'   Skewness: ',F8.3,'(',F8.3,')')
655   FORMAT(1X,'   Kurtosis: ',F8.3,'(',F8.3,')')
660   FORMAT(1X,'Bounds X:',F6.1,',',F6.1)
665   FORMAT(1X,'       Y:',F6.1,',',F6.1)
666   FORMAT(1X,'Isotropic Analysis')
667   FORMAT(1X,'Anisotropic Analysis')
668   FORMAT(1X,'   Direction ',I2)
669   FORMAT(1X,'   Degrees from north: ',F5.1,' (+/-',F5.1,')')
670   FORMAT(6X,'Lag#   Distance        Gamma',15X,'Drift',8X,'Couples')
671   FORMAT(6X,'Lag#   Distance        Gamma',15X,'Drift',8X,'Couples',
     +   '  Direction')
675   FORMAT(5X,I4,2X,F9.2,2X,F17.4,3X,F17.4,3X,I5)
676   FORMAT(5X,I4,2X,F9.2,2X,F17.4,3X,F17.4,3X,I5,3X,F6.1)
```

```
      END
C*******************************************************************
C*******************************************************************

      SUBROUTINE PRIMGRPH(IDIR,NUNIT)

C-----Draws primitive gamma vs. lag 2-dimensional graph
C     on 80 character x 25 line screen
C     IDIR=direction number (2d semivariograms)
C     NUNIT=file unit specifier; 0=screen, 1=current output file (6)

      LOGICAL   THERE,LOG,GRAPH,HAAN
      INTEGER   XPLACE,YPLACE,GRPH(20,70),OUT,XCOL,YCOL,ZCOL
      REAL YAXIS(25)
      CHARACTER*32 FILNAM,PRESOUT,OUTFIL
      CHARACTER*20 FTITLE,ZLABEL
      CHARACTER CHR(15)*1,Q*10
      CHARACTER*1 HEAD1(79),HEAD2(79),HEAD3(79),HEAD4(79)

C-----Common blocks
      COMMON/VARIDAT/NCPL(1000),DIST(1000),GAMA(1000),DRIFT(1000),
     +               DIR(15),TOL(15)
      COMMON/FILINFO/FILNAM,OUTFIL,PRESOUT,FTITLE,
     +               NCOL,NROW,HEAD1,HEAD2,HEAD3,HEAD4
      COMMON/PGMPARAM/NDIM,NR,NLAG,MLAG,NSTP,LOG,A,HAAN,
     +               XMAX,XMIN,YMAX,YMIN,XCOL,YCOL,ZCOL,
     +               ZLABEL,GRAPH
      COMMON/DSTATS/AMEAN,AVAR,NN,STDEV,CVA,CSKEW,CKURT,
     +              SESKEW,SEKURT

      IF(NUNIT.EQ.1) THEN
         OUT=6
      ELSE
         OUT=0
      ENDIF

C-----Initialize
      DO 80 I=1,20
         DO 80 J=1,70
            GRPH(I,J)=0
 80   CONTINUE

C-----FIND GMAX, DMAX
      GMAX=0.
      DO 10 I=1,NLAG
         IF(GAMA(I).GT.GMAX) GMAX=GAMA(I)
 10   CONTINUE
      DMAX=FLOAT(NLAG)
      DMAX=DMAX+0.05*DMAX
      GMAX=GMAX+0.05*GMAX


C-----CREATE SCREEN MATRIX GRPH
      DO 250 I=1,NLAG
         IF (NCPL(I).EQ.0) GOTO 250
         XPLACE=NINT((DIST(I)/DMAX)*70.)
         YPLACE=21-NINT((GAMA(I)/GMAX)*20.)
         GRPH(YPLACE,XPLACE)=GRPH(YPLACE,XPLACE)+1
 250  CONTINUE


C-----CREATE Y-AXIS VECTOR
      K=0
      DO 260 I=20,1,-1
         K=K+1
         IF(INT(FLOAT(I)/4.).EQ.FLOAT(I)/4.) THEN
            YAXIS(K)=FLOAT(I)*GMAX/20.
         ELSE
            YAXIS(K)=-99.
         ENDIF
 260  CONTINUE
      YAXIS(20)=-99.

C-----PRINT TOP 20 ROWS (GRPH MATRIX)
      WRITE(OUT,101)
      WRITE(OUT,*)
      DO 265 I=1,20
         IF (YAXIS(I).NE.-99.) THEN
            WRITE(OUT,100) YAXIS(I),
         ELSE
            WRITE(OUT,101)
         ENDIF
         DO 270 J=1,70
           IF ((GRPH(I,J).GE.1).AND.(GRPH(I,J).LE.9)) THEN
             WRITE(OUT,105) GRPH(I,J),
           ELSEIF (GRPH(I,J).GT.9) THEN
             WRITE(OUT,106) '#',
           ELSEIF (GRPH(I,J).EQ.0) THEN
             WRITE(OUT,106) ' ',
           ENDIF
 270     CONTINUE
          WRITE(OUT,*)
 265  CONTINUE

 100  FORMAT(1X,G6.1,'>'\)
 101  FORMAT(1X,6X,'+'\)
 105  FORMAT(I1\)
 106  FORMAT(A1\)

C------X AXIS AND LINES BELOW
      Q='^+++++++++'
      WRITE(OUT,110)
      WRITE(OUT,111) Q,Q,Q,Q,Q,Q,Q
      WRITE(OUT,115)
      DO 280 I=1,6
         WRITE(OUT,116) FLOAT(I)*DMAX/7.
 280  CONTINUE
      WRITE(OUT,*)
      WRITE(OUT,117)
 110  FORMAT(1X,6X\)
 111  FORMAT(7A10)
 115  FORMAT(1X,6X,' 0      '\)
 116  FORMAT('  ',G8.2\)
 117  FORMAT(1X,6X,30X,'lag (h)')

      IF(NDIM.EQ.1) THEN
         WRITE(OUT,120)
      ELSEIF (NDIM.EQ.2) THEN
         IF (IDIR.EQ.0) THEN
            WRITE(OUT,125)
         ELSE
            WRITE(OUT,130) DIR(IDIR),TOL(IDIR)
         ENDIF
      ENDIF
 120  FORMAT(1X,6X,20X,'Single-Dimension Analysis'\)
 125  FORMAT(1X,6X,23X,'Isotropic Analysis'\)
 130  FORMAT(1X,6X,12X,'Anisotropic Analysis (',
     +        F5.1,' +/-',F4.1,' dgrs N)'\)

C-----Hold screen until <cr>
      IF(OUT.EQ.0) THEN
         WRITE(OUT,'(\)')
         READ(OUT,'(A1)') Q
         DO 300 I=1,25
           WRITE(OUT,*)
 300     CONTINUE
      WRITE(OUT,*)
      ENDIF

      RETURN
      END

C**********************************************************************
C**********************************************************************
C**********************                      ***************************
C********************** End of Semivar.for ***************************
C**********************                      ***************************
C**********************************************************************
C**********************************************************************
```

Semivar.for Output file Semivar.out

```
 ****************************************************************************
 ****************************************************************************

 Semivariogram Analysis - 2D
 Sample run
 Input file: Sample.prn
 Columns X,Y,Z:   1  2  3
 Z Label: Zvariate
 Transform Z: Ln(z+  1.0)
 Step Size:      5



 Isotropic Analysis
 Lag Maximum:  113
        Used:   70
  Statistics (bcktrnsf Haan):
    Mean    :    .3972
    Std. Dev:    .0571
    C.V.:        .1438
    N:         248
    Skewness:     .790(     .155)
    Kurtosis:    4.064(     .308)
 Bounds X:     .2,  79.9
        Y:     .0,  79.4

      Lag#  Distance        Gamma             Drift        Couples
        1      1.90              .0127              .0216      34
        2      5.53              .0135              .0041     620
        3     10.15              .0157             -.0034    1267
        4     15.03              .0159              .0049    1723
        5     20.03              .0183              .0016    2114
        6     25.05              .0179              .0058    2337
        7     30.00              .0189              .0120    2520
        8     34.99              .0199              .0135    2636
        9     39.99              .0201              .0148    2650
       10     44.99              .0227              .0213    2617
       11     49.92              .0233              .0225    2433
       12     54.93              .0244              .0156    2346
       13     59.95              .0249              .0231    2060
       14     64.89              .0231              .0308    1752
       15     68.71              .0219              .0129     756


      +
.3E-01>
      +                                                    1    1
      +                                                1            1
      +                                           1                    1
.2E-01>
      +                                 1    1
      +                   1    1    1
      +
.2E-01>         1   1
      +
      + 1  1
      +
.1E-01>
      +
      +
      +
.5E-02>
      +
      +
      +
      ^+++++++++^+++++++++^+++++++++^+++++++++^+++++++++^+++++++++^+++++++++
       0        11.       21.       32.       42.       53.       63.
                                    lag (h)
                             Isotropic Analysis
```

SAS Command File for Model Fitting;

```
TITLE1 'Semivariogram Best Fit Tests';
TITLE2 'Sample run          : Zvariate            ';
TITLE3 'Input Data';
```

```
DATA TEMP;
   INPUT I H GAMMA DRIFT N;
   CARDS;
      1     1.90           .0127           .0216     34
      2     5.53           .0135           .0041    620
      3    10.15           .0157          -.0034   1267
      4    15.03           .0159           .0049   1723
      5    20.03           .0183           .0016   2114
      6    25.05           .0179           .0058   2337
      7    30.00           .0189           .0120   2520
      8    34.99           .0199           .0135   2636
      9    39.99           .0201           .0148   2650
     10    44.99           .0227           .0213   2617
     11    49.92           .0233           .0225   2433
     12    54.93           .0244           .0156   2346
     13    59.95           .0249           .0231   2060
     14    64.89           .0231           .0308   1752
     15    68.71           .0219           .0129    756
;

PROC PRINT;

TITLE3 'Linear to Range';
PROC NLIN;
   PARAMETERS C0= .2490E-03
              C= .1743E-01
              A= 34.36    ;
   BOUNDS  0<C0, 0<C, 0<A;
   _WEIGHT_=N;
   IF H<A THEN DO;
      MODEL GAMMA=H*((C-C0)/A)+C0;
   END;
   ELSE DO;
      MODEL GAMMA=C;
   END;
   OUTPUT PREDICTED=GHAT_L PARMS=C0_L C_L A_L;
PROC PLOT;
   PLOT GAMMA*H='*' GHAT_L*H='L' /OVERLAY VZERO HZERO;

TITLE3 'Spherical Model';
PROC NLIN;
   PARAMETERS C0= .2490E-03
              C= .1743E-01
              A= 34.36    ;
   BOUNDS  0<C0, 0<C, 0<A;
   _WEIGHT_=N;
   IF H<A THEN DO;
      MODEL GAMMA=C0+(C-C0)*(1.5*(H/A)-0.5*((H/A)**3));
   END;
   ELSE DO;
      MODEL GAMMA=C;
   END;
   OUTPUT PREDICTED=GHAT_S PARMS=C0_S C_S A_S;
PROC PLOT;
   PLOT GAMMA*H='*' GHAT_S*H='S' /OVERLAY VZERO HZERO;

TITLE3 'Exponential Model';
PROC NLIN;
   PARAMETERS C0= .2490E-03
              C= .1743E-01
              A= 22.90    ;
   BOUNDS  0<C0, 0<C, 0<A;
   _WEIGHT_=N;
   MODEL GAMMA=C0+(C-C0)*(1-EXP(-H/A));
   OUTPUT PREDICTED=GHAT_E PARMS=C0_E C_E A_E;
PROC PLOT;
   PLOT GAMMA*H='*' GHAT_E*H='E' /OVERLAY VZERO HZERO;

TITLE3 'Gaussian Model';
PROC NLIN;
   PARAMETERS C0= .2490E-03
              C= .1743E-01
              A= 34.36    ;
   BOUNDS  0<C0, 0<C, 0<A;
   _WEIGHT_=N;
   MODEL GAMMA=C0+(C-C0)*(1-EXP(-(H**2/(A/3**.5)**2)));
   OUTPUT PREDICTED=GHAT_G PARMS=C0_G C_G A_G;
```

```
PROC PLOT;
   PLOT GAMMA*H='*' GHAT_G*H='G' /OVERLAY VZERO HZERO;

PROC RSQUARE;
   MODEL GAMMA=GHAT_L GHAT_S GHAT_E GHAT_G
               /STOP=1;

PROC PRINT;
  VAR  GAMMA H N GHAT_L CO_L C_L A_L
       GHAT_S CO_S C_S A_S
       GHAT_E CO_E C_E A_E
       GHAT_G CO_G C_G A_G;

ENDSAS;
```

File Punctual.hlp; to accompany

Program Punctual.for,
Punctual Kriging Analysis
Version 11.10.86 Copyright (c) 1986 G.P. Robertson

The correct citation for this program is
Robertson, G.P. 1987. Geostatisitics in ecology: interpolating with known variance. Ecology 68:

Later releases of this program available from: Computer Services Laboratory
W.K. Kellogg Biol. Station
Michigan State University
Hickory Corners, MI 49060.

Program parameters:

1. Hardware requirements. This version was written in Microsoft Fortran version 3.3 designed to run under the IBM MS-DOS 2.0+ operating system. Compilation by other compilers compatible with the ANSI 77 standard should be possible.

2. Data Input. Data is read into the program as ASCII characters in free format from a user-specified file. The first 4 lines of the input file are treated as header lines -- they should not contain data. If the first characters of the third line are numeric, they are assumed to represent the number of columns in each data record. A sample header:

```
    Line 1: File xxxxx.prn from supporting files xxyyy.ext and xxxyy.ext;
         2: Small grid samples taken 5/19.....
         3: 15 columns: Sample#, Xcor, Ycor, z1,z2,z3,z4,z5,
         4:          z6,z7,z8,z9,z10,z11,z12.
         5: 1 0 15 4.3 2.11 .0003 -99. 2.3 3 5 4 3 2 2 1.9997
```

   If the column no. value on line 3 is not correct or if there are different numbers of columns in any row, the data will not be read correctly and calculations will be incorrect. Missing values are represented by the value -99. Data size limitations are 15 columns x 1000 rows, though redimensioning key variables will allow indefinite input file sizes.

3. Output. Output data is written to a user-specified output file or to the screen if no output file is specified. All data is ASCII format and can be subsequently printed or read into another program.

4. User-specified prompts:

   a) Main menu; a) read data file; see limitations above;
      b) scan curent data set; use cntrl-s to halt scrolling;
      c) punctual kriging analysis;
      d) neighborhood analysis; use to define optimal neighborhood parameters for punctual analysis; this routine uses jacknifing to explore reduced error with a range of different neighborhood parameters; requires substantial CPU time;
      e) help; this output;
      f) exit program;

   b) input file name; conform to MSDOS convention; supports directory paths;

   c) run title; as desired for labeling output graphs; 64 char. max;

   d) number of dimensions; either 1 or 2;

   e) x,z or x,y,z columns: columns in which x coordinates, y coordinates, and active z-value can be found;

   f) z-label: as desired for labeling output, 64 char. max;

   g) use subset of data set; to use only part of the input (measured) data grid in the interpolation analysis;

   h) transform z-value; if requested, z is logn-transformed with the option of adding a constant to each value before transforming; because of the discontinuous nature of this distribution over the range <-1 to >1, all values <1.0 are ignored for the duration of the analysis; note that the transformation lasts only until the next Punctual menu; the statistics output give back-transformed values (Haan 1977) as requested -- see note below;

i) specify semivariogram model; 4 models are available: linear to specified range, spherical, exponential, and gaussian. Model parameters are C0 (y-intercept), C (sill value), and A (range in the case of the linear and spherical models, some factor of range in exponential and gaussian models). These model parameters can be taken directly from the output of the semivariogram program after running it through SAS's nonlinear estimation routines;

j) neighborhood parameters; unless neighborhood analysis is requested, user must provide a) the maximum number of neighbors M to use for interpolating any given grid point (100 max), and b) the maximum radius within which to search for neighbors (up to the maximum distance separating any two points to be interpolated). All neighbors within the maximum radius are sorted by distance and the nearest M are used for the interpolation.

k) interpolation grid definition; the coordinate points making up the interpolation grid can be either 1) calculated as a rectangular grid with user-specified increments between x and between y interpolation points; 2) read from a separate file containing the coordinate points (the file should contain 4 header lines, and an indefinite number of records containing either 1 or 2 values each for the x and (if two dimensional) y coordinates); 3) the same as the coordinate points for the input data set with normal interpolation (self-fit); or 4) the same as the coordinate points for the input data set but interpolation jacknifed.

l) use subset of interpolation grid; if only part of the interpolation grid is to be calculated; note that points in the measured data grid that are outside the interpolation grid may be used in the interpolatin of these points if they satisfy the neighborhood requirements.

j) specify output file; N = results are not read to output file, appear on screen only;
A = appends results to last output file opened;
f = name of new output file; if file exists, can write over it or append to it as specified;

N.B. Read carefully the annotations on the output file; where the backtransformation of transformed values was requested, Haan's (1977) method is used, where the backtransformed mean = exp([mean of logn values]+0.5*var of logn values)]-A; and the backtransformed variance = ([mean of untransformed values]^2)*(exp[var of logn values]-1). Where A = the constant added to original values before ln transformation in order to bring original values to >1.0 before ln transformation. (See Haan, C.T. 1977. Chapter 6. Some continuous probability distributions. Statistical Methods in Hydrology. The Iowa State Univ. Press, Ames, Iowa.)

```
$storage:2

      PROGRAM PUNCTUAL


C     Version 11.10.86
C     Copyright (c) 1986 G.P. Robertson. All rights reserved.

C     The correct citation for this program is Robertson, G.P. 1987.
C         Geostatistics in ecology: interpolating with known variance.
C         Ecology 68:(in press).
C
C     This program is being maintained at the Computer Services Laboratory
C                                             W.K. Kellogg Biological Station
C                                             Michigan State University
C                                             Hickory Corners, MI 49060.
C
C        Please report significant bugs and well-documented enhancements
C        to this address for inclusion in later versions.

C     Documention for users appears in file Punctual.hlp.

C     Subroutines: DATREAD, SCAN, UCASE, LCASE, KRIGE, NBHDANAL, SETPUN,
C                  DEFPTS, TRANSFORM, BOUNDS, INTBOUNDS, STATS, GETOUT,
C                  DEFMODEL, DEFNEIGH, TEXT1OUT, GETNBRS, SORT, NEXTPT,
C                  SIMUL, CALCBG, CALCFN

C-----Variable list
C
C  NAME        FUNCTION
C  ----        --------
C  FILNAM      input data file
C  OUTFIL      file where output data sent
C  PRESOUT     last output file named (used in appending)
C  FTITLE      run title
C  ZLABEL      z-variable title
C  XCOL        column in filnam with x-coordinate values
C  YCOL        column with y-coordinate values
C  ZCOL        column with z-corrdinate values
C  XMAX        greatest x-coordinate
C  XMIN        least x-coordinate
C  YMAX        greatest y-coordinate
C  YMIN        least y-coodinate
C  GXMAX       greatest interpolation grid x-coord
C  GYMAX       greatest interp. grid y-coord
C  GXMIN
C  GYMIN
C  LOG         log transformation? (logical)
C  HAAN        back-tranform using Haan (1977) (logical)
C  NR          number of valid data elements
C  XINC        x interpolation increment
C  YINC        y interpolation increment
C  RNBHD       radius of neighborhood (user defined)
C  RNBMAX      maximum radius neighborhood (data defined)
C  MAXNBR      maximum no. of neighbors (user defined)
C  MODEL       semivariogram model
C  CO          y-intercept of semivariogram
C  C9          C of semivariogram model
C  A9          A of semivariogram model
C  HRDGRD      true if interpolation grid coord. to be read from file
C  HRDFIL      name of file if hrdgrd true
C  JACK        true if jacknife
C  SELF        true if fitting only input data coordinates


      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4        DATA
      INTEGER       CHOICE,XCOL,YCOL,ZCOL
      CHARACTER*1   HEAD1,HEAD2,HEAD3,HEAD4,LINE(79),YORN
      CHARACTER*64  FILNAM,OUTFIL,LSTOUT,FTITLE,ZLABEL,HRDFIL
      LOGICAL       TWODIM,LOG,HAAN,HERC,CHECK,BOUND,READIN,HRDGRD,
     +              OPEND,JACK,SELF,THERE

      COMMON/DATAIN/DATA(1000,15)
      COMMON/DATACOLS/X(1000),Y(1000),Z(1000)
      COMMON/FILINFO/FILNAM,OUTFIL,LSTOUT,FTITLE,NCOL,NROW,
     +                HEAD1(79),HEAD2(79),HEAD3(79),HEAD4(79)
```

```
      COMMON/GENPARAM/NR,NRGRID,TWODIM,HERC,CHECK,HAAN,LOG,BOUND,
     +                OFF,XCOL,YCOL,ZCOL,ZLABEL,TZMN,TZVAR,UZMN,UZVAR,
     +                BZMN,BZVAR,XMIN,XMAX,YMIN,YMAX,GXMIN,GXMAX,
     +                GYMIN,GYMAX,NRTOT
      COMMON/PUNPARAM/XINC,YINC,RNBHD,RNBMAX,MAXNBR,MODEL,
     +                C0,C9,A9,HRDGRD,HRDFIL,JACK,SELF


C-----Write menu header
      DO 3 I=1,5
         WRITE(*,*)
 3    CONTINUE
      WRITE(*,*)'Program Punctual'
      WRITE(*,*)'  Punctual Kriging Analysis'
      WRITE(*,*)'  Version 11.10.86'
      WRITE(*,*)' '
      WRITE(*,*)'See file Punctual.hlp and source code listing'
      WRITE(*,*)'  for documentation and copyright information'
      WRITE(*,'(//1X,A,$)')'Hit <return> to continue.....'
      READ(*,'(A1)') YORN
      WRITE(*,'(/////////////////////////1X)')

      READIN=.FALSE.
C-----Choose function
 1    WRITE(*,*)
      CHECK=.FALSE.
      IF (.NOT.READIN) THEN
         WRITE(*,400) 'None'
      ELSE
         WRITE(*,400) FILNAM
         WRITE(*,405) NCOL
         WRITE(*,410) NROW
      ENDIF
 400  FORMAT(/1X,'Current file: ',A32)
 405  FORMAT(1X,'   number of columns:',I3)
 410  FORMAT(1X,'   number of rows:    ',I5)

      WRITE(*,*)
      WRITE(*,*)'Choose a function:'
      WRITE(*,*)'    1. Read in data file'
      WRITE(*,*)'    2. Scan current data set'
      WRITE(*,*)'    3. Neighborhood estimation analysis'
      WRITE(*,*)'    4. Punctual kriging analysis'
      WRITE(*,*)'    5. Help'
      WRITE(*,*)'    6. End Program'
      WRITE(*,*)
      WRITE(*,'(4X,A14,$)')'Enter choice: '
   2  READ(*,*,ERR=600) CHOICE

      IF (CHOICE.EQ.1) THEN
         CALL DATREAD(READIN)
         IF(.NOT.READIN) GOTO 1
      ELSEIF (CHOICE.EQ.2) THEN
         IF(.NOT.READIN) CALL DATREAD(READIN)
         IF(.NOT.READIN) GOTO 1
         CALL SCAN
      ELSEIF (CHOICE.EQ.3) THEN
         IF (.NOT.READIN) CALL DATREAD(READIN)
         IF(.NOT.READIN) GOTO 1
         CALL NBHDANAL
      ELSEIF (CHOICE.EQ.4) THEN
         IF (.NOT.READIN) CALL DATREAD(READIN)
         IF(.NOT.READIN) GOTO 1
         CALL KRIGE
      ELSEIF (CHOICE.EQ.5) THEN
C---------SEE IF HELP FILE PRESENT
         INQUIRE(FILE='PUNCTUAL.HLP', EXIST=THERE)
         IF (THERE) THEN
            DO 12 I=1,10
              WRITE(*,*)
12          CONTINUE
            OPEN (7,FILE='PUNCTUAL.HLP',STATUS='OLD')
10          DO 11 J=1,24
              READ (7,'(79A1)',END=20) (LINE(I),I=1,79)
              WRITE(*,'(1X,79A1)') (LINE(I),I=1,79)
11          CONTINUE
            WRITE(*,'(20X,A,$)')'(.....hit <return> to continue.....)'
```

```fortran
             READ(*,'(A1)') YORN
             GOTO 10
 20          CLOSE(7)
             WRITE(*,'(/1X,A,$)')'(.....hit <return> to continue.....)'
             READ(*,'(A1)') YORN
             WRITE(*,'(//////////////////////////1X)')
           ELSE
             WRITE(*,*)'   file Punctual.hlp not in directory...'
         ENDIF
       ELSEIF (CHOICE.EQ.6) THEN
         INQUIRE(6,OPENED=OPEND)
         IF(OPEND) CLOSE(6)
         WRITE(*,*)
         WRITE(*,*)'Program Punctual;'
         WRITE(*,*)'Normal termination.'
         WRITE(*,*)
         GOTO 999
      ELSE
         GOTO 600
      ENDIF

      GOTO 1

C-----Error trapping
 600  WRITE(*,*) '      entry error......'
 602  WRITE(*,'(7X,A,$)') 'try again: '
      GOTO 2

 999  END

C***************************************************************************
C***************************************************************************

      SUBROUTINE DATREAD(READIN)

C     Data is read in in a free format style by columns; if
C     the no. of columns is not on 3rd header record user is asked for
C     value.

C     NROW       rows read in from data file (total potential z-points)
C     NRTOT      no. of measured z-points in input dataset
C     NR         no. of valid z-points in measured grid used
C     NRGRID     no. of valid measured z-points in interpol. grid
C     NCOL       columns of data in data file
C     READIN     data file read yet?
C     NUM        to find # of cols, if any, in HEAD3
C     FILNAM     name of input data file
C     OUTFIL     name of present output file
C     LSTOUT    name of past output file (for appending)
C     FTITLE     title of data file
C     HEAD1etc   First header line of filnam (no x,y,z data in header)

      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4        DATA
      INTEGER       XCOL,YCOL,ZCOL
      CHARACTER*64  TMPNAM,FILNAM,OUTFIL,LSTOUT,FTITLE,ZLABEL
      CHARACTER*1   HEAD1,HEAD2,HEAD3,HEAD4,DATA1(79),YORN
      LOGICAL       TWODIM,THERE,READIN,LOG,HAAN,HERC,CHECK,BOUND

      COMMON/DATAIN/DATA(1000,15)
      COMMON/FILINFO/FILNAM,OUTFIL,LSTOUT,FTITLE,NCOL,NROW,
     +                 HEAD1(79),HEAD2(79),HEAD3(79),HEAD4(79)
      COMMON/GENPARAM/NR,NRGRID,TWODIM,HERC,CHECK,HAAN,LOG,BOUND,
     +                  OFF,XCOL,YCOL,ZCOL,ZLABEL,TZMN,TZVAR,UZMN,UZVAR,
     +                  BZMN,BZVAR,XMIN,XMAX,YMIN,YMAX,GXMIN,GXMAX,
     +                  GYMIN,GYMAX,NRTOT

C-----Get input file
  1   WRITE(*,*)
  6      WRITE(*,100)
  5      READ(*,'(A64)') TMPNAM
         CALL UCASE(TMPNAM)
         IF((TMPNAM.EQ.'QUIT').OR.(TMPNAM.EQ.'EXIT').OR.
     +      (TMPNAM.EQ.'END').OR.(TMPNAM(1:2).EQ.'NO')) RETURN

C-----See if file exists
      INQUIRE(FILE=TMPNAM,EXIST=THERE)
      IF (.NOT.THERE) THEN
         WRITE(*,105)
         WRITE(*,106)
         GOTO 5
      ENDIF
  100 FORMAT(1X,'Enter input filename: ',$)
  105 FORMAT(4X,'File does not exist; ',A20)
  106 FORMAT(4X,'try again or enter ''Quit'': ',$)

C-----Get file or run title
      WRITE(*,'(A18,$)') ' Enter run title: '
      READ(*,'(A64)') FTITLE

C-----Read header
      OPEN (5,FILE=TMPNAM,STATUS='OLD',ERR=605)
      READ(5,'(79A1)',ERR=605) (HEAD1(I), I=1,79)
      READ(5,'(79A1)',ERR=605) (HEAD2(I), I=1,79)
      READ(5,'(79A1)',ERR=605) (HEAD3(I), I=1,79)
      READ(5,'(79A1)',ERR=605) (HEAD4(I), I=1,79)
      WRITE(*,*)
      WRITE(*,*) 'File header (top 4 records in file):'
      WRITE(*,'(1X,79A1)') (HEAD1(I), I=1,79)
      WRITE(*,'(1X,79A1)') (HEAD2(I), I=1,79)
      WRITE(*,'(1X,79A1)') (HEAD3(I), I=1,79)
      WRITE(*,'(1X,79A1)') (HEAD4(I), I=1,79)
      READ(5,'(79A1)',ERR=605) (DATA1(I), I=1,79)
      WRITE(*,*)
      WRITE(*,*) 'First data record:'
      WRITE(*,'(1X,79A1)') (DATA1(I),I=1,79)
      WRITE(*,*)
      BACKSPACE 5

C-----Read number of columns in file
      NUM=0
      IPL=0
      DO 10 I=1,79
         IF (HEAD3(I).EQ.' ') THEN
            IF(IPL.GT.0) GOTO 15
            GOTO 10
         ENDIF
         ICH=ICHAR(HEAD3(I))-48
         IF ((ICH.GE.0).AND.(ICH.LE.9)) THEN
            NUM=(NUM*(10**IPL))+ICH
            IPL=IPL+1
            IF (NUM.GT.999) GOTO 15
         ENDIF
   10 CONTINUE

   15 IF ((NUM.LT.2).OR.(NUM.GT.15)) THEN
   17    WRITE(*,'(A33,$)')' Number of columns in data file: '
   16    READ(*,'(I4)',IOSTAT=IERR) NUM
         IF(IERR.NE.0) THEN
            WRITE(*,*)'    value must be an integer......'
            WRITE(*,'(4X,A11,$)') 'try again: '
            GOTO 16
         ENDIF
         IF (NUM.EQ.0) GOTO 1
         IF ((NUM.LT.2).OR.(NUM.GT.15)) THEN
            WRITE(*,*)'    number of columns out of range......'
            WRITE(*,'(4X,A11,$)') 'try again: '
            GOTO 16
         ENDIF
      ENDIF
      NCOL=NUM

C-----Read data
      WRITE(*,120) NCOL,TMPNAM
      DO 20 J=1,1000
      READ(5,*,END=25,ERR=605) (DATA(J,I),I=1,NCOL)
   20 CONTINUE
         CLOSE(5)
         WRITE(*,165)
  165    FORMAT(' ...>1000 values;',/1X,
     +          '     should remainder be dropped (y/n): ',$)
         READ(*,'(A1)') YORN
         IF((YORN.NE.'Y').AND.(YORN.NE.'y')) GOTO 6
   25 CLOSE(5)
      NROW=J-1
      WRITE(*,122) NROW
  120 FORMAT(/16X,'...reading ',I2,' columns from file ',A32)
  122 FORMAT(16X,'...end of file after ',I4,' values'/)

      FILNAM(1:1)=TMPNAM(1:1)
      CALL LCASE(TMPNAM)
      FILNAM(2:64)=TMPNAM(2:64)
      READIN=.TRUE.

      WRITE(*,*)
      RETURN

C-----Error trapping code
  605 WRITE(*,*)'                 ...file error......',CHAR(7),CHAR(7)
      WRITE(*,110)
      READIN=.FALSE.
  110 FORMAT(4X,'Enter input filename (or ''Quit''): ',$)
      GOTO 5

      END

C***********************************************************************
C***********************************************************************

      SUBROUTINE SCAN

      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 DATA
      CHARACTER*64 FILNAM,OUTFIL,LSTOUT,FTITLE
      CHARACTER*1  HEAD1,HEAD2,HEAD3,HEAD4

      COMMON/DATAIN/DATA(1000,15)
      COMMON/FILINFO/FILNAM,OUTFIL,LSTOUT,FTITLE,NCOL,NROW,
     +               HEAD1(79),HEAD2(79),HEAD3(79),HEAD4(79)

C-----Find out how many rows to scan
      WRITE(*,*)
    1 WRITE(*,'(A,$)')' Number of rows to scan: '
   16 READ(*,'(I4)',ERR=600) N
      IF(N.GT.NROW) N=NROW
      WRITE(*,*)
      IF(N.LE.0) RETURN

C-----Print n lines of file
C-----Start with 4 header lines
      WRITE(*,'(1X,79A1)') (HEAD1(I), I=1,79)
      WRITE(*,'(1X,79A1)') (HEAD2(I), I=1,79)
      WRITE(*,'(1X,79A1)') (HEAD3(I), I=1,79)
      WRITE(*,'(1X,79A1)') (HEAD4(I), I=1,79)

C-----Continue with all data lines
      DO 10 I=1,N
         WRITE(*,*)
         DO 20 J=1,NCOL
            A=FLOAT(J)
            IF ((A/8).EQ.INT(A/8)) THEN
               WRITE(*,*)
               WRITE(*,'(1X,A,$)') '          '
            ENDIF
            WRITE(*,100) DATA(I,J)
  100       FORMAT(' ',G10.3,$)
   20    CONTINUE
   10    CONTINUE
      WRITE(*,*)

      RETURN

C-----Error trapping code
  600 WRITE(*,*)'   Value must be an integer;'
      WRITE(*,'(4X,A11,$)') 'try again: '
      GOTO 16
      END

C***********************************************************************
C***********************************************************************
```

```
      SUBROUTINE UCASE(STRING)

      CHARACTER*64 STRING

C-----Convert lower case letters to uppercase
      DO 10 I=1,64
         IC=ICHAR(STRING(I:I))
         IF((IC.GE.97).AND.(IC.LE.122)) THEN
            STRING(I:I)=CHAR(IC-32)
         ENDIF
 10   CONTINUE

      RETURN
      END

C***********************************************************************
C***********************************************************************

      SUBROUTINE LCASE(STRING)

      CHARACTER*64 STRING

C-----Convert uppercase letters to lowercase
      DO 10 I=1,64
         IC=ICHAR(STRING(I:I))
         IF((IC.GE.65).AND.(IC.LE.90)) THEN
           STRING(I:I)=CHAR(IC+32)
         ENDIF
 10   CONTINUE

      RETURN
      END

C***********************************************************************
C***********************************************************************

      SUBROUTINE KRIGE

      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER      XCOL,YCOL,ZCOL,OUT
      CHARACTER*1  HEAD1,HEAD2,HEAD3,HEAD4,Q,YORN
      CHARACTER*64 FILNAM,OUTFIL,LSTOUT,HRDFIL,FTITLE,ZLABEL
      LOGICAL      TWODIM,CHECK,BOUND,HERC,LOG,HAAN,ERROR,
     +             HRDGRD,YESOUT,JACK,SELF

      DIMENSION FN(101,101),BG(101),BF(101),NBRLOC(100)

      COMMON/DATACOLS/X(1000),Y(1000),Z(1000)
      COMMON/FILINFO/FILNAM,OUTFIL,LSTOUT,FTITLE,NCOL,NROW,
     +               HEAD1(79),HEAD2(79),HEAD3(79),HEAD4(79)
      COMMON/GENPARAM/NR,NRGRID,TWODIM,HERC,CHECK,HAAN,LOG,BOUND,
     +                OFF,XCOL,YCOL,ZCOL,ZLABEL,TZMN,TZVAR,UZMN,UZVAR,
     +                BZMN,BZVAR,XMIN,XMAX,YMIN,YMAX,GXMIN,GXMAX,
     +                GYMIN,GYMAX,NRTOT
      COMMON/PUNPARAM/XINC,YINC,RNBHD,RNBMAX,MAXNBR,MODEL,
     +                C0,C9,A9,HRDGRD,HRDFIL,JACK,SELF

C-----If first time through, see if need to call setpun
      IF(XCOL.GT.0) THEN
         WRITE(*,105)
 105     FORMAT(/1X,'Reset run parameters? (y/n): ',$)
         READ(*,'(A1)') YORN
      ELSE
         YORN='Y'
      ENDIF

 31   IF ((YORN.EQ.'Y').OR.(YORN.EQ.'y')) THEN
C--------Set run parameters: x,y,z cols, data grid, etc.
         CALL SETPUN

      ELSEIF((YORN.NE.'N').AND.(YORN.NE.'n')) THEN
         WRITE(*,*)'          answer Y or N only......'
         WRITE(*,'(7X,A11,$)') 'try again: '
         READ(*,'(A1)') YORN
         GOTO 31
      ENDIF

C-----Ask for how to define neighbor search
      CALL DEFNEIGH(.FALSE.)

C-----Determine how to get interpolation points
      CALL DEFPTS

C-----Determine interpolation grid boundaries (not dataset bounds)
      CALL INTBOUNDS(HRDGRD)

C-----Calculate stats
      CALL STATS

C-----Get output file
      CALL GETOUT(OUT)
      IF(OUT.NE.0) YESOUT=.TRUE.

C-----Start writing to output file
      CALL TEXT1OUT(OUT,'P')
      IF(JACK.OR.SELF) THEN
         IF(JACK) THEN
            WRITE(OUT,*) 'Jacknife (Cross-Validation) Analysis'
            WRITE(OUT,'(1X,A,A)') 'NB: Measured values for a given ',
     +             'coordinate point are NOT used in the interpolated'
            WRITE(OUT,*) '     estimate for that coordinate point.'
         ELSE
            WRITE(OUT,*) 'Self-fit Analysis'
            WRITE(OUT,'(1X,A,A)') 'NB: Measured values for a given ',
     +                'coordinate point ARE used in the interpolated'
            WRITE(OUT,*) '     estimate for that coordinate point.'
         ENDIF
         WRITE(OUT,'(1X,A,A)') 'NB: Analysis includes only those ',
     +            'measured points within the interpolation bounds'
         IF(LOG.AND.(.NOT.HAAN)) WRITE(OUT,'(5X,A,A)') 'Neither ',
     +        'measured nor interpolated values are backtransformed'
         IF(TWODIM) THEN
            WRITE(OUT,280)
         ELSE
            WRITE(OUT,282)
         ENDIF
      ELSE
         IF(TWODIM) THEN
            WRITE(OUT,290)
         ELSE
            WRITE(OUT,292)
         ENDIF
      ENDIF
 280  FORMAT(/1X,4X,'X',10X,'Y',10X,'Measured Z',8X,'Estimated Z',
     +       8X,'Std Dev',4X,'Neighbors')
 282  FORMAT(/1X,4X,'X',10X,'Measured Z',8X,'Estimated Z',
     +       8X,'Std Dev',7X,'Neighbors')
 290  FORMAT(1X,4X,'X',10X,'Y',12X,'Est Z',8X,'Std Dev',
     +       11X,'Neighbors')
 292  FORMAT(1X,4X,'X',12X,'Est Z',8X,'Std Dev',
     +       11X,'Neighbors')

C-----Give first part of status report
      IF(YESOUT) THEN
         Q=CHAR(8)
         WRITE(*,*)
         WRITE(*,'(1X,A29,A16,$)') '    ...calculating grid point',
     +        '                '
         IF(TWODIM) WRITE(*,602) '        '
 602     FORMAT('+',A8,$)
      ENDIF

C-----Initialize gridpoints
      IF (HRDGRD) THEN
 40      OPEN(7,FILE=HRDFIL,STATUS='OLD',IOSTAT=IERR)
         IF (IERR.NE.0) THEN
           WRITE(*,*) '           ...file error file ',HRDFIL,CHAR(7),CHAR(7)
           WRITE(*,'(1X,A,$)') '      ...continue? (y/n): '
           READ(*,'(A1)') YORN
           IF((YORN.EQ.'Y').OR.(YORN.EQ.'y')) GOTO 40
           RETURN
         ENDIF
      ELSE
         IF(TWODIM) THEN
            XPT=GXMIN
            YPT=GYMIN-YINC
         ELSE
            XPT=GXMIN-XINC
            YPT=GYMIN
         ENDIF
      ENDIF

C-----Initialize stat accumulators
      I=0
      NZ=0
      ZESUM=0
      ZESUM2=0
      ZVSUM=0
      ZSSUM=0
      ZISUM=0
      ZISUM2=0
      ZEZI=0

C================================================================C
C                                                                C
C                   START OF MAIN KRIGING LOOP                   C
C                                                                C
C================================================================C

C-----Get gridpoint; if XPT=-99, finish up
 30   I=I+1
      IF(JACK.OR.SELF)THEN
         IF(I.GT.NR) GOTO 77
         XPT=X(I)
         IF((XPT.LT.GXMIN).OR.(XPT.GT.GXMAX)) GOTO 30
         IF(TWODIM) THEN
            YPT=Y(I)
            IF((YPT.LT.GYMIN).OR.(YPT.GT.GYMAX)) GOTO 30
         ENDIF
         ZI=Z(I)
      ELSE
         CALL NEXTPT(XPT,YPT)
         IF (XPT.EQ.-99.) GOTO 77
      ENDIF
      ZE=-99
      ZV=-99
      ZS=-99

C-----Status report to screen
         IF(YESOUT) THEN
            IF (.NOT.TWODIM) THEN
               WRITE(*,600) (Q,K=1,16),XPT,I
 600           FORMAT(16A,F7.1,' (',I6,')',$)
            ELSE
               WRITE(*,601) (Q,K=1,24),XPT,YPT,I
 601           FORMAT(24A,F7.1,',',F7.1,' (',I6,')',$)
            ENDIF
          ENDIF

C-----Get nearest 100 neighbors for point I
  76     CALL GETNBRS(XPT,YPT,NNBRS,NBRLOC)
         IF(NNBRS.EQ.0) GOTO 99
         NEW=NNBRS+1

C-----Calculate the matrix FN
         CALL CALCFN(FN,NNBRS,NBRLOC,TZVAR)

C-----Compute the matrix BG
         CALL CALCBG(XPT,YPT,BG,NNBRS,NBRLOC)
            DO 65 J=1,NEW
               BF(J)=BG(J)
 65         CONTINUE
         CALL SIMUL(FN,BG,NEW,ERROR)
         IF (ERROR) GOTO 99

C-----Determine the Z-value for the input coordinate
         ZE=CALCZ(NNBRS,NBRLOC,BG)
```

```
C-----Determine the Estimation Variance for this coordinate
         ZV=CALCVAR(NNBRS,BG,BF,TZVAR)
         IF(ZV.LT.0) THEN
            ZV=-99.
            ZS=-99.
            GOTO 99
         ELSE
            ZS=ZV**.5
         ENDIF
C-----Backtransform and write the results to the output file
         IF(LOG.AND.HAAN) THEN
            ZE= DEXP(ZE+0.5*ZV)-OFF
            ZV= UZMN**2 *(DEXP(TZVAR))*(1-DEXP(-ZV))
            IF(ZV.LT.0) THEN
               ZE=-99.
               ZV=-99.
               ZS=-99.
               GOTO 99
            ELSE
               ZS=ZV**.5
            ENDIF
         ENDIF

C--------Accumulate average ZE & ZV stats
         NZ=NZ+1
         ZESUM=ZESUM+ZE
         ZESUM2=ZESUM2+ZE**2
         ZSSUM=ZSSUM+ZS
         IF(JACK.OR.SELF) THEN
            IF(LOG.AND.HAAN) ZI= DEXP(ZI)-OFF
            ZISUM=ZISUM+ZI
            ZISUM2=ZISUM2+ZI**2
            ZEZI=ZEZI+ZE*ZI
         ENDIF

  99  IF((JACK.OR.SELF).AND.TWODIM) THEN
            WRITE(OUT,250) XPT,YPT,ZI,ZE,ZS,NNBRS
      ELSEIF((JACK.OR.SELF).AND.(.NOT.TWODIM)) THEN
            WRITE(OUT,252) XPT,ZI,ZE,ZS,NNBRS
      ELSEIF((.NOT.JACK).AND.(.NOT.SELF).AND.TWODIM) THEN
            WRITE(OUT,260) XPT,YPT,ZE,ZS,NNBRS
      ELSEIF((.NOT.JACK).AND.(.NOT.SELF).AND.(.NOT.TWODIM)) THEN
            WRITE(OUT,262) XPT,ZE,ZS,NNBRS
      ENDIF

 250  FORMAT(1X,G10.4,1X,G10.4,1X,G16.5,1X,G16.5,1X,G16.5,1X,I5)
 252  FORMAT(1X,G10.4,1X,G16.5,1X,G16.5,1X,G16.5,1X,I5)
 260  FORMAT(1X,G10.4,1X,G10.4,1X,G16.5,1X,G16.5,1X,I5)
 262  FORMAT(1X,G10.4,1X,G16.5,1X,G16.5,1X,I5)

C-----Return to the top and get the next point
      GOTO 30

C==================================================================C
C                                                                  C
C                     END OF MAIN KRIGING LOOP                     C
C                                                                  C
C==================================================================C

 77   WRITE(OUT,*)

C-----Calculate summary stats
C     Note use of T-1 rather than T below
C     Note that if jack first zimn= for successfully estimated points;
C        2nd zimn for all measured Z's in interpolation grid
      IF(NZ.GT.1) THEN
         T=DBLE(NZ)
         ZEVAR=(ZESUM2-(T*ZEMN**2))/(T-1)
         IF((JACK.OR.SELF)) THEN
            CORR=-99.
            ZIMN=ZISUM/T
            ZIVAR=(ZISUM2-(T*ZIMN**2))/(T-1)
            COV=(ZEZI-(ZISUM*ZESUM)/T)/(T-1)
            IF((ZEVAR.GE.0).AND.(ZIVAR.GE.0))
     +         CORR=COV/((ZEVAR**.5)*(ZIVAR**.5))
            WRITE(OUT,*)
            WRITE(OUT,*) 'Covariance between measured and ',
     +                'successfully estimated Z''s: ',SNGL(COV)
            WRITE(OUT,*) 'Correlation coefficient (r): ',SNGL(CORR)
         ENDIF
      ELSE
         WRITE(OUT,780)
 780     FORMAT(//1X,'Less than 2 estimated Z''s successfully ',
     +            'calculated.'//)
      ENDIF

      RETURN

      END

C************************************************************************
C************************************************************************

      SUBROUTINE NBHDANAL

      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER       XCOL,YCOL,ZCOL,OUT
      CHARACTER*1  HEAD1,HEAD2,HEAD3,HEAD4,YORN,Q,JTYPE
      CHARACTER*64 FILNAM,OUTFIL,LSTOUT,HRDFIL,FTITLE,ZLABEL
      LOGICAL       TWODIM,CHECK,BOUND,HERC,LOG,HAAN,ERROR,
     +              HRDGRD,YESOUT,FRESH,JACK,SELF

      DIMENSION FN(101,101),BG(101),BF(101),NBRLOC(100)

      COMMON/DATACOLS/X(1000),Y(1000),Z(1000)
      COMMON/FILINFO/FILNAM,OUTFIL,LSTOUT,FTITLE,NCOL,NROW,
     +              HEAD1(79),HEAD2(79),HEAD3(79),HEAD4(79)
      COMMON/GENPARAM/NR,NRGRID,TWODIM,HERC,CHECK,HAAN,LOG,BOUND,
     +               OFF,XCOL,YCOL,ZCOL,ZLABEL,TZMN,TZVAR,UZMN,UZVAR,
     +               BZMN,BZVAR,XMIN,XMAX,YMIN,YMAX,GXMIN,GXMAX,
     +               GYMIN,GYMAX,NRTOT
      COMMON/PUNPARAM/XINC,YINC,RNBHD,RNBMAX,MAXNBR,MODEL,
     +               C0,C9,A9,HRDGRD,HRDFIL,JACK,SELF

C-----If first time through, see if need to call setpun
      FRESH=.FALSE.
      IF(XCOL.EQ.0) THEN
         FRESH=.TRUE.
      ELSE
         WRITE(*,105)
 105     FORMAT(/1X,'Reset run parameters? (y/n): ',$)
         READ(*,'(A1)') YORN
      ENDIF

  31  IF (FRESH.OR.(YORN.EQ.'Y').OR.(YORN.EQ.'y')) THEN
         CALL SETPUN
         CALL DEFNEIGH(.TRUE.)
         CALL INTBOUNDS(.FALSE.)
         CALL STATS
      ELSEIF((YORN.NE.'N').AND.(YORN.NE.'n')) THEN
         WRITE(*,*)'          answer Y or N only......'
         WRITE(*,'(7X,A11,$)') 'try again: '
         READ(*,'(A1)') YORN
         GOTO 31
      ENDIF

      HRDGRD=.FALSE.
      SELF=.FALSE.
      JACK=.TRUE.
C-----Get output file; if different from last or new setpun, write head
      CALL GETOUT(OUT)
      IF(OUT.NE.0) YESOUT=.TRUE.
      IF((OUTFIL.NE.LSTOUT).OR.FRESH) CALL TEXT1OUT(OUT,'N')

C-----Start writing NBHDANAL info to output file
      WRITE(OUT,*)
      WRITE(OUT,*) 'Neighborhood Estimation Analysis'
      WRITE(OUT,'(1X,A,A)') 'NB: Points are jacknifed. Measured ',
     +    'values for a given coordinate point are NOT used in'
      WRITE(OUT,*) '   the interpolated estimate for that point.'
      WRITE(OUT,'(1X,A,A)') 'NB: Analysis includes only those ',
     +         'measured points within the interpolation bounds.'
      IF(LOG.AND.(.NOT.HAAN)) WRITE(OUT,*) '    Neither measured ',
     +            'nor interpolated values are backtransformed.'
```

```
      WRITE(OUT,286)
      WRITE(OUT,288)
 286  FORMAT(/1X,'Max  ',1x,'Max         ',
     +       1X,'Measured            ',4X,'Interpolated        ',
     +       1x,'        ',1X,'          ',
     +       1x,'Absolute Error      ',4x,'Reduced Error       ',
     +       3x,' r ')
 288  FORMAT(1X,'N''brs',1x,'Radius     ',
     +       1X,'Mean       Std Dev ',4x,'Mean       Std Dev ',
     +       1x,'  N  ',1X,'Covariance ',
     +       1x,'Mean       Std Dev ',4x,'Mean       Std Dev ')
      WRITE(OUT,*)
      WRITE(OUT,'(35X,A,A)') '--- Maximum Nunber of Neighbors ',
     +                       'Incremented ---'

C-----Give first part of status report
      IF(YESOUT) THEN
         Q=CHAR(8)
         WRITE(*,*)
         WRITE(*,'(1X,A29,A24,$)') '    ...calculating grid point',
     +         '                              '
         IF(TWODIM) WRITE(*,602) '        '
 602     FORMAT('+',A8,$)
      ENDIF

C===================================================================C
C                                                                   C
C                START OF MAIN NEIGBORHOOD LOOP                     C
C                                                                   C
C===================================================================C

C-----Start loop through different neighborhood parameters
      JJ=0
      DO 510 II=3,12
         MAXNBR=MIN(NR-1,100)
         RNBHD=RNBMAX

         IF(II.LE.6) THEN
            MAXNBR=2**II
            IF(MAXNBR.GT.NR-1) MAXNBR=NR-1
            IF(MAXNBR.EQ.MAXOLD) THEN
               JJ=JJ+NRGRID
               GOTO 510
            ENDIF
            MAXOLD=MAXNBR
         ELSEIF(II.EQ.7) THEN
            MAXNBR=100
            IF(MAXNBR.GT.NR-1) MAXNBR=NR-1
            IF(MAXNBR.EQ.MAXOLD) THEN
               JJ=JJ+NRGRID
               GOTO 510
            ENDIF
            MAXOLD=MAXNBR
         ELSEIF(II.EQ.8) THEN
            WRITE(OUT,*)
            WRITE(OUT,'(38X,A)')
     +            '--- Neighborhood Radius Incremented ---'
            RNBHD=RNBMAX*.05
         ELSEIF(II.EQ.9) THEN
            RNBHD=RNBMAX*.10
         ELSEIF(II.EQ.10) THEN
            RNBHD=RNBMAX*.25
         ELSEIF(II.EQ.11) THEN
            RNBHD=RNBMAX*.50
         ELSEIF(II.EQ.12) THEN
            RNBHD=RNBMAX*.75
         ENDIF

C-----Initialize Cross-validation stats
      ABSUM=0
      ABSUM2=0
      REDSUM=0
      REDS2=0
      ZESUM=0
      ZESUM2=0
      ZISUM=0
      ZISUM2=0
      ZEZI=0
      NZ=0

C-----Calculate zest and zvar for every point in INTERPOLATION grid
      DO 500 I=1,NR
         ZE=-99
         ZV=-99
         ZS=-99
         ZI=Z(I)
         XPT=X(I)
         IF((XPT.LT.GXMIN).OR.(XPT.GT.GXMAX)) GOTO 500
         IF(TWODIM) THEN
            YPT=Y(I)
            IF((YPT.LT.GYMIN).OR.(YPT.GT.GYMAX)) GOTO 500
         ENDIF

C-----Status report to screen
         JJ=JJ+1
         IF(YESOUT) THEN
            IF (.NOT.TWODIM) THEN
               WRITE(*,600) (Q,K=1,24),XPT,JJ,NRGRID*16
 600           FORMAT(24A,F7.1,' (',I5,' of ',I5,')',$)
            ELSE
               WRITE(*,601) (Q,K=1,32),XPT,YPT,JJ,NRGRID*16
 601           FORMAT(32A,F7.1,',',F7.1,' (',I5,' of ',I5,')',$)
            ENDIF
          ENDIF

C-----Get nearest MAXNBR neighbors for point I
  76     CALL GETNBRS(XPT,YPT,NNBRS,NBRLOC)
         IF(NNBRS.EQ.0) GOTO 500
         NEW=NNBRS+1

C-----Calculate the matrix FN
         CALL CALCFN(FN,NNBRS,NBRLOC,TZVAR)

C-----Compute the matrix BG
         CALL CALCBG(XPT,YPT,BG,NNBRS,NBRLOC)
            DO 65 J=1,NEW
               BF(J)=BG(J)
 65         CONTINUE
         CALL SIMUL(FN,BG,NEW,ERROR)
         IF (ERROR) GOTO 500

C-----Determine the Z-value for the input coordinate
         ZE=CALCZ(NNBRS,NBRLOC,BG)

C-----Determine the Estimation Variance for this coordinate
         ZV=CALCVAR(NNBRS,BG,BF,TZVAR)
         IF(ZV.LT.0) THEN
             ZV=-99.
             ZE=-99.
             ZS=-99.
             GOTO 500
          ELSE
             ZS=ZV**.5
          ENDIF

C-----Backtransform and write the results to the output file
         IF(LOG.AND.HAAN) THEN
            ZI= DEXP(ZI)-OFF
            ZE= DEXP(ZE+0.5*ZV)-OFF
            ZV= UZMN**2 *(DEXP(TZVAR))*(1-DEXP(-ZV))
            IF(ZV.LT.0) THEN
               ZV=-99.
               ZE=-99.
               ZS=-99.
               GOTO 500
            ELSE
               ZS=ZV**.5
            ENDIF
         ENDIF

C-----Cross-validation stats
         NZ=NZ+1
         ZESUM=ZESUM+ZE
         ZESUM2=ZESUM2+ZE**2
         ZISUM=ZISUM+ZI
         ZISUM2=ZISUM2+ZI**2
         ZEZI=ZEZI+ZE*ZI
         ABSUM=ABSUM+(ZE-ZI)
         ABSUM2=ABSUM2+(ZE-ZI)**2
         IF(ZV.GT.0) THEN
            T=(ZE-ZI)/ZV**.5
            REDSUM=REDSUM+T
            REDS2=REDS2+T**2
         ENDIF

C-----Return to the top and get the next point
 500  CONTINUE

C-----Calculate neighborhood estimation stats for this nbr defn
C     Note use of T-1 rather than T for variance terms
      IF(NZ.GT.1) THEN
         ZESTD=-99.
         ZISTD=-99.
         REDSTD=-99.
         CORR=-99.
         T=DBLE(NZ)
         ZEMN=ZESUM/T
         ZEVAR=(ZESUM2-(T*ZEMN**2))/(T-1)
         IF(ZEVAR.GE.0) ZESTD=ZEVAR**.5
         ZIMN=ZISUM/T
         ZIVAR=(ZISUM2-(T*ZIMN**2))/(T-1)
         IF(ZIVAR.GE.0) ZISTD=ZIVAR**.5
         COV=(ZEZI-(ZISUM*ZESUM)/T)/(T-1)
         IF((ZESTD.NE.-99.).AND.(ZISTD.NE.-99.))
     +      CORR=COV/(ZESTD*ZISTD)
         ABSMN=ABSUM/T
         ABSVAR=(ABSUM2-(T*ABSMN**2))/(T-1)
         IF(ABSVAR.GE.0) ABSSTD=ABSVAR**.5
         REDMN=REDSUM/T
         REDVAR=(REDS2-(T*REDMN**2))/(T-1)
         IF(REDVAR.GE.0) REDSTD=REDVAR**.5
C--------Print neighborhood estimation stats
         WRITE(OUT,290) MAXNBR,RNBHD,ZIMN,ZISTD,ZEMN,ZESTD,
     +                  NZ,COV,ABSMN,ABSSTD,REDMN,REDSTD,CORR
 290     FORMAT(1X,I4,1X,5(G11.3),1X,I4,1X,5(G11.3),1X,F7.3)
      ENDIF

C-----Get next neighborhood defininition
 510  CONTINUE


C==============================================================C
C                                                              C
C                END OF MAIN NEIGHBORHOOD LOOP                 C
C                                                              C
C==============================================================C

 77   RETURN
      END

C***********************************************************************
C***********************************************************************

      SUBROUTINE SETPUN

      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4       DATA
      INTEGER      XCOL,YCOL,ZCOL
      CHARACTER*1  YORN,ANSW,HEAD1,HEAD2,HEAD3,HEAD4
      CHARACTER*64 FILNAM,OUTFIL,LSTOUT,HRDFIL,FTITLE,ZLABEL
      LOGICAL      TWODIM,CHECK,BOUND,HERC,LOG,HAAN,THERE,
     +             HRDGRD,JACK,SELF

      COMMON/DATAIN/DATA(1000,15)
      COMMON/DATACOLS/X(1000),Y(1000),Z(1000)
      COMMON/FILINFO/FILNAM,OUTFIL,LSTOUT,FTITLE,NCOL,NROW,
     +               HEAD1(79),HEAD2(79),HEAD3(79),HEAD4(79)
      COMMON/GENPARAM/NR,NRGRID,TWODIM,HERC,CHECK,HAAN,LOG,BOUND,
     +                OFF,XCOL,YCOL,ZCOL,ZLABEL,TZMN,TZVAR,UZMN,UZVAR,
     +                BZMN,BZVAR,XMIN,XMAX,YMIN,YMAX,GXMIN,GXMAX,
     +                 GYMIN,GYMAX,NRTOT
      COMMON/PUNPARAM/XINC,YINC,RNBHD,RNBMAX,MAXNBR,MODEL,
     +                 C0,C9,A9,HRDGRD,HRDFIL,JACK,SELF

C-----Get the number of dimensions
      WRITE(*,*)
      WRITE(*,'(A38,$)')    ' Enter number of dimensions (1 or 2): '
    5 READ(*,*,IOSTAT=IERR) N
      IF((IERR.NE.0).OR.((N.NE.1).AND.(N.NE.2))) THEN
         WRITE(*,'(4X,A,$)') 'enter 1 or 2 only: '
         GOTO 5
      ELSEIF(N.EQ.1) THEN
         TWODIM=.FALSE.
      ELSEIF (N.EQ.2) THEN
         TWODIM=.TRUE.
      ENDIF

C-----Ask user for column numbers for X, Y, and Z and get z laBel
    1 IF(.NOT.TWODIM) THEN
         WRITE(*,'(A38,$)') '               column numbers for X,Z: '
    2    READ(*,*,IOSTAT=IERR) XCOL,ZCOL
         IF(IERR.NE.0) THEN
            WRITE(*,650)
            GOTO 2
         ENDIF
         YCOL=XCOL
         IF ((XCOL.GT.NCOL).OR.(ZCOL.GT.NCOL)) THEN
            WRITE(*,*) '             value out of range...'
            WRITE(*,'(1X,A,$)') '             re-enter: '
            GOTO 2
         ENDIF
      ELSE
         WRITE(*,'(A38,$)') '             column numbers for X,Y,Z: '
    3    READ(*,*,IOSTAT=IERR) XCOL,YCOL,ZCOL
         IF(IERR.NE.0) THEN
            WRITE(*,650)
            GOTO 3
         ENDIF
         IF ((XCOL.GT.NCOL).OR.(YCOL.GT.NCOL).OR.(ZCOL.GT.NCOL)) THEN
            WRITE(*,*) '             value out of range...'
            WRITE(*,'(1X,A,$)') '             re-enter: '
            GOTO 3
         ENDIF
      ENDIF
      WRITE(*,'(A,$)') '                          label for Z: '
      READ(*,'(A20)') ZLABEL
      WRITE(*,*)

C-----Put data from proper columns into work arrays
      K=0
      DO 15 I=1,NROW
         IF (DATA(I,XCOL).EQ.-99.0) GOTO 15
         IF (TWODIM.AND.(DATA(I,YCOL).EQ.-99.0)) GOTO 15
         IF (DATA(I,ZCOL).EQ.-99.0) GOTO 15
         K=K+1
         X(K)=DATA(I,XCOL)
         Y(K)=0
         IF(TWODIM) Y(K)=DATA(I,YCOL)
         Z(K)=DATA(I,ZCOL)
   15 CONTINUE
      IF(K.LE.2) THEN
         WRITE(*,700)
  700    FORMAT(14X,'too few valid data in these columns.....',/)
         GOTO 1
      ENDIF
      NRTOT=K

C-----See if user wants only subset of dataset
      CALL BOUNDS

C-----See if user wants a logarithmic transformation of Z
      CALL TRANSFORM

C-----Get the Semivariogram model
      CALL DEFMODEL
      RETURN

C-----Error trapping code
 650  FORMAT(4X,'numeric input only: ',$)
      END

C******************************************************************************
C******************************************************************************

      SUBROUTINE DEFPTS

      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER      XCOL,YCOL,ZCOL
      CHARACTER*1  YORN,ANSW,HEAD1,HEAD2,HEAD3,HEAD4
      CHARACTER*64 FILNAM,OUTFIL,LSTOUT,HRDFIL,FTITLE,ZLABEL
      LOGICAL      TWODIM,CHECK,BOUND,HERC,LOG,HAAN,THERE,
     +             HRDGRD,JACK,SELF

      COMMON/GENPARAM/NR,NRGRID,TWODIM,HERC,CHECK,HAAN,LOG,BOUND,
     +                OFF,XCOL,YCOL,ZCOL,ZLABEL,TZMN,TZVAR,UZMN,UZVAR,
     +                BZMN,BZVAR,XMIN,XMAX,YMIN,YMAX,GXMIN,GXMAX,
     +                GYMIN,GYMAX,NRTOT
      COMMON/PUNPARAM/XINC,YINC,RNBHD,RNBMAX,MAXNBR,MODEL,
     +                C0,C9,A9,HRDGRD,HRDFIL,JACK,SELF

C-----Define how to acquire interpolation points
      JACK=.FALSE.
      HRDGRD=.FALSE.
      SELF=.FALSE.
  28  WRITE(*,510)
 510  FORMAT(' [C]alculate kriged points, [J]acknife,'/
     +       '   [S]elf-fit or [R]ead from file: ',$)
  23  READ(*,'(A1)') ANSW

      IF ((ANSW.EQ.'C').OR.(ANSW.EQ.'c')) THEN
         WRITE(*,110)
 110     FORMAT(1X,'   specify X-axis increment: ',$)
  22     READ(*,*,IOSTAT=IERR) XINC
         IF(IERR.NE.0) THEN
            WRITE(*,650)
            GOTO 22
         ENDIF
         YINC=0
         IF (TWODIM) THEN
            WRITE(*,120)
 120        FORMAT(1X,'   specify Y-axis increment: ',$)
  24        READ(*,*,IOSTAT=IERR) YINC
            IF(IERR.NE.0) THEN
               WRITE(*,650)
               GOTO 24
            ENDIF
          ENDIF

      ELSEIF ((ANSW.EQ.'J').OR.(ANSW.EQ.'j')) THEN
         JACK=.TRUE.

      ELSEIF ((ANSW.EQ.'S').OR.(ANSW.EQ.'s')) THEN
         SELF=.TRUE.

      ELSEIF ((ANSW.EQ.'R').OR.(ANSW.EQ.'r')) THEN
         HRDGRD=.TRUE.
         WRITE(*,500)
 500     FORMAT('    Specify name of file with grid points: ',$)
         READ(*,'(A20)') HRDFIL
         INQUIRE(FILE=HRDFIL,EXIST=THERE)
         IF (.NOT.THERE) THEN
            WRITE(*,*) '    file not in default directory;'
            GOTO 28
         ENDIF
      ELSE
         WRITE(*,650)
         GOTO 23
      ENDIF
      WRITE(*,*)

      RETURN
```

```
C-----Error trapping code
 650  FORMAT(4X,'entry error; try again: ',$)
      END

C*************************************************************************
C*************************************************************************

      SUBROUTINE TRANSFORM

      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER      XCOL,YCOL,ZCOL
      CHARACTER*1  YORN
      CHARACTER*64 ZLABEL
      LOGICAL      TWODIM,CHECK,BOUND,HERC,LOG,HAAN

      COMMON/DATACOLS/X(1000),Y(1000),Z(1000)
      COMMON/GENPARAM/NR,NRGRID,TWODIM,HERC,CHECK,HAAN,LOG,BOUND,
     +                OFF,XCOL,YCOL,ZCOL,ZLABEL,TZMN,TZVAR,UZMN,UZVAR,
     +                BZMN,BZVAR,XMIN,XMAX,YMIN,YMAX,GXMIN,GXMAX,
     +                GYMIN,GYMAX,NRTOT

  20  WRITE(*,105)
 105  FORMAT(' Transform Z to logn(z)? (y/n): ',$)
  31  READ(*,'(A1)') YORN
      IF ((YORN.EQ.'N').OR.(YORN.EQ.'n')) THEN
         LOG=.FALSE.
         HAAN=.FALSE.
      ELSEIF ((YORN.EQ.'Y').OR.(YORN.EQ.'y')) THEN
         LOG=.TRUE.
         WRITE(*,106)
 106     FORMAT('    Specify a in the expression Z=ln(z+a): ',$)
  22     READ(*,*,IOSTAT=IERR) OFF
         IF(IERR.NE.0) THEN
            WRITE(*,650)
            GOTO 22
         ENDIF

         IF((OFF.LT.0).OR.(OFF.GT.9999)) THEN
            WRITE(*,650)
            GOTO 22
         ENDIF

C-----Count points out of range in SPECIFIED data grid (not entire)
         NOUTS=0
         DO 15 I=1,NR
            IF(Z(I)+OFF.LT.1.0) NOUTS=NOUTS+1
  15     CONTINUE
         IF(NOUTS.GT.0) THEN
            IF(NR-NOUTS.LT.3) THEN
               WRITE(*,150) NR-NOUTS
 150           FORMAT(4X,'too few data (',I2,
     +                 ') are >1 before log-n transformation......')
               WRITE(*,'(4X,A,A,$)') ' continue with ',
     +                          'transformation? (y/n): '
               GOTO 31
            ELSE
               WRITE(*,100) NOUTS
 100           FORMAT(4X,'N.BG. ',I4,
     +                 ' Z-values are out of range (<1.0) and')
               WRITE(*,101) NR-NOUTS
 101           FORMAT(4X,'  will be ignored; there remain',I4,
     +                 ' values;')
  16           WRITE(*,102)
 102           FORMAT(4X,'  is this ok? (y/n): ',$)
               READ(*,'(A1)') YORN
               IF((YORN.NE.'Y').AND.(YORN.NE.'y')) GOTO 20
               WRITE(*,*)
            ENDIF
         ENDIF

         WRITE(*,107)
 107     FORMAT('    Specify if want [H]aan or [N]o '
     +      'backtranformation: ',$)
  21     READ(*,'(A1)') YORN
         IF ((YORN.EQ.'H').OR.(YORN.EQ.'h')) THEN
            HAAN=.TRUE.
         ELSEIF ((YORN.EQ.'N').OR.(YORN.EQ.'n')) THEN
```

```
            HAAN=.FALSE.
         ELSE
            WRITE(*,'(7X,A24,$)') 'answer H or N only......'
            GOTO 21
         ENDIF

C-----Transform data in work arrays
         K=0
         DO 18 I=1,NR
            IF(Z(I)+OFF.LT.1.0) GOTO 18
            K=K+1
            X(K)=X(I)
            Y(K)=Y(I)
            Z(K)=DLOG(Z(I)+OFF)
  18     CONTINUE
         NR=K
      ELSE
         WRITE(*,'(4X,A,$)') '   answer y or n only: '
         GOTO 31
      ENDIF
      WRITE(*,*)
      RETURN

C-----Error trapping code
 650  FORMAT(7X,'value out of range......',/14X,'try again: ',$)

      END

C***************************************************************************
C***************************************************************************

      SUBROUTINE BOUNDS

C-----Gets bounds for data set; see INTBOUNDS for interpolation bounds

      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER       XCOL,YCOL,ZCOL
      CHARACTER*1   YORN
      CHARACTER*64  ZLABEL
      LOGICAL       TWODIM,CHECK,BOUND,HERC,LOG,HAAN

      COMMON/DATACOLS/X(1000),Y(1000),Z(1000)
      COMMON/GENPARAM/NR,NRGRID,TWODIM,HERC,CHECK,HAAN,LOG,BOUND,
     +                 OFF,XCOL,YCOL,ZCOL,ZLABEL,TZMN,TZVAR,UZMN,UZVAR,
     +                 BZMN,BZVAR,XMIN,XMAX,YMIN,YMAX,GXMIN,GXMAX,
     +                 GYMIN,GYMAX,NRTOT

C-----Determine Xmin,XMAX,Ymin,Ymax (if onedim, all y(i)'s = 0)
      XMIN=X(1)
      XMAX=X(1)
      YMIN=Y(1)
      YMAX=Y(1)
      IF(.NOT.TWODIM) THEN
         DO 10 I=2,NRTOT
               XMIN=DMIN1(XMIN,X(I))
               XMAX=DMAX1(XMAX,X(I))
  10     CONTINUE
      ELSE
         DO 20 I=2,NRTOT
               XMIN=DMIN1(XMIN,X(I))
               XMAX=DMAX1(XMAX,X(I))
               YMIN=DMIN1(YMIN,Y(I))
               YMAX=DMAX1(YMAX,Y(I))
  20     CONTINUE
      ENDIF
 49   TXMIN=XMIN
      TXMAX=XMAX
      TYMIN=YMIN
      TYMAX=YMAX
      NRTMP=NRTOT
      BOUND=.FALSE.

C-----Write bounds to screen
 50   IF(TWODIM) THEN
         WRITE(*,*)'X-Y bounds for dataset:'
         WRITE(*,'(4X,A3,F11.3,A3,F11.3)')'X: ',TXMIN,' , ',TXMAX
         WRITE(*,'(4X,A3,F11.3,A3,F11.3)')'Y: ',TYMIN,' , ',TYMAX
      ELSE
         WRITE(*,*)'Bounds for dataset:'
         WRITE(*,'(4X,A3,F11.3,A3,F11.3)')'X: ',TXMIN,' , ',TXMAX
      ENDIF
      WRITE(*,250) NRTMP
 250  FORMAT (4X,'Number of valid points: ',I4)

C-----See if user wants subset
      WRITE(*,'(4X,A,$)') 'Respecify bounds? (y/n): '
 45   READ(*,'(A1)') YORN
 46   IF ((YORN.EQ.'Y').OR.(YORN.EQ.'y')) THEN
         WRITE(*,220)
 220     FORMAT(7X,'Enter min X and max X (2 values): ',$)
  16     READ(*,*,IOSTAT=IERR)  TXMIN, TXMAX
         IF((IERR.NE.0).OR.(TXMIN.GE.TXMAX)) THEN
            WRITE(*,650)
            GOTO 16
         ENDIF
         IF(TXMIN.LT.XMIN) TXMIN=XMIN
         IF(TXMAX.GT.XMAX) TXMAX=XMAX

         IF (TWODIM) THEN
            WRITE(*,230)
 230        FORMAT(7X,'Enter min Y and max Y (2 values): ',$)
  17        READ(*,*,IOSTAT=IERR) TYMIN,TYMAX
            IF((IERR.NE.0).OR.(TYMIN.GE.TYMAX)) THEN
               WRITE(*,650)
               GOTO 17
            ENDIF
            IF(TYMIN.LT.YMIN) TYMIN=YMIN
            IF(TYMAX.GT.YMAX) TYMAX=YMAX
          ENDIF

C-----See if >3 points fall within specified range
C     NB if onedim all Y(L)'s = 0, as does TYMIN, TYMAX
         NRTMP=0
         DO 15 L=1,NRTOT
            IF ((X(L).GE.TXMIN).AND.(X(L).LE.TXMAX).AND.
     +          (Y(L).GE.TYMIN).AND.(Y(L).LE.TYMAX)) NRTMP=NRTMP+1
 15      CONTINUE
         IF(NRTMP.LE.2) THEN
            WRITE(*,150) NRTMP,CHAR(7),CHAR(7)
 150        FORMAT(14X,'...too few data (',I2,') within range.....',
     +                 A,A)
            WRITE(*,*)
            GOTO 49
         ENDIF
         GOTO 50

      ELSEIF ((YORN.EQ.'N').OR.(YORN.EQ.'n')) THEN
C-----Fill X,Y,Z vectors if boundaries changed
         IF((XMIN.NE.TXMIN).OR.(XMAX.NE.TXMAX).OR.(YMIN.NE.TYMIN)
     +      .OR.(YMAX.NE.TYMAX)) THEN
            XMIN=TXMIN
            XMAX=TXMAX
            YMIN=TYMIN
            YMAX=TYMAX
            K=0
            DO 35 L=1,NRTOT
               IF ((X(L).GE.XMIN).AND.(X(L).LE.XMAX).AND.
     +            (Y(L).GE.YMIN).AND.(Y(L).LE.YMAX)) THEN
                  K=K+1
                  X(K)=X(L)
                  Y(K)=Y(L)
                  Z(K)=Z(L)
               ENDIF
 35         CONTINUE
            BOUND=.TRUE.
            NRTMP=K
         ENDIF
         NR=NRTMP

      ELSE
         WRITE(*,*) '   answer either y or n......'
         WRITE(*,'(4X,A,$)') 'try again: '
         GOTO 45
      ENDIF
      WRITE(*,*)
 999  RETURN

C-----Error trapping code
 650  FORMAT(7X,'entry error......',/14X,'try again: ',$)

      END

C***************************************************************************
C***************************************************************************

      SUBROUTINE INTBOUNDS(EARLY)

      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER       XCOL,YCOL,ZCOL
      CHARACTER*1   YORN
      CHARACTER*64  ZLABEL
      LOGICAL       TWODIM,CHECK,BOUND,HERC,LOG,HAAN,EARLY

      COMMON/DATACOLS/X(1000),Y(1000),Z(1000)
      COMMON/GENPARAM/NR,NRGRID,TWODIM,HERC,CHECK,HAAN,LOG,BOUND,
     +                OFF,XCOL,YCOL,ZCOL,ZLABEL,TZMN,TZVAR,UZMN,UZVAR,
     +                BZMN,BZVAR,XMIN,XMAX,YMIN,YMAX,GXMIN,GXMAX,
     +                GYMIN,GYMAX,NRTOT

C-----Set grid boundaries to dataset bounds
      GXMIN=XMIN
      GXMAX=XMAX
      GYMIN=YMIN
      GYMAX=YMAX

      IF(EARLY) THEN
         NRGRID=0
         RETURN
      ENDIF

C-----Write bounds to screen
      WRITE(*,*)'Interpolation Bounds:'
      IF(TWODIM) THEN
         WRITE(*,'(4X,A3,F11.3,A3,F11.3)')'X: ',GXMIN,' , ',GXMAX
         WRITE(*,'(4X,A3,F11.3,A3,F11.3)')'Y: ',GYMIN,' , ',GYMAX
      ELSE
         WRITE(*,'(4X,A3,F11.3,A3,F11.3)')'X: ',GXMIN,' , ',GXMAX
      ENDIF

C-----See how many measured points are within interpolation grid
 55   NRGRID=0
      DO 15 I=1,NR
         IF(TWODIM) THEN
            IF((Y(I).GE.GYMIN).AND.(Y(I).LE.GYMAX).AND.
     +        (X(I).GE.GXMIN).AND.(X(I).LE.GXMAX)) NRGRID=NRGRID+1
         ELSE
            IF((X(I).GE.GXMIN).AND.(X(I).LE.GXMAX)) NRGRID=NRGRID+1
         ENDIF
 15   CONTINUE
      WRITE(*,250) NRGRID
 250  FORMAT (4X,'Number of measured points within bounds: ',I4)

      IF(NRGRID.EQ.0) THEN
         YORN='Y'
         WRITE(*,*) 'Respecify bounds:'
      ELSE
         WRITE(*,'(4X,A,$)') 'Respecify bounds? (y/n): '
         READ(*,'(A1)') YORN
      ENDIF
 45   IF ((YORN.EQ.'Y').OR.(YORN.EQ.'y')) THEN
         WRITE(*,220)
 220     FORMAT(7X,'Enter min X and max X (2 values): ',$)
 16      READ(*,*,IOSTAT=IERR)  GXMIN, GXMAX
         IF(IERR.NE.0) THEN
            WRITE(*,605)
            GOTO 16
         ENDIF
         IF (TWODIM) THEN
            WRITE(*,230)
 230        FORMAT(7X,'Enter min Y and max Y (2 values): ',$)
```

```
  17       READ(*,*,IOSTAT=IERR) GYMIN,GYMAX
           IF((IERR.NE.0).OR.(GYMIN.GE.GYMAX)) THEN
              WRITE(*,605)
              GOTO 17
           ENDIF
        ENDIF
        GOTO 55

      ELSEIF ((YORN.NE.'N').AND.(YORN.NE.'n')) THEN
         WRITE(*,*) '      answer either y or n......'
         WRITE(*,'(7X,A,$)') 'try again: '
         READ(*,'(A1)') YORN
         GOTO 45
      ENDIF

      WRITE(*,*)
 999  RETURN

C-----Error trapping code
 605  FORMAT(7X,'value(s) out of range......',/14X,'try again: ',$)

      END

C***********************************************************************
C***********************************************************************

      SUBROUTINE STATS

      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER      XCOL,YCOL,ZCOL
      CHARACTER*1  YORN,HEAD1,HEAD2,HEAD3,HEAD4
      CHARACTER*64 HRDFIL,ZLABEL
      LOGICAL      TWODIM,CHECK,BOUND,HERC,LOG,HAAN,THERE,
     +             HRDGRD,JACK,SELF

      COMMON/DATACOLS/X(1000),Y(1000),Z(1000)
      COMMON/GENPARAM/NR,NRGRID,TWODIM,HERC,CHECK,HAAN,LOG,BOUND,
     +                OFF,XCOL,YCOL,ZCOL,ZLABEL,TZMN,TZVAR,UZMN,UZVAR,
     +                BZMN,BZVAR,XMIN,XMAX,YMIN,YMAX,GXMIN,GXMAX,
     +                GYMIN,GYMAX,NRTOT
      COMMON/PUNPARAM/XINC,YINC,RNBHD,RNBMAX,MAXNBR,MODEL,
     +                C0,C9,A9,HRDGRD,HRDFIL,JACK,SELF

C-----NB: STATS is called AFTER transformations
C     TZMN, TZVAR = transformed general mean and var of z
C     UZMN, UZVAR = untransformed

C-----Calculate stats
      SUMZ=0
      SUMZ2=0
      DO 10 I=1,NR
         SUMZ=SUMZ+Z(I)
         SUMZ2=SUMZ2+Z(I)**2
 10   CONTINUE

C-----Calculate mean, variance
      T=DBLE(NR)
      TZMN=SUMZ/T
      IF(T.GT.1) TZVAR=(SUMZ2-(T*TZMN**2))/(T-1)

C-----If transformed, get untransformed mean, var
      IF(LOG) THEN
         SUMZ=0
         SUMZ2=0
         DO 20 I=1,NR
            SUMZ=SUMZ+DEXP(Z(I))-OFF
            SUMZ2=SUMZ2+ (DEXP(Z(I))-OFF)**2
  20     CONTINUE
         UZMN=SUMZ/T
         IF(T.GT.1) UZVAR=(SUMZ2-(T*UZMN**2))/(T-1)
         IF(HAAN) THEN
            BZMN=DEXP(TZMN+0.5*TZVAR)-OFF
            BZVAR=(UZMN**2)*(DEXP(TZVAR)-1)
         ELSE
            BZMN=DEXP(TZMN)-OFF
            BZVAR=DEXP(TZVAR)
         ENDIF
      ELSE
         UZMN=TZMN
         UZVAR=TZVAR
      ENDIF


      RETURN
      END

C***********************************************************************
C***********************************************************************

      SUBROUTINE GETOUT(OUT)
C     Gets name of output file, opens

      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER      XCOL,YCOL,ZCOL,OUT
      CHARACTER*1  YORN,HEAD1,HEAD2,HEAD3,HEAD4
      CHARACTER*64 FILNAM,OUTFIL,FTITLE,ZLABEL,LSTOUT
      LOGICAL      TWODIM,THERE,OPEND,CHECK,HERC,LOG,HAAN,BOUND

      COMMON/FILINFO/FILNAM,OUTFIL,LSTOUT,FTITLE,NCOL,NROW,
     +               HEAD1(79),HEAD2(79),HEAD3(79),HEAD4(79)
      COMMON/GENPARAM/NR,NRGRID,TWODIM,HERC,CHECK,HAAN,LOG,BOUND,
     +                OFF,XCOL,YCOL,ZCOL,ZLABEL,TZMN,TZVAR,UZMN,UZVAR,
     +                BZMN,BZVAR,XMIN,XMAX,YMIN,YMAX,GXMIN,GXMAX,
     +                GYMIN,GYMAX,NRTOT

C-----Get output filename
 10   OUT=6
      INQUIRE (OUT,OPENED=OPEND,NAME=LSTOUT)
      WRITE(*,100)
 100  FORMAT(' Output file ([N]one, [A]ppend, or filename): ',$)
      READ(*,'(A64)') OUTFIL
      IF ((OUTFIL.EQ.'N').OR.(OUTFIL.EQ.'n')) THEN
         OUTFIL='NONE'
         CLOSE(OUT)
         OUT=0
      ELSEIF ((OUTFIL.EQ.'A').OR.(OUTFIL.EQ.'a')) THEN
         IF (.NOT.OPEND) THEN
           WRITE(*,*) '   there''s no output file to append to...'
           GOTO 10
         ELSE
           OUTFIL=LSTOUT
           WRITE(OUT,'(////////////)')
         ENDIF
      ELSE
         INQUIRE(FILE=OUTFIL,EXIST=THERE)
         INQUIRE(OUT,OPENED=OPEND)
         IF (THERE) THEN
            WRITE(*,110)
 110        FORMAT(' File exists;'
     +         /3X,'Do you want to write over it? (y/n): ',$)
 12         READ(*,'(A1)') YORN
            IF ((YORN.EQ.'N').OR.(YORN.EQ.'n')) THEN
               GOTO 10
            ELSEIF ((YORN.NE.'Y').AND.(YORN.NE.'y')) THEN
               WRITE(*,115)
 115           FORMAT('    answer either y or n: ',$)
               GOTO 12
            ENDIF
         ENDIF
         IF (OPEND) CLOSE(OUT)
         OPEN(OUT,FILE=OUTFIL,STATUS='NEW',IOSTAT=IERR)
         IF(IERR.NE.0) GOTO 600
      ENDIF

      WRITE(*,*)
 70   RETURN

C-----Error trapping
 600  WRITE(*,*)'                  ...file error......',CHAR(7),CHAR(7)
      WRITE(*,*)
      GOTO 10

      END
```

```
C***************************************************************************
C***************************************************************************

      SUBROUTINE DEFMODEL

      IMPLICIT REAL*8 (A-H,O-Z)
      CHARACTER*64 HRDFIL
      LOGICAL      HRDGRD,JACK,SELF

      COMMON/PUNPARAM/XINC,YINC,RNBHD,RNBMAX,MAXNBR,MODEL,
     +              C0,C9,A9,HRDGRD,HRDFIL,JACK,SELF

      WRITE(*,400)
  400 FORMAT(1X,'Specify semivariogram model;',/1x,
     +   '   Models available:',/1x,
     +   '     1 - Linear to range',/1X,
     +   '         for H(i) < A, ',
     +             'Gamma(i) = C0+[(C-C0)/A]*H(i)',/1X,
     +   '                  > A, ',
     +             'Gamma(i) = C',/1X,
     +   '     2 - Spherical',/1X,
     +   '         for H < A, ',
     +             'Gamma(i) = C0+(C-C0)*[1.5*(H(i)/A)',
     +             '-(0.5*(H(i)/A)**3)]',/1X,
     +   '               > A, ',
     +             'Gamma(i) = C',/1X,
     +   '     3 - Exponential',/1X,
     +   '         Gamma(i) = C0+(C-C0)*[1-EXP(-H(i)/Ao)]',/1X,
     +   '     4 - Gaussian',/1X,
     +   '         Gamma(i) = C0-(C-C0)*',
     +             '[1-EXP(-H(i)**2/(Ao/3**.5)**2)]')
      WRITE(*,'(1X,A17,$)') '   Enter Choice: '
   28 READ(*,*,IOSTAT=IERR) MODEL
      IF(IERR.NE.0) THEN
         WRITE(*,605)
         GOTO 28
      ENDIF

      C9=0
      C0=0
      A9=0
      IF ((MODEL.GE.1).OR.(MODEL.LE.4)) THEN
         WRITE(*,'(4X,A,$)')     'Enter C0 (nugget variance): '
   34    READ(*,*,IOSTAT=IERR) C0
         IF(IERR.NE.0) THEN
            WRITE(*,605)
            GOTO 34
         ENDIF
         WRITE(*,'(4X,A,$)')     '                  C (sill): '
   36    READ(*,*,IOSTAT=IERR) C9
         IF(IERR.NE.0) THEN
            WRITE(*,605)
            GOTO 36
         ENDIF
         IF (MODEL.LT.3) THEN
            WRITE(*,'(4X,A,$)')     '                 A (range): '
         ELSE
            WRITE(*,'(4X,A,$)')     '                       Ao: '
         ENDIF
   38    READ(*,*,IOSTAT=IERR) A9
         IF(IERR.NE.0) THEN
            WRITE(*,605)
            GOTO 38
         ENDIF

      ELSE
         WRITE(*,605)
         GOTO 28
      ENDIF

      WRITE(*,*)
      RETURN

C-----Error trapping code
  605 FORMAT(7X,'value out of range......',/14X,'try again: ',$)

      END
```

```
C***********************************************************************
C***********************************************************************

      SUBROUTINE DEFNEIGH(EARLY)

      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER XCOL,YCOL,ZCOL
      CHARACTER*64 HRDFIL,ZLABEL
      LOGICAL      TWODIM,CHECK,BOUND,HERC,LOG,HAAN,HRDGRD,
     +             EARLY,JACK,SELF

      COMMON/GENPARAM/NR,NRGRID,TWODIM,HERC,CHECK,HAAN,LOG,BOUND,
     +                OFF,XCOL,YCOL,ZCOL,ZLABEL,TZMN,TZVAR,UZMN,UZVAR,
     +                BZMN,BZVAR,XMIN,XMAX,YMIN,YMAX,GXMIN,GXMAX,
     +                GYMIN,GYMAX,NRTOT
      COMMON/PUNPARAM/XINC,YINC,RNBHD,RNBMAX,MAXNBR,MODEL,
     +                C0,C9,A9,HRDGRD,HRDFIL,JACK,SELF

C-----Define maximum neighborhood search radius
      DISTX=XMAX-XMIN
      DISTY=YMAX-YMIN
      IF(.NOT.TWODIM) THEN
         RNBMAX=DSQRT(DISTX**2)
      ELSE
         RNBMAX=DSQRT(DISTX**2+DISTY**2)
      ENDIF
      WRITE(*,*) 'Neighborhood Definition'
      WRITE(*,210) RNBMAX
 210  FORMAT(1X,'Maximum neighborhood radius is ',G12.3,';')
      RNBHD=RNBMAX
      MAXNBR=MIN(NR-1,100)
      IF(EARLY) GOTO 999

      WRITE(*,560)
  42  READ(*,*,ERR=630) TRNBHD
 560  FORMAT(1X,'   specify maximum search radius: ',$)
      IF(TRNBHD.LE.0) GOTO 630
      IF(TRNBHD.LT.RNBHD) RNBHD=TRNBHD
      WRITE(*,*)

C-----Define max no. of neighbors to include in neighborhood search
      MAXNBR=MIN(NR,100)
      WRITE(*,552) MAXNBR
 552  FORMAT(1X,'Number of potential neighbors for any search: ',
     +       I3)
      WRITE(*,550) MIN(100,MAXNBR)
 550  FORMAT(1X,'   specify maximum (to ',I3,'): ',$)
  32  READ(*,*,ERR=620) MAXNBR
      IF(MAXNBR.LT.1) GOTO 620
      IF(MAXNBR.GT.100) THEN
         WRITE(*,*)'        ......n set to 100;'
         MAXNBR=100
      ENDIF
      IF(MAXNBR.GT.NR) MAXNBR=NR

 999  WRITE(*,*)
      RETURN

C-----Error trapping code
 620  WRITE(*,650)
      GOTO 32
 630  WRITE(*,650)
      GOTO 42
 650  FORMAT(4X,'value out of range......',/4X,'try again: ',$)

      END

C***********************************************************************
C***********************************************************************

      SUBROUTINE TEXT1OUT(OUT,KTYPE)

C-----KTYPE specifies type of kriging: P=punctual,
C                                     N=neighborhood analysis,
C                                     B=block
```

```
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER       XCOL,YCOL,ZCOL,OUT
      CHARACTER*1   HEAD1,HEAD2,HEAD3,HEAD4,Q,KTYPE
      CHARACTER*64  FILNAM,OUTFIL,LSTOUT,HRDFIL,FTITLE,ZLABEL
      LOGICAL       TWODIM,CHECK,BOUND,HERC,LOG,HAAN,
     +              HRDGRD,JACK,SELF,NBANAL

      COMMON/FILINFO/FILNAM,OUTFIL,LSTOUT,FTITLE,NCOL,NROW,
     +                HEAD1(79),HEAD2(79),HEAD3(79),HEAD4(79)
      COMMON/GENPARAM/NR,NRGRID,TWODIM,HERC,CHECK,HAAN,LOG,BOUND,
     +                 OFF,XCOL,YCOL,ZCOL,ZLABEL,TZMN,TZVAR,UZMN,UZVAR,
     +                 BZMN,BZVAR,XMIN,XMAX,YMIN,YMAX,GXMIN,GXMAX,
     +                 GYMIN,GYMAX,NRTOT
      COMMON/PUNPARAM/XINC,YINC,RNBHD,RNBMAX,MAXNBR,MODEL,
     +                 C0,C9,A9,HRDGRD,HRDFIL,JACK,SELF


C-----Start writing to output file
      IF(KTYPE.EQ.'B') THEN
         WRITE(OUT,*) 'Block Kriging Output'
      ELSEIF(KTYPE.EQ.'P') THEN
         WRITE(OUT,*) 'Punctual Kriging Output'
      ELSE
         WRITE(OUT,*) 'Punctual Kriging Neighborhood Analysis Output'
      ENDIF
      IF(FTITLE.NE.'      ') WRITE(OUT,*) 'Run title: ',FTITLE
      WRITE(OUT,*) 'Input file: ',FILNAM
      WRITE(OUT,204) NCOL,NROW
 204  FORMAT(1X,'(',I2,' columns x ',I4,' rows)')
      IF(.NOT.TWODIM) THEN
         WRITE(OUT,208) XCOL,ZCOL
 208     FORMAT(' X,Z Cols: ',I2,', ',I2)
      ELSE
         WRITE(OUT,210) XCOL,YCOL,ZCOL
 210     FORMAT(' X,Y,Z Cols: ',I2,', ',I2,', ',I2)
      ENDIF
      IF(ZLABEL.NE.'      ') WRITE(OUT,*) 'Z col label: ',ZLABEL
      WRITE(OUT,*)'Defined Boundaries'
      WRITE(OUT,*) '   Data (Measured) Points'
      WRITE(OUT,660) XMIN,XMAX
      IF(TWODIM) WRITE(OUT,665) YMIN,YMAX
      WRITE(OUT,*) '   Interpolated Points'
      WRITE(OUT,660) GXMIN,GXMAX
      IF(TWODIM) WRITE(OUT,665) GYMIN,GYMAX
 660  FORMAT(1X,'      X:',G14.3,' -',G14.3)
 665  FORMAT(1X,'      Y:',G14.3,' -',G14.3)

      WRITE(OUT,*) 'Number of Measured Points'
      WRITE(OUT,215) NROW
      WRITE(OUT,216) NRTOT
      WRITE(OUT,217) NR
      IF(.NOT.HRDGRD) WRITE(OUT,218) NRGRID
 215  FORMAT(4X,'total in input dataset: ',I4)
 216  FORMAT(4X,'total valid in dataset: ',I4)
 217  FORMAT(4X,'within dataset bounds: ',I4)
 218  FORMAT(4X,'within interpolation bounds: ',I4)

      IF(LOG) THEN
         WRITE(OUT,*) 'Z-values log-n transformed:'
         WRITE(OUT,220) OFF
 220     FORMAT('    Ln(x+a) where a =',F6.1)
      ENDIF
      WRITE(OUT,*) 'General Statistics'
      WRITE(OUT,*) '(For all measured points within data boundaries)'
      IF(.NOT.LOG) THEN
         WRITE(OUT,630) UZMN
         WRITE(OUT,635) UZVAR
         WRITE(OUT,640) UZVAR**.5
         WRITE(OUT,645) NR
 630     FORMAT(1X,'   Mean:     ',G14.4)
 635     FORMAT(1X,'   Variance: ',G14.4)
 640     FORMAT(1X,'   Std. Dev: ',G14.4)
 645     FORMAT(1X,'   N:        ',I10)
      ELSE
         IF(HAAN) THEN
            WRITE(OUT,725)
 725        FORMAT(14X,'Untransformed',3X,'Transformed',3X,
     +                 'Backtransformed (Haan)')
         ELSE
            WRITE(OUT,726)
726         FORMAT(14X,'Untransformed',3X,'Transformed',3X,
     +                 'Backtransformed (Std)')
         ENDIF
         WRITE(OUT,730) UZMN,TZMN,BZMN
         WRITE(OUT,735) UZVAR,TZVAR,BZVAR
         WRITE(OUT,740) UZVAR**.5, TZVAR**.5, BZVAR**.5
         WRITE(OUT,745) NR,NR,NR
730      FORMAT(1X,'   Mean:     ',G14.4,1X,G14.4,1X,G14.4)
735      FORMAT(1X,'   Variance: ',G14.4,1X,G14.4,1X,G14.4)
740      FORMAT(1X,'   Std. Dev: ',G14.4,1X,G14.4,1X,G14.4)
745      FORMAT(1X,'   N:        ',I10,1X,I10,1X,I10)
      ENDIF
      WRITE(OUT,*)
      IF(KTYPE.EQ.'B') THEN
         WRITE(OUT,'(1X,A16,F10.3)')'Block width: ',WIDTH
         WRITE(OUT,'(1X,A16,F10.3)')'      variance: ',BLVAR
      ELSE
         IF(HRDGRD) WRITE(OUT,*) 'Interpolation gridpoints from ',
     +                          'file ',HRDFIL
         IF(JACK.OR.SELF) WRITE(OUT,*) 'Interpolation gridpoints same ',
     +                      'as input file gridpoints'
      ENDIF
      WRITE(OUT,*)
      IF(KTYPE.NE.'N') THEN
         WRITE(OUT,*) 'Neighborhood Definition'
         WRITE(OUT,*) '  Parameter         Maximum    Used'
         IF(JACK) THEN
            WRITE(OUT,225) NR-1,MAXNBR
         ELSE
            WRITE(OUT,225) NR,MAXNBR
         ENDIF
225      FORMAT(1X,    '  No. of neighbors:',I5,9X,I5)
         WRITE(OUT,226) RNBMAX,RNBHD
226      FORMAT(1X,    '  Search radius:   ',G11.3,1X,G11.3)
         WRITE(OUT,*)
      ENDIF

      WRITE(OUT,*) 'Semivariogram model'
      IF (MODEL.EQ.1) THEN
         WRITE(OUT,*)'Linear to range:'
         WRITE(OUT,*)'  for H(i) < A, Gamma(i)= C0+[(C-C0)/A]*H(i)'
         WRITE(OUT,*)'           > A, Gamma(i)= C'
      ELSEIF(MODEL.EQ.2) THEN
         WRITE(OUT,*)'Spherical:'
         WRITE(OUT,*)'  for H(i) < A, Gamma(i)= C0+(C-C0)*',
     +               '[1.5*(H(i)/A]'
         WRITE(OUT,*)'           > A, Gamma(i)= C'
      ELSEIF(MODEL.EQ.3) THEN
         WRITE(OUT,*)'Exponential'
         WRITE(OUT,*)'  Gamma(i)= C0+(C-C0)*[1-EXP(-H(i)/Ao)]'
      ELSEIF(MODEL.EQ.4) THEN
         WRITE(OUT,*)'Gaussian'
         WRITE(OUT,*)'  Gamma(i) =',
     +   ' C0-(C-C0)*[1-EXP(-H(i)**2/(A/3**.5)**2)]'
      ENDIF
      WRITE(OUT,*)'               where C0 = ',SNGL(C0)
      WRITE(OUT,*)'                      C = ',SNGL(C9)
      IF(MODEL.LT.3) THEN
         WRITE(OUT,*)'                   A = ',SNGL(A9)
      ELSE
         WRITE(OUT,*)'                  Ao = ',SNGL(A9)
      ENDIF
      WRITE(OUT,*)

      WRITE(OUT,*) 'Interpolation Output'
      IF(LOG) THEN
         IF(HAAN) THEN
            WRITE(OUT,*) 'N.B. Krige estimates backtransformed ',
     +                   '(Haan 1977)'
         ELSE
            WRITE(OUT,*) 'N.B. Krige estimates not backtransformed'
         ENDIF
      ENDIF
      RETURN
      END
C***************************************************************************
C***************************************************************************

      SUBROUTINE GETNBRS(XPT,YPT,NNBRS,NBRLOC)

      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4        DIST(1000),DNEAR(100)
      INTEGER       XCOL,YCOL,ZCOL,NBRLOC(100)
      LOGICAL       TWODIM,LOG,HERC,CHECK,HAAN,HRDGRD,BOUND,JACK,SELF
      CHARACTER*64 HRDFIL,ZLABEL

      COMMON/DATACOLS/X(1000),Y(1000),Z(1000)
      COMMON/GENPARAM/NR,NRGRID,TWODIM,HERC,CHECK,HAAN,LOG,BOUND,
     +                OFF,XCOL,YCOL,ZCOL,ZLABEL,TZMN,TZVAR,UZMN,UZVAR,
     +                BZMN,BZVAR,XMIN,XMAX,YMIN,YMAX,GXMIN,GXMAX,
     +                GYMIN,GYMAX,NRTOT
      COMMON/PUNPARAM/XINC,YINC,RNBHD,RNBMAX,MAXNBR,MODEL,
     +                CO,C9,A9,HRDGRD,HRDFIL,JACK,SELF

C-----Find 100 or MAXNBR of nearest neighbors; reject distances > RNBHD;
C     if Jack=.true. XPT,YPT itself ignored (dist.le.1d-7)

C-----Create array DIST of distance of every element from block center
      DO 110 I=1,NR
         DISTX=(XPT-X(I))**2
         DISTY=(YPT-Y(I))**2
         DIST(I)=SNGL((DISTX+DISTY)**.5)
 110  CONTINUE

C-----Create array NBRLOC(MAXNBRS) of addresses of neighborhood points
C       array DNEAR holds distance for each address; jump out at maxnbrs
      NNBRS=0
      DO 130 I=1,NR
         IF((DIST(I).GT.RNBHD).OR.(JACK.AND.(DIST(I).LE.1D-7))) GOTO 130
            NNBRS=NNBRS+1
            NBRLOC(NNBRS)=I
            DNEAR(NNBRS)=DIST(I)
            IF(NNBRS.EQ.MAXNBR) GOTO 140
 130  CONTINUE

C-----Do an initial sort on DNEAR
 140  CALL SORT(DNEAR,NBRLOC,NNBRS)

C-----If no. of neighbors within neighborhood so far =maxneighbors, then
C     may be more neighbors out there within neighborhood. Now go
C     through remainder of data set to see which of remaining items
C     to include in DNEAR;
C     Since have already found the max no. of nearest neighbors, even
C     if future points are within the neighborhood radius they must be
C     less than the greatest radius thus far encountered to be included
C     as a neighbor; therefore no need to test whether that point is in
C     the neighborhood radius.  If there are fewer points within the
C     neighborhood radius than the maxnbr of neighbors, there's no need
C     to sort further.

      IF(NNBRS.EQ.MAXNBR) THEN
         DO 150 I=NNBRS+1,NR
            IF(DIST(I).GT.DNEAR(NNBRS)) GOTO 150
            IF(JACK.AND.(DIST(I).LE.1D-7)) GOTO 150
            NBRLOC(NNBRS)=I
            DNEAR(NNBRS)=DIST(I)
            CALL SORT(DNEAR,NBRLOC,NNBRS)
 150     CONTINUE
      ENDIF

      RETURN
      END

C***************************************************************************
C***************************************************************************

      SUBROUTINE SORT(DATA1,DATA2,N)

C-----Generalized sort routine; sorts on Data1 for N elements; ascending
C       must specify real or integer
```

```
      REAL*4 DATA1(1),SAVE1
      INTEGER DATA2(1),SAVE2

      IF(N.GE.2) THEN
         DO 20 I=2,N
           DO 10 J=1,I-1
             IF(DATA1(I).GE.DATA1(J)) GOTO 10
             SAVE1=DATA1(I)
             DATA1(I)=DATA1(J)
             DATA1(J)=SAVE1
             SAVE2=DATA2(I)
             DATA2(I)=DATA2(J)
             DATA2(J)=SAVE2
 10        CONTINUE
 20      CONTINUE
      ENDIF

      RETURN
      END

C**************************************************************************
C**************************************************************************

      SUBROUTINE NEXTPT(XPT,YPT)

      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER  XCOL,YCOL,ZCOL
      LOGICAL  TWODIM,LOG,HAAN,HERC,CHECK,HRDGRD,END,BOUND,JACK,SELF
      CHARACTER*64 HRDFIL,ZLABEL

      COMMON/GENPARAM/NR,NRGRID,TWODIM,HERC,CHECK,HAAN,LOG,BOUND,
     +                OFF,XCOL,YCOL,ZCOL,ZLABEL,TZMN,TZVAR,UZMN,UZVAR,
     +                BZMN,BZVAR,XMIN,XMAX,YMIN,YMAX,GXMIN,GXMAX,
     +                GYMIN,GYMAX,NRTOT
      COMMON/PUNPARAM/XINC,YINC,RNBHD,RNBMAX,MAXNBR,MODEL,
     +                C0,C9,A9,HRDGRD,HRDFIL,JACK,SELF

C-----Determine the next coordinate pair to be kriged
      IF (HRDGRD) THEN
         IF (.NOT.TWODIM) THEN
            YPT=-99.
            READ(7,*,END=99,IOSTAT=IERR) XPT
            IF(IERR.NE.0) THEN
               WRITE(*,*)' File error, file ',HRDFIL,';'
               GOTO 99
            ENDIF
            IF((XPT.GE.GXMIN).AND.(XPT.LE.GXMAX)) NRGRID=NRGRID+1
         ELSE
            READ(7,*,END=99,IOSTAT=IERR) XPT,YPT
            IF(IERR.NE.0) THEN
               WRITE(*,*)' File error, file ',HRDFIL,';'
               GOTO 99
            ENDIF

            IF((YPT.GE.GYMIN).AND.(YPT.LE.GYMAX).AND.
     +        (XPT.GE.GXMIN).AND.(XPT.LE.GXMAX)) NRGRID=NRGRID+1
         ENDIF
         RETURN

 99      CLOSE(7)
         XPT=-99.
         RETURN

      ELSEIF (TWODIM) THEN
         IF (YPT+YINC.LE.GYMAX) THEN
            YPT=YPT+YINC
         ELSE
            IF (XPT+XINC.LE.GXMAX) THEN
               XPT=XPT+XINC
               YPT=GYMIN
            ELSE
               XPT=-99
            ENDIF
         ENDIF

      ELSE
```

```
         IF ((XPT+XINC).LE.GXMAX) THEN
            XPT=XPT+XINC
         ELSE
            XPT=-99.
         ENDIF

      ENDIF
  999 RETURN

      END

C************************************************************************
C************************************************************************

      DOUBLE PRECISION FUNCTION GAMMA(H)

      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER       XCOL,YCOL,ZCOL
      LOGICAL       HRDGRD,CHECK,TWODIM,LOG,HAAN,HERC,BOUND,JACK,SELF
      CHARACTER*64 HRDFIL,ZLABEL

      COMMON/GENPARAM/NR,NRGRID,TWODIM,HERC,CHECK,HAAN,LOG,BOUND,
     +                 OFF,XCOL,YCOL,ZCOL,ZLABEL,TZMN,TZVAR,UZMN,UZVAR,
     +                 BZMN,BZVAR,XMIN,XMAX,YMIN,YMAX,GXMIN,GXMAX,
     +                 GYMIN,GYMAX,NRTOT
      COMMON/PUNPARAM/XINC,YINC,RNBHD,RNBMAX,MAXNBR,MODEL,
     +                 C0,C9,A9,HRDGRD,HRDFIL,JACK,SELF

C-----Determine the value GAMMA
C     NB: Although if h.ge.A9 below evaluates to GAMMA=TZVAR-C0,
C     float point processing mandates using full equation to avoid crash

      IF(MODEL.EQ.1) THEN
         IF (H.LT.A9) GAMMA=TZVAR-(C0+((C9-C0)/A9)*H)
         IF (H.GE.A9) GAMMA=TZVAR-(C0+((C9-C0)/A9)*A9)

      ELSEIF(MODEL.EQ.2) THEN
         IF(H.LT.A9) GAMMA=TZVAR-(C0+(C9-C0)*(1.5*(H/A9)-
     +                        0.5*((H/A9)**3)))
         IF(H.GE.A9) GAMMA=TZVAR-(C0+(C9-C0)*(1.5*(A9/A9)-
     +                        0.5*((A9/A9)**3)))

      ELSEIF(MODEL.EQ.3) THEN
         GAMMA=TZVAR-(C0+(C9-C0)*(1-DEXP(-H/A9)))

      ELSEIF(MODEL.EQ.4) THEN
         GAMMA=TZVAR-(C0-(C9-C0)*(1-DEXP(-(H**2/(A9/3**.5)**2))))

      ENDIF

  999 RETURN
      END

C************************************************************************
C************************************************************************

      SUBROUTINE SIMUL(FN,BG,NRANK,ERROR)

      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8  FN(101,101),BG(101)
      LOGICAL ERROR

      ERROR=.FALSE.

C-----Perform outer loop from row 1 to 1 < the rank(NRANK) of the matrix
      DO 5 LL=1,NRANK-1
         IF (ABS(FN(LL,LL)).GT.1D-7) THEN
            PIVOT = 1.0/FN(LL,LL)
            DO 10 I=LL+1,NRANK
               R=FN(I,LL)*PIVOT
               DO 15 J=I,NRANK
                  FN(J,I)=FN(J,I)-(FN(J,LL)*R)
 15            CONTINUE
               BG(I)=BG(I)-(BG(LL)*R)
 10         CONTINUE
         ELSE
            ERROR=.TRUE.
            RETURN
         ENDIF
 5    CONTINUE

      DO 20 LL=NRANK,1,-1
         IF (FN(LL,LL).NE.0.0) THEN
            BG(LL)=BG(LL)/FN(LL,LL)
            IF (LL.GT.1) THEN
               DO 25 J=1,LL-1
                  BG(J)=BG(J)-(BG(LL)*FN(LL,J))
 25            CONTINUE
            ENDIF
         ELSE
            ERROR=.TRUE.
            RETURN
         ENDIF
 20   CONTINUE

 999  RETURN
      END

C***********************************************************************
C***********************************************************************

      SUBROUTINE CALCBG(XPT,YPT,BG,NNBRS,NBRLOC)

      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8       BG(101)
      INTEGER      NBRLOC(100)

      COMMON/DATACOLS/X(1000),Y(1000),Z(1000)

C-----Determine the matrix BG
      BG(NNBRS+1) = 1.0
      DO 10 I=1,NNBRS
         IP=NBRLOC(I)
         DISTX=X(IP)-XPT
         DISTY=Y(IP)-YPT
         DD=DSQRT(DISTX**2+DISTY**2)
         BG(I) = GAMMA(DD)
 10   CONTINUE

      RETURN
      END

C***********************************************************************
C***********************************************************************

      SUBROUTINE CALCFN(FN,NNBRS,NBRLOC,TZVAR)

      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8       FN(101,101)
      INTEGER      NBRLOC(100)
      LOGICAL      JACK,SELF,HRDGRD
      CHARACTER*64 HRDFIL

      COMMON/DATACOLS/X(1000),Y(1000),Z(1000)
      COMMON/PUNPARAM/XINC,YINC,RNBHD,RNBMAX,MAXNBR,MODEL,
     +                C0,C9,A9,HRDGRD,HRDFIL,JACK,SELF

C-----Initialize the matrix
      DO 10 I=1,NNBRS+1
         DO 15 J=1,NNBRS+1
            FN(I,J)=0.0
 15      CONTINUE
 10   CONTINUE

      FN(1,1) = TZVAR-C0

C-----Start calculations for FN
      DO 20 I=2,NNBRS
         J2=I-1
         IP1=NBRLOC(I)
         DO 25 J=1,J2
            IP2=NBRLOC(J)
            DISTX = X(IP1) - X(IP2)
            DISTY = Y(IP1) - Y(IP2)
```

```
          DD = DSQRT(DISTX**2+DISTY**2)
          FN(I,J) = GAMMA(DD)
          FN(J,I) = FN(I,J)
 25     CONTINUE
 20   CONTINUE

C-----Finish off the outer row and column of FN
      DO 30 I=1,NNBRS
         FN(I,I) = FN(1,1)
         FN(NNBRS+1,I) = 1.0
         FN(I,NNBRS+1) = 1.0
 30   CONTINUE
      FN(NNBRS+1,NNBRS+1) = 0.0

      RETURN
      END

C***************************************************************************
C***************************************************************************

      DOUBLE PRECISION FUNCTION CALCZ(NNBRS,NBRLOC,BG)

      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8  BG(101)
      INTEGER NBRLOC(100)

      COMMON/DATACOLS/X(1000),Y(1000),Z(1000)

C-----Calculate the Z estimate
      ZX=0.0
      DO 10 I=1,NNBRS
         IKR=NBRLOC(I)
         ZX=BG(I)*Z(IKR)+ZX
 10   CONTINUE

C-----Final Result
      CALCZ=ZX

      RETURN
      END

C***************************************************************************
C***************************************************************************

      DOUBLE PRECISION FUNCTION CALCVAR(NNBRS,BG,BF,TZVAR)

      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8      BG(101),BF(101)

C-----Determine the estimation variance
      TCVAR=0.0
      DO 10 I=1,NNBRS+1
         TCVAR=TCVAR-(BG(I)*BF(I))
 10   CONTINUE

C-----Final estimation variance
      CALCVAR=TCVAR+TZVAR

      RETURN
      END

C***************************************************************************
C***************************************************************************
C**********************                       ****************************
C********************** End of Punctual.for ****************************
C**********************                       ****************************
C***************************************************************************
C***************************************************************************
```

Punctual Kriging output file Punctual.out

```
 Punctual Kriging Output
 Run title: Sample run: Punctual krig
 Input file: Sample.prn
 ( 3 columns x  256 rows)
 X,Y,Z Cols:  1,  2,  3
 Z col label: Zvariate
 Defined Boundaries
    Data (Measured) Points
       X:      .200    -      79.9
       Y:      .000    -      79.4
    Interpolated Points
       X:      .200    -      79.9
       Y:      .000    -      79.4
 Number of Measured Points
    total in input dataset:  256
    total valid in dataset:  248
    within dataset bounds:  248
    within interpolation bounds:  248
 Z-values log-n transformed:
    Ln(x+a) where a =   1.0
 General Statistics
 (For all measured points within data boundaries)
              Untransformed  Transformed  Backtransformed (Haan)
    Mean:          .3977         .3242          .3972
    Variance:     .4575E-01     .2054E-01      .3284E-02
    Std. Dev:     .2139         .1433          .5730E-01
    N:              248        248        248


 Neighborhood Definition
    Parameter          Maximum     Used
    No. of neighbors:  248           32
    Search radius:      113.        113.

 Semivariogram model
 Spherical:
   for H(i) < A, Gamma(i)= C0+(C-C0)*[1.5*(H(i)/A]
            > A, Gamma(i)= C
                 where C0 =    1.227000E-002
                        C =   2.388000E-002
                        A =          67.7400000

 Interpolation Output
 N.B. Krige estimates backtransformed (Haan 1977)
      X          Y              Est Z          Std Dev            Neighbors
   .2000      .0000           .60135             .46148E-01      32
   .2000      9.000           .42850             .46396E-01      32
   .2000      18.00           .51865             .46693E-01      32
   .2000      27.00           .71307             .46469E-01      32
   .2000      36.00           .55273             .46127E-01      32
   .2000      45.00           .42876             .46104E-01      32
   .2000      54.00           .66840             .46238E-01      32
   .2000      63.00           .58036             .45223E-01      32
   .2000      72.00           .37591             .46132E-01      32
   9.200      .0000           .40787             .46699E-01      32
   9.200      9.000           .50066             .44847E-01      32
   9.200      18.00           .45957             .45795E-01      32
   9.200      27.00           .50209             .45380E-01      32
   9.200      36.00           .56083             .45865E-01      32
   9.200      45.00           .39099             .45094E-01      32
   9.200      54.00           .41370             .45845E-01      32
   9.200      63.00           .44520             .44686E-01      32
   9.200      72.00           .41106             .46104E-01      32
   18.20      .0000           .28832             .46831E-01      32
   18.20      9.000           .19571             .45190E-01      32
   18.20      18.00           .33388             .45774E-01      32
   18.20      27.00           .25824             .45394E-01      32
   18.20      36.00           .24784             .45596E-01      32
   18.20      45.00           .37807             .45872E-01      32
   18.20      54.00           .67118             .45119E-01      32
   18.20      63.00           .50163             .45547E-01      32
   18.20      72.00           .58608             .45387E-01      32
   27.20      .0000           .14714             .46340E-01      32
   27.20      9.000           .16204             .45867E-01      32
   27.20      18.00           .15237             .45484E-01      32
```

27.20	27.00	.22282	.45688E-01	32
27.20	36.00	.24220	.46080E-01	32
27.20	45.00	.28579	.44969E-01	32
27.20	54.00	.28433	.45233E-01	32
27.20	63.00	.44551	.44572E-01	32
27.20	72.00	.46334	.45463E-01	32
36.20	.0000	.24031	.46828E-01	32
36.20	9.000	.26554	.46124E-01	32
36.20	18.00	.15954	.45094E-01	32
36.20	27.00	.16166	.45831E-01	32
36.20	36.00	.35656	.44780E-01	32
36.20	45.00	.30276	.45486E-01	32
36.20	54.00	.33613	.45362E-01	32
36.20	63.00	.41905	.45899E-01	32
36.20	72.00	.38987	.45499E-01	32
45.20	.0000	.19481	.45633E-01	32
45.20	9.000	.17672	.44970E-01	32
45.20	18.00	.52518	.45244E-01	32
45.20	27.00	.32074	.44998E-01	32
45.20	36.00	.26264	.46036E-01	32
45.20	45.00	.38807	.45834E-01	32
45.20	54.00	.32501	.45084E-01	32
45.20	63.00	.36678	.45333E-01	32
45.20	72.00	.42684	.45531E-01	32
54.20	.0000	.21460	.46069E-01	32
54.20	9.000	.24851	.44885E-01	32
54.20	18.00	.32794	.45064E-01	32
54.20	27.00	.34807	.45889E-01	32
54.20	36.00	.30492	.45440E-01	32
54.20	45.00	.27368	.45614E-01	32
54.20	54.00	.23761	.45645E-01	32
54.20	63.00	.77784	.45412E-01	32
54.20	72.00	.28997	.45614E-01	32
63.20	.0000	.21475	.45653E-01	32
63.20	9.000	.26617	.46345E-01	32
63.20	18.00	.20112	.45296E-01	32
63.20	27.00	.34343	.45785E-01	32
63.20	36.00	.60978	.46059E-01	32
63.20	45.00	.45818	.45619E-01	32
63.20	54.00	.62323	.44642E-01	32
63.20	63.00	.52122	.45728E-01	32
63.20	72.00	1.1983	.45185E-01	32
72.20	.0000	.28995	.45931E-01	32
72.20	9.000	.32103	.45080E-01	32
72.20	18.00	.33788	.45603E-01	32
72.20	27.00	.50718	.45670E-01	32
72.20	36.00	.51250	.45498E-01	32
72.20	45.00	.53438	.46004E-01	32
72.20	54.00	.42795	.45607E-01	32
72.20	63.00	.67666	.45466E-01	32
72.20	72.00	.62523	.45169E-01	32

File Block.hlp; to accompany

Program Block.for,
Block Kriging Analysis
Version 11.10.86 Copyright (c) 1986 G.P. Robertson

The correct citation for this program is
Robertson, G.P. 1987. Geostatisitics in ecology: interpolating with known variance. Ecology 68:

Later releases of this program available from: Computer Services Laboratory
W.K. Kellogg Biol. Station
Michigan State University
Hickory Corners, MI 49060.

Program parameters:

1. Hardware requirements. This version was written in Microsoft Fortran version 3.3 designed to run under the IBM MS-DOS 2.0+ operating system. Compilation by other compilers compatible with the ANSI 77 standard should be possible.

2. Data Input. Data is read into the program as ASCII characters in free format from a user-specified file. The first 4 lines of the input file are treated as header lines -- they should not contain data. If the first characters of the third line are numeric, they are assumed to represent the number of columns in each data record. A sample header:

```
Line 1: File xxxxx.prn from supporting files xxyyy.ext and xxxyy.ext;
     2: Small grid samples taken 5/19.....
     3: 15 columns: Sample#, Xcor, Ycor, z1,z2,z3,z4,z5,
     4:         z6,z7,z8,z9,z10,z11,z12.
     5: 1 0 15 4.3 2.11 .0003 -99. 2.3 3 5 4 3 2 2 1.9997
```

If the column no. value on line 3 is not correct or if there are different numbers of columns in any row, the data will not be read correctly and calculations will be incorrect. Missing values are represented by the value -99. Data size limitations are 15 columns x 1000 rows, though redimensioning key variables will allow indefinite input file sizes.

3. Output. Output data is written to a user-specified output file or to the screen if no output file is specified. All data is ASCII format and can be subsequently printed or read into another program.

4. User-specified prompts:

a) Main menu; a) read data file; see limitations above;
b) scan curent data set; use cntrl-s to halt scrolling;
c) block kriging analysis;
d) help; this output;
e) exit program;

b) input file name; conform to MSDOS convention;

c) run title; as desired for labeling output graphs; 64 char. max;

d) number of dimensions; either 1 or 2;

e) x,z or x,y,z columns: columns in which x coordinates, y coordinates, and active z-value can be found;

f) z-label: as desired for labeling output, 64 char. max;

g) use subset of data set; to use only part of the input (measured) data grid in the interpolation analysis;

h) transform z-value; if requested, z is logn-transformed with the option of adding a constant to each value before transforming; because of the discontinuous nature of this distribution over the range <-1 to >1, all values <1.0 are ignored for the duration of the analysis; note that the transformation lasts only until the next Punctual menu; the statistics output give back-transformed values (Haan 1977) if requested -- see note below;

i) specify semivariogram model; 4 models are available: linear to specified range, spherical, exponential, and gaussian. Model parameters are C0 (y-intercept), C (sill value), and A (range in the case of the linear and spherical models) or Ao (some factor of range in exponential and gaussian models). These model parameters can be taken directly from the output of the semivariogram program after running it through SAS's nonlinear estimation routines;

j) neighborhood parameters; user must provide a) the maximum number of neighbors M to use for interpolating any given grid point (100 max), and b) the maximum radius within which to search for neighbors (up to the maximum distance separating any two points to be interpolated). All neighbors within the maximum radius are sorted by distance and the nearest M are used for the interpolation. Higher M's increase processing time substantially.

k) use subset of interpolation grid; if only part of the interpolation grid is to be calculated; note that points in the measured data grid that are outside the interpolation grid may be used in the interpolatin of these points if they satisfy the neighborhood requirements.

l) specify output file;
N = results are not read to output file, appear on screen only;
A = appends results to last output file opened;
f = name of new output file; if file exists, can write over it or append to it as specified;

N.B. Read carefully the annotations on the output file; where the backtransformation of transformed values was requested, Haan's (1977) method is used, where the backtransformed mean = exp([mean of logn values]+0.5*var of logn values)]-A; and the backtransformed variance = ([mean of untransformed values]^2)*(exp[var of logn values]-1). Where A = the constant added to original values before ln transformation in order to bring original values to >1.0 before ln transformation. (See Haan, C.T. 1977. Chapter 6. Some continuous probability distributions. Statistical Methods in Hydrology. The Iowa State Univ. Press, Ames, Iowa.)

```
$storage:2

      PROGRAM BLOCK

C     Version 11.10.86
C     Copyright (c) 1986 G.P. Robertson. All rights reserved.

C     The correct citation for this program is Robertson, G.P. 1987.
C         Geostatistics in ecology: interpolating with known variance.
C         Ecology 68:(in press).
C
C     This program is being maintained at the Computer Services Laboratory
C                                             W.K. Kellogg Biological Station
C                                             Michigan State University
C                                             Hickory Corners, MI 49060.
C
C        Please report significant bugs and well-documented enhancements
C        to this address for inclusion in later versions.

C     Documention for users appears in file Block.hlp.

C     Subroutines: DATREAD, SCAN, UCASE, LCASE, BLKRIG, SETBLK, BOUNDS,
C                  TRANSFORM, INTBOUNDS, DEFBMODEL, DEFBNEIGH, STATS, GETOUT,
C                  TEXTBOUT, BLKVAR, BLCOV, GETBNBRS, SORT, GAMMAB, SIMULB

C-----Variable list
C
C  NAME         FUNCTION
C  ----         --------
C  FILNAM       input data file
C  OUTFIL       file where output data sent
C  PRESOUT      last output file named (used in appending)
C  FTITLE       run title
C  ZLABEL       z-variable title
C  XCOL         column in filnam with x-coordinate values
C  YCOL         column with y-coordinate values
C  ZCOL         column with z-corrdinate values
C  XMAX         greatest x-coordinate
C  XMIN         least x-coordinate
C  YMAX         greatest y-coordinate
C  YMIN         least y-coodinate
C  GXMAX        greatest interpolation grid x-coord
C  GYMAX        greatest interp. grid y-coord
C  GXMIN
C  GYMIN
C  LOG          log transformation? (logical)
C  HAAN         back-tranform using Haan (1977) (logical)
C  NR           number of valid data elements
C  RNBHD        radius of neighborhood (user defined)
C  RNBMAX       maximum radius neighborhood (data defined)
C  MAXNBR       maximum no. of neighbors (user defined)
C  MODEL        semivariogram model
C  C0           y-intercept of semivariogram
C  C9           C of semivariogram model
C  A9           A of semivariogram model
C  NXBL         no. of blocks in x direction
C  NYBL                          y direction
C  XBL          x coordinate for center of current block
C  YBL          y coordinate for center of current block

      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4        DATA
      INTEGER       CHOICE,XCOL,YCOL,ZCOL,BL
      CHARACTER*64  FILNAM,OUTFIL,LSTOUT,FTITLE,ZLABEL
      CHARACTER*1   HEAD1,HEAD2,HEAD3,HEAD4,LINE(79)
      LOGICAL       TWODIM,LOG,HAAN,HERC,BOUND,READIN,CHECK,OPEND,THERE

      COMMON/DATAIN/DATA(1000,15)
      COMMON/DATACOLS/X(1000),Y(1000),Z(1000)
      COMMON/FILINFO/FILNAM,OUTFIL,LSTOUT,FTITLE,NCOL,NROW,
     +               HEAD1(79),HEAD2(79),HEAD3(79),HEAD4(79)
      COMMON/GENPARAM/NR,NRGRID,TWODIM,HERC,CHECK,HAAN,LOG,BOUND,
     +                OFF,XCOL,YCOL,ZCOL,ZLABEL,TZMN,TZVAR,UZMN,UZVAR,
     +                BZMN,BZVAR,XMIN,XMAX,YMIN,YMAX,GXMIN,GXMAX,
     +                GYMIN,GYMAX,NRTOT
      COMMON/BLKPARAM/RNBHD,RNBMAX,MAXNBR,MODEL,C0,C9,A9,
     +                WIDTH,NXBL,NYBL,BLVAR
```

```
C-----Write menu header
      DO 3 I=1,5
         WRITE(*,*)
 3    CONTINUE
      WRITE(*,*)'Program Block'
      WRITE(*,*)'  Block Kriging Analysis'
      WRITE(*,*)'  Version 11.10.86'
      WRITE(*,*)' '
      WRITE(*,*)'See file Block.hlp and source code listing'
      WRITE(*,*)'  for documentation and copyright information'
      WRITE(*,'(//1X,A,$)')'Hit <return> to continue.....'
      READ(*,'(A1)') YORN
      WRITE(*,'(//////////////////////////1X)')

      READIN=.FALSE.
C-----Choose function
 1    WRITE(*,*)
      CHECK=.FALSE.
      IF (.NOT.READIN) THEN
         WRITE(*,400) 'None'
      ELSE
         WRITE(*,400) FILNAM
         WRITE(*,405) NCOL
         WRITE(*,410) NROW
      ENDIF
 400  FORMAT(/1X,'Current file: ',A16)
 405  FORMAT(1X,'   number of columns:',I3)
 410  FORMAT(1X,'   number of rows:   ',I5)

      WRITE(*,*)
      WRITE(*,*)'Choose a function:'
      WRITE(*,*)'    1. Read in data file'
      WRITE(*,*)'    2. Scan current data set'
      WRITE(*,*)'    3. Block kriging analysis'
      WRITE(*,*)'    4. Help'
      WRITE(*,*)'    5. End Program'
      WRITE(*,*)
      WRITE(*,'(4X,A14,$)')'Enter choice: '
    2 READ(*,*,ERR=600) CHOICE

      IF (CHOICE.EQ.1) THEN
         CALL DATREAD(READIN)
         IF(.NOT.READIN) GOTO 1
      ELSEIF (CHOICE.EQ.2) THEN
         IF(.NOT.READIN) CALL DATREAD(READIN)
         IF(.NOT.READIN) GOTO 1
         CALL SCAN
      ELSEIF (CHOICE.EQ.3) THEN
         IF (.NOT.READIN) CALL DATREAD(READIN)
         IF(.NOT.READIN) GOTO 1
         CALL BLKRIG
      ELSEIF (CHOICE.EQ.4) THEN
C---------SEE IF HELP FILE PRESENT
          INQUIRE(FILE='BLOCK.HLP', EXIST=THERE)
          IF (THERE) THEN
            DO 12 I=1,10
              WRITE(*,*)
 12         CONTINUE
            OPEN (7,FILE='BLOCK.HLP',STATUS='OLD')
 10         DO 11 J=1,24
              READ (7,'(79A1)',END=20) (LINE(I),I=1,79)
              WRITE(*,'(1X,79A1)') (LINE(I),I=1,79)
 11         CONTINUE
            WRITE(*,'(20X,A,$)')'(.....hit <return> to continue.....)'
            READ(*,'(A1)') YORN
            GOTO 10
 20         CLOSE(7)
            WRITE(*,'(/20X,A,$)')'(.....hit <return> to continue.....)'
            READ(*,'(A1)') YORN
            WRITE(*,'(//////////////////////////1X)')
         ELSE
           WRITE(*,*)'    file Block.hlp not in directory...'
         ENDIF
       ELSEIF (CHOICE.EQ.5) THEN
         INQUIRE(6,OPENED=OPEND)
         IF(OPEND) CLOSE(6)
         WRITE(*,*)
         WRITE(*,*)'Program Block;'
         WRITE(*,*)'Normal termination.'
         WRITE(*,*)
         GOTO 999
      ELSE
         GOTO 600
      ENDIF

      GOTO 1

C-----Error trapping
 600  WRITE(*,*) '      entry error......'
 602  WRITE(*,'(7X,A,$)') 'try again: '
      GOTO 2

 999  END

C***********************************************************************
C***********************************************************************

      SUBROUTINE DATREAD(READIN)

C     Data is read in in a free format style by columns; if
C     the no. of columns is not on 3rd header record user is asked for
C     value.

      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4       DATA
      INTEGER      XCOL,YCOL,ZCOL
      CHARACTER*64 TMPNAM,FILNAM,OUTFIL,LSTOUT,FTITLE,ZLABEL
      CHARACTER*1  HEAD1,HEAD2,HEAD3,HEAD4,DATA1(79),YORN
      LOGICAL      TWODIM,THERE,READIN,LOG,HERC,CHECK,HAAN,BOUND

      COMMON/DATAIN/DATA(1000,15)
      COMMON/FILINFO/FILNAM,OUTFIL,LSTOUT,FTITLE,NCOL,NROW,
     +               HEAD1(79),HEAD2(79),HEAD3(79),HEAD4(79)
      COMMON/GENPARAM/NR,NRGRID,TWODIM,HERC,CHECK,HAAN,LOG,BOUND,
     +                OFF,XCOL,YCOL,ZCOL,ZLABEL,TZMN,TZVAR,UZMN,UZVAR,
     +                BZMN,BZVAR,XMIN,XMAX,YMIN,YMAX,GXMIN,GXMAX,
     +                GYMIN,GYMAX,NRTOT

C-----Get input file
  1   WRITE(*,*)
  6      WRITE(*,100)
  5      READ(*,'(A64)') TMPNAM
         CALL UCASE(TMPNAM)
         IF((TMPNAM.EQ.'QUIT').OR.(TMPNAM.EQ.'EXIT').OR.
     +      (TMPNAM.EQ.'END').OR.(TMPNAM(1:2).EQ.'NO')) RETURN

C-----See if file exists
      INQUIRE(FILE=TMPNAM,EXIST=THERE)
      IF (.NOT.THERE) THEN
         WRITE(*,105)
         WRITE(*,106)
         GOTO 5
      ENDIF
 100  FORMAT(1X,'Enter input filename: ',$)
 105  FORMAT(4X,'File does not exist; ',A20)
 106  FORMAT(4X,'try again or enter ''Quit'': ',$)

C-----Get file or run title
      WRITE(*,'(A18,$)') ' Enter run title: '
      READ(*,'(A64)') FTITLE

C-----Read header
      OPEN (5,FILE=TMPNAM,STATUS='OLD',ERR=605)
      READ(5,'(79A1)',ERR=605) (HEAD1(I), I=1,79)
      READ(5,'(79A1)',ERR=605) (HEAD2(I), I=1,79)
      READ(5,'(79A1)',ERR=605) (HEAD3(I), I=1,79)
      READ(5,'(79A1)',ERR=605) (HEAD4(I), I=1,79)
      WRITE(*,*)
      WRITE(*,*) 'File header (top 4 records in file):'
      WRITE(*,'(1X,79A1)') (HEAD1(I), I=1,79)
      WRITE(*,'(1X,79A1)') (HEAD2(I), I=1,79)
      WRITE(*,'(1X,79A1)') (HEAD3(I), I=1,79)
      WRITE(*,'(1X,79A1)') (HEAD4(I), I=1,79)
      READ(5,'(79A1)',ERR=605) (DATA1(I), I=1,79)
      WRITE(*,*)
      WRITE(*,*) 'First data record:'
      WRITE(*,'(1X,79A1)') (DATA1(I),I=1,79)
      WRITE(*,*)
      BACKSPACE 5

C-----Read number of columns in file
      NUM=0
      IPL=0
      DO 10 I=1,79
         IF (HEAD3(I).EQ.' ') THEN
            IF(IPL.GT.0) GOTO 15
            GOTO 10
         ENDIF
         ICH=ICHAR(HEAD3(I))-48
         IF ((ICH.GE.0).AND.(ICH.LE.9)) THEN
            NUM=(NUM*(10**IPL))+ICH
            IPL=IPL+1
            IF (NUM.GT.999) GOTO 15
         ENDIF
 10   CONTINUE

 15   IF ((NUM.LT.2).OR.(NUM.GT.15)) THEN
 17      WRITE(*,'(A33,$)')' Number of columns in data file: '
 16      READ(*,'(I4)',IOSTAT=IERR) NUM
         IF(IERR.NE.0) THEN
            WRITE(*,*)'   value must be an integer......'
            WRITE(*,'(4X,A11,$)') 'try again: '
            GOTO 16
         ENDIF
         IF (NUM.EQ.0) THEN
            CLOSE(5)
            GOTO 1
         ELSEIF ((NUM.LT.2).OR.(NUM.GT.15)) THEN
            WRITE(*,*)'   number of columns out of range......'
            WRITE(*,'(4X,A11,$)') 'try again: '
            GOTO 16
         ENDIF
      ENDIF
      NCOL=NUM

C-----Read data
      WRITE(*,120) NCOL,TMPNAM
      DO 20 J=1,1000
      READ(5,*,END=25,ERR=605) (DATA(J,I),I=1,NCOL)
  20  CONTINUE
         CLOSE(5)
         WRITE(*,165)
 165     FORMAT(' ...>1000 values;',/1X,
     +          '    should remainder be dropped (y/n): ',$)
         READ(*,'(A1)') YORN
         IF((YORN.NE.'Y').AND.(YORN.NE.'y')) GOTO 6

 25   CLOSE(5)
      NROW=J-1
      WRITE(*,122) NROW
 120  FORMAT(/16X,'...reading ',I2,' columns from file ',A32)
 122  FORMAT(16X,'...end of file after ',I4,' values'/)

      FILNAM(1:1)=TMPNAM(1:1)
      CALL LCASE(TMPNAM)
      FILNAM(2:64)=TMPNAM(2:64)
      READIN=.TRUE.

      WRITE(*,*)
      RETURN

C-----Error trapping code
 605  WRITE(*,*)'                ...file error......',CHAR(7),CHAR(7)
      WRITE(*,110)
      READIN=.FALSE.
 110  FORMAT(4X,'Enter input filename (or ''Quit''): ',$)
      GOTO 5

      END
```

```
C***********************************************************************
C***********************************************************************

      SUBROUTINE SCAN

      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 DATA
      CHARACTER*64 FILNAM,OUTFIL,LSTOUT,FTITLE
      CHARACTER*1  HEAD1,HEAD2,HEAD3,HEAD4

      COMMON/DATAIN/DATA(1000,15)
      COMMON/FILINFO/FILNAM,OUTFIL,LSTOUT,FTITLE,NCOL,NROW,
     +               HEAD1(79),HEAD2(79),HEAD3(79),HEAD4(79)

C-----Find out how many rows to scan
      WRITE(*,*)
 1    WRITE(*,'(A,$)')' Number of rows to scan: '
 16   READ(*,'(I4)',ERR=600) N
      IF(N.GT.NROW) N=NROW
      WRITE(*,*)
      IF(N.LE.0) RETURN

C-----Print n lines of file
C-----Start with 4 header lines
      WRITE(*,'(1X,79A1)') (HEAD1(I), I=1,79)
      WRITE(*,'(1X,79A1)') (HEAD2(I), I=1,79)
      WRITE(*,'(1X,79A1)') (HEAD3(I), I=1,79)
      WRITE(*,'(1X,79A1)') (HEAD4(I), I=1,79)

C-----Continue with all data lines
      DO 10 I=1,N
         WRITE(*,*)
         DO 20 J=1,NCOL
            A=FLOAT(J)
            IF ((A/8).EQ.INT(A/8)) THEN
               WRITE(*,*)
               WRITE(*,'(1X,A,$)') '          '
            ENDIF
            WRITE(*,100) DATA(I,J)
 100        FORMAT(' ',G10.3,$)
 20      CONTINUE
 10      CONTINUE
      WRITE(*,*)

      RETURN

C-----Error trapping code
 600  WRITE(*,*)'   Value must be an integer;'
      WRITE(*,'(4X,A11,$)') 'try again: '
      GOTO 16
      END

C***********************************************************************
C***********************************************************************

      SUBROUTINE UCASE(STRING)

      CHARACTER*64 STRING

C-----Convert lower case letters to uppercase
      DO 10 I=1,64
         IC=ICHAR(STRING(I:I))
         IF((IC.GE.97).AND.(IC.LE.122)) THEN
            STRING(I:I)=CHAR(IC-32)
         ENDIF
 10   CONTINUE

      RETURN
      END

C***********************************************************************
C***********************************************************************

      SUBROUTINE LCASE(STRING)

      CHARACTER*64 STRING
```

```
C-----Convert uppercase letters to lowercase
      DO 10 I=1,64
         IC=ICHAR(STRING(I:I))
         IF((IC.GE.65).AND.(IC.LE.90)) THEN
           STRING(I:I)=CHAR(IC+32)
         ENDIF
 10   CONTINUE

      RETURN
      END

C***********************************************************************
C***********************************************************************

      SUBROUTINE BLKRIG

      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER      XCOL,YCOL,ZCOL,OUT
      CHARACTER*1  HEAD1,HEAD2,HEAD3,HEAD4,Q,YORN
      CHARACTER*64 FILNAM,OUTFIL,LSTOUT,HRDFIL,FTITLE,ZLABEL
      LOGICAL      TWODIM,CHECK,BOUND,HERC,LOG,HAAN,ERROR,
     +             YESOUT

      DIMENSION NBRLOC(100),AV(10201),RV(101),GV(101),COV(2,2)

      COMMON/DATACOLS/X(1000),Y(1000),Z(1000)
      COMMON/FILINFO/FILNAM,OUTFIL,LSTOUT,FTITLE,NCOL,NROW,
     +               HEAD1(79),HEAD2(79),HEAD3(79),HEAD4(79)
      COMMON/GENPARAM/NR,NRGRID,TWODIM,HERC,CHECK,HAAN,LOG,BOUND,
     +                OFF,XCOL,YCOL,ZCOL,ZLABEL,TZMN,TZVAR,UZMN,UZVAR,
     +                BZMN,BZVAR,XMIN,XMAX,YMIN,YMAX,GXMIN,GXMAX,
     +                GYMIN,GYMAX,NRTOT
      COMMON/BLKPARAM/RNBHD,RNBMAX,MAXNBR,MODEL,C0,C9,A9,
     +                WIDTH,NXBL,NYBL,BLVAR

C-----If first time through, see if need to call setpun
      IF(XCOL.GT.0) THEN
         WRITE(*,105)
 105     FORMAT(/1X,'Reset run parameters? (y/n): ',$)
         READ(*,'(A1)') YORN
      ELSE
         YORN='Y'
      ENDIF

C-----Set run parameters: x,y,z cols, data grid, etc.
 31   IF ((YORN.EQ.'Y').OR.(YORN.EQ.'y')) THEN
         CALL SETBLK

      ELSEIF((YORN.NE.'N').AND.(YORN.NE.'n')) THEN
         WRITE(*,*)'            answer Y or N only......'
         WRITE(*,'(7X,A11,$)') 'try again: '
         READ(*,'(A1)') YORN
         GOTO 31
      ENDIF

C-----Ask for how to define neighbor search
      CALL DEFBNEIGH(.FALSE.)

C-----Get output file
      CALL GETOUT(OUT)
      IF(OUT.NE.0) YESOUT=.TRUE.

C-----Calculate stats
      CALL STATS

C-----Calculate block variance
      CALL BLKVAR

C-----Start writing to output file
      CALL TEXTBOUT(OUT)
      WRITE(OUT,'(1X,A,A)') 'NB: Analysis includes only those ',
     +         'measured points within interpolation bounds'
      IF(LOG.AND.(.NOT.HAAN)) WRITE(OUT,'(5X,A,A)') 'Neither ',
     +     'measured nor interpolated values are backtransformed'
      WRITE(OUT,290)
 290  FORMAT(1X,4X,'X',10X,'Y',12X,'Est Z',8X,'Std Dev',
     +       11X,'Neighbors')
```

```
C-----Give first part of status report
      IF(YESOUT) THEN
         Q=CHAR(8)
         WRITE(*,'(1X,A25,A24,$)') '    ...calculating block',
     +        '                              '
         IF(TWODIM) WRITE(*,602) '        '
 602     FORMAT(A8,$)
      ENDIF

C-----Initialize stat accumulators
      JJ=0

C-----Initialize x and y offset coordinates (to find block center)
      XMID=GXMIN+0.5*(GXMAX-GXMIN)
      XOFF=XMID-(0.5*NXBL*WIDTH)-(0.5*WIDTH)
      YMID=GYMIN+0.5*(GYMAX-GYMIN)
      YOFF=YMID-(0.5*NYBL*WIDTH)-(0.5*WIDTH)


C=================================================================C
C                                                                 C
C                   START OF MAIN BLOCK LOOP                      C
C                                                                 C
C=================================================================C


C-----Work through grid by blocks; start in southeast corner
      DO 100 I=1,NXBL
          XBL=XOFF+I*WIDTH

          DO 100 J=1,NYBL
             YBL=YOFF+J*WIDTH
              ZE=-99.
              ZV=-99.
              ZS=-99.

C-----Status report to screen
         JJ=JJ+1
         IF(YESOUT) THEN
            IF (.NOT.TWODIM) THEN
               WRITE(*,600) (Q,K=1,24),XBL,JJ,NXBL
 600           FORMAT(24A,F7.1,' (',I5,' of ',I5,')',$)
            ELSE
               WRITE(*,601) (Q,K=1,32),XBL,YBL,JJ,NXBL*NYBL
 601           FORMAT(32A,F7.1,',',F7.1,' (',I5,' of ',I5,')',$)
            ENDIF
         ENDIF

C-----Get nearest Maxnbr neighbors to center of block
  76     CALL GETBNBRS(XBL,YBL,NNBRS,NBRLOC)
         IF(NNBRS.EQ.0) GOTO 99

C-----Determine covar between each sample and the current block
         COV(1,1)=XBL
         COV(2,1)=YBL
         NNBR1=NNBRS+1
         DO 240 K=1,NNBRS
            IP=NBRLOC(K)
            COV(1,2)=X(IP)
            COV(2,2)=Y(IP)
            CALL BLCOV(COV,VOC)
            RV(K)=VOC
            GV(K)=VOC
            AV((K-1)*NNBR1+NNBR1)=1.0
            AV((NNBR1-1)*NNBR1+K)=1.0
 240     CONTINUE
         AV(NNBR1*NNBR1)=0.0

C-----Determine covar among samples in block neighborhood
         DO 250 K=1,NNBRS
            IP=NBRLOC(K)
            DO 260 L=K,NNBRS
               IP1=NBRLOC(L)
               DISTX=X(IP)-X(IP1)
               DISTY=Y(IP)-Y(IP1)
               H=DSQRT(DISTX**2+DISTY**2)
```

```
                GMH=GAMMAB(H)
                AV((K-1)*NNBR1+L)=GMH
                AV((L-1)*NNBR1+K)=GMH
 260         CONTINUE
 250      CONTINUE
          RV(NNBR1)=1.0

C-----Go for the simultaneous equations
          CALL SIMULB(RV,AV,NNBR1,1,IER)
          IF (IER.NE.0) GOTO 99

C-----Calculate krig estimate and variance
          ZV=0.0
          ZE=0.0
          DO 490 K=1,NNBRS
             IP=NBRLOC(K)
             ZE=ZE+Z(IP)*RV(K)
             ZV=ZV+GV(K)*RV(K)
 490      CONTINUE
          ZV=RV(NNBR1)+ZV-BLVAR

C-----If variance negative, set values to -99.
          IF(ZV.LT.0) THEN
             ZE=-99.
             ZS=-99.
          ELSE
C-----------Backtransform ZE and ZV as per Haan (1977) if HAAN
             IF(LOG.AND.HAAN) THEN
                ZE=DEXP(ZE+0.5*ZV)-OFF
                ZV=UZMN**2*(DEXP(TZVAR))*(1-DEXP(-ZV))
             ENDIF
             ZS=DSQRT(ZV)
          ENDIF
C-----Write results to output
  99      IF(TWODIM) THEN
             WRITE(OUT,460) XBL,YBL,ZE,ZS,NNBRS
          ELSE
             WRITE(OUT,462) XBL,ZE,ZS,NNBRS
          ENDIF

 460      FORMAT(1X,G10.4,1X,G10.4,1X,G16.5,1X,G16.5,1X,I5)
 462      FORMAT(1X,G10.4,1X,G16.5,1X,G16.5,1X,I5)


C==================================================================C
C                                                                  C
C                    END OF MAIN BLOCK LOOP                        C
C                                                                  C
C==================================================================C

C-----Go back to top and get new block coordinates
 100  CONTINUE

      WRITE(OUT,*)
      WRITE(OUT,*)

      RETURN

 999  END

C***********************************************************************
C***********************************************************************

      SUBROUTINE SETBLK

      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4       DATA
      INTEGER      XCOL,YCOL,ZCOL
      CHARACTER*1  YORN,ANSW,HEAD1,HEAD2,HEAD3,HEAD4
      CHARACTER*64 FILNAM,OUTFIL,LSTOUT,HRDFIL,FTITLE,ZLABEL
      LOGICAL      TWODIM,CHECK,BOUND,HERC,LOG,HAAN,THERE

      COMMON/DATAIN/DATA(1000,15)
      COMMON/DATACOLS/X(1000),Y(1000),Z(1000)
      COMMON/FILINFO/FILNAM,OUTFIL,LSTOUT,FTITLE,NCOL,NROW,
     +                HEAD1(79),HEAD2(79),HEAD3(79),HEAD4(79)
      COMMON/GENPARAM/NR,NRGRID,TWODIM,HERC,CHECK,HAAN,LOG,BOUND,
     +                OFF,XCOL,YCOL,ZCOL,ZLABEL,TZMN,TZVAR,UZMN,UZVAR,
     +                BZMN,BZVAR,XMIN,XMAX,YMIN,YMAX,GXMIN,GXMAX,
     +                GYMIN,GYMAX,NRTOT
      COMMON/BLKPARAM/RNBHD,RNBMAX,MAXNBR,MODEL,C0,C9,A9,
     +                WIDTH,NXBL,NYBL,BLVAR

C-----Get the number of dimensions
      WRITE(*,*)
      WRITE(*,'(A38,$)')    ' Enter number of dimensions (1 or 2): '
 5    READ(*,*,IOSTAT=IERR) N
      IF((IERR.NE.0).OR.((N.NE.1).AND.(N.NE.2))) THEN
         WRITE(*,'(4X,A,$)') 'enter 1 or 2 only: '
         GOTO 5
      ELSEIF(N.EQ.1) THEN
         TWODIM=.FALSE.
      ELSEIF (N.EQ.2) THEN
         TWODIM=.TRUE.
      ENDIF

C-----Ask user for column numbers for X, Y, and Z and get z label
 1    IF(.NOT.TWODIM) THEN
         WRITE(*,'(A38,$)') '               column numbers for X,Z: '
 2       READ(*,*,IOSTAT=IERR) XCOL,ZCOL
         IF(IERR.NE.0) THEN
            WRITE(*,650)
            GOTO 2
         ENDIF
         YCOL=XCOL
         IF ((XCOL.GT.NCOL).OR.(ZCOL.GT.NCOL)) THEN
            WRITE(*,*) '            value out of range...'
            WRITE(*,'(1X,A,$)') '            re-enter: '
            GOTO 2
         ENDIF
      ELSE
         WRITE(*,'(A38,$)') '             column numbers for X,Y,Z: '
 3       READ(*,*,IOSTAT=IERR) XCOL,YCOL,ZCOL
         IF(IERR.NE.0) THEN
            WRITE(*,650)
            GOTO 3
         ENDIF
         IF ((XCOL.GT.NCOL).OR.(YCOL.GT.NCOL).OR.(ZCOL.GT.NCOL)) THEN
            WRITE(*,*) '            value out of range...'
            WRITE(*,'(1X,A,$)') '            re-enter: '
            GOTO 3
         ENDIF
      ENDIF
      WRITE(*,'(A,$)') '                            label for Z: '
      READ(*,'(A20)') ZLABEL
      WRITE(*,*)

C-----Put data from proper columns into work arrays
      K=0
      DO 15 I=1,NROW
         IF (DATA(I,XCOL).EQ.-99.0) GOTO 15
         IF (TWODIM.AND.(DATA(I,YCOL).EQ.-99.0)) GOTO 15
         IF (DATA(I,ZCOL).EQ.-99.0) GOTO 15
         K=K+1
         X(K)=DATA(I,XCOL)
         Y(K)=0
         IF(TWODIM) Y(K)=DATA(I,YCOL)
         Z(K)=DATA(I,ZCOL)
 15   CONTINUE
      IF(K.LE.2) THEN
         WRITE(*,700)
 700     FORMAT(14X,'too few valid data in these columns.....',/)
         GOTO 1
      ENDIF
      NRTOT=K

C-----See if user wants only subset of dataset
      CALL BOUNDS

C-----See if user wants a logarithmic transformation of Z
      CALL TRANSFORM

C-----Determine interpolation grid boundaries (not dataset bounds)
      CALL INTBOUNDS(.FALSE.)
```

```
C-----Determine block size
 18   WRITE(*,'(4X,A,$)')'Enter block width: '
 16   READ(*,*,IOSTAT=IERR) WIDTH
      IF((IERR.NE.0).OR.(WIDTH.LE.0)) THEN
         WRITE(*,650)
         GOTO 16
      ENDIF
C-----Determine no of blocks across and up interpolation grid
      GXLEN=DABS(GXMAX-GXMIN)
      GYLEN=DABS(GYMAX-GYMIN)
      NXBL=NINT((GXLEN/WIDTH)+0.49)
      NYBL=NINT((GYLEN/WIDTH)+0.49)
      IF(TWODIM) THEN
         WRITE(*,424) NXBL,NYBL
      ELSE
         WRITE(*,425) NXBL
      ENDIF
 424  FORMAT(4X,'Kriged area will be',I3,' blocks wide (E-W) x',
     +   I3,' long (N-S);')
 425  FORMAT(4X,'Kriged area will be',I3,' blocks long;')
      WRITE(*,'(4X,A,$)') 'Respecify width? (y/n): '
  19  READ(*,'(A1)') YORN
      IF ((YORN.EQ.'Y').OR.(YORN.EQ.'y')) THEN
         GOTO 18
      ELSEIF ((YORN.NE.'N').AND.(YORN.NE.'n')) THEN
         WRITE(*,655)
         GOTO 19
      ENDIF
      WRITE(*,*)

C-----Get the Semivariogram model
      CALL DEFBMODEL

      RETURN

C-----Error trapping code
 650  FORMAT(4X,'numeric input only: ',$)
 655  FORMAT(4X,'entry error; try again: ',$)
      END

C*************************************************************************
C*************************************************************************

      SUBROUTINE BOUNDS

C-----Gets bounds for data set; see INTBOUNDS for interpolation bounds

      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER      XCOL,YCOL,ZCOL
      CHARACTER*1  YORN
      CHARACTER*64 ZLABEL
      LOGICAL      TWODIM,CHECK,BOUND,HERC,LOG,HAAN

      COMMON/DATACOLS/X(1000),Y(1000),Z(1000)
      COMMON/GENPARAM/NR,NRGRID,TWODIM,HERC,CHECK,HAAN,LOG,BOUND,
     +                OFF,XCOL,YCOL,ZCOL,ZLABEL,TZMN,TZVAR,UZMN,UZVAR,
     +                BZMN,BZVAR,XMIN,XMAX,YMIN,YMAX,GXMIN,GXMAX,
     +                GYMIN,GYMAX,NRTOT

C-----Determine Xmin,XMAX,Ymin,Ymax (if onedim, all y(i)'s = 0)
      XMIN=X(1)
      XMAX=X(1)
      YMIN=Y(1)
      YMAX=Y(1)
      IF(.NOT.TWODIM) THEN
         DO 10 I=2,NRTOT
              XMIN=DMIN1(XMIN,X(I))
              XMAX=DMAX1(XMAX,X(I))
 10      CONTINUE
      ELSE
         DO 20 I=2,NRTOT
              XMIN=DMIN1(XMIN,X(I))
              XMAX=DMAX1(XMAX,X(I))
              YMIN=DMIN1(YMIN,Y(I))
              YMAX=DMAX1(YMAX,Y(I))
 20      CONTINUE
```

```
      ENDIF
 49   TXMIN=XMIN
      TXMAX=XMAX
      TYMIN=YMIN
      TYMAX=YMAX
      NRTMP=NRTOT
      BOUND=.FALSE.

C-----Write bounds to screen
 50   IF(TWODIM) THEN
         WRITE(*,*)'X-Y bounds for dataset:'
         WRITE(*,'(4X,A3,F11.3,A3,F11.3)')'X: ',TXMIN,' , ',TXMAX
         WRITE(*,'(4X,A3,F11.3,A3,F11.3)')'Y: ',TYMIN,' , ',TYMAX
      ELSE
         WRITE(*,*)'Bounds for dataset:'
         WRITE(*,'(4X,A3,F11.3,A3,F11.3)')'X: ',TXMIN,' , ',TXMAX
      ENDIF
      WRITE(*,250) NRTMP
 250  FORMAT (4X,'Number of valid points: ',I4)

C-----See if user wants subset
      WRITE(*,'(4X,A,$)') 'Respecify bounds? (y/n): '
 45   READ(*,'(A1)') YORN
 46   IF ((YORN.EQ.'Y').OR.(YORN.EQ.'y')) THEN
         WRITE(*,220)
 220     FORMAT(7X,'Enter min X and max X (2 values): ',$)
  16     READ(*,*,IOSTAT=IERR)  TXMIN, TXMAX
         IF((IERR.NE.0).OR.(TXMIN.GE.TXMAX)) THEN
            WRITE(*,650)
            GOTO 16
         ENDIF
         IF(TXMIN.LT.XMIN) TXMIN=XMIN
         IF(TXMAX.GT.XMAX) TXMAX=XMAX

         IF (TWODIM) THEN
            WRITE(*,230)
 230        FORMAT(7X,'Enter min Y and max Y (2 values): ',$)
  17        READ(*,*,IOSTAT=IERR) TYMIN,TYMAX
            IF((IERR.NE.0).OR.(TYMIN.GE.TYMAX)) THEN
               WRITE(*,650)
               GOTO 17
            ENDIF
            IF(TYMIN.LT.YMIN) TYMIN=YMIN
            IF(TYMAX.GT.YMAX) TYMAX=YMAX
          ENDIF

C-----See if >3 points fall within specified range
C     NB if onedim all Y(L)'s = 0, as does TYMIN, TYMAX
         NRTMP=0
         DO 15 L=1,NRTOT
            IF ((X(L).GE.TXMIN).AND.(X(L).LE.TXMAX).AND.
     +          (Y(L).GE.TYMIN).AND.(Y(L).LE.TYMAX)) NRTMP=NRTMP+1
 15      CONTINUE
         IF(NRTMP.LE.2) THEN
            WRITE(*,150) NRTMP,CHAR(7),CHAR(7)
 150        FORMAT(14X,'...too few data (',I2,') within range.....',
     +                A,A)
            WRITE(*,*)
            GOTO 49
         ENDIF
         GOTO 50

      ELSEIF ((YORN.EQ.'N').OR.(YORN.EQ.'n')) THEN
C-----Fill X,Y,Z vectors if boundaries changed
         IF((XMIN.NE.TXMIN).OR.(XMAX.NE.TXMAX).OR.(YMIN.NE.TYMIN)
     +      .OR.(YMAX.NE.TYMAX)) THEN
            XMIN=TXMIN
            XMAX=TXMAX
            YMIN=TYMIN
            YMAX=TYMAX
            K=0
            DO 35 L=1,NRTOT
               IF ((X(L).GE.XMIN).AND.(X(L).LE.XMAX).AND.
     +            (Y(L).GE.YMIN).AND.(Y(L).LE.YMAX)) THEN
                  K=K+1
                  X(K)=X(L)
                  Y(K)=Y(L)
                  Z(K)=Z(L)
               ENDIF
 35         CONTINUE
            BOUND=.TRUE.
            NRTMP=K
         ENDIF
         NR=NRTMP

      ELSE
         WRITE(*,*) '   answer either y or n......'
         WRITE(*,'(4X,A,$)') 'try again: '
         GOTO 45
      ENDIF

      WRITE(*,*)
 999  RETURN

C-----Error trapping code
 650  FORMAT(7X,'entry error......',/14X,'try again: ',$)

      END

C***********************************************************************
C***********************************************************************

      SUBROUTINE TRANSFORM

      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER      XCOL,YCOL,ZCOL
      CHARACTER*1  YORN
      CHARACTER*64 ZLABEL
      LOGICAL      TWODIM,CHECK,BOUND,HERC,LOG,HAAN

      COMMON/DATACOLS/X(1000),Y(1000),Z(1000)
      COMMON/GENPARAM/NR,NRGRID,TWODIM,HERC,CHECK,HAAN,LOG,BOUND,
     +                OFF,XCOL,YCOL,ZCOL,ZLABEL,TZMN,TZVAR,UZMN,UZVAR,
     +                BZMN,BZVAR,XMIN,XMAX,YMIN,YMAX,GXMIN,GXMAX,
     +                GYMIN,GYMAX,NRTOT

  20  WRITE(*,105)
 105  FORMAT(' Transform Z to logn(z)? (y/n): ',$)
  31  READ(*,'(A1)') YORN
      IF ((YORN.EQ.'N').OR.(YORN.EQ.'n')) THEN
         LOG=.FALSE.
         HAAN=.FALSE.
      ELSEIF ((YORN.EQ.'Y').OR.(YORN.EQ.'y')) THEN
         LOG=.TRUE.
         WRITE(*,106)
 106     FORMAT('    Specify a in the expression Z=ln(z+a): ',$)
  22     READ(*,*,IOSTAT=IERR) OFF
         IF(IERR.NE.0) THEN
            WRITE(*,650)
            GOTO 22
         ENDIF

         IF((OFF.LT.0).OR.(OFF.GT.9999)) THEN
            WRITE(*,650)
            GOTO 22
         ENDIF

C-----Count points out of range in SPECIFIED data grid (not entire)
         NOUTS=0
         DO 15 I=1,NR
            IF(Z(I)+OFF.LT.1.0) NOUTS=NOUTS+1
 15      CONTINUE
         IF(NOUTS.GT.0) THEN
            IF(NR-NOUTS.LT.3) THEN
               WRITE(*,150) NR-NOUTS
 150           FORMAT(4X,'too few data (',I2,
     +                   ') are >1 before log-n transformation......')
               WRITE(*,'(4X,A,A,$)') ' continue with ',
     +                        'transformation? (y/n): '
               GOTO 31
            ELSE
               WRITE(*,100) NOUTS
 100           FORMAT(4X,'N.BG. ',I4,
     +                   ' Z-values are out of range (<1.0) and')
```

```
               WRITE(*,101) NR-NOUTS
  101          FORMAT(4X,'  will be ignored; there remain',I4,
     +                        ' values;')
  16           WRITE(*,102)
  102          FORMAT(4X,'  is this ok? (y/n): ',$)
               READ(*,'(A1)') YORN
               IF((YORN.NE.'Y').AND.(YORN.NE.'y')) GOTO 20
               WRITE(*,*)
            ENDIF
         ENDIF

         WRITE(*,107)
  107    FORMAT('    Specify if want [H]aan or [N]o '
     +       'backtranformation: ',$)
   21    READ(*,'(A1)') YORN
         IF ((YORN.EQ.'H').OR.(YORN.EQ.'h')) THEN
            HAAN=.TRUE.
         ELSEIF ((YORN.EQ.'N').OR.(YORN.EQ.'n')) THEN
            HAAN=.FALSE.
         ELSE
            WRITE(*,'(7X,A24,$)') 'answer H or N only......'
            GOTO 21
         ENDIF

C-----Transform data in work arrays
         K=0
         DO 18 I=1,NR
            IF(Z(I)+OFF.LT.1.0) GOTO 18
            K=K+1
            X(K)=X(I)
            Y(K)=Y(I)
            Z(K)=DLOG(Z(I)+OFF)
   18    CONTINUE
         NR=K
      ELSE
         WRITE(*,'(4X,A,$)') '   answer y or n only: '
         GOTO 31
      ENDIF
      WRITE(*,*)
      RETURN

C-----Error trapping code
 650  FORMAT(7X,'value out of range......',/14X,'try again: ',$)

      END

C***********************************************************************
C***********************************************************************

      SUBROUTINE INTBOUNDS(EARLY)

      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER      XCOL,YCOL,ZCOL
      CHARACTER*1  YORN
      CHARACTER*64 ZLABEL
      LOGICAL      TWODIM,CHECK,BOUND,HERC,LOG,HAAN,EARLY

      COMMON/DATACOLS/X(1000),Y(1000),Z(1000)
      COMMON/GENPARAM/NR,NRGRID,TWODIM,HERC,CHECK,HAAN,LOG,BOUND,
     +                OFF,XCOL,YCOL,ZCOL,ZLABEL,TZMN,TZVAR,UZMN,UZVAR,
     +                BZMN,BZVAR,XMIN,XMAX,YMIN,YMAX,GXMIN,GXMAX,
     +                GYMIN,GYMAX,NRTOT

C-----Set grid boundaries to dataset bounds
      GXMIN=XMIN
      GXMAX=XMAX
      GYMIN=YMIN
      GYMAX=YMAX

C-----If early quit after setting gxmax=xmax, etc.
      IF(EARLY) THEN
         NRGRID=0
         RETURN
      ENDIF

C-----Write bounds to screen
```

```
 55   GXLEN=ABS(GXMAX-GXMIN)
      GYLEN=ABS(GYMAX-GYMIN)
      WRITE(*,*)'Interpolation Bounds:  --Min--       --Max--',
     +          '    --Length--'
      IF(TWODIM) THEN
         WRITE(*,'(17X,3(A3,F11.3))')'X: ',GXMIN,' , ',GXMAX,
     +          '  ',GXLEN
         WRITE(*,'(17X,3(A3,F11.3))')'Y: ',GYMIN,' , ',GYMAX,
     +          '  ',GYLEN
      ELSE
         WRITE(*,'(17X,3(A3,F11.3))')'X: ',GXMIN,' , ',GXMAX,
     +          '  ',GXLEN
      ENDIF

C-----See how many measured points are within interpolation grid
      NRGRID=0
      DO 15 I=1,NR
         IF(TWODIM) THEN
            IF((Y(I).GE.GYMIN).AND.(Y(I).LE.GYMAX).AND.
     +        (X(I).GE.GXMIN).AND.(X(I).LE.GXMAX)) NRGRID=NRGRID+1
         ELSE
            IF((X(I).GE.GXMIN).AND.(X(I).LE.GXMAX)) NRGRID=NRGRID+1
         ENDIF
 15   CONTINUE
      WRITE(*,250) NRGRID
 250  FORMAT (4X,'Number of measured points within bounds: ',I4)

      IF(NRGRID.EQ.0) THEN
         YORN='Y'
         WRITE(*,*) 'Respecify bounds:'
      ELSE
         WRITE(*,'(4X,A,$)') 'Respecify bounds? (y/n): '
         READ(*,'(A1)') YORN
      ENDIF
 45   IF ((YORN.EQ.'Y').OR.(YORN.EQ.'y')) THEN
         WRITE(*,220)
 220     FORMAT(7X,'Enter min X and max X (2 values): ',$)
 16      READ(*,*,IOSTAT=IERR)  GXMIN, GXMAX
         IF(IERR.NE.0) THEN
            WRITE(*,605)
            GOTO 16
         ENDIF
         IF (TWODIM) THEN
            WRITE(*,230)
 230        FORMAT(7X,'Enter min Y and max Y (2 values): ',$)
 17         READ(*,*,IOSTAT=IERR) GYMIN,GYMAX
            IF((IERR.NE.0).OR.(GYMIN.GE.GYMAX)) THEN
               WRITE(*,605)
               GOTO 17
            ENDIF
         ENDIF
         GOTO 55

      ELSEIF ((YORN.NE.'N').AND.(YORN.NE.'n')) THEN
         WRITE(*,*) '      answer either y or n......'
         WRITE(*,'(7X,A,$)') 'try again: '
         READ(*,'(A1)') YORN
         GOTO 45
      ENDIF

      WRITE(*,*)
 999  RETURN

C-----Error trapping code
 605  FORMAT(7X,'value(s) out of range......',/14X,'try again: ',$)

      END

C***********************************************************************
C***********************************************************************

      SUBROUTINE DEFBMODEL

      IMPLICIT REAL*8 (A-H,O-Z)
      CHARACTER*64 HRDFIL
      LOGICAL      HRDGRD,JACK,SELF
```

```
       COMMON/BLKPARAM/RNBHD,RNBMAX,MAXNBR,MODEL,C0,C9,A9,
      +                WIDTH,NXBL,NYBL,BLVAR

       WRITE(*,400)
  400  FORMAT(1X,'Specify semivariogram model;',/1x,
      +    '   Models available:',/1x,
      +    '     1 - Linear to range',/1X,
      +    '         for H(i) < A, ',
      +               'Gamma(i) = C0+[(C-C0)/A]*H(i)',/1X,
      +    '                  > A, ',
      +               'Gamma(i) = C',/1X,
      +    '     2 - Spherical',/1X,
      +    '         for H < A, ',
      +               'Gamma(i) = C0+(C-C0)*[1.5*(H(i)/A)',
      +               '-(0.5*(H(i)/A)**3)]',/1X,
      +    '               > A, ',
      +               'Gamma(i) = C',/1X,
      +    '     3 - Exponential',/1X,
      +    '         Gamma(i) = C0+(C-C0)*[1-EXP(-H(i)/Ao)]',/1X,
      +    '     4 - Gaussian',/1X,
      +    '         Gamma(i) = C0-(C-C0)*',
      +              '[1-EXP(-H(i)**2/(Ao/3**.5)**2)]')
      WRITE(*,'(1X,A17,$)') '   Enter Choice: '
   28 READ(*,*,IOSTAT=IERR) MODEL
      IF(IERR.NE.0) THEN
         WRITE(*,605)
         GOTO 28
      ENDIF

      C9=0
      C0=0
      A9=0
      IF ((MODEL.GE.1).OR.(MODEL.LE.4)) THEN
         WRITE(*,'(4X,A,$)')     'Enter C0 (nugget variance): '
   34    READ(*,*,IOSTAT=IERR) C0
         IF(IERR.NE.0) THEN
            WRITE(*,605)
            GOTO 34
         ENDIF
         WRITE(*,'(4X,A,$)')     '                  C (sill): '
   36    READ(*,*,IOSTAT=IERR) C9
         IF(IERR.NE.0) THEN
            WRITE(*,605)
            GOTO 36
         ENDIF
         IF (MODEL.LT.3) THEN
            WRITE(*,'(4X,A,$)')     '                A (range): '
         ELSE
            WRITE(*,'(4X,A,$)')     '                       Ao: '
         ENDIF
   38    READ(*,*,IOSTAT=IERR) A9
         IF(IERR.NE.0) THEN
            WRITE(*,605)
            GOTO 38
         ENDIF

      ELSE
         WRITE(*,605)
         GOTO 28
      ENDIF

      WRITE(*,*)
      RETURN

C-----Error trapping code
  605 FORMAT(7X,'value out of range......',/14X,'try again: ',$)

      END

C***********************************************************************
C***********************************************************************

      SUBROUTINE DEFBNEIGH(EARLY)

      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER XCOL,YCOL,ZCOL
      CHARACTER*64 HRDFIL,ZLABEL
      LOGICAL      TWODIM,CHECK,BOUND,HERC,LOG,HAAN,EARLY

      COMMON/GENPARAM/NR,NRGRID,TWODIM,HERC,CHECK,HAAN,LOG,BOUND,
     +                OFF,XCOL,YCOL,ZCOL,ZLABEL,TZMN,TZVAR,UZMN,UZVAR,
     +                BZMN,BZVAR,XMIN,XMAX,YMIN,YMAX,GXMIN,GXMAX,
     +                GYMIN,GYMAX,NRTOT

      COMMON/BLKPARAM/RNBHD,RNBMAX,MAXNBR,MODEL,C0,C9,A9,
     +                WIDTH,NXBL,NYBL,BLVAR

C-----Define maximum neighborhood search radius
      DISTX=XMAX-XMIN
      DISTY=YMAX-YMIN
      IF(.NOT.TWODIM) THEN
         RNBMAX=DSQRT(DISTX**2)
      ELSE
         RNBMAX=DSQRT(DISTX**2+DISTY**2)
      ENDIF
      WRITE(*,*) 'Neighborhood Definition'
      WRITE(*,210) RNBMAX
 210  FORMAT(1X,'Maximum neighborhood radius is ',G12.3,';')
      RNBHD=RNBMAX
      MAXNBR=MIN(NR-1,100)
      IF(EARLY) GOTO 999

      WRITE(*,560)
  42  READ(*,*,ERR=630) TRNBHD
 560  FORMAT(1X,'   specify maximum search radius: ',$)
      IF(TRNBHD.LE.0) GOTO 630
      IF(TRNBHD.LT.RNBHD) RNBHD=TRNBHD
      WRITE(*,*)

C-----Define max no. of neighbors to include in neighborhood search
      MAXNBR=MIN(NR,100)
      WRITE(*,552) MAXNBR
 552  FORMAT(1X,'Number of potential neighbors for any search: ',
     +      I3)
      WRITE(*,550) MIN(100,MAXNBR)
 550  FORMAT(1X,'   specify maximum (to ',I3,'): ',$)
  32  READ(*,*,ERR=620) MAXNBR
      IF(MAXNBR.LT.1) GOTO 620
      IF(MAXNBR.GT.100) THEN
         WRITE(*,*)'        ......n set to 100;'
         MAXNBR=100
      ENDIF
      IF(MAXNBR.GT.NR) MAXNBR=NR

 999  WRITE(*,*)
      RETURN

C-----Error trapping code
 620  WRITE(*,650)
      GOTO 32
 630  WRITE(*,650)
      GOTO 42
 650  FORMAT(4X,'value out of range......',/4X,'try again: ',$)

      END

C************************************************************************
C************************************************************************

      SUBROUTINE STATS

      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER      XCOL,YCOL,ZCOL
      LOGICAL      TWODIM,CHECK,HERC,LOG,HAAN,BOUND
      CHARACTER*64 ZLABEL

      COMMON/DATACOLS/X(1000),Y(1000),Z(1000)
      COMMON/GENPARAM/NR,NRGRID,TWODIM,HERC,CHECK,HAAN,LOG,BOUND,
     +                OFF,XCOL,YCOL,ZCOL,ZLABEL,TZMN,TZVAR,UZMN,UZVAR,
     +                BZMN,BZVAR,XMIN,XMAX,YMIN,YMAX,GXMIN,GXMAX,
     +                GYMIN,GYMAX,NRTOT

C-----NB: STATS is called AFTER transformations
C     TZMN, TZVAR = transformed general mean and var of z
```

```
C     UZMN, UZVAR = untransformed

C-----Calculate stats
      SUMZ=0
      SUMZ2=0
      DO 10 I=1,NR
         SUMZ=SUMZ+Z(I)
         SUMZ2=SUMZ2+Z(I)**2
 10   CONTINUE

C-----Calculate mean, variance
      T=DBLE(NR)
      TZMN=SUMZ/T
      IF(T.GT.1) TZVAR=(SUMZ2-(T*TZMN**2))/(T-1)

C-----If transformed, get untransformed mean, var
      IF(LOG) THEN
         SUMZ=0
         SUMZ2=0
         DO 20 I=1,NR
            SUMZ=SUMZ+DEXP(Z(I))-OFF
            SUMZ2=SUMZ2+ (DEXP(Z(I))-OFF)**2
 20      CONTINUE
         UZMN=SUMZ/T
         IF(T.GT.1) UZVAR=(SUMZ2-(T*UZMN**2))/(T-1)
         IF(HAAN) THEN
            BZMN=DEXP(TZMN+0.5*TZVAR)-OFF
            BZVAR=(UZMN**2)*(DEXP(TZVAR)-1)
         ELSE
            BZMN=DEXP(TZMN)-OFF
            BZVAR=DEXP(TZVAR)
         ENDIF
      ELSE
         UZMN=TZMN
         UZVAR=TZVAR
      ENDIF


      RETURN
      END

C***********************************************************************
C***********************************************************************

      SUBROUTINE GETOUT(OUT)
C     Gets name of output file, opens

      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER      XCOL,YCOL,ZCOL,OUT
      CHARACTER*1  YORN,HEAD1,HEAD2,HEAD3,HEAD4
      CHARACTER*64 FILNAM,OUTFIL,FTITLE,ZLABEL,LSTOUT
      LOGICAL      TWODIM,THERE,OPEND,CHECK,HERC,LOG,HAAN,BOUND

      COMMON/FILINFO/FILNAM,OUTFIL,LSTOUT,FTITLE,NCOL,NROW,
     +               HEAD1(79),HEAD2(79),HEAD3(79),HEAD4(79)
      COMMON/GENPARAM/NR,NRGRID,TWODIM,HERC,CHECK,HAAN,LOG,BOUND,
     +                OFF,XCOL,YCOL,ZCOL,ZLABEL,TZMN,TZVAR,UZMN,UZVAR,
     +                BZMN,BZVAR,XMIN,XMAX,YMIN,YMAX,GXMIN,GXMAX,
     +                GYMIN,GYMAX,NRTOT

C-----Get output filename
 10   OUT=6
      INQUIRE (OUT,OPENED=OPEND,NAME=LSTOUT)
      WRITE(*,100)
 100  FORMAT(' Output file ([N]one, [A]ppend, or filename): ',$)
      READ(*,'(A64)') OUTFIL
      IF ((OUTFIL.EQ.'N').OR.(OUTFIL.EQ.'n')) THEN
         OUTFIL='NONE'
         CLOSE(OUT)
         OUT=0
      ELSEIF ((OUTFIL.EQ.'A').OR.(OUTFIL.EQ.'a')) THEN
         IF (.NOT.OPEND) THEN
           WRITE(*,*) '   there''s no output file to append to...'
           GOTO 10
         ELSE
           OUTFIL=LSTOUT
           WRITE(OUT,'(////////////)')
         ENDIF
      ELSE
         INQUIRE(FILE=OUTFIL,EXIST=THERE)
         INQUIRE(OUT,OPENED=OPEND)
         IF (THERE) THEN
            WRITE(*,110)
 110        FORMAT(' File exists;'
     +         /3X,'Do you want to write over it? (y/n): ',$)
  12        READ(*,'(A1)') YORN
            IF ((YORN.EQ.'N').OR.(YORN.EQ.'n')) THEN
               GOTO 10
            ELSEIF ((YORN.NE.'Y').AND.(YORN.NE.'y')) THEN
               WRITE(*,115)
 115           FORMAT('    answer either y or n: ',$)
               GOTO 12
            ENDIF
         ENDIF
         IF (OPEND) CLOSE(OUT)
         OPEN(OUT,FILE=OUTFIL,STATUS='NEW',IOSTAT=IERR)
         IF(IERR.NE.0) GOTO 600
      ENDIF

      WRITE(*,*)
 70   RETURN

C-----Error trapping
 600  WRITE(*,*)'                ...file error......',CHAR(7),CHAR(7)
      WRITE(*,*)
      GOTO 10

      END

C***********************************************************************
C***********************************************************************

      SUBROUTINE TEXTBOUT(OUT)

      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER      XCOL,YCOL,ZCOL,OUT
      CHARACTER*1  HEAD1,HEAD2,HEAD3,HEAD4,Q
      CHARACTER*64 FILNAM,OUTFIL,LSTOUT,HRDFIL,FTITLE,ZLABEL
      LOGICAL      TWODIM,CHECK,BOUND,HERC,LOG,HAAN,
     +             HRDGRD,JACK,SELF,NBANAL

      COMMON/FILINFO/FILNAM,OUTFIL,LSTOUT,FTITLE,NCOL,NROW,
     +               HEAD1(79),HEAD2(79),HEAD3(79),HEAD4(79)
      COMMON/GENPARAM/NR,NRGRID,TWODIM,HERC,CHECK,HAAN,LOG,BOUND,
     +                OFF,XCOL,YCOL,ZCOL,ZLABEL,TZMN,TZVAR,UZMN,UZVAR,
     +                BZMN,BZVAR,XMIN,XMAX,YMIN,YMAX,GXMIN,GXMAX,
     +                GYMIN,GYMAX,NRTOT
      COMMON/BLKPARAM/RNBHD,RNBMAX,MAXNBR,MODEL,C0,C9,A9,
     +                WIDTH,NXBL,NYBL,BLVAR

C-----Start writing to output file
      WRITE(OUT,*) 'Block Kriging Output'
      IF(FTITLE.NE.'     ') WRITE(OUT,*) 'Run title: ',FTITLE
      WRITE(OUT,*) 'Input file: ',FILNAM
      WRITE(OUT,204) NCOL,NROW
 204  FORMAT(1X,'(',I2,' columns x ',I4,' rows)')
      IF(.NOT.TWODIM) THEN
         WRITE(OUT,208) XCOL,ZCOL
 208     FORMAT(' X,Z Cols: ',I2,', ',I2)
      ELSE
         WRITE(OUT,210) XCOL,YCOL,ZCOL
 210     FORMAT(' X,Y,Z Cols: ',I2,', ',I2,', ',I2)
      ENDIF
      IF(ZLABEL.NE.'     ') WRITE(OUT,*) 'Z col label: ',ZLABEL
      WRITE(OUT,*)'Defined Boundaries'
      WRITE(OUT,*) '   Data (Measured) Points'
      WRITE(OUT,660) XMIN,XMAX
      IF(TWODIM) WRITE(OUT,665) YMIN,YMAX
      WRITE(OUT,*) '   Interpolated Points'
      WRITE(OUT,660) GXMIN,GXMAX
      IF(TWODIM) WRITE(OUT,665) GYMIN,GYMAX
660   FORMAT(1X,'      X:',G14.3,' -',G14.3)
665   FORMAT(1X,'      Y:',G14.3,' -',G14.3)
```

```
      WRITE(OUT,*) 'Number of Measured Points'
      WRITE(OUT,215) NROW
      WRITE(OUT,216) NRTOT
      WRITE(OUT,217) NR
      IF(.NOT.HRDGRD) WRITE(OUT,218) NRGRID
215   FORMAT(4X,'total in input dataset: ',I4)
216   FORMAT(4X,'total valid in dataset: ',I4)
217   FORMAT(4X,'within dataset bounds: ',I4)
218   FORMAT(4X,'within interpolation bounds: ',I4)

      IF(LOG) THEN
         WRITE(OUT,*) 'Z-values log-n transformed:'
         WRITE(OUT,220) OFF
220      FORMAT('    Ln(x+a) where a =',F6.1)
      ENDIF
      WRITE(OUT,*) 'General Statistics'
      WRITE(OUT,*) '(For all measured points within data boundaries)'
      IF(.NOT.LOG) THEN
         WRITE(OUT,630) UZMN
         WRITE(OUT,635) UZVAR
         WRITE(OUT,640) UZVAR**0.5
         WRITE(OUT,645) NR
630      FORMAT(1X,'   Mean:      ',G14.4)
635      FORMAT(1X,'   Variance: ',G14.4)
640      FORMAT(1X,'   Std. Dev: ',G14.4)
645      FORMAT(1X,'   N:         ',I10)
      ELSE
         IF(HAAN) THEN
            WRITE(OUT,725)
725         FORMAT(14X,'Untransformed',3X,'Transformed',3X,
     +                  'Backtransformed (Haan)')
         ELSE
            WRITE(OUT,726)
726         FORMAT(14X,'Untransformed',3X,'Transformed',3X,
     +                  'Backtransformed (Std)')
         ENDIF
         WRITE(OUT,730) UZMN,TZMN,BZMN
         WRITE(OUT,735) UZVAR,TZVAR,BZVAR
         WRITE(OUT,740) UZVAR**.5, TZVAR**.5, BZVAR**.5
         WRITE(OUT,745) NR,NR,NR
730      FORMAT(1X,'   Mean:      ',G14.4,1X,G14.4,1X,G14.4)
735      FORMAT(1X,'   Variance: ',G14.4,1X,G14.4,1X,G14.4)
740      FORMAT(1X,'   Std. Dev: ',G14.4,1X,G14.4,1X,G14.4)
745      FORMAT(1X,'   N:         ',I10,1X,I10,1X,I10)
      ENDIF
      WRITE(OUT,*)
      WRITE(OUT,'(1X,A16,F10.3)')'Block width: ',WIDTH
      WRITE(OUT,'(1X,A16,F10.3)')'      variance: ',BLVAR
      WRITE(OUT,*)
      WRITE(OUT,*) 'Neighborhood Definition'
      WRITE(OUT,*) '     Parameter          Maximum     Used'
        IF(JACK) THEN
           WRITE(OUT,225) NR-1,MAXNBR
        ELSE
           WRITE(OUT,225) NR,MAXNBR
        ENDIF
225   FORMAT(1X,   '   No. of neighbors:',I5,9X,I5)
      WRITE(OUT,226) RNBMAX,RNBHD
226   FORMAT(1X,   '   Search radius:   ',G11.3,1X,G11.3)
      WRITE(OUT,*)

      WRITE(OUT,*) 'Semivariogram model'
      IF (MODEL.EQ.1) THEN
         WRITE(OUT,*)'Linear to range:'
         WRITE(OUT,*)'  for H(i) < A, Gamma(i)= C0+[(C-C0)/A]*H(i)'
         WRITE(OUT,*)'           > A, Gamma(i)= C'
      ELSEIF(MODEL.EQ.2) THEN
         WRITE(OUT,*)'Spherical:'
         WRITE(OUT,*)'  for H(i) < A, Gamma(i)= C0+(C-C0)*',
     +                 '[1.5*(H(i)/A]'
         WRITE(OUT,*)'           > A, Gamma(i)= C'
      ELSEIF(MODEL.EQ.3) THEN
         WRITE(OUT,*)'Exponential'
         WRITE(OUT,*)'  Gamma(i)= C0+(C-C0)*[1-EXP(-H(i)/Ao)]'
      ELSEIF(MODEL.EQ.4) THEN
         WRITE(OUT,*)'Gaussian'
         WRITE(OUT,*)'  Gamma(i) =',
     +   ' C0-(C-C0)*[1-EXP(-H(i)**2/(A/3**.5)**2)]'
      ENDIF
      WRITE(OUT,*)'                 where C0 = ',SNGL(C0)
      WRITE(OUT,*)'                        C = ',SNGL(C9)
      IF(MODEL.LT.3) THEN
         WRITE(OUT,*)'                      A = ',SNGL(A9)
      ELSE
         WRITE(OUT,*)'                     Ao = ',SNGL(A9)
      ENDIF
      WRITE(OUT,*)

      WRITE(OUT,*) 'Interpolation Output'
      IF(LOG) THEN
         IF(HAAN) THEN
            WRITE(OUT,*) 'N.B. Krige estimates backtransformed ',
     +                   '(Haan 1977)'
         ELSE
            WRITE(OUT,*) 'N.B. Krige estimates not backtransformed'
         ENDIF
      ENDIF
      RETURN
      END

C***************************************************************************
C***************************************************************************

      SUBROUTINE BLKVAR

      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION COV(2,2)

      COMMON/BLKPARAM/RNBHD,RNBMAX,MAXNBR,MODEL,C0,C9,A9,
     +                WIDTH,NXBL,NYBL,BLVAR

C-----Calculate the block variance
      VOC2=0.0
      WD2=WIDTH/2.
      WD4=WIDTH/4.
      WD8=WIDTH/8.
      DO 60 I=1,4
         DO 60 J=1,4
            COV(1,1)=WD2+(I-3)*WD4+WD8
            COV(2,1)=WD2+(J-3)*WD4+WD8
            COV(1,2)=WD2
            COV(2,2)=WD2
            CALL BLCOV(COV,VOC)
            VOC2=VOC2+VOC
 60   CONTINUE
      BLVAR=VOC2/16.

      END

C***************************************************************************
C***************************************************************************

      SUBROUTINE BLCOV(B,VOC)

C-----Calculates the block covariance, i.e. covar between a sample and
C     a block

      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION B(2,2)

      COMMON/BLKPARAM/RNBHD,RNBMAX,MAXNBR,MODEL,C0,C9,A9,
     +                WIDTH,NXBL,NYBL,BLVAR

      VOC=0.0
      WD8=WIDTH/8.
      WD4=WIDTH/4.
      DO 10 I=1,4
         DO 10 J=1,4
            DISTX=B(1,2)-B(1,1)+(I-3)*WD4+WD8
            DISTY=B(2,2)-B(2,1)+(J-3)*WD4+WD8
            DISTX=DSQRT(DISTX**2)
            DISTY=DSQRT(DISTY**2)
            H=DSQRT(DISTX**2+DISTY**2)
            VOC=VOC+GAMMAB(H)
 10   CONTINUE
      VOC=VOC/16.
      RETURN
      END

C*************************************************************************
C*************************************************************************

      SUBROUTINE GETBNBRS(XPT,YPT,NNBRS,NBRLOC)

      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4        DIST(1000),DNEAR(100)
      INTEGER       XCOL,YCOL,ZCOL,NBRLOC(100)
      LOGICAL       TWODIM,LOG,HERC,CHECK,HAAN,HRDGRD,BOUND,JACK,SELF
      CHARACTER*64 HRDFIL,ZLABEL

      COMMON/DATACOLS/X(1000),Y(1000),Z(1000)
      COMMON/GENPARAM/NR,NRGRID,TWODIM,HERC,CHECK,HAAN,LOG,BOUND,
     +                OFF,XCOL,YCOL,ZCOL,ZLABEL,TZMN,TZVAR,UZMN,UZVAR,
     +                BZMN,BZVAR,XMIN,XMAX,YMIN,YMAX,GXMIN,GXMAX,
     +                GYMIN,GYMAX,NRTOT
      COMMON/BLKPARAM/RNBHD,RNBMAX,MAXNBR,MODEL,C0,C9,A9,
     +                WIDTH,NXBL,NYBL,BLVAR

C-----Find 100 or MAXNBR of nearest neighbors; reject distances > RNBHD;
C     if Jack=.true. XPT,YPT itself ignored (dist.le.1d-7)

C-----Create array DIST of distance of every element from block center
      DO 110 I=1,NR
         DISTX=(XPT-X(I))**2
         DISTY=(YPT-Y(I))**2
         DIST(I)=SNGL((DISTX+DISTY)**0.5)
 110  CONTINUE

C-----Create array NBRLOC(MAXNBRS) of addresses of neighborhood points
C       array DNEAR holds distance for each address; jump out at maxnbrs
      NNBRS=0
      DO 130 I=1,NR
         IF((DIST(I).GT.RNBHD).OR.(JACK.AND.(DIST(I).LE.1D-7))) GOTO 130
            NNBRS=NNBRS+1
            NBRLOC(NNBRS)=I
            DNEAR(NNBRS)=DIST(I)
            IF(NNBRS.EQ.MAXNBR) GOTO 140
 130  CONTINUE

C-----Do an initial sort on DNEAR
 140  CALL SORT(DNEAR,NBRLOC,NNBRS)

C-----If no. of neighbors within neighborhood so far =maxneighbors, then
C     may be more neighbors out there within neighborhood. Now go
C     through remainder of data set to see which of remaining items
C     to include in DNEAR;
C     Since have already found the max no. of nearest neighbors, even
C     if future points are within the neighborhood radius they must be
C     less than the greatest radius thus far encountered to be included
C     as a neighbor; therefore no need to test whether that point is in
C     the neighborhood radius.  If there are fewer points within the
C     neighborhood radius than the maxnbr of neighbors, there's no need
C     to sort further.

      IF(NNBRS.EQ.MAXNBR) THEN
         DO 150 I=NNBRS+1,NR
            IF(DIST(I).GT.DNEAR(NNBRS)) GOTO 150
            IF(JACK.AND.(DIST(I).LE.1D-7)) GOTO 150
            NBRLOC(NNBRS)=I
            DNEAR(NNBRS)=DIST(I)
            CALL SORT(DNEAR,NBRLOC,NNBRS)
 150     CONTINUE
      ENDIF

      RETURN
      END

C*************************************************************************
C*************************************************************************

      SUBROUTINE SORT(DATA1,DATA2,N)
```

```
C-----Generalized sort routine; sorts on Data1 for N elements; ascending
C       must specify real or integer

      REAL*4 DATA1(1),SAVE1
      INTEGER DATA2(1),SAVE2

      IF(N.GE.2) THEN
         DO 20 I=2,N
           DO 10 J=1,I-1
             IF(DATA1(I).GE.DATA1(J)) GOTO 10
             SAVE1=DATA1(I)
             DATA1(I)=DATA1(J)
             DATA1(J)=SAVE1
             SAVE2=DATA2(I)
             DATA2(I)=DATA2(J)
             DATA2(J)=SAVE2
 10        CONTINUE
 20      CONTINUE
      ENDIF

      RETURN
      END

C***********************************************************************
C***********************************************************************

      DOUBLE PRECISION FUNCTION GAMMAB(H)

      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/BLKPARAM/RNBHD,RNBMAX,MAXNBR,MODEL,C0,C9,A9,
     +                WIDTH,NXBL,NYBL,BLVAR

C-----Determine the value GAMMAB
C     NB: (1) For block.for there is not TZVAR;
C         (2) Although if h.ge.A9 below evaluates to GAMMAB=C9,
C             float point processing mandates using full equation
C             to avoid crash

      IF(MODEL.EQ.1) THEN
         IF (H.LT.A9) THEN
            GAMMAB=C0+((C9-C0)/A9)*H
         ELSE
            GAMMAB=C0+((C9-C0)/A9)*A9
         ENDIF

      ELSEIF(MODEL.EQ.2) THEN
         IF(H.LT.A9) THEN
            GAMMAB=C0+(C9-C0)*(1.5*(H/A9)-0.5*((H/A9)**3))
         ELSE
            GAMMAB=C0+(C9-C0)*(1.5*(A9/A9)-0.5*((A9/A9)**3))
         ENDIF

      ELSEIF(MODEL.EQ.3) THEN
         GAMMAB=C0+(C9-C0)*(1-DEXP(-H/A9))

      ELSEIF(MODEL.EQ.4) THEN
         GAMMAB=C0-(C9-C0)*(1-DEXP(-(H**2/(A9/3**.5)**2)))

      ENDIF

 999  RETURN
      END

C***********************************************************************
C***********************************************************************

      SUBROUTINE SIMULB(SOLNV,VMAT,NROWS,NCOLS,ISERR)

C-----Generic simultaneous equation subroutine; matrix (in vector
C        form) VMAT with NROWS rows and NCOLS columns


      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION VMAT(10201),SOLNV(101)
      IF(NROWS.LE.0) GOTO 23

C-----Find greatest element in matrix VMAT
      ISERR=0.
      PIVOT=0.
      NRSQ=NROWS*NROWS
      NM=NCOLS*NROWS
      DO 30 L=1,NRSQ
        UX=DABS(VMAT(L))
        IF((UX-PIVOT).GT.0) THEN
          PIVOT=UX
          I=L
        ENDIF
   30 CONTINUE
      TOL=PIVOT*1D-14

C-----VMAT(I) is pivot element
C     Start elimination
      LST=1
      DO 17 K=1,NROWS

C--------Test for singularity
         IF(PIVOT.LE.0) GOTO 23
         IF(ISERR.EQ.0) THEN
            IF((PIVOT-TOL).LE.0) ISERR=K-1
         ENDIF
         PIVTMP=1./VMAT(I)
         J=(I-1)/NROWS
         I=I-J*NROWS-K
         J=J+1-K

C--------I+K=row index, J+K column index of pivot
         DO 8 L=K,NM,NROWS
           LL=L+I
           UX=PIVTMP*SOLNV(LL)
           SOLNV(LL)=SOLNV(L)
    8    SOLNV(L)=UX

         IF((K-NROWS).GE.0) GOTO 18
C--------Column interchange
         LEND=LST+NROWS-K
         IF(J.LE.0) GOTO 12
         II=J*NROWS
         DO 11 L=LST,LEND
           UX=VMAT(L)
           LL=L+II
           VMAT(L)=VMAT(LL)
   11    VMAT(LL)=UX

C--------Row interchange
   12    DO 13 L=LST,NRSQ,NROWS
           LL=L+I
           UX=PIVTMP*VMAT(LL)
           VMAT(LL)=VMAT(L)
   13    VMAT(L)=UX

         VMAT(LST)=J

C--------Element reduction
         PIVOT=0.
         LST=LST+1
         J=0
         DO 16 II=LST,LEND
           PIVTMP=-VMAT(II)
           IST=II+NROWS
           J=J+1
           DO 15 L=IST,NRSQ,NROWS
             LL=L-J
             VMAT(L)=VMAT(L)+PIVTMP*VMAT(LL)
             UX=DABS(VMAT(L))
             IF((UX-PIVOT).LE.0) GOTO 15
             PIVOT=UX
             I=L
   15      CONTINUE
           DO 16 L=K,NM,NROWS
           LL=L+J
   16    SOLNV(LL)=SOLNV(LL)+PIVTMP*SOLNV(L)
   17 LST=LST+NROWS

 C-----End of elimination loop
 C     Back substitution and interchange
   18 I=NROWS-1
      IF(I.LT.0) THEN
         GOTO 23
      ELSEIF(I.EQ.0) THEN
         GOTO 999
      ENDIF
      IST=NRSQ+NROWS
      LST=NROWS+1
      DO 21 I=2,NROWS
         II=LST-I
         IST=IST-LST
         L=IST-NROWS
         L=VMAT(L)+.5
         DO 21 J=II,NM,NROWS
           UX=SOLNV(J)
           LL=J
           DO 20 K=IST,NRSQ,NROWS
             LL=LL+1
  20       UX=UX-VMAT(K)*SOLNV(LL)
           K=J+L
           SOLNV(J)=SOLNV(K)
  21  SOLNV(K)=UX
 999  RETURN

C-----ERROR RETURN
  23  ISERR=-1
      RETURN
      END

C***************************************************************************
C***************************************************************************
C***********************                    *******************************
C*********************** End of Block.for   *******************************
C***********************                    *******************************
C***************************************************************************
C***************************************************************************
```

Block Kriging output file Block.out

```
 Block Kriging Output
 Run title: Sample run: Block krig
 Input file: Sample.prn
 ( 3 columns x  256 rows)
 X,Y,Z Cols:  1,  2,  3
 Z col label: Zvariate
 Defined Boundaries
    Data (Measured) Points
       X:      .200     -      79.9
       Y:      .000     -      79.4
    Interpolated Points
       X:      .200     -      79.9
       Y:      .000     -      79.4
 Number of Measured Points
    total in input dataset:  256
    total valid in dataset:  248
    within dataset bounds:  248
    within interpolation bounds:  248
 Z-values log-n transformed:
    Ln(x+a) where a =   1.0
 General Statistics
 (For all measured points within data boundaries)
                Untransformed   Transformed   Backtransformed (Haan)
    Mean:          .3977           .3242            .3972
    Variance:      .4575E-01       .2054E-01        .3284E-02
    Std. Dev:      .2139           .1433            .5730E-01
    N:               248         248          248

    Block width:       9.000
       variance:        .013

 Neighborhood Definition
    Parameter          Maximum     Used
    No. of neighbors:  248           32
    Search radius:      113.        113.

 Semivariogram model
 Spherical:
   for H(i) < A, Gamma(i)= C0+(C-C0)*[1.5*(H(i)/A]
             > A, Gamma(i)= C
                  where C0 =    1.223000E-002
                         C =    2.388000E-002
                         A =          67.4000000

 Interpolation Output
 N.B. Krige estimates backtransformed (Haan 1977)
 NB: Analysis includes only those measured points within interpolation bounds
     X          Y              Est Z          Std Dev          Neighbors
  4.050      3.700            .46586          .45497E-02       32
  4.050      12.70            .45981          .51034E-02       32
  4.050      21.70            .56957          .39460E-02       32
  4.050      30.70            .56284          .42379E-02       32
  4.050      39.70            .48219          .42634E-02       32
  4.050      48.70            .42794          .46395E-02       32
  4.050      57.70            .59150          .41908E-02       32
  4.050      66.70            .42529          .41011E-02       32
  4.050      75.70            .42511          .57216E-02       32
  13.05      3.700            .36360          .43491E-02       32
  13.05      12.70            .39530          .36808E-02       32
  13.05      21.70            .42609          .41374E-02       32
  13.05      30.70            .49415          .44140E-02       32
  13.05      39.70            .48394          .51906E-02       32
  13.05      48.70            .38028          .35475E-02       32
  13.05      57.70            .42653          .45741E-02       32
  13.05      66.70            .36980          .40318E-02       32
  13.05      75.70            .53382          .42825E-02       32
  22.05      3.700            .26696          .55903E-02       32
  22.05      12.70            .24184          .51751E-02       32
  22.05      21.70            .25311          .40353E-02       32
  22.05      30.70            .20363          .43857E-02       32
  22.05      39.70            .26198          .38291E-02       32
  22.05      48.70            .40016          .53946E-02       32
  22.05      57.70            .52241          .43833E-02       32
  22.05      66.70            .34981          .41509E-02       32
  22.05      75.70            .38165          .42016E-02       32
```

31.05	3.700	.20668	.39233E-02	32
31.05	12.70	.15555	.45398E-02	32
31.05	21.70	.15535	.38454E-02	32
31.05	30.70	.22233	.46591E-02	32
31.05	39.70	.22904	.53551E-02	32
31.05	48.70	.29200	.42685E-02	32
31.05	57.70	.23814	.37448E-02	32
31.05	66.70	.43219	.44103E-02	32
31.05	75.70	.20059	.39754E-02	32
40.05	3.700	.21486	.46902E-02	32
40.05	12.70	.20863	.41117E-02	32
40.05	21.70	.25605	.49221E-02	32
40.05	30.70	.25574	.44884E-02	32
40.05	39.70	.29365	.39438E-02	32
40.05	48.70	.33950	.40117E-02	32
40.05	57.70	.35914	.51949E-02	32
40.05	66.70	.38169	.44430E-02	32
40.05	75.70	.30198	.50451E-02	32
49.05	3.700	.22574	.50342E-02	32
49.05	12.70	.29065	.54955E-02	32
49.05	21.70	.42234	.42988E-02	32
49.05	30.70	.31701	.39542E-02	32
49.05	39.70	.30163	.35378E-02	32
49.05	48.70	.32119	.49528E-02	32
49.05	57.70	.39267	.41866E-02	32
49.05	66.70	.47397	.48287E-02	32
49.05	75.70	.43438	.40315E-02	32
58.05	3.700	.19644	.43980E-02	32
58.05	12.70	.21018	.39721E-02	32
58.05	21.70	.19843	.51475E-02	32
58.05	30.70	.34206	.32641E-02	32
58.05	39.70	.31223	.48271E-02	32
58.05	48.70	.24406	.50115E-02	32
58.05	57.70	.48925	.40151E-02	32
58.05	66.70	.46939	.45747E-02	32
58.05	75.70	.48811	.48518E-02	32
67.05	3.700	.32848	.55666E-02	32
67.05	12.70	.32230	.38613E-02	32
67.05	21.70	.29881	.37960E-02	32
67.05	30.70	.73168	.39564E-02	32
67.05	39.70	.66855	.56101E-02	32
67.05	48.70	.50951	.37756E-02	32
67.05	57.70	.49633	.37947E-02	32
67.05	66.70	.70230	.39821E-02	32
67.05	75.70	.76921	.51138E-02	32
76.05	3.700	.41477	.56992E-02	32
76.05	12.70	.42373	.54190E-02	32
76.05	21.70	.44370	.47367E-02	32
76.05	30.70	.51424	.50012E-02	32
76.05	39.70	.47327	.46386E-02	32
76.05	48.70	.50996	.44133E-02	32
76.05	57.70	.61693	.59607E-02	32
76.05	66.70	.73286	.39922E-02	32
76.05	75.70	.62508	.57781E-02	32

Sample input file Sample.prn

```
File Sample.prn
Sample data for 2 dimensional geostat analysis
3 columns: Xcor, Ycor, Zvariate
Semivar model (r2=.95): spherical, C0=.01227, C=.02388, Ao=67.74
     1.7       1.3     0.61
     2.3       7.0     0.37
     4.5      11.9     0.42
     3.1      19.5     0.45
     3.4      23.4     0.92
     2.7      29.4     0.60
     1.6      32.6     0.60
     0.2      40.0     0.45
     4.1      44.5     0.43
     0.8      48.7     0.38
     3.2      52.8     0.57
     2.2      58.1     0.94
     0.6      64.0     0.51
     4.2      69.6     0.39
     2.4      71.8     0.34
     4.9      76.9     0.41
     7.8       3.5     0.37
     9.4       8.4     0.47
     6.7      10.2     0.61
     6.7      16.3     0.46
     6.8      20.7     0.39
     6.5      27.2     0.44
     9.8      33.1     0.49
     5.8      39.6     0.54
     5.7      42.7     0.38
     9.5      46.1     0.28
     9.7      51.1     0.47
     6.2      57.5     0.39
     9.4      63.3     0.44
     6.8      65.6     0.37
     5.6      71.1     0.46
     9.8      77.4     0.61
    10.2       4.6     0.56
    12.2       6.5     0.45
    12.2      10.8     0.44
    13.9      15.8     0.50
    13.2      21.7     0.44
    10.8      26.0     0.50
    11.6      32.7     0.91
    13.2      35.8     0.55
    13.1      43.1     0.55
    10.7      45.3     0.53
    13.9      51.5     0.26
    13.4      57.2     0.33
    14.5      63.6     0.31
    14.6      68.8     0.30
    14.4      71.0     0.24
    14.0      77.5     0.64
    15.0       2.3     0.25
    17.1      10.0     0.22
    19.4      10.3     0.13
    16.6      15.1     0.18
    15.9      23.8     0.41
    17.0      28.8     0.25
    19.8      33.4     0.09
    19.1      38.4     0.24
    15.7      44.5     0.42
    15.0      45.7     0.32
    17.3      53.4     0.72
    19.8      56.8     0.54
    16.3      62.0     0.61
    16.9      66.9     0.52
    18.1      70.4     0.73
    16.5      78.3     0.72
    22.9       4.1     0.42
    21.6       6.6     0.23
    20.2      13.8     0.41
    21.8      18.9     0.39
    20.2      21.6     0.23
    20.2      25.4     0.22
    22.9      33.5     0.28
```

22.7	37.4	0.21
22.7	40.5	0.27
24.6	46.0	0.34
24.4	50.0	0.36
21.0	58.7	0.90
22.4	62.9	0.13
22.1	69.3	0.23
21.7	74.1	0.33
24.3	76.6	0.16
28.9	2.1	0.06
29.7	8.2	0.12
29.4	14.8	0.07
26.9	19.8	0.14
28.0	21.8	0.16
29.3	27.2	0.23
25.0	32.0	0.22
25.9	38.6	-99.00
27.7	44.3	0.25
27.7	48.9	0.37
28.0	53.1	0.24
28.1	58.3	0.26
27.0	62.9	0.44
27.3	67.0	0.42
26.2	73.3	0.54
26.8	79.2	0.28
32.6	3.2	0.21
31.7	6.0	0.41
30.3	10.7	0.17
34.0	18.4	0.18
33.6	23.4	0.15
32.2	27.1	-99.00
31.1	34.3	0.27
33.2	37.8	0.16
33.8	44.0	0.29
34.1	46.1	0.39
34.1	52.6	0.15
32.1	58.8	0.13
31.5	61.1	0.34
33.4	67.4	0.42
32.9	74.8	0.02
30.8	78.6	0.08
38.0	4.2	0.38
39.1	5.0	0.17
39.1	11.9	0.23
35.3	18.2	0.14
38.0	23.0	0.13
35.4	29.9	0.15
36.6	31.6	0.25
36.5	36.4	0.37
38.2	42.5	0.18
39.6	48.8	0.34
37.0	52.4	0.47
39.1	56.4	0.39
39.1	63.6	0.46
35.2	69.9	0.67
39.0	71.7	0.29
36.0	77.6	0.23
43.5	1.6	0.15
44.8	9.7	0.15
40.5	12.8	0.19
43.4	15.4	0.23
44.1	23.9	0.41
44.4	26.8	0.31
42.9	31.1	0.27
40.5	36.2	0.34
43.1	42.3	0.38
41.8	47.0	0.27
43.9	54.6	-99.00
40.2	55.1	-99.00
40.1	64.0	0.35
41.4	65.2	0.28
42.4	72.9	0.30
44.5	79.4	0.41
47.0	1.5	0.22
48.2	8.4	0.28
47.1	10.9	0.23
45.6	19.3	0.67

46.9	22.1	0.67
48.5	25.8	-99.00
47.1	31.0	0.25
47.4	39.2	0.20
49.8	40.9	0.40
49.2	45.2	0.49
45.6	54.8	0.31
46.8	56.7	0.29
46.6	62.9	0.39
49.3	66.0	0.49
46.8	73.4	0.52
46.0	77.6	0.39
50.9	.0	0.24
54.8	8.8	0.24
55.0	12.5	0.19
54.5	17.0	0.35
53.3	20.8	0.24
50.7	26.4	0.18
51.7	30.7	0.49
51.8	38.0	0.20
54.0	40.3	0.32
51.9	46.6	0.31
50.8	52.3	0.23
50.1	56.2	0.32
52.8	63.5	0.94
50.2	66.3	-99.00
51.4	74.4	0.31
52.8	76.1	0.63
56.9	2.9	0.19
56.5	7.6	0.17
58.2	12.4	0.21
57.7	18.5	0.18
57.2	20.3	0.19
58.0	29.9	0.25
55.2	30.4	0.71
56.1	35.5	0.30
56.1	42.5	0.22
56.1	48.8	0.20
56.6	54.8	0.12
56.6	56.2	0.71
59.7	60.9	0.32
59.3	67.0	0.17
56.2	72.7	0.22
59.8	79.2	0.33
61.5	0.6	0.15
61.6	5.8	-99.00
64.2	13.7	0.15
63.0	16.7	0.19
64.4	23.3	0.17
61.2	29.2	0.18
61.6	31.5	0.33
60.2	35.0	0.33
60.9	44.1	0.27
64.6	48.7	0.31
63.1	53.7	0.61
62.7	57.3	0.72
60.5	62.7	0.45
60.4	65.5	0.65
62.0	71.6	1.14
64.4	76.8	1.00
65.1	3.5	0.39
69.5	8.3	0.40
68.5	13.3	0.49
68.0	17.3	0.40
67.6	24.7	0.45
69.8	28.7	0.52
65.9	31.2	1.50
66.7	35.3	-99.00
67.4	41.6	0.85
66.2	47.0	0.79
68.8	53.5	0.56
65.0	58.0	0.43
70.0	63.8	0.58
66.3	65.8	0.54
65.0	70.0	1.25
66.6	79.0	0.52
73.1	1.7	0.28

71.3	9.4	0.30
74.9	11.3	0.25
70.7	19.1	0.26
71.5	21.8	0.34
72.3	29.6	0.59
73.5	32.9	0.47
70.5	35.3	0.56
74.4	41.2	0.37
71.3	48.7	0.41
70.3	52.7	0.33
70.2	59.7	0.40
70.2	61.2	0.60
73.5	68.7	0.57
72.5	73.1	0.58
74.0	76.2	0.43
75.0	4.9	0.50
79.1	7.3	0.51
75.4	11.6	0.51
79.0	18.7	0.44
77.9	23.9	0.54
75.8	29.6	0.51
75.5	30.9	0.53
79.3	39.6	0.51
79.9	43.7	0.56
76.6	47.9	0.56
77.4	53.0	0.54
75.0	56.5	0.59
75.1	64.2	0.85
77.5	66.6	0.80
77.3	72.0	0.71
76.9	74.5	0.85