7567 Volume II

Develop Documentation/Prepare Remedial Action Concept Plan for Building 24 Contamination Plume at Picatinny Arsenal
Appendices

Contract Number DAAA15-87-D-0220
Task Order 5

submitted to
Peer Consultants, Inc.
1160 Rockville Pike, Suite 202
Rockville, Maryland, 20852
and
U.S. Army Toxic and Hazardous Material Agency
Aberdeen Proving Ground, Maryland, 21010-5401

submitted by
Engineering Technologies Associates, Inc.
3458 Ellicott Center Drive
Ellicott City, Maryland, 21045
(301) 461-9920

September 28, 1989

Distribution Unlimited
Approved for Public Release

20070206265

Best Available Copy

Appendix A - MODFLOW Modifications

Appendix A - MODFLOW Modifications

The MODFLOW model calculates the head in the node with a pumping well. This head is not the actual head in the well, however. The head losses caused by ground water flowing to a small diameter withdrawal point are not accounted for. It is typically necessary to predict drawdown in the well as well as drawdown in the aquifer. For this reason, several new routines were written for the MODFLOW model. These routines allow the user to calculate drawdowns in the wells. These drawdowns include the effects of flow convergence and well efficiency. It is also possible to create a file of time versus drawdown at a well and have the output show an alphanumeric well name.

Convergence loss is calculated using the semiempirical equation developed by Prickett and Lonnquist (1971). The convergence loss is

$$h = 0.3665(Q/T)\log(del/4.81r_w)$$

where

h = head loss
Q = pumping rate
T = effective transmissivity
del = geometric mean of node length and width
r_w = well radius

Effective transmissivity is the input transmissivity for the node if the well is in a confined aquifer. If the well is in an unconfined aquifer, effective transmissivity is calculated as the product of the geometric mean of the anisotropy ratio, hydraulic conductivity, and saturated thickness.

Well efficiency is a measure of head losses caused by the gravel pack, and well screen. Well efficiency typically is predicted with an equation of the form

$$h = A(Q)^a$$

where

A = coefficient
a = exponent, typically between 1 and 3

These equations were added to the MODFLOW model. Input data is in the same format as in the original MODFLOW model; additional input variables appear on each line of the well module input data. The additional input parameters that may be used are

ALOSS - coefficient in well loss equation (A)

AEXP - exponent in well loss equation ([a])
RWELL - radius of well (r_w)
IWELL - unit number for time versus head output
WELNAM - 8 character well name that will appear on output

All or none of the above parameters may be used with the modified version of the model. If the above parameters are not entered the model will merely operate as before. Well drawdowns are output by the modified version of MODFLOW whenever heads are printed.

Well Package Input
(modified for well drawdown)

Input for the Well (WEL) Package is read from the unit specified in IUNIT(2).

FOR EACH SIMULATION

WEL1AL

1. Data: MXWELL IWELCB
   Format: I10 I10

FOR EACH STRESS PERIOD

WEL1RP

2. Data: ITMP
   Format: I10

3. Data:	Layer	Row	Column	Q	ALOSS	AEXP	RWELL	IWELL	WELNAM
Format:	I10	I10	I10	F10.0	F8.0	F8.0	F8.0	I6	A8

(Input item 3 normally consists of one record for each well. If ITMP is negative or zero, item 3 is not read.)

Explanation of Fields Used in Input Instructions

MXWELL--is the maximum number of wells used at any time.

IWELCB--is a flag and a unit number.

If IWELCB > 0, it is the unit number on which cell-by-cell flow terms will be recorded whenever ICBCFL (see Output Control) is set.

If IWELCB = 0, cell-by-cell flow terms will not be printed or recorded.

If IWELCB < 0, well recharge will be printed whenever ICBCFL is set.

ITMP--is a flag and a counter.

If ITMP < 0, well data from the last stress period will be reused.

If ITMP > 0, ITMP will be the number of wells active during the current stress period.

Layer--is the layer number of the model cell that contains the well.

Row--is the row number of the model cell that contains the well.

Column--is the column number of the model cell that contains the well.

Q--is the volumetric recharge rate. A positive value indicates recharge and a negative value indicates discharge.

ALOSS--well loss equation coefficient to calculate head loss due to well inefficiency. Equation for head loss due to well inefficiency is $ALOSS(Q)^{AEXP}$. Optional input.

AEXP--well loss equation exponent to calcualate head loss due to well inefficiency. Equation for head loss due to well inefficiency is $ALOSS(Q)^{AEXP}$. Optional input, necessary if ALOSS is greater than zero.

RWELL--well radius. Optional input.

IWELL--unit number for time versus head output at this well. Equal to zero if no time versus head output file to be written.

WELNAM--well name that will appear on output, maximum of eight characters.

Modifications to MAIN

The MAIN module of MODFLOW was modified by adding a call to new subroutine BAS2OT.

```
$LARGE
$DEBUG
$NOFLOATCALLS
CHRI  ******************************************************************
CHRI  Code distributed by:
CHRI           INTERNATIONAL GROUND WATER MODELING CENTER
CHRI          Holcomb Research Institute, Butler University
CHRI       Indianapolis, Indiana 46208, USA. Phone: (317)283-9458
CHRI  Date: January 1985
C     ******************************************************************
C     MAIN CODE FOR MODULAR MODEL --    6/1/83
C          BY MICHAEL G. MCDONALD AND ARLEN W. HARBAUGH
C-----VERSION 1116 28DEC1983 MAIN1
C     ******************************************************************
C
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
      INTEGER*4 LENX,ISUM
      COMMON X(65000)
      DIMENSION HEADNG(32),VBVL(4,20),IUNIT(24)
CF66
C     DIMENSION VBNM(4,20)
CF66
CF77
      CHARACTER*4 VBNM(4,20)
CF77
      DOUBLE PRECISION DUMMY
      EQUIVALENCE (DUMMY,X(1))
C     ------------------------------------------------------------------
C
C1------SET SIZE OF X ARRAY. REMEMBER TO REDIMENSION X.
      LENX=65000
C
C2------ASSIGN BASIC INPUT UNIT AND PRINTER UNIT.
      INBAS=1
      IOUT=6
C
C3------DEFINE PROBLEM__ROWS,COLUMNS,LAYERS,STRESS PERIODS,PACKAGES
      CALL BAS1DF(ISUM,HEADNG,NPER,ITMUNI,TOTIM,NCOL,NROW,NLAY,
     1             NODES,INBAS,IOUT,IUNIT)
C
C4------ALLOCATE SPACE IN "X" ARRAY.
      CALL BAS1AL(ISUM,LENX,LCHNEW,LCHOLD,LCIBOU,LCCR,LCCC,LCCV,
     1              LCHCOF,LCRHS,LCDELR,LCDELC,LCSTRT,LCBUFF,LCIOFL,
     2              INBAS,ISTRT,NCOL,NROW,NLAY,IOUT)
      IF(IUNIT(1).GT.0) CALL BCF1AL(ISUM,LENX,LCSC1,LCHY,
     1     LCBOT,LCTOP,LCSC2,LCTRPY,IUNIT(1),ISS,
     2     NCOL,NROW,NLAY,IOUT,IBCFCB)
      IF(IUNIT(2).GT.0) CALL WEL1AL(ISUM,LENX,LCWELL,MXWELL,NWELLS,
     1                 IUNIT(2),IOUT,IWELCB)
      IF(IUNIT(3).GT.0) CALL DRN1AL(ISUM,LENX,LCDRAI,NDRAIN,MXDRN,
     1                 IUNIT(3),IOUT,IDRNCB)
      IF(IUNIT(8).GT.0) CALL RCH1AL(ISUM,LENX,LCIRCH,LCRECH,NRCHOP,
     1            NCOL,NROW,IUNIT(8),IOUT,IRCHCB)
```

```
      IF(IUNIT(5).GT.0) CALL EVT1AL(ISUM,LENX,LCIEVT,LCEVTR,LCEXDP,
     1            LCSURF,NCOL,NROW,NEVTOP,IUNIT(5),IOUT,IEVTCB)
      IF(IUNIT(4).GT.0) CALL RIV1AL(ISUM,LENX,LCRIVR,MXRIVR,NRIVER,
     1            IUNIT(4),IOUT,IRIVCB)
      IF(IUNIT(7).GT.0) CALL GHB1AL(ISUM,LENX,LCBNDS,NBOUND,MXBND,
     1            IUNIT(7),IOUT,IGHBCB)
      IF(IUNIT(9).GT.0) CALL SIP1AL(ISUM,LENX,LCEL,LCFL,LCGL,LCV,
     1          LCHDCG,LCLRCH,LCW,MXITER,NPARM,NCOL,NROW,NLAY,
     2          IUNIT(9),IOUT)
      IF(IUNIT(11).GT.0) CALL SOR1AL(ISUM,LENX,LCA,LCRES,LCHDCG,LCLRCH,
     1           LCIEQP,MXITER,NCOL,NROW,NLAY,NSLICE,MBW,IUNIT(11),IOUT)
C
C5------IF THE "X" ARRAY IS NOT BIG ENOUGH THEN STOP.
      IF(ISUM-1.GT.LENX) STOP
C
C6------READ AND PREPARE INFORMATION FOR ENTIRE SIMULATION.
      CALL BAS1RP(X(LCIBOU),X(LCHNEW),X(LCSTRT),X(LCHOLD),
     1       ISTRT,INBAS,HEADNG,NCOL,NROW,NLAY,NODES,VBVL,X(LCIOFL),
     2       IUNIT(12),IHEDFM,IDDNFM,IHEDUN,IDDNUN,IOUT)
      IF(IUNIT(1).GT.0) CALL BCF1RP(X(LCIBOU),X(LCHNEW),X(LCSC1),
     1          X(LCHY),X(LCCR),X(LCCC),X(LCCV),X(LCDELR),
     2           X(LCDELC),X(LCBOT),X(LCTOP),X(LCSC2),X(LCTRPY),
     3           IUNIT(1),ISS,NCOL,NROW,NLAY,NODES,IOUT)
      IF(IUNIT(9).GT.0) CALL SIP1RP(NPARM,MXITER,ACCL,HCLOSE,X(LCW),
     1          IUNIT(9),IPCALC,IPRSIP,IOUT)
      IF(IUNIT(11).GT.0) CALL SOR1RP(MXITER,ACCL,HCLOSE,IUNIT(11),
     1        IPRSOR,IOUT)
C
C7------SIMULATE EACH STRESS PERIOD.
      DO 300 KPER=1,NPER
C
C7A-----READ STRESS PERIOD TIMING INFORMATION.
      CALL BAS1ST(NSTP,DELT,TSMULT,PERTIM,KPER,INBAS,IOUT)
C
C7B-----READ AND PREPARE INFORMATION FOR STRESS PERIOD.
      IF(IUNIT(2).GT.0) CALL WEL1RP(X(LCWELL),NWELLS,MXWELL,IUNIT(2),
     1             IOUT)
      IF(IUNIT(3).GT.0) CALL DRN1RP(X(LCDRAI),NDRAIN,MXDRN,IUNIT(3),
     1                 IOUT)
      IF(IUNIT(8).GT.0) CALL RCH1RP(NRCHOP,X(LCIRCH),X(LCRECH),
     1            X(LCDELR),X(LCDELC),NROW,NCOL,NLAY,IUNIT(8),IOUT)
      IF(IUNIT(5).GT.0) CALL EVT1RP(NEVTOP,X(LCIEVT),X(LCEVTR),
     1            X(LCEXDP),X(LCSURF),X(LCDELR),X(LCDELC),NCOL,NROW,
     1            NLAY,IUNIT(5),IOUT)
      IF(IUNIT(4).GT.0) CALL RIV1RP(X(LCRIVR),NRIVER,MXRIVR,IUNIT(4),
     1            IOUT)
      IF(IUNIT(7).GT.0) CALL GHB1RP(X(LCBNDS),NBOUND,MXBND,IUNIT(7),
     1              IOUT)
C
C7C-----SIMULATE EACH TIME STEP.
      DO 200 KSTP=1,NSTP
C
C7C1----CALCULATE TIME STEP LENGTH. SET HOLD=HNEW..
      CALL BAS1AD(DELT,TSMULT,TOTIM,PERTIM,X(LCHNEW),X(LCHOLD),KSTP,
     1             NCOL,NROW,NLAY)
C
C7C2----ITERATIVELY FORMULATE AND SOLVE THE EQUATIONS.
      DO 100 KITER=1,MXITER
C
C7C2A---FORMULATE THE FINITE DIFFERENCE EQUATIONS.
      CALL BAS1FM(X(LCHCOF),X(LCRHS),NCOL,NROW,NLAY,NODES)
      IF(IUNIT(1).GT.0) CALL BCF1FM(X(LCHCOF),X(LCRHS),X(LCHOLD),
     1          X(LCSC1),X(LCHNEW),X(LCIBOU),X(LCCR),X(LCCC),X(LCCV),
     2          X(LCHY),X(LCTRPY),X(LCBOT),X(LCTOP),X(LCSC2),
     3          X(LCDELR),X(LCDELC),DELT,ISS,KITER,KSTP,KPER,NCOL,
     4          NROW,NLAY,IOUT)
      IF(IUNIT(2).GT.0) CALL WEL1FM(NWELLS,MXWELL,X(LCRHS),X(LCWELL),
     1           X(LCIBOU),NCOL,NROW,NLAY)
      IF(IUNIT(3).GT.0) CALL DRN1FM(NDRAIN,MXDRN,X(LCDRAI),X(LCHNEW),
     1         X(LCHCOF),X(LCRHS),X(LCIBOU),NCOL,NROW,NLAY)
      IF(IUNIT(8).GT.0) CALL RCH1FM(NRCHOP,X(LCIRCH),X(LCRECH),
     1            X(LCRHS),X(LCIBOU),NCOL,NROW,NLAY)
      IF(IUNIT(5).GT.0) CALL EVT1FM(NEVTOP,X(LCIEVT),X(LCEVTR),
     1            X(LCEXDP),X(LCSURF),X(LCRHS),X(LCHCOF),X(LCIBOU),
     1            X(LCHNEW),NCOL,NROW,NLAY)
      IF(IUNIT(4).GT.0) CALL RIV1FM(NRIVER,MXRIVR,X(LCRIVR),X(LCHNEW),
     1            X(LCHCOF),X(LCRHS),X(LCIBOU),NCOL,NROW,NLAY)
      IF(IUNIT(7).GT.0) CALL GHB1FM(NBOUND,MXBND,X(LCBNDS),X(LCHCOF),
     1            X(LCRHS),X(LCIBOU),NCOL,NROW,NLAY)
C
C7C2B---MAKE ONE CUT AT AN APPROXIMATE SOLUTION.
      IF(IUNIT(9).GT.0) CALL SIP1AP(X(LCHNEW),X(LCIBOU),X(LCCR),X(LCCC),
     1     X(LCCV),X(LCHCOF),X(LCRHS),X(LCEL),X(LCFL),X(LCGL),X(LCV),
     2     X(LCW),X(LCHDCG),X(LCLRCH),NPARM,KITER,HCLOSE,ACCL,ICNVG,
     3     KSTP,KPER,IPCALC,IPRSIP,MXITER,NSTP,NCOL,NROW,NLAY,NODES,
     4     IOUT)
      IF(IUNIT(11).GT.0) CALL SOR1AP(X(LCHNEW),X(LCIBOU),X(LCCR),
     1     X(LCCC),X(LCCV),X(LCHCOF),X(LCRHS),X(LCA),X(LCRES),X(LCIEQP),
     2     X(LCHDCG),X(LCLRCH),KITER,HCLOSE,ACCL,ICNVG,KSTP,KPER,IPRSOR,
     3     MXITER,NSTP,NCOL,NROW,NLAY,NSLICE,MBW,IOUT)
C
C7C2C---IF CONVERGENCE CRITERION HAS BEEN MET STOP ITERATING.
      IF(ICNVG.EQ.1) GO TO 110
  100 CONTINUE
      KITER=MXITER
  110 CONTINUE
C
C7C3----DETERMINE WHICH OUTPUT IS NEEDED.
      CALL BAS1OC(NSTP,KSTP,KPER,ISTRT,ICNVG,X(LCIOFL),NLAY,
     1  IBUDFL,ICBCFL,IHDDFL,IUNIT(12),IOUT)
C
C7C4----CALCULATE BUDGET TERMS. SAVE CELL-BY-CELL FLOW TERMS.
      MSUM=1
      IF(IUNIT(1).GT.0) CALL BCF1BD(VBNM,VBVL,MSUM,X(LCHNEW),
     1     X(LCIBOU),X(LCHOLD),X(LCSC1),X(LCCR),X(LCCC),X(LCCV),
     2     X(LCTOP),X(LCSC2),DELT,ISS,NCOL,NROW,NLAY,KSTP,KPER,
     3     IBCFCB,ICBCFL,X(LCBUFF),IOUT)
      IF(IUNIT(2).GT.0) CALL WEL1BD(NWELLS,MXWELL,VBNM,VBVL,MSUM,
     1     X(LCWELL),X(LCIBOU),DELT,NCOL,NROW,NLAY,KSTP,KPER,IWELCB,
     2     ICBCFL,X(LCBUFF),IOUT)
      IF(IUNIT(3).GT.0) CALL DRN1BD(NDRAIN,MXDRN,VBNM,VBVL,MSUM,
     1     X(LCDRAI),DELT,X(LCHNEW),NCOL,NROW,NLAY,X(LCIBOU),KSTP,KPER,
     2     IDRNCB,ICBCFL,X(LCBUFF),IOUT)
      IF(IUNIT(8).GT.0) CALL RCH1BD(NRCHOP,X(LCIRCH),X(LCRECH),
     1     X(LCIBOU),NROW,NCOL,NLAY,DELT,VBVL,VBNM,MSUM,KSTP,KPER,
     2     IRCHCB,ICBCFL,X(LCBUFF),IOUT)
      IF(IUNIT(5).GT.0) CALL EVT1BD(NEVTOP,X(LCIEVT),X(LCEVTR),
     1     X(LCEXDP),X(LCSURF),X(LCIBOU),X(LCHNEW),NCOL,NROW,NLAY,
     2     DELT,VBVL,VBNM,MSUM,KSTP,KPER,IEVTCB,ICBCFL,X(LCBUFF),IOUT)
      IF(IUNIT(4).GT.0) CALL RIV1BD(NRIVER,MXRIVR,X(LCRIVR),X(LCIBOU),
     1     X(LCHNEW),NCOL,NROW,NLAY,DELT,VBVL,VBNM,MSUM,
     2     KSTP,KPER,IRIVCB,ICBCFL,X(LCBUFF),IOUT)
      IF(IUNIT(7).GT.0) CALL GHB1BD(NBOUND,MXBND,VBNM,VBVL,MSUM,
     1     X(LCBNDS),DELT,X(LCHNEW),NCOL,NROW,NLAY,X(LCIBOU),KSTP,KPER,
     2     IGHBCB,ICBCFL,X(LCBUFF),IOUT)
C
C7C5---PRINT AND OR SAVE HEADS AND DRAWDOWNS. PRINT OVERALL BUDGET.
C
      CALL BAS1OT(X(LCHNEW),X(LCSTRT),ISTRT,X(LCBUFF),X(LCIOFL),
     1        MSUM,X(LCIBOU),VBNM,VBVL,KSTP,KPER,DELT,
     2        PERTIM,TOTIM,ITMUNI,NCOL,NROW,NLAY,ICNVG,
     3     IHDDFL,IBUDFL,IHEDFM,IHEDUN,IDDNFM,IDDNUN,IOUT)
C
C      PRINT WELL DRAWDOWNS
C
      CALL BAS2OT(X(LCHNEW),X(LCIOFL),
     1        TOTIM,NCOL,NROW,NLAY,
     2     IHDDFL,IOUT,
     3     X(LCWELL),X(LCCC),X(LCCR),X(LCDELR),X(LCDELC),
     4     X(LCTRPY),X(LCIBOU),X(LCHY),X(LCBOT),X(LCTOP),
     5     NWELLS,MXWELL)
C
C7C6----IF ITERATION FAILED TO CONVERGE THEN STOP.
      IF(ICNVG.EQ.0) STOP
  200 CONTINUE
  300 CONTINUE
C
C8------END PROGRAM
      STOP
C
      END
```

Module BAS2OT

BAS2OT checks the head print flag, IHDDFL, and if set, calls routine SBAS2W.

```
      SUBROUTINE BAS2OT(HNEW,IOFLG,
     1  TOTIM,NCOL,NROW,NLAY,
     2  IHDDFL,IOUT,WELL,CC,CR,
     3  DELR,DELC,TRPY,IBOUND,HY,BOT,TOP,NWELLS,MXWELL)
C
C     WRITTEN BY DONALD KOCH, NOV 1986 TO CALL A ROUTINE TO
C       COMPUTE WELL HEADS (INCLUDING CONVERGEANCE LOSS AND WELL EFFICIENCY
C       LOSS
C
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
      DOUBLE PRECISION HNEW
C
      DIMENSION HNEW(NCOL,NROW,NLAY),
     1          IOFLG(NLAY,4),
     2          CC(NCOL,NROW,NLAY),CR(NCOL,NROW,NLAY),
     3          WELL(8,MXWELL),DELR(NCOL),DELC(NROW),TRPY(NLAY),
     4          HY(NCOL,NROW,NLAY),IBOUND(NCOL,NROW,NLAY),
     5          TOP(NCOL,NROW,NLAY),BOT(NCOL,NROW,NLAY)
CF66
C     ------------------------------------------------------------------
C
C3------IF HEAD AND DRAWDOWN FLAG (IHDDFL) IS SET WRITE WELL
C3------DRAWDOWNS IN ACCORDANCE WITH FLAGS IN IOFLG.
      IF(IHDDFL.EQ.0) GO TO 100
C
C        CALL TO WELL ROUTINE - DON KOCH   NOV. 1986
C
       CALL SBAS2W(HNEW,WELL,CC,CR,DELR,DELC,TRPY,IBOUND,
     1   HY,BOT,TOP,NWELLS,MXWELL,NCOL,NROW,NLAY,IOUT,TOTIM)
C
C6------RETURN
 100  RETURN
      END
```

Module SBAS2W

This routine computes the head in a pumping well including convergence losses and well efficiency loss. The routine first calculates node transmissivity. For layer type 2 or 4 (confined aquifer only), this involves back calculating the input transmissivity value. For layer type 1 or 3, subroutine SBAS2H is called, which calculates effective transmissivity from the anisotropy ratio, hydraulic conductivity and saturated thickness. The convergence loss is calculated with the following equation.

$$h = 0.3665(Q/T)\log_{10}(del/4.81r_w)$$

where

h = head loss
Q = pumping rate
T = effective transmissivity
del = geometric mean of node length and width
r_w = well radius

Well efficiency is a measure of head losses caused by the gravel pack, and well screen. Well efficiency typically is predicted with an equation of the form

$$h = A(Q)^a$$

where

A = coefficient
a = exponent, typically between 1 and 3

The routine prints to the output unit, a header; and for each well, the well number, the well name (if entered), convergence loss, well efficiency loss, and head in each well. If IWELL is greater than zero, time and head in the well are written to unit IWELL, whenever, heads are printed. If the well runs dry (only possible in layer type 1 or 3 when the elevation of the bottom of the aquifer is known), the well discharge is set to zero.

```
      SUBROUTINE SBAS2W(HNEW,WELL,CC,CR,DELR,DELC,TRPY,IBOUND,
     1HY,BOT,TOP,NWELLS,MXWELL,NCOL,NROW,NLAY,IOUT,TOTIM)
C
C
C
C-----VERSION 2.0 NOV 1986 SBAS2W
C     *****************************************************************
C    THIS ROUTINE COMPUTES THE HEAD IN A PUMPING WELL INCLUDING
C     CONVERGEANCE LOSSES IN A GRID NODE AND WELL EFFICIENCY LOSS
C     HEAD IN EACH WELL IS PRINTED TO THE OUTPUT AND IF A UNIT NUMBER
C     IS SPECIFIED, THE TIME AND HEAD ARE WRITTEN TO A SEPARATE FILE.
C
C     CONVERGEANCE LOSS IS CALCULATED USING THE SEMIEMPIRICAL RESULTS
C      OF PRICKETT AND LONNQUIST, "SELECTED DIGITAL COMPUTER TECHNIQUES
C      FOR GROUNDWATER RESOURCE EVALUATION", BULLETIN 55, ILLINOIS STATE
C      WATER SURVEY, 1971, P. 61.  THE BASIC VARIABLES ARE THE EFFECTIVE
C      RADIUS OF THE WELL (RWELL), THE TRANSMISSIVITY (TRANS), THE PUMPING
C      RATE (Q), AND THE SIZE OF THE NODE (GEOMETRIC MEAN OF DELR AND DELC).
C
C      THE EQUATION IS
C             CLOSS = 0.3665 * (Q/TRANS) *LOG10 (DEL/(4.81 * RWELL))
C
C
C     WELL EFFICIENCY LOSS IS CALCULATED AS
C
C       WLOSS = ALOSS * Q ** AEXP
C
C    WELL NAMES MAY ALSO BE USED, VARIABALE WELNAM IS DIMENSIONED
C     FOR A MAXIMUM OF 100 WELLS, IF MORE WELLS ARE DESIRED, THE
C     DIMENSION MUST BE CHANGED IN THIS SUBROUTINE AND IN SUBROUTINE
C       WEL1RP
C
C---SPECIFICATIONS
      CHARACTER*8 WELNAM
      DOUBLE PRECISION HNEW
      DIMENSION HNEW(NCOL,NROW,NLAY),WELL(8,MXWELL),CC(NCOL,NROW,NLAY)
     1,TRPY(NLAY),CR(NCOL,NROW,NLAY),DELR(NCOL),DELC(NROW),
     2IBOUND(NCOL,NROW,NLAY),HY(NCOL,NROW,NLAY),TOP(NCOL,NROW,NLAY),
     3BOT(NCOL,NROW,NLAY)
      COMMON/WELL2/WELNAM(100)
      COMMON/FLWCOM/LAYCON(80)
C-----------------------------------------------------------
C
C   PRINT A HEADER
      WRITE(IOUT,1002)
 1002 FORMAT(/1H ,11X,'WELL NO.   WELL NAME  CONVERGENCE',
     1'  EFFICIENCY   HEAD'/38X,'LOSS        LOSS     IN WELL')
      DO 5 II=1,NWELLS
           K=WELL(1,II)
           I=WELL(2,II)
           J=WELL(3,II)
           Q=WELL(4,II)
           ALOSS=WELL(5,II)
           AEXP=WELL(6,II)
```

```
          RWELL=WELL(7,II)
          IWELL=WELL(8,II)
C
C  CALCULATE PROPER TOP AND BOTTOM INDICES
           KB=0
           KT=0
           DO 55 KCOUNT=1,K
             IF(LAYCON(KCOUNT) .EQ. 1 .OR. LAYCON(KCOUNT) .EQ. 3)
     1        KB=KB+1
             IF(LAYCON(KCOUNT) .EQ. 2 .OR. LAYCON(KCOUNT) .EQ. 3)
     1        KT=KT+1
   55      CONTINUE
C
C         WRITE(*,*)II,I,J,K,KT,KB
C
C      CALCULATE THE NODE TRANSMISSIVITY ?????
          TRANS=0.
          ICOUNT=0
          TEMP1=0.
          IF(I .GT. 1)TEMP1=CC(J,I-1,K)*(DELC(I)+DELC(I-1))/2./DELR(J)
          IF(TEMP1 .GT. 0)ICOUNT=ICOUNT+1
          TEMP2=0.
          IF(J .GT. 1)TEMP2=CR(J-1,I,K)*(DELR(J)+DELR(J-1))/2./DELC(I)
          IF(TEMP2 .GT. 0)ICOUNT=ICOUNT+1
          IF(CC(J,I,K) .GT. 0)ICOUNT=ICOUNT+1
          IF(CR(J,I,K) .GT. 0)ICOUNT=ICOUNT+1
          IF(ICOUNT .GT. 0)TRANS=(CC(J,I,K)*(DELC(I+1)+DELC(I))/2./
     1     DELR(J)+CR(J,I,K)*(DELR(J+1)+DELR(J))/2./DELC(I)+TEMP1+
     2     TEMP2)/ICOUNT
C     . IF UNCONFINED AQUIFER AND THERE IS AN HY ARRAY STORED CALCULATE A
C            NEW TRANS THAT IS ACCURATE
        IF(LAYCON(K) .EQ. 1 .OR. LAYCON(K) .EQ. 3)CALL
     1     SBAS2H(HNEW,IBOUND,CC,HY,TRPY,BOT,TOP,K,I,J,KB,KT,NCOL,NROW,
     2             NLAY,TRANS)
C      CALCULATE CONVERGEANCE LOSS BASED ON GEOMETRIC AVERAGE NODE SIZE AND
C        EFFECTIVE RADIUS OF WELL
          CLOSS=0
          IF(RWELL .GT. 0 .AND. TRANS .GT. 0)CLOSS=0.3665*Q/TRANS*
     1       ALOG10(SQRT(DELR(J)*DELC(I))/4.81/RWELL)
C      CALCULATE WELL EFFICIENCY LOSS
          WLOSS=0
          IF(AEXP .GT. 0)WLOSS=SIGN(ALOSS*ABS(Q)**AEXP,Q)
C      CALCULATE HEAD IN WELL
          WHEAD=HNEW(J,I,K)+WLOSS+CLOSS
C      PRINT RESULTS
          WRITE(IOUT,1001)II,WELNAM(II),CLOSS,WLOSS,WHEAD
 1001     FORMAT(15X,I5,5X,A8,3G12.4)
C     IF IWELL NONZERO THEN PRINT TIME AND DRAWDOWN TO OUTPUT FILE
          IF(IWELL .GT. 0)WRITE(IWELL,1003)TOTIM,WHEAD
 1003     FORMAT(2G20.8)
C     CHECK TO SEE IF WELL WATER LEVEL IS BELOW BOTTOM OF AQUIFER, IF SO
C       TURN OFF WELL
          IF(LAYCON(K) .NE. 1 .AND. LAYCON(K) .NE. 3)GO TO 5
          IF(WHEAD .LT. BOT(J,I,KB) .OR. TRANS .LE. 0)WELL(4,II)=0.
          IF(WHEAD .LT. BOT(J,I,KB) .OR. TRANS .LE. 0)WRITE(IOUT,1004)
    1         WELNAM(II)
1004      FORMAT(15X,' WELL ',A8,' HAS GONE DRY, IT HAS BEEN TURNED',
    1     ' OFF')
   5 CONTINUE
     RETURN
     END
```

Module SBAS2H

Subroutine SBAS2H calculates effective transmissivity from saturated thickness, hydraulic conductivity, and the anisotropy ratio. For an anisotropic aquifer, the effective transmissivity is the geometric mean of the directional transmissivities.

```
      SUBROUTINE SBAS2H(HNEW,IBOUND,CC,HY,TRPY,
     1BOT,TOP,K,I,J,KB,KT,NCOL,NROW,NLAY,TRANS)
C
C
C     ******************************************************************
C     COMPUTE TRANSMISSIVITY FROM SATURATED THICKNESS AND HYDRAULIC
C     CONDUCTIVITY
C     ******************************************************************
C
C       THIS ROUTINE WRITTEN BY DONALD KOCH, ENGINEERING TECHNOLOGIES
C        ASSOCIATES, NOVEMBER 1986
C
C      SPECIFICATIONS:
C     ------------------------------------------------------------------
      DOUBLE PRECISION HNEW
C
      DIMENSION HNEW(NCOL,NROW,NLAY),IBOUND(NCOL,NROW,NLAY),
     1               CC(NCOL,NROW,NLAY),
     2               HY(NCOL,NROW,NLAY), TRPY(NLAY),
     3               BOT(NCOL,NROW,NLAY),TOP(NCOL,NROW,NLAY)
C
      COMMON /FLWCOM/LAYCON(80)
C     ------------------------------------------------------------------
C
C1------TRANSMISSIVITY WILL BE STORED TEMPORARILY IN THE CC ARRAY.
C2------IF CELL IS INACTIVE THEN SET T=0 & MOVE ON TO NEXT CELL.
      IF(IBOUND(J,I,K).NE.0) GO TO 10
      CC(J,I,K)=0.
      GO TO 200
C
C3------CALCULATE SATURATED THICKNESS.
   10 HD=HNEW(J,I,K)
      IF(LAYCON(K).EQ.1) GO TO 50
      IF(HD.GT.TOP(J,I,KT)) HD=TOP(J,I,KT)
   50 THCK=HD-BOT(J,I,KB)
C
C4------CHECK TO SEE IF SATURATED THICKNESS IS GREATER THAN ZERO.
      IF(THCK.LE.0.) GO TO 100
C
C5------IF SATURATED THICKNESS>0 THEN T=K*THICKNESS.
      CC(J,I,K)=THCK*HY(J,I,KB)
      GO TO 200
C
C6------WHEN SATURATED THICKNESS < 0, SET
C6------TRANSMISSIVITY=0
  100 CC(J,I,K)=0.
  200 CONTINUE
C       COMPUTE GEOMETRIC MEAN TRANSMISSIVITY IN CELL
      TRANS=SQRT(TRPY(K))*CC(J,I,K)
      RETURN
      END
```

Appendix B - User's Manual for PREMOD3D

I. Introduction

The PREMOD3D program takes the output from the United States Geological Survey ground water flow program (McDonald,M.G. and Harbaugh,A.W., A Modular Three-Dimensional Finite-Difference Ground-Water Flow Model, U.S. Geological Survey, 1984; MODFLOW) and prepares the flow input file for the three dimensional, random walk, solute transport program (RAND3D). PREMOD3D reads in the MODFLOW data input and binary head output files and calculates ground water velocities, and sink locations and strengths. The program is written in Fortran and designed to be run on an IBM PC microcomputer under MSDOS. The following sections of this user's manual describe the requirements for the use of the model, the procedures used to calculate velocities and flows, and detailed user instructions.

II. Description

The function of the PREMOD3D program is to take the output of the MODFLOW model and to prepare the velocity files for the RAND3D model. The program is written in Fortran and compiled using the Microsoft Fortran 3.31 compiler. Many of the subroutines in the program were taken from the MODFLOW model source code. The program calculates the velocity vectors in the x, y, and z directions across each node. The program also calculates the velocity of the water table including the change in elevation since the previous time step, recharge, and evapotranspiration. The program calculates the position and flow rate of each sink in the model. Sinks are automatically created at pumping wells, gaining rivers, gaining drains, and gaining general head boundaries. The program creates a velocity file with the .RND extension for each time step processed. This velocity file may be input to the RAND3D model. Options in the program permit selection of a subset of the flow model to be used for solute transport calculations, and stress periods to be skipped in the velocity file generation process.

The PREMOD3D program uses the same inputs as the MODFLOW model, plus the head file created by the MODFLOW model. Additional program inputs include a three letter code used in constructing file names for the output files and the rows and columns of the flow model to be used in the creation of the velocity file. The user is prompted for each stress period whether or not that stress period is to be processed into a velocity file.

The following equations are used to compute velocities.

$$VI=CR_{J,I,K}(H_{J,I,K}-H_{J+1,I,K})*2*CONV/7.48/(DELR_{J+1}+DELR_{J})$$

$$VJ=CC_{J,I-1,K}(H_{J,I,K}-H_{J,I-1,K})*2*CONV/7.48/(DELC_{I+1}+DELC_{I})$$

$$VZ=CV_{J,I,K-1}(H_{J,I,K}-H_{J,I,K-1})*CONV/7.48 \qquad \text{for } k=2 \text{ to } NL$$

$$VZ=(H_{J,I,K}-HOLD_{J,I,K})/DELT*CONV/7.48 - RECH_{J,I} - RIV/DELR_{J}/DELC_{I} - ET_{J,I} \qquad \text{for } k=1$$

where

VI = velocity in row direction from middle of node j,i,k to middle of node j+1,i,k
VJ = velocity in column direction from middle of node j,i,k to middle of node j,i-1,k (ft/day)
VZ = vertical velocity from node j,i,k to to j,i,k-1 (ft/day)
$CR_{j,i,k}$ = hydraulic conductivity from node j,i,k to node j+1,i,k (ft/day)
$CC_{j,i,k}$ = hydraulic conductivity from node j,i,k to node j,i+1,k (ft/day)
$CV_{j,i,k}$ = vertical leakance from node j,i,k to node j,i,k+1 (ft/day)
i = column index from right to left
j = row index from top to bottom
k = layer index from land surface down
$H_{j,i,k}$ = head in node j,i,k (ft)
$HOLD_{j,i,k}$ = head at previous time step in node j,i,k (ft)
$DELR_{j}$ = column width (ft)
$DELC_{i}$ = row width (ft)
DELT = time step (days)
CONV = conversion factor entered by user to convert from MODFLOW flow rates to gallons per day.
$RECH_{j,i}$ = recharge to top layer of model, positive for flow into aquifer (ft/day)
$ET_{j,i}$ = evapotranspiration from top layer of model, negative for flow out of aquifer (ft/day)
$RIV_{j,i}$ = river flow to top layer of model, positive for flow into aquifer (ft^3/day)

Notice that the PREMOD3D program reverses the sign convention for the rows and the layers. In RAND3D, the row numbering convention is from bottom to top, and the layer number convention is from the bottom layer to the surface (water table) layer.

Sink locations are computed by the PREMOD3D program based on the inputs of wells, rivers, drains, and general head boundaries. There are a set of subroutines that read in MODFLOW data (the same subroutines that are used in the MODFLOW program) and then create sinks using the same logic that was used to calculate the sink terms of the ground water flow differential equation being solved in MODFLOW. The basic procedure is that each well, drain, river, or general head boundary is tested to see if water is leaving the model during the current stress period. If water is leaving, the flow rate is calculated, and the location of the sink and flow rate are written into the output file.

Several features and options supported by MODFLOW are not supported by PREMOD3D and RAND3D. The following list describes the limitations of PREMOD3D.

- Only layer types 1 (water table) and 3 (fully convertible) are supported. This limitation is necessary because the RAND3D model requires top and bottom elevations for all confined aquifer units.
- The top layer of the model must be a layer type 1 (water table).
- Recharge and evapotranspiration are only allowed to and from the top layer of the model (recharge and evapotranspiration modules option 1).
- The top layer must stay active (all nodes must contain water at all times).
- PREMOD3D assumes that all time steps within a stress period are to be made into velocity files. The user may select which stress periods are to be processed, however.
- The maximum number of grid nodes (rows x columns x layers) that may be handled by the current version of PREMOD3D is approximately 9000, which corresponds to the approximate limit of nodes that may be simulated with MODFLOW on a 640K memory IBM PC under MSDOS. This limit may be expanded by redimensioning the X array and resetting variable LENX in the program.

III. User Instructions

The first step in preparing a velocity file for input to the RAND3D model is to run MODFLOW while saving the heads. Heads are saved by using the basic output module of the program. The MODFLOW user's manual explains how to set up the input to use this feature of the MODFLOW model. It is necessary to save heads from each time step of each stress period that is to be simulated. The heads are saved in a binary file (Fortran unformatted records). It is also recommended that the MODFLOW model be run using feet as the length unit and days as the time unit (PREMOD3D has only been tested for these input parameters). The following instructions assume that the user knows how to run the MODFLOW model and is familiar with the input/output conventions.

The next step is to run the PREMOD3D program. Enter "PREMOD3D" at the appropriate prompt.

```
C:\MODFLOW>a:premod3d
```

Then enter the file names of the MODFLOW input data in exactly the same fashion and order as was used for the MODFLOW program. Unit 6 is an output file. It should not be necessary to save this file for PREMOD3D. It shows the same outputs as generated by the MODFLOW program for the reading of input data and the allocation of space in memory. Unit 1 is the input to the basic module. The other unit numbers are assigned by the user in the basic module input. It will be necessary to enter the basic output control module input file just as it was entered for the MODFLOW model. The PREMOD3D program ignores all inputs except the input of the unit number on which heads were saved. PREMOD3D reads heads from this unit.

```
File name missing or blank - Please enter name
UNIT 6? x.lst
UNIT 1? b:prob1.dat
UNIT 2? b:prob2.dat
UNIT 3? b:prob3.dat
UNIT 5? b:prob5.dat
UNIT 4? b:prob4.dat
UNIT 16? b:probsshd.dat
UNIT 8? b:prob8.dat
```

After all the input files have been assigned, the user is prompted for a three letter code that will be used to synthesize velocity file names. The program automatically assigns file names using the image, "xxx###.RND" where xxx is the three letter code entered by the user, and ### is a number made up from the stress period and the time step. It would start with 101 for the first time step of stress period 1, 102 for the second time step of stress period 1 and so on. The first time step of stress period 2 would be 201. MODFLOW simulations with up to nine stress periods with up to 99 time steps each may be converted by PREMOD3D.

```
  ENTER A THREE LETTER CODE FOR THE OUTPUT FILE
abc
```

The user is next prompted for a unit number for the output. Enter any unused unit number (be careful with this one, it is possible to overwrite your own input data).

```
  ENTER A UNIT NUMBER FOR THE OUTPUT
19
```

The user is next prompted for a conversion factor to change model flow rates to gallons per day. The MODFLOW model permits any consistent set of units. The RAND3D model assumes all flow rates are in gpd and all velocities are in ft/day. The conversion factor is used to multiply MODFLOW flow rates to get gpd. Typically, the MODFLOW model is run using ft for length, and days for time. MODFLOW model flows are thus given in ft^3/day. The proper conversion factor is thus 7.48.

```
  ENTER A CONVERSION FACTOR TO CHANGE MODEL UNITS TO    GAL-
DAY UNITS
7.48
```

It is possible to generate velocity files for a subset of the flow model. PREMOD3D automatically calculates new row and column numbers and the coördinates (in feet) of the lower left corner of the velocity file grid (assuming that the lower left corner of the flow model grid is 0,0). Enter the desired range of rows and columns, inclusive. The model makes no corrections when subsetting a flow model; there will be noflow boundaries in the RAND3D model at the edges. The program does not correct the calculated velocities at these locations; particles will attempt to cross the no flow boundary and bounce off in an unrealistic fashion. It is the user's responsibility to setup a realistic model.

```
  ENTER RANGE OF COLUMNS TO USE IN TRANSPORT MODEL
1,20
  ENTER RANGE OF ROWS TO USE IN TRANSPORT MODEL
1,20
```

Then the PREMOD3D program begins to loop through the stress periods. At the beginning of each stress period, the user is asked if solute transport is to be simulated in the stress period, i.e. does the user wish to create velocity files for each time step in this stress period. If the answer is yes, the program begins to read in the previously saved head data for the first time step of the stress period after prompting the user for the name of the file on which heads were saved (using the unit number previously assigned in the basic output control module input). If the answer is no, the program cycles to the next stress period, reads more input data, and the user will be prompted again.

```
 DO YOU WISH TO SIMULATE SOLUTE TRANSPORT IN STRESS
PERIOD    1  (Y/N) y
UNIT 15? b:probwlhd.dat
```

At this point, no more user input is required (unless additional stress periods are to be converted to velocity files). The program opens a file for each time step, writes the header data, fills the file with the computed velocity vectors, sink locations and flow rates, and continues to the next time step.

```
  FILE abc101.RND OPENED ON UNIT            19
  FILE abc102.RND OPENED ON UNIT            19
  FILE abc103.RND OPENED ON UNIT            19
  FILE abc104.RND OPENED ON UNIT            19
  FILE abc105.RND OPENED ON UNIT            19
  FILE abc106.RND OPENED ON UNIT            19
  FILE abc107.RND OPENED ON UNIT            19
  FILE abc108.RND OPENED ON UNIT            19
  FILE abc109.RND OPENED ON UNIT            19
  FILE abc110.RND OPENED ON UNIT            19
Stop - Program terminated.

C:\MODFLOW>
```

The beginning and end of a velocity file are shown on the following page. The top line of the file shows the number of columns, number of rows, number of layers, grid spacing in the column direction (ft), grid spacing in the row direction (ft), and the x,y,z coordinates of the lower left bottom corner of the model grid system (ft). The following lines show for each grid node: location by column, row, layer; saturated thickness of this layer (ft); velocity in x direction (column) (ft/day); velocity in y direction (row) (ft/day); and velocity in z direction (up is positive) (ft/day); elevation of the bottom of the layer (ft); and elevation of the top of the layer (ft). In the top layer, the top elevation will be zero unless there is a river node at this location in which case the top elevation is the elevation of the bottom of the river. There is one line like this for each grid node. At the end of the file there is one line for each sink in the model. For each sink, the location by x,y coordinates (ft) and layer number and flow rate in gpd is shown.

Example Velocity file (abc110.rnd)

```
20 20  3     100.0      100.0        .0         .0     490.0
 1 20  3   40.0      -.984E-01   .000       .413E-01   660.        695.
 2 20  3   42.0      -.869E-01   .000      -.288E-02   660.        .000
 3 20  3   43.7      -.771E-01   .000      -.339E-02   660.        .000
 4 20  3   45.3      -.686E-01   .000      -.386E-02   660.        .000
 5 20  3   46.7      -.611E-01   .000      -.427E-02   660.        .000
 6 20  3   47.9      -.546E-01   .000      -.464E-02   660.        .000
 7 20  3   49.0      -.487E-01   .000      -.498E-02   660.        .000
 8 20  3   50.0      -.435E-01   .000      -.527E-02   660.        .000
 9 20  3   50.8      -.388E-01   .000      -.552E-02   660.        .000
10 20  3   51.6      -.346E-01   .000      -.573E-02   660.        .000
11 20  3   52.3      -.307E-01   .000      -.591E-02   660.        .000
12 20  3   52.9      -.271E-01   .000      -.606E-02   660.        .000
13 20  3   53.4      -.236E-01   .000      -.617E-02   660.        .000
14 20  3   53.9      -.202E-01   .000      -.626E-02   660.        .000
15 20  3   54.3      -.169E-01   .000      -.633E-02   660.        .000
16 20  3   54.7      -.135E-01   .000      -.637E-02   660.        .000
17 20  3   54.9      -.102E-01   .000      -.640E-02   660.        .000
18 20  3   55.1      -.680E-02   .000      -.642E-02   660.        .000
19 20  3   55.3      -.340E-02   .000      -.643E-02   660.        .000
20 20  3   55.3       .000       .000      -.644E-02   660.        .000
 1 19  3   40.0      -.983E-01   .000       .413E-01   660.        695.
 2 19  3   42.0      -.868E-01  -.110E-03  -.288E-02   660.        .000
 3 19  3   43.7      -.770E-01  -.220E-03  -.340E-02   660.        .000
                .
                .
                .
                .
                .
                .
12  1  1   100.      -.174E-02   .183E-04  -.115E-03   500.        600.
13  1  1   100.      -.160E-02   .183E-04  -.148E-03   500.        600.
14  1  1   100.      -.142E-02   .366E-04  -.176E-03   500.        600.
15  1  1   100.      -.123E-02   .183E-04  -.201E-03   500.        600.
16  1  1   100.      -.101E-02   .183E-04  -.221E-03   500.        600.
17  1  1   100.      -.769E-03   .183E-04  -.238E-03   500.        600.
18  1  1   100.      -.522E-03   .000      -.250E-03   500.        600.
19  1  1   100.      -.266E-03   .183E-04  -.259E-03   500.        600.
20  1  1   100.       .000       .183E-04  -.262E-03   500.        600.
      1150.0          950.0          3.0   .1440E+05
        50.0         1950.0          3.0   3260.
        50.0         1850.0          3.0   3260.
        50.0         1750.0          3.0   3255.
        50.0         1650.0          3.0   3246.
        50.0         1550.0          3.0   3232.
        50.0         1450.0          3.0   3219.
        50.0         1350.0          3.0   3205.
        50.0         1250.0          3.0   3191.
        50.0         1150.0          3.0   3182.
        50.0         1050.0          3.0   3173.
        50.0          950.0          3.0   3173.
```

50.0	850.0	3.0	3173.
50.0	750.0	3.0	3178.
50.0	650.0	3.0	3187.
50.0	550.0	3.0	3196.
50.0	450.0	3.0	3209.
50.0	350.0	3.0	3219.
50.0	250.0	3.0	3232.
50.0	150.0	3.0	3237.
50.0	50.0	3.0	3241.

PREMOD3D

```
$DEBUG
$LARGE
$NOFLOATCALLS
C
C  THIS PROGRAM READS IN INPUT DATA AND SAVED HEADS FROM THE USGS MODFLOW
C   3D GROUND WATER MODEL AND CREATES INPUT FILES FOR THE RAND3D MODEL
C
C  WRITTEN BY DONALD KOCH, ENGINEERING TECHNOLOGIES ASSOCIATES, NOVEMBER 1988
C
C  MOST OF THE CODE IS TAKEN FROM THE USGS MODULAR MODEL
C         BY MICHAEL G. MCDONALD AND ARLEN W. HARBAUGH
C-----VERSION 1116 28DEC1983 MAIN1
C     ******************************************************************
C
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
      INTEGER*4 LENX
      COMMON X(90000)
      DIMENSION HEADNG(32),VBVL(4,20),IUNIT(24)
CF77
      CHARACTER*4 VBNM(4,20)
      CHARACTER*1 ANSWER
      CHARACTER*3 CODE
CF77
      DOUBLE PRECISION DUMMY
      EQUIVALENCE (DUMMY,X(1))
C     ------------------------------------------------------------------
C
C1------SET SIZE OF X ARRAY. REMEMBER TO REDIMENSION X.
      LENX=90000
C
C2------ASSIGN BASIC INPUT UNIT AND PRINTER UNIT.
      INBAS=1
      IOUT=6
C
C3------DEFINE PROBLEM__ROWS,COLUMNS,LAYERS,STRESS PERIODS,PACKAGES
      CALL BAS1DF(ISUM,HEADNG,NPER,ITMUNI,TOTIM,NCOL,NROW,NLAY,
     1             NODES,INBAS,IOUT,IUNIT)
C
C4------ALLOCATE SPACE IN "X" ARRAY.
      CALL BAS1AL(ISUM,LENX,LCHNEW,LCHOLD,LCIBOU,LCCR,LCCC,LCCV,
     1              LCHCOF,LCRHS,LCDELR,LCDELC,LCSTRT,LCBUFF,LCIOFL,
     2              INBAS,ISTRT,NCOL,NROW,NLAY,IOUT)
      IF(IUNIT(1).GT.0) CALL BCF1AL(ISUM,LENX,LCSC1,LCHY,
     1     LCBOT,LCTOP,LCSC2,LCTRPY,IUNIT(1),ISS,
     2     NCOL,NROW,NLAY,IOUT,IBCFCB)
      IF(IUNIT(2).GT.0) CALL WEL1AL(ISUM,LENX,LCWELL,MXWELL,NWELLS,
     1                  IUNIT(2),IOUT,IWELCB)
      IF(IUNIT(3).GT.0) CALL DRN1AL(ISUM,LENX,LCDRAI,NDRAIN,MXDRN,
     1                  IUNIT(3),IOUT,IDRNCB)
      IF(IUNIT(8).GT.0) CALL RCH1AL(ISUM,LENX,LCIRCH,LCRECH,NRCHOP,
     1            NCOL,NROW,IUNIT(8),IOUT,IRCHCB)
      IF(IUNIT(5).GT.0) CALL EVT1AL(ISUM,LENX,LCIEVT,LCEVTR,LCEXDP,
     1            LCSURF,NCOL,NROW,NEVTOP,IUNIT(5),IOUT,IEVTCB)
      IF(IUNIT(4).GT.0) CALL RIV1AL(ISUM,LENX,LCRIVR,MXRIVR,NRIVER,
     1            IUNIT(4),IOUT,IRIVCB)
      IF(IUNIT(7).GT.0) CALL GHB1AL(ISUM,LENX,LCBNDS,NBOUND,MXBND,
     1            IUNIT(7),IOUT,IGHBCB)
C
C5------IF THE "X" ARRAY IS NOT BIG ENOUGH THEN STOP.
      IF(ISUM-1.GT.LENX) STOP
C
C6------READ AND PREPARE INFORMATION FOR ENTIRE SIMULATION.
      CALL BAS1RP(X(LCIBOU),X(LCHNEW),X(LCSTRT),X(LCHOLD),
     1       ISTRT,INBAS,HEADNG,NCOL,NROW,NLAY,NODES,VBVL,X(LCIOFL),
     2       IUNIT(12),IHEDFM,IDDNFM,IHEDUN,IDDNUN,IOUT)
      IF(IUNIT(1).GT.0) CALL BCF1RP(X(LCIBOU),X(LCHNEW),X(LCSC1),
     1          X(LCHY),X(LCCR),X(LCCC),X(LCCV),X(LCDELR),
     2           X(LCDELC),X(LCBOT),X(LCTOP),X(LCSC2),X(LCTRPY),
     3           IUNIT(1),ISS,NCOL,NROW,NLAY,NODES,IOUT)
C
C      GET A THREE LETTER CODE FOR THE OUTPUT FILE NAME FROM THE USER
C
       WRITE(*,*)'  ENTER A THREE LETTER CODE FOR THE OUTPUT FILE'
       READ(*,1003)CODE
 1003  FORMAT(A3)
       WRITE(*,*)'  ENTER A UNIT NUMBER FOR THE OUTPUT'
       READ(*,*)IVOUT
       WRITE(IOUT,*)'  OUTPUT FILE CODE = ',CODE
       WRITE(IOUT,*)'  OUTPUT UNIT NUMBER = ',IVOUT
C             QUERY USER FOR CONVERSION FACTOR - TO CONVERT MODEL UNITS TO
C                        GAL-DAY UNITS
C
       WRITE(*,*)'  ENTER A CONVERSION FACTOR TO CHANGE MODEL UNITS TO
     1  GAL-DAY UNITS'
       READ(*,*)CONVRT
       WRITE(IOUT,*)'  CONVERSION FACTOR = ',CONVRT
C
C          ENTER SECTION OF MODEL TO BE USED IN TRANSPORT SIMULATION
C
    5  WRITE(*,*)'  ENTER RANGE OF COLUMNS TO USE IN TRANSPORT MODEL'
       READ(*,*)LLX,LUX
       IF(LLX .LT. 1 .OR. LLX .GT. NCOL)GO TO 401
       IF(LUX .LT. 1 .OR. LUX .GT. NCOL)GO TO 401
       IF(LUX .LE. LLX)GO TO 401
       WRITE(*,*)'  ENTER RANGE OF ROWS TO USE IN TRANSPORT MODEL'
       READ(*,*)LLY,LUY
       IF(LLY .LT. 1 .OR. LLY .GT. NROW)GO TO 401
       IF(LUY .LT. 1 .OR. LUY .GT. NROW)GO TO 401
       IF(LUY .LE. LLY)GO TO 401
       WRITE(IOUT,*)'  COLUMNS IN TRANSPORT MODEL FROM ',LLX,' TO ',
     1 LUX
       WRITE(IOUT,*)'  ROWS IN TRANSPORT MODEL FROM ',LLY,' TO ',LUY
C
C7------SIMULATE EACH STRESS PERIOD.
```

```
      DO 300 KPER=1,NPER
C
C7A-----READ STRESS PERIOD TIMING INFORMATION.
      CALL BAS1ST(NSTP,DELT,TSMULT,PERTIM,KPER,INBAS,IOUT)
C
C7B-----READ AND PREPARE INFORMATION FOR STRESS PERIOD.
      IF(IUNIT(2).GT.0) CALL WEL1RP(X(LCWELL),NWELLS,MXWELL,IUNIT(2),
     1             IOUT)
      IF(IUNIT(3).GT.0) CALL DRN1RP(X(LCDRAI),NDRAIN,MXDRN,IUNIT(3),
     1                 IOUT)
      IF(IUNIT(8).GT.0) CALL RCH1RP(NRCHOP,X(LCIRCH),X(LCRECH),
     1     X(LCDELR),X(LCDELC),NROW,NCOL,NLAY,IUNIT(8),IOUT)
      IF(IUNIT(5).GT.0) CALL EVT1RP(NEVTOP,X(LCIEVT),X(LCEVTR),
     1            X(LCEXDP),X(LCSURF),X(LCDELR),X(LCDELC),NCOL,NROW,
     2            NLAY,IUNIT(5),IOUT)
      IF(IUNIT(4).GT.0) CALL RIV1RP(X(LCRIVR),NRIVER,MXRIVR,IUNIT(4),
     1            IOUT)
      IF(IUNIT(7).GT.0) CALL GHB1RP(X(LCBNDS),NBOUND,MXBND,IUNIT(7),
     1              IOUT)
C
C        QUERY USER IF THIS STRESS PERIOD IS TO BE USED, IF NOT JUMP
C           TO END OF STRESS PERIOD LOOP
C
      WRITE(*,1002)KPER
 1002 FORMAT(1X,' DO YOU WISH TO SIMULATE SOLUTE TRANSPORT IN STRESS
     1 PERIOD ',I3,'  (Y/N) '\)
      READ(*,1001)ANSWER
 1001 FORMAT(A1)
      IF(ANSWER .EQ. 'N' .OR. ANSWER .EQ. 'n')GO TO 300
      WRITE(IOUT,*)' TIME STEPS IN STRESS PERIOD ',KPER,' CONVERTED '
C
C       TIME STEP LOOP
      DO 200 KSTP=1,NSTP
C
C-----READ IN HEADS FROM EACH TIME STEP.
C
      CALL BASPRE(X(LCHNEW),X(LCBUFF),X(LCIOFL),KSTP,KPER,NCOL,
     1       NROW,NLAY,IOUT,IHEDFM,IHEDUN,IPFLAG,PERTIM,TOTIM)
C
C             OPEN OUTPUT FILE ON UNIT IVOUT WITH NAME CODE+KSTP+".RND"
C
      CALL OUTDAT(IVOUT,CODE,KPER,KSTP,IOUT)
C
C              WRITE HEADER TO FILE
C
       CALL HEADER(NCOL,NROW,NLAY,X(LCDELR),X(LCDELC),IVOUT,LLX,LUX,
     1   LLY,LUY,X(LCRHS),X(LCHCOF),X(LCBOT))
C
C                  CALCULATE APPARENT VELOCITY OF WATER TABLE FROM RECHARGE
C                   AND PUT INTO ZVEL AND X(LCRHS)
C
       IF(IUNIT(8) .GT. 0)CALL PRERCH(NRCHOP,X(LCIRCH),X(LCRECH)
     1          ,X(LCRHS),X(LCIBOU),NROW,NCOL,NLAY)
C
```

```
C
C                        SWITCH HNEW TO HOLD
      CALL HSWITCH(X(LCHNEW),X(LCHOLD),X(LCRHS),NCOL,NROW,NLAY)
C                       END OF TIME PERIOD LOOP
  200 CONTINUE
C                       END OF STRESS PERIOD LOOP
  300 CONTINUE
C
C8------END PROGRAM
      STOP
C
C                   ERROR MESSAGE FOR BAD COLUMNS OR ROW SELECTION
  401 WRITE(*,*)'  OUTSIDE MODEL - TRY AGAIN'
      GO TO 5
C
      END
C
      SUBROUTINE HSWITCH(HNEW,HOLD,ZVEL,NCOL,NROW,NLAY)
C
C        THIS SUBROUTINE REPLACES HOLD WITH HNEW AT THE END OF THE TIME LOOP
C
C        WRITTEN BY DONALD KOCH, ENGINEERING TECHNOLOGIES ASSOCIATES, INC
C                     DECEMBER 1988
C
      DOUBLE PRECISION HNEW
      DIMENSION HNEW(NCOL,NROW,NLAY),HOLD(NCOL,NROW,NLAY),
     1 ZVEL(NCOL,NROW)
C
      DO 5 K=1,NLAY
        DO 10 I=1,NROW
           DO 15 J=1,NCOL
               HOLD(J,I,K)=HNEW(J,I,K)
   15      CONTINUE
   10   CONTINUE
    5 CONTINUE
C
C            ZERO OUT ZVEL ARRAY FOR NEW TIME STEP
      DO 20 I=1,NROW
        DO 25 J=1,NCOL
            ZVEL(J,I)=0
   25   CONTINUE
   20 CONTINUE
      RETURN
      END
C
      SUBROUTINE HEADER(NCOL,NROW,NLAY,DELR,DELC,IVOUT,LLX,LUX,
     1   LLY,LUY,ZVEL,RBOT,BOT)
C
C       WRITE HEADER RECORD INTO IVOUT AND INITIALIZE ZVEL AND RBOT
C        CALCULATE LOWER LEFT CORNER
C        WRITTEN BY DONALD KOCH, ENGINEERING TECHNOLOGIES ASSOCIATES, INC
C                     DECEMBER 1988
C
      DIMENSION DELR(NCOL),DELC(NROW),ZVEL(NCOL,NROW),RBOT(NCOL,NROW),
C                   CALCULATE APPARENT VELOCITY OF WATER TABLE FROM
C                      EVAPOTRANSPIRATION AND PUT INTO ZVEL AND X(LCRHS)
C
      IF(IUNIT(5).GT.0)CALL PREEVT(NEVTOP,X(LCIEVT),X(LCEVTR),
     1            X(LCEXDP),X(LCSURF),X(LCRHS),X(LCIBOU),X(LCHNEW),
     2            NCOL,NROW,NLAY)
C
C                   CALCULATE APPARENT VELOCITY OF WATER TABLE FROM
C                      RIVER LEAKAGE AND PUT INTO ZVELAND X(LCRHS)
C
      IF(IUNIT(4) .GT. 0)CALL PRERIV(NRIVER,MXIRVR,X(LCRIVR),
     1 X(LCHNEW),X(LCIBOU),X(LCHCOF),X(LCDELR),X(LCDELC),X(LCRHS),
     2 NCOL,NROW,NLAY,IVOUT,CONVRT)
C
C        CALCULATE VELOCITIES AND WRITE TO FILE
C
      CALL VELOCITY(X(LCHNEW),X(LCHOLD),X(LCIBOU),X(LCCR),X(LCCC),
     1  X(LCCV),X(LCHY),X(LCTRPY),X(LCBOT),X(LCTOP),X(LCDELR),
     2  X(LCDELC),DELT,ISS,KSTP,KPER,NCOL,NROW,NLAY,IOUT,IVOUT,
     3  X(LCRHS),X(LCHCOF),LLX,LUX,LLY,LUY,CONVRT)
C
C         CALCULATE SINK LOCATIONS AND WRITE TO FILE
C
C         SINKS AT CONSTANT HEAD NODES WHERE WATER LEAVES MODEL
C
C      CALL CHDPRE(
C
C                 CREATE SINKS AT WELLS
C
      IF(IUNIT(2) .GT. 0)CALL WELPRE(NWELLS,MXWELL,X(LCWELL),
     1 X(LCIBOU),X(LCDELR),X(LCDELC),NCOL,NROW,NLAY,IVOUT,CONVRT,
     2 LLX,LUX,LLY,LUY)
C
C                      CREATE SINKS AT DRAINS
C
      IF(IUNIT(3).GT.0) CALL DRNPRE(NDRAIN,MXDRN,X(LCDRAI),
     1 X(LCHNEW),X(LCIBOU),X(LCDELR),X(LCDELC),NCOL,NROW,NLAY,
     2 IVOUT,CONVRT,LLX,LUX,LLY,LUY)
C
C                       CREATE SINKS AT RIVERS
C
      IF(IUNIT(4) .GT. 0)CALL RIVPRE(NRIVER,MXIRVR,X(LCRIVR),
     1 X(LCHNEW),X(LCIBOU),X(LCDELR),X(LCDELC),NCOL,NROW,
     2 NLAY,IVOUT,CONVRT,LLX,LUX,LLY,LUY)
C
C                        CREATE SINKS AT GENERALIZED HEAD BOUNDARIES
C
      IF(IUNIT(7).GT.0) CALL GHBPRE(NBOUND,MXBND,X(LCBNDS),
     1  X(LCHNEW),X(LCDELR),X(LCDELC),NCOL,NROW,NLAY,IVOUT,CONVRT,
     2  LLX,LUX,LLY,LUY)
C
C                           CLOSE OUTPUT UNIT
C
      CLOSE(UNIT=IVOUT)
     1BOT(NCOL,NROW,NLAY)
C
C                 CHECK FOR UNEQUAL GRID SPACING
      IF(DELR(LLX) .NE. DELR(LUX))GO TO 100
      IF(DELC(LLY) .NE. DELC(LUY))GO TO 100
C                 ASSIGN COORDINATES OF LOWER LEFT CORNER OF GRID
C                     FIRST FIND LOWEST POINT OF MODEL
      MINBOT=BOT(1,1,1)
      K=NLAY
      DO 15 J=1,NCOL
       DO 15 I=1,NROW
         IF(MINBOT .GT. BOT(J,I,K))MINBOT=BOT(J,I,K)
   15 CONTINUE
      ZLL=MINBOT-10.
C                     THEN CALC LOWER LEFT CORNER POSITION
      CALL XYZPOS(LLX,LUY,1,XLL,YLL,Z,DELR,DELC,NROW,NCOL,NLAY)
      XLL=XLL-DELR(LLX)/2.
      YLL=YLL-DELC(LUY)/2.
      WRITE(IVOUT,1001)LUX-LLX+1,LUY-LLY+1,NLAY,DELR(LLX),DELC(LLY),
     1  XLL,YLL,ZLL
 1001 FORMAT(3I3,5F10.1)
C
C                 INITIALIZE ZVEL AND RBOT
C
      DO 5 J=1,NCOL
        DO 5 I=1,NROW
            ZVEL(J,I)=0.0
            RBOT(J,I)=0.0
    5 CONTINUE
C                              RETURN
      RETURN
C                             ERROR FOR UNEQUAL GRID SPACING
  100 WRITE(*,*)'  UNEQUAL GRID SPACING - CANNOT BE USED WITH RAND3D'
    . STOP 3333
      END
      SUBROUTINE OUTDAT(IVOUT,CODE,KPER,KSTEP,IOUT)
C
C   THIS SUBROUTINE CREATES NAMES FOR AND OPENS THE VELOCITY AND
C     SINK OUTPUT FILES
C
C        WRITTEN BY DONALD KOCH, ENGINEERING TECHNOLOGIES ASSOCIATES, INC
C                   - DECEMBER 1988
C
C     IVOUT - OUTPUT UNIT FOR VELOCITY FILE
C     CODE  - 3 LETTER CODE TO BE USED IN FILES
C     VOTFIL - NAME OF VELOCITY OUTPUT FILE
C     KSTEP - TIME STEP NUMBER
C     KPER - PERIOD NUMBER
C     KNUM - OUTPUT FILE NUMBER
C
      CHARACTER*3 CODE
      CHARACTER*10 VOTFIL
      DATA VOTFIL/'XXX000.RND'/
C
```

```
      KNUM=KPER*100+KSTEP
      WRITE(VOTFIL(1:6),1001)CODE,KNUM
C     WRITE(*,*)VOTFIL,KSTEP
 1001 FORMAT(A3,I3.3)
      OPEN(IVOUT,FILE=VOTFIL,STATUS='NEW')
      WRITE(*,*)'  FILE ',VOTFIL,' OPENED ON UNIT ',IVOUT
      WRITE(IOUT,*)'  FILE ',VOTFIL,' OPENED ON UNIT ',IVOUT
      RETURN
      END
      SUBROUTINE XYZPOS(J,I,K,X,Y,Z,DELR,DELC,NROW,NCOL,NLAYER)
C
C    THIS SUBROUTINE COMPUTES THE POSITION OF A SINK IN FEET FROM
C      THE LOWER LEFT CORNER OF THE GRID WHEN GIVEN THE POSITION IN
C        J,I,K COORDINATES
C
C          WRITTEN BY DONALD KOCH, ETA, DEC 1988
C
C    VARIABLES
C         J - COLUMN
C         I - ROW
C         K - LAYER
C         X - DISTANCE FROM LOWER LEFT CORNER ALONG ROWS
C         Y - DISTANCE FROM LOWER LEFT CORNER ALONG COLUMNS
C         Z - DISTANCE FROM BOTTOM
C         DELR - ARRAY OF WIDTH OF ROWS
C         DELC - ARRAY OF WIDTH OF COLUMNS
C         NROW - NUMBER OF ROWS
C         NCOL - NUMBER OF COLUMNS
C         NLAYER - NUMBER OF LAYERS
C
      DIMENSION DELR(NCOL),DELC(NROW)
C
C                 COMPUTE X
      X=0
      IF(J .GT. NCOL .OR. J .LT. 1)GO TO 20
      DO 5 IJ=1,J-1
          X=X+DELR(IJ)
    5 CONTINUE
      X=X+DELR(J)/2
C              COMPUTE Y
      Y=0
      IF(I .GT. NROW .OR. I .LT. 1)GO TO 20
      DO 10 II=NROW,I+1,-1
          Y=Y+DELC(II)
   10 CONTINUE
      Y=Y+DELC(I)/2
C              COMPUTE Z IN TERMS OF LAYERING
      IF(K .GT. NLAYER .OR. K .LT. 1)GO TO 20
      Z=NLAYER-K+1
C
      RETURN
   20 WRITE(*,*)J,I,K,' THIS POSITION OUTSIDE MODEL GRID'
      STOP 2222
      END
```

```
$DEBUG
$LARGE
$NOFLOATCALLS
      SUBROUTINE BCF1AL(ISUM,LENX,LCSC1,LCHY,LCBOT,
     1     LCTOP,LCSC2,LCTRPY,IN,ISS,NCOL,NROW,NLAY,IOUT,IBCFCB)
C
C-----VERSION 0931 08DEC1983 BCF1AL
C
C     ******************************************************************
C     ALLOCATE ARRAY STORAGE FOR BLOCK-CENTERED FLOW PACKAGE
C     ******************************************************************
C
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
      INTEGER*4 LENX
      COMMON /FLWCOM/LAYCON(80)
C     ------------------------------------------------------------------
C
C1------IDENTIFY PACKAGE
      WRITE(IOUT,1)IN
    1 FORMAT(1H0,'BCF1 -- BLOCK-CENTERED FLOW PACKAGE, VERSION 1',
     1', 12/08/83',' INPUT READ FROM UNIT',I3)
C
C2------READ AND PRINT ISS (STEADY-STATE FLAG) AND IBCFCB (FLAG FOR
C2------PRINTING OR UNIT# FOR RECORDING CELL-BY-CELL FLOW TERMS)
      READ(IN,2) ISS,IBCFCB
    2 FORMAT(2I10)
      IF(ISS.EQ.0) WRITE(IOUT,3)
    3 FORMAT(1X,'TRANSIENT SIMULATION')
      IF(ISS.NE.0) WRITE(IOUT,4)
    4 FORMAT(1X,'STEADY-STATE SIMULATION')
      IF(IBCFCB.GT.0) WRITE(IOUT,9) IBCFCB
    9 FORMAT(1X,'CELL-BY-CELL FLOWS WILL BE RECORDED ON UNIT',I3)
      IF(IBCFCB.LT.0) WRITE(IOUT,88)
   88 FORMAT(1X,'CONSTANT HEAD CELL-BY-CELL FLOWS WILL BE PRINTED')
C
C3------READ TYPE CODE FOR EACH LAYER AND COUNT TOPS AND BOTTOMS
      IF(NLAY.LE.80) GO TO 50
      WRITE(IOUT,11)
   11 FORMAT(1H0,'YOU HAVE SPECIFIED MORE THAN 80 MODEL LAYERS'/1X,
     1  'SPACE IS RESERVED FOR A MAXIMUM OF 80 LAYERS IN ARRAY LAYCON')
      STOP
C
C3A-----READ LAYER TYPE CODES.
   50 READ(IN,51) (LAYCON(I),I=1,NLAY)
   51 FORMAT(40I2)
C        BOTTOM IS READ FOR TYPES 1,3    TOP IS READ FOR TYPES 2,3
      WRITE(IOUT,52)
   52 FORMAT(1X,5X,'LAYER  AQUIFER TYPE',/1X,5X,19('-'))
C
C3B-----INITIALIZE TOP AND BOTTOM COUNTERS.
      NBOT=0
      NTOP=0
C
```

```
C3C------PRINT LAYER TYPE AND COUNT TOPS AND BOTTOMS NEEDED.
      DO 100 I=1,NLAY
C
C3C1----PRINT LAYER NUMBER AND LAYER TYPE CODE.
      L=LAYCON(I)
      WRITE(IOUT,7) I,L
    7 FORMAT(1X,I9,I10)
C
C3C2----ONLY THE TOP LAYER CAN BE UNCONFINED(LAYCON=1).
      IF(L.NE.1 .OR. I.EQ.1) GO TO 70
      WRITE(IOUT,8)
    8 FORMAT(1H0,'AQUIFER TYPE 1 IS ONLY ALLOWED IN TOP LAYER')
      STOP
C
C3C3----LAYER TYPES 1 AND 3 NEED A BOTTOM. ADD 1 TO KB.
   70 IF(L.EQ.1 .OR. L.EQ.3) NBOT=NBOT+1
C
C3C4----LAYER TYPES 2 AND 3 NEED A TOP. ADD 1 TO KT.
      IF(L.EQ.2 .OR. L.EQ.3) NTOP=NTOP+1
  100 CONTINUE
C
C
C
C4------COMPUTE DIMENSIONS FOR ARRAYS.
      NRC=NROW*NCOL
      ISIZ=NRC*NLAY
C
C5------ALLOCATE SPACE FOR ARRAYS. IF RUN IS TRANSIENT(ISS=0)
C5------THEN SPACE MUST BE ALLOCATED FOR STORAGE.
      ISOLD=ISUM
      LCSC1=ISUM
      IF(ISS.EQ.0) ISUM=ISUM+ISIZ
      LCSC2=ISUM
      IF(ISS.EQ.0) ISUM=ISUM+NRC*NTOP
      LCTRPY=ISUM
      ISUM=ISUM+NLAY
      LCBOT=ISUM
      ISUM=ISUM+NRC*NBOT
      LCHY=ISUM
      ISUM=ISUM+NRC*NBOT
      LCTOP=ISUM
      ISUM=ISUM+NRC*NTOP
C
C6------PRINT THE AMOUNT OF SPACE USED BY THE BCF PACKAGE.
      ISP=ISUM-ISOLD
      WRITE(IOUT,101) ISP
  101 FORMAT(1X,I6,' ELEMENTS IN X ARRAY ARE USED BY BCF')
      ISUM1=ISUM-1
      WRITE(IOUT,102) ISUM1,LENX
  102 FORMAT(1X,I6,' ELEMENTS OF X ARRAY USED OUT OF',I7)
      IF(ISUM1.GT.LENX) WRITE(IOUT,103)
  103 FORMAT(1X,'   ***X ARRAY MUST BE DIMENSIONED LARGER***')
C
C7------RETURN
```

```
      RETURN
      END
      SUBROUTINE BCF1RP(IBOUND,HNEW,SC1,HY,CR,CC,CV,DELR,DELC,
     1        BOT,TOP,SC2,TRPY,IN,ISS,NCOL,NROW,NLAY,NODES,IOUT)
C
C-----VERSION 1003 03MAY1983 BCF1RP
C
C     ******************************************************************
C     READ AND INITIALIZE DATA FOR BLOCK-CENTERED FLOW PACKAGE
C     ******************************************************************
C
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
      DOUBLE PRECISION HNEW
C
      DIMENSION HNEW(NODES),SC1(NODES),HY(NODES),CR(NODES),CC(NODES),
     1          CV(NODES),DELR(NCOL),DELC(NROW),BOT(NODES),TOP(NODES),
     1          SC2(NODES),TRPY(NLAY),IBOUND(NODES)
CF66
C     DIMENSION ANAME(6,10)
CF66
CF77
      CHARACTER*4 ANAME(6,10)
CF77
C
      COMMON /FLWCOM/LAYCON(80)
C
      DATA ANAME(1,1),ANAME(2,1),ANAME(3,1),ANAME(4,1),ANAME(5,1),
     1  ANAME(6,1) /'    ','PRIM','ARY ','STOR','AGE ','COEF'/
      DATA ANAME(1,2),ANAME(2,2),ANAME(3,2),ANAME(4,2),ANAME(5,2),
     1  ANAME(6,2) /'    ','TRAN','SMIS','. AL','ONG ','ROWS'/
      DATA ANAME(1,3),ANAME(2,3),ANAME(3,3),ANAME(4,3),ANAME(5,3),
     1  ANAME(6,3) /'   H','YD. ','COND','. AL','ONG ','ROWS'/
      DATA ANAME(1,4),ANAME(2,4),ANAME(3,4),ANAME(4,4),ANAME(5,4),
     1  ANAME(6,4) /'VERT',' HYD',' CON','D /T','HICK','NESS'/
      DATA ANAME(1,5),ANAME(2,5),ANAME(3,5),ANAME(4,5),ANAME(5,5),
     1  ANAME(6,5) /'    ','    ','    ','    ','  BO','TTOM'/
      DATA ANAME(1,6),ANAME(2,6),ANAME(3,6),ANAME(4,6),ANAME(5,6),
     1  ANAME(6,6) /'    ','    ','    ','    ','    ',' TOP'/
      DATA ANAME(1,7),ANAME(2,7),ANAME(3,7),ANAME(4,7),ANAME(5,7),
     1  ANAME(6,7) /'  SE','COND','ARY ','STOR','AGE ','COEF'/
      DATA ANAME(1,8),ANAME(2,8),ANAME(3,8),ANAME(4,8),ANAME(5,8),
     1  ANAME(6,8) /'COLU','MN T','O RO','W AN','ISOT','ROPY'/
      DATA ANAME(1,9),ANAME(2,9),ANAME(3,9),ANAME(4,9),ANAME(5,9),
     1  ANAME(6,9) /'    ','    ','    ','    ','    ','DELR'/
      DATA ANAME(1,10),ANAME(2,10),ANAME(3,10),ANAME(4,10),ANAME(5,10),
     1  ANAME(6,10) /'    ','    ','    ','    ','    ','DELC'/
C     ------------------------------------------------------------------
C
C1------CALCULATE NUMBER OF NODES IN A LAYER AND READ TRPY,DELR,DELC
      NIJ=NCOL*NROW
C
      CALL U1DREL(TRPY,ANAME(1,8),NLAY,IN,IOUT)
      CALL U1DREL(DELR,ANAME(1,9),NCOL,IN,IOUT)
      CALL U1DREL(DELC,ANAME(1,10),NROW,IN,IOUT)
C
C2------READ ALL PARAMETERS FOR EACH LAYER
      KT=0
      KB=0
      DO 200 K=1,NLAY
C
C2A-----FIND ADDRESS OF EACH LAYER IN THREE DIMENSION ARRAYS.
      IF(LAYCON(K).EQ.1 .OR. LAYCON(K).EQ.3) KB=KB+1
      IF(LAYCON(K).EQ.2 .OR. LAYCON(K).EQ.3) KT=KT+1
      LOC=1+(K-1)*NIJ
      LOCB=1+(KB-1)*NIJ
      LOCT=1+(KT-1)*NIJ
C
C2B-----READ PRIMARY STORAGE COEFFICIENT INTO ARRAY SC1 IF TRANSIENT
      IF(ISS.EQ.0)CALL U2DREL(SC1(LOC),ANAME(1,1),NROW,NCOL,K,IN,IOUT)
C
C2C-----READ TRANSMISSIVITY INTO ARRAY CC IF LAYER TYPE IS 0 OR 2
      IF(LAYCON(K).EQ.3 .OR. LAYCON(K).EQ.1) GO TO 100
      CALL U2DREL(CC(LOC),ANAME(1,2),NROW,NCOL,K,IN,IOUT)
      GO TO 110
C
C2D-----READ HYDRAULIC CONDUCTIVITY(HY) AND BOTTOM ELEVATION(BOT)
C2D-----IF LAYER TYPE IS 1 OR 3
  100 CALL U2DREL(HY(LOCB),ANAME(1,3),NROW,NCOL,K,IN,IOUT)
      CALL U2DREL(BOT(LOCB),ANAME(1,5),NROW,NCOL,K,IN,IOUT)
C
C2E-----READ VERTICAL HYCOND/THICK INTO ARRAY CV IF NOT BOTTOM LAYER
C2E----- READ AS HYCOND/THICKNESS -- CONVERTED TO CONDUCTANCE LATER
  110 IF(K.EQ.NLAY) GO TO 120
      CALL U2DREL(CV(LOC),ANAME(1,4),NROW,NCOL,K,IN,IOUT)
C
C2F-----READ SECONDARY STORAGE COEFFICIENT INTO ARRAY SC2 IF TRANSIENT
C2F-----AND LAYER TYPE IS 2 OR 3
  120 IF(LAYCON(K).NE.3 .AND. LAYCON(K).NE.2) GO TO 200
      IF(ISS.EQ.0)CALL U2DREL(SC2(LOCT),ANAME(1,7),NROW,NCOL,K,IN,IOUT)
C
C2G-----READ TOP ELEVATION(TOP) IF LAYER TYPE IS 2 OR 3
      CALL U2DREL(TOP(LOCT),ANAME(1,6),NROW,NCOL,K,IN,IOUT)
  200 CONTINUE
C
C3------PREPARE AND CHECK BCF DATA
      CALL SBCF1N(HNEW,IBOUND,SC1,SC2,CR,CC,CV,HY,TRPY,DELR,DELC,ISS,
     1         NCOL,NROW,NLAY,IOUT)
C
C4------RETURN
      RETURN
      END
      SUBROUTINE VELOCITY(HNEW,HOLD,IBOUND,CR,CC,CV,HY,TRPY,
     1                 BOT,TOP,DELR,DELC,DELT,ISS,KSTP,KPER,
     2                 NCOL,NROW,NLAY,IOUT,IVOUT,ZVEL,RBOT,
     3                 LLX,LUX,LLY,LUY,CONV)
C
C     *****************************************************************
```

```
C     CALCULATE VELOCITIES AND WRITE TO FILE ON UNIT IFOUT
C     *****************************************************************
C
C        SPECIFICATIONS:
C     -----------------------------------------------------------------
      DOUBLE PRECISION HNEW
C
      DIMENSION HOLD(NCOL,NROW,NLAY),HNEW(NCOL,NROW,NLAY),
     1    IBOUND(NCOL,NROW,NLAY),CR(NCOL,NROW,NLAY),
     2    CC(NCOL,NROW,NLAY),CV(NCOL,NROW,NLAY),HY(NCOL,NROW,NLAY),
     3    TRPY(NLAY),BOT(NCOL,NROW,NLAY),TOP(NCOL,NROW,NLAY),DELR(NCOL),
     4    DELC(NROW),ZVEL(NCOL,NROW),RBOT(NCOL,NROW)
C
      COMMON /FLWCOM/LAYCON(80)
C     -----------------------------------------------------------------
      KB=0
      KT=0
C
C1------FOR EACH LAYER: CALCULATE HARMONIC MEAN HY
      DO 100 K=1,NLAY
C
C ----------CANNOT USE LAYER TYPE 0 and 2 - INADEQUATE INFO FOR TRANSPORT
C           MODEL
      IF(LAYCON(K) .EQ. 0 .OR. LAYCON(K) .EQ. 2)GOTO 310
C
      IF(LAYCON(K).EQ.3 .OR. LAYCON(K).EQ.2) KT=KT+1
      IF(LAYCON(K).EQ.3 .OR. LAYCON(K).EQ.1) KB=KB+1
C
C---------FILL CC ARRAY WITH HYDRAULIC CONDUCTIVITIES
C         FOR LAYER TYPE 1 AND 3
C
       DO 5 I=1,NROW
       DO 5 J=1,NCOL
         CC(J,I,K)=HY(J,I,KB)
    5 CONTINUE
C
C  -----CALCULATE HARMONIC MEAN HYDRAULIC CONDUCTIVITIES AND
C  -----PUT IN CC AND CR
      CALL SBCFPR(CR,CC,TRPY,K,NCOL,NROW,NLAY)
  100 CONTINUE
C
      KT=0
      DO 200 K=1,NLAY
      IF(LAYCON(K).EQ.3 .OR. LAYCON(K).EQ.2) KT=KT+1
        DO 140 I=LLY,LUY
        DO 140 J=LLX,LUX
C                                IF THE CELL IS EXTERNAL THEN SKIP IT.
         IF(IBOUND(J,I,K).EQ.0 .OR. HNEW(J,I,K) .GE. 1.E30) THEN
               VI=0.0
               VJ=0.0
               VZ=0.0
               THICK=-99.
         ELSE
C                                COMPUTE VELOCITY IN ROW DIRECTION
```

```
          IF(J .EQ. NCOL .OR. IBOUND(J+1,I,K) .EQ. 0)THEN
             VI=0.0
          ELSE
             VI=CR(J,I,K)*(HNEW(J,I,K)-HNEW(J+1,I,K))*2./
     1          (DELR(J+1)+DELR(J))*CONV/7.48
          ENDIF
C                             COMPUTE VELOCITY IN COLUMN DIRECTION
          IF(I .EQ. 1 .OR. IBOUND(J,I-1,K) .EQ. 0)THEN
              VJ=0.0
          ELSE
               VJ=CC(J,I-1,K)*(HNEW(J,I,K)-HNEW(J,I-1,K))*2./
     1            (DELC(I+1)+DELC(I))*CONV/7.48
          ENDIF
C                              COMPUTE VERTICAL VELOCITY
          IF(K .EQ. 1)THEN
              VZ=ZVEL(J,I)/7.48*CONV
              IF(ISS .EQ. 0)VZ=VZ+(HNEW(J,I,K)-HOLD(J,I,K))/DELT
     1           *CONV/7.48
              THICK=HNEW(J,I,K)-BOT(J,I,K)
              IF(RBOT(J,I) .GT. 0) THEN
                 TP=RBOT(J,I)
              ELSE
                 TP=0.0
              ENDIF
          ELSE
              VZ=CV(J,I,K-1)*(HNEW(J,I,K)-HNEW(J,I,K-1))*CONV/7.48
              TP=TOP(J,I,KT)
C
C                          ZERO VERTICAL VELOCITY IF LAYER UNDER WATER TABLE
C                                         CONDITIONS
C              IF(HNEW(J,I,K) .LT. TP)VZ=0.0
C
C                          CODE FOR THICKNESS IN PRICKETT RAND3D
C             IF(K .EQ. 2)THICK =HNEW(J,I,K-1)-(TOP(J,I,KT)+
C    1           BOT(J,I,K))/2
C             IF(K .EQ. 3)THICK=(TOP(J,I,KT-1)+BOT(J,I,K-1))/2
C    1           -BOT(J,I,K)
C
C                          COMPUTE THICKNESS FOR ETA RAND3D
              THICK=TOP(J,I,KT)-BOT(J,I,K)
         ENDIF
        ENDIF
C                               WRITE VELOCITY
          NEWROW=LUY-LLY+1
  120     WRITE(IVOUT,1001)J-LLX+1,NEWROW-I+LLY,NLAY-K+1,THICK,VI,VJ,
     1      VZ,BOT(J,I,K),TP
 1001     FORMAT(3I3,6(1X,G10.3))
  140   CONTINUE
  200 CONTINUE
C                                    WRITE GRID SIZE DATA TO FILE
C      WRITE(IVOUT,1005)(DELR(J),J=1,NCOL)
C 1005 FORMAT(10F8.2)
C      WRITE(IVOUT,1001)(DELC(I),I=1,NROW)
C                                         RETURN
```

```
      RETURN
C
C                       ERROR MESSAGE FOR LAYER TYPE 0 - CANNOT BE USED
  310 WRITE(*,*)' CANNOT USE LAYER TYPE ZERO OR TWO FOR TRANSPORT '
      STOP 0011
      END
      SUBROUTINE SBCFPR(CR,CC,TRPY,K,NCOL,NROW,NLAY)
C
C
C     ******************************************************************
C     COMPUTE BRANCH HYDRAULIC CONDUCTIVITY USING HARMONIC MEAN OF BLOCK
C     HYDRAULIC CONDUCTIVITIES BLOCK HY IS IN CC UPON ENTRY
C     ******************************************************************
C
C      SPECIFICATIONS:
C     ------------------------------------------------------------------
C
      DIMENSION CR(NCOL,NROW,NLAY), CC(NCOL,NROW,NLAY)
     2   , TRPY(NLAY)
C
C     ------------------------------------------------------------------
      YX=TRPY(K)*2.
C
C1------FOR EACH CELL CALCULATE BRANCH HY FROM THAT CELL
C1------TO THE ONE ON THE RIGHT AND THE IN FRONT.
      DO 40 I=1,NROW
      DO 40 J=1,NCOL
      T1=CC(J,I,K)
C
C2------IF T=0 THEN SET CONDUCTANCE EQUAL TO 0. GO ON TO NEXT CELL.
      IF(T1.NE.0.) GO TO 10
      CR(J,I,K)=0.
      GO TO 40
C
C3------IF THIS IS NOT THE LAST COLUMN(RIGHTMOST) THEN CALCULATE
C3------BRANCH HY IN THE ROW DIRECTION (CR) TO THE RIGHT.
   10 IF(J.EQ.NCOL) GO TO 30
      T2=CC(J+1,I,K)
      CR(J,I,K)=2.*T2*T1/(T1+T2)
C
C4------IF THIS IS NOT THE LAST ROW(FRONTMOST) THEN CALCULATE
C4------BRANCH HY IN THE COLUMN DIRECTION (CC) TO THE FRONT.
   30 IF(I.EQ.NROW) GO TO 40
      T2=CC(J,I+1,K)
      CC(J,I,K)=YX*T2*T1/(T1+T2)
   40 CONTINUE
C
C5------RETURN
      RETURN
      END
      SUBROUTINE SBCF1B(HNEW,IBOUND,CR,CC,CV,TOP,NCOL,NROW,NLAY,
     1      KSTP,KPER,IBCFCB,BUFF,IOUT)
C
C-----VERSION 1004 03MAY1983 SBCF1B
```

```
C
C     ******************************************************************
C     COMPUTE FLOW ACROSS EACH CELL WALL
C     ******************************************************************
C
C     SPECIFICATIONS:
C     ------------------------------------------------------------------
      DOUBLE PRECISION HNEW,HD
C
      DIMENSION HNEW(NCOL,NROW,NLAY), IBOUND(NCOL,NROW,NLAY),
     1     CR(NCOL,NROW,NLAY), CC(NCOL,NROW,NLAY),
     2     CV(NCOL,NROW,NLAY), TOP(NCOL,NROW,NLAY),
     3     BUFF(NCOL,NROW,NLAY)
C
      COMMON /FLWCOM/LAYCON(80)
C
CF66
C     DIMENSION TEXT(12)
CF66
CF77
      CHARACTER*4 TEXT(12)
CF77
C
      DATA TEXT(1),TEXT(2),TEXT(3),TEXT(4),TEXT(5),TEXT(6),TEXT(7),
     1   TEXT(8),TEXT(9),TEXT(10),TEXT(11),TEXT(12)
     2   /'FLOW',' RIG','HT F','ACE ',
     2    'FLOW',' FRO','NT F','ACE ','FLOW',' LOW','ER F','ACE '/
C     ------------------------------------------------------------------
C
      NCM1=NCOL-1
      IF(NCM1.LT.1) GO TO 405
C
C1-----CLEAR THE BUFFER
      DO 310 K=1,NLAY
      DO 310 I=1,NROW
      DO 310 J=1,NCOL
      BUFF(J,I,K)=0.
  310 CONTINUE
C
C2-----FOR EACH CELL CALCULATE FLOW THRU RIGHT FACE & STORE IN BUFFER
      DO 400 K=1,NLAY
      DO 400 I=1,NROW
      DO 400 J=1,NCM1
      IF((IBOUND(J,I,K).LE.0) .AND. (IBOUND(J+1,I,K).LE.0)) GO TO 400
      HDIFF=HNEW(J,I,K)-HNEW(J+1,I,K)
      BUFF(J,I,K)=HDIFF*CR(J,I,K)
  400 CONTINUE
C
C3-----RECORD CONTENTS OF BUFFER
      CALL UBUDSV(KSTP,KPER,TEXT(1),IBCFCB,BUFF,NCOL,NROW,NLAY,IOUT)
C
C4-----CLEAR THE BUFFER
  405 NRM1=NROW-1
      IF(NRM1.LT.1) GO TO 505
      DO 410 K=1,NLAY
      DO 410 I=1,NROW
      DO 410 J=1,NCOL
      BUFF(J,I,K)=0.
  410 CONTINUE
C
C5-----FOR EACH CELL CALCULATE FLOW THRU FRONT FACE & STORE IN BUFFER
      DO 500 K=1,NLAY
      DO 500 I=1,NRM1
      DO 500 J=1,NCOL
      IF((IBOUND(J,I,K).LE.0) .AND. (IBOUND(J,I+1,K).LE.0)) GO TO 500
      HDIFF=HNEW(J,I,K)-HNEW(J,I+1,K)
      BUFF(J,I,K)=HDIFF*CC(J,I,K)
  500 CONTINUE
C
C6-----RECORD CONTENTS OF BUFFER.
      CALL UBUDSV(KSTP,KPER,TEXT(5),IBCFCB,BUFF,NCOL,NROW,NLAY,IOUT)
  505 NLM1=NLAY-1
      IF(NLM1.LT.1) GO TO 1000
C
C7-----CLEAR THE BUFFER
      DO 510 K=1,NLAY
      DO 510 I=1,NROW
      DO 510 J=1,NCOL
      BUFF(J,I,K)=0.
  510 CONTINUE
C
C8-----FOR EACH CELL CALCULATE FLOW THRU LOWER FACE & STORE IN BUFFER
      KT=0
      DO 600 K=1,NLM1
      IF(LAYCON(K).EQ.3 .OR. LAYCON(K).EQ.2) KT=KT+1
      DO 600 I=1,NROW
      DO 600 J=1,NCOL
      IF((IBOUND(J,I,K).LE.0) .AND. (IBOUND(J,I,K+1).LE.0)) GO TO 600
      HD=HNEW(J,I,K+1)
      IF(LAYCON(K+1).NE.3 .AND. LAYCON(K+1).NE.2) GO TO 580
      TMP=HD
      IF(TMP.LT.TOP(J,I,KT+1)) HD=TOP(J,I,KT+1)
  580 HDIFF=HNEW(J,I,K)-HD
      BUFF(J,I,K)=HDIFF*CV(J,I,K)
  600 CONTINUE
C
C9-----RECORD CONTENTS OF BUFFER.
      CALL UBUDSV(KSTP,KPER,TEXT(9),IBCFCB,BUFF,NCOL,NROW,NLAY,IOUT)
C
C10----RETURN
 1000 RETURN
      END
      SUBROUTINE SBCF1F(HNEW,IBOUND,CR,CC,CV,DELR,DELC,
     1   TOP,DELT,NCOL,NROW,NLAY,KSTP,KPER,IBD,IBCFCB,ICBCFL,
     2   BUFF,IVOUT,CONVRT)
C-----VERSION 1123 29MAY1983 SBCF1F
C
C     ******************************************************************
```

```
C     COMPUTE FLOW FROM CONSTANT HEAD NODES
C     ******************************************************************
C
C     SPECIFICATIONS:
C     ------------------------------------------------------------------
      DOUBLE PRECISION HNEW,HD
C
      DIMENSION HNEW(NCOL,NROW,NLAY), IBOUND(NCOL,NROW,NLAY),
     1     CR(NCOL,NROW,NLAY), CC(NCOL,NROW,NLAY),
     2     CV(NCOL,NROW,NLAY), DELR(NCOL),DELC(NROW),
     3     TOP(NCOL,NROW,NLAY),BUFF(NCOL,NROW,NLAY)
C
      COMMON /FLWCOM/LAYCON(80)
C
CF66
C     DIMENSION TEXT(4),VBNM(4,20)
CF66
CF77
      CHARACTER*4 TEXT(4),VBNM(4,20)
CF77
C
      DATA TEXT(1),TEXT(2),TEXT(3),TEXT(4) /'   C','ONST','ANT ','HEAD'/
C     ------------------------------------------------------------------
C
C1------CLEAR BUDGET ACCUMULATORS
      CHIN=0.
      CHOUT=0.
C
C2------CLEAR BUFFER IF CELL-BY-CELL FLOW TERM FLAG(IBD) IS SET
      IF(IBD.EQ.0) GO TO 8
      DO 5 K=1,NLAY
      DO 5 I=1,NROW
      DO 5 J=1,NCOL
      BUFF(J,I,K)=0.
    5 CONTINUE
C
C3------FOR EACH CELL IF IT IS CONSTANT HEAD COMPUTE FLOW ACROSS 6
C3-----FACES.
    8 KT=0
      DO 200 K=1,NLAY
      LC=LAYCON(K)
      IF(LC.EQ.3 .OR. LC.EQ.2) KT=KT+1
      DO 200 I=1,NROW
      DO 200 J=1,NCOL
C
C4-----IF CELL IS NOT CONSTANT HEAD SKIP IT & GO ON TO NEXT CELL.
      IF (IBOUND(J,I,K).GE.0)GO TO 200
C
C5-----CLEAR FIELDS FOR SIX FLOW RATES.
      X1=0.
      X2=0.
      X3=0.
      X4=0.
      X5=0.
      X6=0.
C6-----FOR EACH FACE OF THE CELL CALCULATE FLOW THROUGH THAT FACE
C6-----OUT OF THE CONSTANT HEAD CELL AND INTO THE FLOW DOMAIN.
C6-----COMMENTS 7-11 APPEAR ONLY IN THE SECTION HEADED BY COMMENT 6A
C6-----BUT THEY APPLY IN A SIMILAR MANNER TO THE SECTIONS HEADED
C6-----BY COMMENTS 6B-6F.
C
C6A----CALCULATE FLOW THROUGH THE LEFT FACE
C
C7-----IF THERE IS NOT A VARIABLE HEAD CELL ON THE OTHER SIDE OF THIS
C7-----FACE THEN GO ON TO THE NEXT FACE.
      IF(J.EQ.1) GO TO 30
      IF(IBOUND(J-1,I,K).LE.0)GO TO 30
      HDIFF=HNEW(J,I,K)-HNEW(J-1,I,K)
C
C8-----CALCULATE FLOW THROUGH THIS FACE INTO THE ADJACENT CELL.
      X1=HDIFF*CR(J-1,I,K)
C
C9-----TEST TO SEE IF FLOW IS POSITIVE OR NEGATIVE
      IF (X1) 10,30,20
C
C10----IF NEGATIVE ADD TO CHOUT(FLOW OUT OF DOMAIN TO CONSTANT HEAD).
   10 CHOUT=CHOUT-X1
      GO TO 30
C
C11----IF POSITIVE ADD TO CHIN(FLOW INTO DOMAIN FROM CONSTANT HEAD).
   20 CHIN=CHIN+X1
C
C6B----CALCULATE FLOW THROUGH THE RIGHT FACE
   30 IF(J.EQ.NCOL) GO TO 60
      IF(IBOUND(J+1,I,K).LE.0) GO TO 60
      HDIFF=HNEW(J,I,K)-HNEW(J+1,I,K)
      X2=HDIFF*CR(J,I,K)
      IF(X2)40,60,50
   40 CHOUT=CHOUT-X2
      GO TO 60
   50 CHIN=CHIN+X2
C
C6C----CALCULATE FLOW THROUGH THE BACK FACE.
   60 IF(I.EQ.1) GO TO 90
      IF (IBOUND(J,I-1,K).LE.0) GO TO 90
      HDIFF=HNEW(J,I,K)-HNEW(J,I-1,K)
      X3=HDIFF*CC(J,I-1,K)
      IF(X3) 70,90,80
   70 CHOUT=CHOUT-X3
      GO TO 90
   80 CHIN=CHIN+X3
C
C6D----CALCULATE FLOW THROUGH THE FRONT FACE.
   90 IF(I.EQ.NROW) GO TO 120
      IF(IBOUND(J,I+1,K).LE.0) GO TO 120
      HDIFF=HNEW(J,I,K)-HNEW(J,I+1,K)
      X4=HDIFF*CC(J,I,K)
      IF (X4) 100,120,110
```

```
  100 CHOUT=CHOUT-X4
      GO TO 120
  110 CHIN=CHIN+X4
C
C6E----CALCULATE FLOW THROUGH THE UPPER FACE
  120 IF(K.EQ.1) GO TO 150
      IF (IBOUND(J,I,K-1).LE.0) GO TO 150
      HD=HNEW(J,I,K)
      IF(LC.NE.3 .AND. LC.NE.2) GO TO 122
      TMP=HD
      IF(TMP.LT.TOP(J,I,KT)) HD=TOP(J,I,KT)
  122 HDIFF=HD-HNEW(J,I,K-1)
      X5=HDIFF*CV(J,I,K-1)
      IF(X5) 130,150,140
  130 CHOUT=CHOUT-X5
      GO TO 150
  140 CHIN=CHIN+X5
C
C6F----CALCULATE FLOW THROUGH THE LOWER FACE.
  150 IF(K.EQ.NLAY) GO TO 180
      IF(IBOUND(J,I,K+1).LE.0) GO TO 180
      HD=HNEW(J,I,K+1)
      IF(LAYCON(K+1).NE.3 .AND. LAYCON(K+1).NE.2) GO TO 152
      TMP=HD
      IF(TMP.LT.TOP(J,I,KT+1)) HD=TOP(J,I,KT+1)
  152 HDIFF=HNEW(J,I,K)-HD
      X6=HDIFF*CV(J,I,K)
      IF(X6) 160,180,170
  160 CHOUT=CHOUT-X6
      GO TO 180
  170 CHIN=CHIN+X6
C
C12-----SUM UP FLOWS THROUGH SIX SIDES OF CONSTANT HEAD CELL.
 180  RATE=X1+X2+X3+X4+X5+X6
C
C13-----PRINT THE INDIVIDUAL RATES IF REQUESTED(IBCFCB<0).
      IF(IBCFCB.LT.0.AND.ICBCFL.NE.0) WRITE(IOUT,900) (TEXT(N),N=1,4),
     1    KPER,KSTP,K,I,J,RATE
  900 FORMAT(1H0,4A4,'   PERIOD',I3,'   STEP',I3,'   LAYER',I3,
     1    '   ROW',I4,'   COL',I4,'   RATE ',G15.7)
C
C                       CALCULATE POSITION OF SINK
      CALL XYZPOS(J,I,K,X,Y,Z,DELR,DELC,NROW,NCOL,NLAYER)
      IF(RATE .LT. 0)WRITE(IVOUT,1001)X,Y,Z,RATE*CONVRT
 1001 FORMAT(3I3,F12.4)
  200 CONTINUE
C17----RETURN
      RETURN
      END
      SUBROUTINE HYCALC(CC,HY,TRPY,IBOUND
     1,BOT,TOP,K,KB,KT,NCOL,NROW,NLAY)
C
C     ******************************************************************
C     COMPUTE HYDRAULIC CONDUCTIVITY FROM TRANSMISSIVITY AND THICKNESS
```

```
C     *****************************************************************
C
C      SPECIFICATIONS:
C     -----------------------------------------------------------------
      DOUBLE PRECISION HNEW
C
      DIMENSION CC(NCOL,NROW,NLAY), HY(NCOL,NROW,NLAY),
     1 TRPY(NLAY),IBOUND(NCOL,NROW,NLAY),
     2  BOT(NCOL,NROW,NLAY),TOP(NCOL,NROW,NLAY)
C
      COMMON /FLWCOM/LAYCON(80)
C     -----------------------------------------------------------------
C
C1------CALCULATE TRANSMISSIVITY AT EACH ACTIVE CELL. TRANSMISSIVITY
C1------WILL BE STORED TEMPORARILY IN THE CC ARRAY.
      DO 200 I=1,NROW
      DO 200 J=1,NCOL
C
C3------CALCULATE THICKNESS.
      THCK=TOP(J,I,KT)-BOT(J,I,KB)

C4------CHECK TO SEE IF THICKNESS IS GREATER THAN ZERO.
      IF(THCK.LE.0.) GO TO 200
C
C5------IF THICKNESS>0 THEN K=T/THICKNESS.
      CC(J,I,K)=CC(J,I,K)/THCK
      GO TO 200
C
C6------WHEN THICKNESS < 0, PRINT A MESSAGE AND SET
C6------TRANSMISSIVITY,  AND VERTICAL CONDUCTANCE =0
C 100 WRITE(IOUT,150) J,I,K
C 150 FORMAT(1H0,10('*'),'NODE',3I4,' (COL,ROW,LAYER) IS DRY')
C     HNEW(J,I,K)=1.E30
C     CC(J,I,K)=0.
C     IBOUND(J,I,K)=0
C     IF(K.LT.NLAY) CV(J,I,K)=0.
C     IF(K.GT.1) CV(J,I,K-1)=0.
  200 CONTINUE
C
C8------RETURN
      RETURN
      END
      SUBROUTINE SBCF1N(HNEW,IBOUND,SC1,SC2,CR,CC,CV,HY,TRPY,DELR,DELC,
     1         ISS,NCOL,NROW,NLAY,IOUT)
C
C-----VERSION 1007 03MAY1983 SBCF1N
C
C     *****************************************************************
C     INITIALIZE AND CHECK BCF DATA
C     *****************************************************************
C
C        SPECIFICATIONS:
C     -----------------------------------------------------------------
C
```

```
      DOUBLE PRECISION HNEW,HCNV
C
      DIMENSION HNEW(NCOL,NROW,NLAY),IBOUND(NCOL,NROW,NLAY)
     1    ,SC1(NCOL,NROW,NLAY),CR(NCOL,NROW,NLAY)
     2    ,CC(NCOL,NROW,NLAY),CV(NCOL,NROW,NLAY)
     3    ,HY(NCOL,NROW,NLAY),TRPY(NLAY),DELR(NCOL),DELC(NROW)
     4    ,SC2(NCOL,NROW,NLAY)
C
      COMMON /FLWCOM/LAYCON(80)
C     ------------------------------------------------------------------
C
C1------IF IBOUND=0, SET CV=0., CC=0., AND HY=0.
      KB=0
      DO 30 K=1,NLAY
      IF(LAYCON(K).EQ.3 .OR. LAYCON(K).EQ.1) KB=KB+1
      DO 30 I=1,NROW
      DO 30 J=1,NCOL
      IF(IBOUND(J,I,K).NE.0) GO TO 30
      IF(K.NE.NLAY) CV(J,I,K)=0.
      IF(K.NE.1) CV(J,I,K-1)=0.
      CC(J,I,K)=0.
      IF(LAYCON(K).EQ.3 .OR. LAYCON(K).EQ.1) HY(J,I,KB)=0.
   30 CONTINUE
C
C2------INSURE THAT EACH ACTIVE CELL HAS AT LEAST ONE NON-ZERO
C2------TRANSMISSIVE PARAMETER. IF NOT, CONVERT CELL TO NOFLOW.
      HCNV=888.88
      KB=0
      DO 60 K=1,NLAY
      IF(LAYCON(K).EQ.1 .OR. LAYCON(K).EQ.3) GO TO 55
C2A-----WHEN LAYER TYPE 0 OR 2, TRANSMISSIVITY OR CV MUST BE NONZERO
      DO 54 I=1,NROW
      DO 54 J=1,NCOL
      IF(IBOUND(J,I,K).EQ.0) GO TO 54
      IF(CC(J,I,K).NE.0.) GO TO 54
      IF(K.EQ.NLAY) GO TO 51
      IF(CV(J,I,K).NE.0.) GO TO 54
   51 IF(K.EQ.1) GO TO 53
      IF(CV(J,I,K-1).NE.0.) GO TO 54
   53 IBOUND(J,I,K)=0
      HNEW(J,I,K)=HCNV
      WRITE(IOUT,52) K,I,J
   52 FORMAT(1X,'NODE (LAYER,ROW,COL)',3I4,
     1      ' ELIMINATED BECAUSE ALL CONDUCTANCES TO NODE ARE 0')
   54 CONTINUE
      GO TO 60
C
C2B-----WHEN LAYER TYPE IS 1 OR 3, HY OR CV MUST BE NONZERO
   55 KB=KB+1
      DO 59 I=1,NROW
      DO 59 J=1,NCOL
      IF(IBOUND(J,I,K).EQ.0) GO TO 59
      IF(HY(J,I,KB).NE.0.) GO TO 59
      IF(K.EQ.NLAY) GO TO 56
      IF(CV(J,I,K).NE.0.) GO TO 59
   56 IF(K.EQ.1) GO TO 57
      IF(CV(J,I,K-1).NE.0.) GO TO 59
   57 IBOUND(J,I,K)=0
      HNEW(J,I,K)=HCNV
      CC(J,I,K)=0.
      WRITE(IOUT,52) K,I,J
   59 CONTINUE
   60 CONTINUE
C
C3------CALCULATE HOR. CONDUCTANCE(CR AND CC) FOR CONSTANT T LAYERS
      DO 65 K=1,NLAY
      IF(LAYCON(K).EQ.3 .OR. LAYCON(K).EQ.1) GO TO 65
      WRITE(*,*)'  CANNOT USE LAYER TYPE 0 OR 2 FOR VELOCITY CALC'
      STOP 11111
      CALL SBCF1C(CR,CC,TRPY,DELR,DELC,K,NCOL,NROW,NLAY)
   65 CONTINUE
C
C4------MULTIPLY VERTICAL LEAKANCE BY AREA TO MAKE CONDUCTANCE
C       MODIFIED BY DON KOCH, DEC 1988 FOR PREMOD3D USE
      IF(NLAY.EQ.1) GO TO 69
      K1=NLAY-1
      DO 68 K=1,K1
      DO 68 I=1,NROW
      DO 68 J=1,NCOL
      CV(J,I,K)=CV(J,I,K)
   68 CONTINUE
C
C5------IF TRANSIENT MULTIPLY PRIMARY STORAGE FACTOR BY DELR &
C5------DELC TO GET PRIMARY STORAGE CAPACITY(SC1).
   69 IF(ISS.NE.0) GO TO 100
      KT=0
      DO 80 K=1,NLAY
      DO 70 I=1,NROW
      DO 70 J=1,NCOL
      SC1(J,I,K)=SC1(J,I,K)*DELR(J)*DELC(I)
   70 CONTINUE
C
C6------IF LAYER IS CONF/UNCONF MULTIPLY SECONDARY STORAGE FACTOR
C6------BY DELR AND DELC TO GET SECONDARY STORAGE CAPACITY(SC2).
      IF(LAYCON(K).NE.3 .AND. LAYCON(K).NE.2) GO TO 80
      KT=KT+1
      DO 75 I=1,NROW
      DO 75 J=1,NCOL
      SC2(J,I,KT)=SC2(J,I,KT)*DELR(J)*DELC(I)
   75 CONTINUE
C
   80 CONTINUE
C
C7------RETURN
  100 RETURN
      END
```

```
$DEBUG
$LARGE
$NOFLOATCALLS
      SUBROUTINE BASPRE(HNEW,BUFF,IOFLG,KSTP,KPER,NCOL,NROW,
     1    NLAY,IOUT,IHEDFM,IHEDUN,IPFLG,PERTIM,TOTIM)
C
C     ******************************************************
C     READ PREVIOUSLY SAVED HEADS FROM BINARY HEAD SAVE FILE
C     ******************************************************
C
C         MODIFED FOR PREMOD3D BY DON KOCH, ENGINEERING TECHNOLOGIES
C           ASSOCIATES FROM USGS MODFLOW - SBAS1H SUBROUTINE
C
C     K1-STEP # IN INPUT FILEC
C     K2-PERIOD # IN INPUT FILE
C     IHEDUN- UNIT # TO READ INPUT FROM
C
C
C        SPECIFICATIONS
C     ------------------------------------------------------
      DOUBLE PRECISION HNEW
C
      DIMENSION HNEW(NCOL,NROW,NLAY),IOFLG(NLAY,4),
     1     BUFF(NCOL,NROW,NLAY)
CF77
      CHARACTER*4 TEXT(4)
CF77
C
      DATA TEXT(1),TEXT(2),TEXT(3),TEXT(4) /'    ','    ','    ',
     1      'HEAD'/
C     ------------------------------------------------------
C
C1------FOR EACH LAYER: READ HEAD
      DO 49 K=1,NLAY
C
C2-----READ HEAD FOR EACH LAYER FROM UNIT=IHEDUN
      CALL ULAREAD(BUFF,TEXT(1),K1,K2,PERTIM,TOTIM,NCOL,NROW,ILAYER,
     1           IHEDUN)
C
C3-----CHECK THAT PERIOD, STEP AND TIME MATCH
      IF(K1 .NE. KSTP)GOTO 55
      IF(K2 .NE. KPER)GOTO 55
      IF(K .NE. ILAYER)GOTO 55
C
C4------COPY HEADS FOR BUFFER INTO  THIS LAYER .
      DO 44 I=1,NROW
      DO 44 J=1,NCOL
      HNEW(J,I,K)=BUFF(J,I,1)
   44 CONTINUE
C
   49 CONTINUE
C5------FOR EACH LAYER: PRINT HEAD IF REQUESTED.
      DO 39 K=1,NLAY
C
```

```
C6------TEST IOFLG TO SEE IF HEAD SHOULD BE PRINTED.
      IF(IOFLG(K,1).EQ.0) GO TO 39
      IPFLG=1
C
C7------COPY HEADS FOR THIS LAYER INTO BUFFER.
      DO 32 I=1,NROW
      DO 32 J=1,NCOL
      BUFF(J,I,1)=HNEW(J,I,K)
   32 CONTINUE
C
C8------CALL UTILITY MODULE TO PRINT CONTENTS OF BUFFER.
      IF(IHEDFM.LT.0) CALL ULAPRS(BUFF,TEXT(1),KSTP,KPER,NCOL,NROW,K,
     1          -IHEDFM,IOUT)
      IF(IHEDFM.GE.0) CALL ULAPRW(BUFF,TEXT(1),KSTP,KPER,NCOL,NROW,K,
     1            IHEDFM,IOUT)
   39 CONTINUE
C
C10-----RETURN
   50 RETURN
C
C11------- PRINT ERROR MESSAGE IF NOT READING CORRECT DATA
   55 WRITE(*,*)'   NOT READING CORRECT DATA IN FILE '
      STOP 0001
      END
      SUBROUTINE ULAREAD(BUF,TEXT,KSTP,KPER,PERTIM,TOTIM,NCOL,
     1                  NROW,ILAY,ICHN)
C
C     *****************************************************************
C     READ 1 LAYER ARRAY FROM BINARY FILE
C     *****************************************************************
C
C        SPECIFICATIONS:
C     -----------------------------------------------------------------
      DIMENSION BUF(NCOL,NROW)
CF77
      CHARACTER*4 TEXT(4)
CF77
C     -----------------------------------------------------------------
C
C1------READ AN UNFORMATTED RECORD CONTAINING IDENTIFYING
C1------INFORMATION.
      READ(ICHN) KSTP,KPER,PERTIM,TOTIM,TEXT,NCOL,NROW,ILAY
C
C2------READ AN UNFORMATTED RECORD CONTAINING ARRAY VALUES
C2------THE ARRAY IS DIMENSIONED (NCOL,NROW)
      READ(ICHN) ((BUF(IC,IR),IC=1,NCOL),IR=1,NROW)
C
C3------RETURN
      RETURN
      END
      SUBROUTINE UBUDSV(KSTP,KPER,TEXT,IBDCHN,BUFF,NCOL,NROW,NLAY,IOUT)
C
C
C-----VERSION 1305 28DEC1983 UBUDSV
```

```
C     ******************************************************************
C     RECORD CELL-BY-CELL FLOW TERMS FOR ONE COMPONENT OF FLOW.
C     ******************************************************************
C
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
      DIMENSION BUFF(NCOL,NROW,NLAY)
CF66
C     DIMENSION TEXT(4)
CF66
CF77
      CHARACTER*4 TEXT(4)
CF77
C     ------------------------------------------------------------------
C
C1------WRITE AN UNFORMATTED RECORD CONTAINING IDENTIFYING
C1------INFORMATION.
      WRITE(IOUT,1) TEXT,IBDCHN,KSTP,KPER
    1 FORMAT(1X,'"',4A4,'" BUDGET VALUES WILL BE SAVED ON UNIT',I3,
     1     ' AT END OF TIME STEP',I3,', STRESS PERIOD',I3)
C
      WRITE(IBDCHN) KSTP,KPER,TEXT,NCOL,NROW,NLAY
C
C2------WRITE AN UNFORMATTED RECORD CONTAINING VALUES FOR
C2------EACH CELL IN THE GRID. THE ARRAY IS DIMENSIONED
C2------(NCOL,NROW,NLAY)
      WRITE(IBDCHN) BUFF
C
C3------RETURN
      RETURN
      END
      SUBROUTINE UCOLNO(NLBL1,NLBL2,NSPACE,NCPL,NDIG,IOUT)
C
C
C-----VERSION 1446 20APR1983 UCOLNO
C     ******************************************************************
C     OUTPUT COLUMN NUMBERS ABOVE A MATRIX PRINTOUT
C        NLBL1 IS THE START COLUMN LABEL (NUMBER)
C        NLBL2 IS THE STOP COLUMN LABEL (NUMBER)
C        NSPACE IS NUMBER OF BLANK SPACES TO LEAVE AT START OF LINE
C        NCPL IS NUMBER OF COLUMN NUMBERS PER LINE
C        NDIG IS NUMBER OF CHARACTERS IN EACH COLUMN FIELD
C        IOUT IS OUTPUT CHANNEL
C     ******************************************************************
C
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
CF66
C     DIMENSION BF(130),DG(10)
CF66
CF77
      CHARACTER*4 BF(130),DG(10),DOT,SPACE
CF77
C
```

```
      DATA DG(1),DG(2),DG(3),DG(4),DG(5),DG(6),DG(7),DG(8),DG(9),DG(10)/
     1         '0   ','1   ','2   ','3   ','4   ','5   ','6   ',
     2         '7   ','8   ','9   '/
      DATA DOT,SPACE/'.   ','    '/
C     -----------------------------------------------------------------
C
C1------CALCULATE # OF COLUMNS TO BE PRINTED (NLBL), WIDTH
C1------OF A LINE (NTOT), NUMBER OF LINES (NWRAP).
      WRITE(IOUT,1)
    1 FORMAT(1X)
      NLBL=NLBL2-NLBL1+1
      N=NLBL
      IF(NLBL.GT.NCPL) N=NCPL
      NTOT=NSPACE+N*NDIG
      IF(NTOT.GT.130) GO TO 50
      NWRAP=(NLBL-1)/NCPL + 1
      J1=NLBL1-NCPL
      J2=NLBL1-1
C
C2------BUILD AND PRINT EACH LINE
      DO 40 N=1,NWRAP
C
C3------CLEAR THE BUFFER (BF).
      DO 20 I=1,130
      BF(I)=SPACE
   20 CONTINUE
      NBF=NSPACE
C
C4------DETERMINE FIRST (J1) AND LAST (J2) COLUMN # FOR THIS LINE.
      J1=J1+NCPL
      J2=J2+NCPL
      IF(J2.GT.NLBL2) J2=NLBL2
C5------LOAD THE COLUMN #'S INTO THE BUFFER.
      DO 30 J=J1,J2
      NBF=NBF+NDIG
      I2=J/10
      I1=J-I2*10+1
      BF(NBF)=DG(I1)
      IF(I2.EQ.0) GO TO 30
      I3=I2/10
      I2=I2-I3*10+1
      BF(NBF-1)=DG(I2)
      IF(I3.EQ.0) GO TO 30
      BF(NBF-2)=DG(I3+1)
   30 CONTINUE
C
C6------PRINT THE CONTENTS OF THE BUFFER (I.E. PRINT THE LINE).
      WRITE(IOUT,31) (BF(I),I=1,NBF)
   31 FORMAT(1X,130A1)
C
   40 CONTINUE
C
C7------PRINT A LINE OF DOTS (FOR ESTHETIC PURPOSES ONLY).
   50 NTOT=NTOT+5
```

```
      IF(NTOT.GT.130) NTOT=130
      WRITE(IOUT,51) (DOT,I=1,NTOT)
   51 FORMAT(1X,130A1)
C
C8------RETURN
      RETURN
      END
      SUBROUTINE ULAPRS(BUF,TEXT,KSTP,KPER,NCOL,NROW,ILAY,IPRN,IOUT)
C
C
C-----VERSION 1448 20APR1983 ULAPRS
C     ******************************************************************
C     PRINT A 1 LAYER ARRAY IN STRIPS
C     ******************************************************************
C
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
      DIMENSION BUF(NCOL,NROW)
CF66
C     DIMENSION TEXT(4)
CF66
CF77
      CHARACTER*4 TEXT(4)
CF77
C     ------------------------------------------------------------------
C
C1------MAKE SURE THE FORMAT CODE (IP OR IPRN) IS BETWEEN 1
C1------AND 12.
      IP=IPRN
      IF(IP.LT.1 .OR. IP.GT.12) IP=12
C
C2------DETERMINE THE NUMBER OF VALUES (NCAP) PRINTED ON ONE LINE.
      IF(IP.EQ.1) NCAP=11
      IF(IP.EQ.2) NCAP=9
      IF(IP.GT.2 .AND. IP.LT.7) NCAP=15
      IF(IP.GT.6 .AND. IP.LT.12) NCAP=20
      IF(IP.EQ.12) NCAP=10
C
C3------CALCULATE THE NUMBER OF STRIPS (NSTRIP).
      NCPF=129/NCAP
      ISP=0
      IF(NCAP.GT.12) ISP=3
      NSTRIP=(NCOL-1)/NCAP + 1
      J1=1-NCAP
      J2=0
C
C4------LOOP THROUGH THE STRIPS.
      DO 2000 N=1,NSTRIP
C
C5------CALCULATE THE FIRST(J1) & THE LAST(J2) COLUMNS FOR THIS STRIP
      J1=J1+NCAP
      J2=J2+NCAP
      IF(J2.GT.NCOL) J2=NCOL
C
```

```
C6-------PRINT TITLE ON EACH STRIP
      WRITE(IOUT,1) TEXT,ILAY,KSTP,KPER
    1 FORMAT(1H1,10X,4A4,' IN LAYER',I3,' AT END OF TIME STEP',I3,
     1     ' IN STRESS PERIOD',I3/11X,71('-'))
C
C7------PRINT COLUMN NUMBERS ABOVE THE STRIP
      CALL UCOLNO(J1,J2,ISP,NCAP,NCPF,IOUT)
C
C8------LOOP THROUGH THE ROWS PRINTING COLS J1 THRU J2 WITH FORMAT IP
      DO 1000 I=1,NROW
      GO TO(10,20,30,40,50,60,70,80,90,100,110,120), IP
C
C------------FORMAT 10G10.3
   10 WRITE(IOUT,11) I,(BUF(J,I),J=J1,J2)
   11 FORMAT(1H0,I3,2X,1PG10.3,10(1X,G10.3))
      GO TO 1000
C
C------------FORMAT 8G13.6
   20 WRITE(IOUT,21) I,(BUF(J,I),J=J1,J2)
   21 FORMAT(1H0,I3,2X,1PG13.6,8(1X,G13.6))
      GO TO 1000
C
C------------FORMAT 15F7.1
   30 WRITE(IOUT,31) I,(BUF(J,I),J=J1,J2)
   31 FORMAT(1H0,I3,1X,15(1X,F7.1))
      GO TO 1000
C
C------------FORMAT 15F7.2
   40 WRITE(IOUT,41) I,(BUF(J,I),J=J1,J2)
   41 FORMAT(1H0,I3,1X,15(1X,F7.2))
      GO TO 1000
C
C------------FORMAT 15F7.3
   50 WRITE(IOUT,51) I,(BUF(J,I),J=J1,J2)
   51 FORMAT(1H0,I3,1X,15(1X,F7.3))
      GO TO 1000
C
C------------FORMAT 15F7.4
   60 WRITE(IOUT,61) I,(BUF(J,I),J=J1,J2)
   61 FORMAT(1H0,I3,1X,15(1X,F7.4))
      GO TO 1000
C
C------------FORMAT 20F5.0
   70 WRITE(IOUT,71) I,(BUF(J,I),J=J1,J2)
   71 FORMAT(1H0,I3,1X,20(1X,F5.0))
      GO TO 1000
C
C------------FORMAT 20F5.1
   80 WRITE(IOUT,81) I,(BUF(J,I),J=J1,J2)
   81 FORMAT(1H0,I3,1X,20(1X,F5.1))
      GO TO 1000
C
C------------FORMAT 20F5.2
   90 WRITE(IOUT,91) I,(BUF(J,I),J=J1,J2)
   91 FORMAT(1H0,I3,1X,20(1X,F5.2))
      GO TO 1000
C
C------------FORMAT 20F5.3
  100 WRITE(IOUT,101) I,(BUF(J,I),J=J1,J2)
  101 FORMAT(1H0,I3,1X,20(1X,F5.3))
      GO TO 1000
C
C------------FORMAT 20F5.4
  110 WRITE(IOUT,111) I,(BUF(J,I),J=J1,J2)
  111 FORMAT(1H0,I3,1X,20(1X,F5.4))
      GO TO 1000
C
C------------FORMAT 9G11.4
  120 WRITE(IOUT,121) I,(BUF(J,I),J=J1,J2)
  121 FORMAT(1H0,I3,2X,1PG11.4,9(1X,G11.4))
C
 1000 CONTINUE
 2000 CONTINUE
C
C9------RETURN
      RETURN
      END
      SUBROUTINE ULAPRW(BUF,TEXT,KSTP,KPER,NCOL,NROW,ILAY,IPRN,IOUT)
C
C
C-----VERSION 1245 04MAY1983 ULAPRW
C     ******************************************************************
C     PRINT 1 LAYER ARRAY
C     ******************************************************************
C
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
      DIMENSION BUF(NCOL,NROW)
CF66
C     DIMENSION TEXT(4)
CF66
CF77
      CHARACTER*4 TEXT(4)
CF77
C     ------------------------------------------------------------------
C
C1------PRINT A HEADER
      IF(ILAY.LE.0) GO TO 5
      WRITE(IOUT,1) TEXT,ILAY,KSTP,KPER
    1 FORMAT(1H1,10X,4A4,' IN LAYER',I3,' AT END OF TIME STEP',I3,
     1     ' IN STRESS PERIOD',I3/11X,71('-'))
C
C2------MAKE SURE THE FORMAT CODE (IP OR IPRN) IS
C2------BETWEEN 1 AND 12.
    5 IP=IPRN
      IF(IP.LT.1 .OR. IP.GT.12) IP=12
C
C3------CALL THE UTILITY MODULE UCOLNO TO PRINT COLUMN NUMBERS.
```

```
      IF(IP.EQ.1) CALL UCOLNO(1,NCOL,0,11,11,IOUT)
      IF(IP.EQ.2) CALL UCOLNO(1,NCOL,0,9,14,IOUT)
      IF(IP.GT.2 .AND. IP.LT.7) CALL UCOLNO(1,NCOL,3,15,8,IOUT)
      IF(IP.GT.6 .AND. IP.LT.12) CALL UCOLNO(1,NCOL,3,20,6,IOUT)
      IF(IP.EQ.12) CALL UCOLNO(1,NCOL,0,10,12,IOUT)
C
C4------LOOP THROUGH THE ROWS PRINTING EACH ONE IN ITS ENTIRETY.
      DO 1000 I=1,NROW
      GO TO(10,20,30,40,50,60,70,80,90,100,110,120), IP
C
C------------ FORMAT 11G10.3
   10 WRITE(IOUT,11) I,(BUF(J,I),J=1,NCOL)
   11 FORMAT(1H0,I3,2X,1PG10.3,10(1X,G10.3)/(5X,11(1X,G10.3)))
      GO TO 1000
C
C------------ FORMAT 9G13.6
   20 WRITE(IOUT,21) I,(BUF(J,I),J=1,NCOL)
   21 FORMAT(1H0,I3,2X,1PG13.6,8(1X,G13.6)/(5X,9(1X,G13.6)))
      GO TO 1000
C
C------------ FORMAT 15F7.1
   30 WRITE(IOUT,31) I,(BUF(J,I),J=1,NCOL)
   31 FORMAT(1H0,I3,1X,15(1X,F7.1)/(5X,15(1X,F7.1)))
      GO TO 1000
C
C------------ FORMAT 15F7.2
   40 WRITE(IOUT,41) I,(BUF(J,I),J=1,NCOL)
   41 FORMAT(1H0,I3,1X,15(1X,F7.2)/(5X,15(1X,F7.2)))
      GO TO 1000
C
C------------ FORMAT 15F7.3
   50 WRITE(IOUT,51) I,(BUF(J,I),J=1,NCOL)
   51 FORMAT(1H0,I3,1X,15(1X,F7.3)/(5X,15(1X,F7.3)))
      GO TO 1000
C
C------------ FORMAT 15F7.4
   60 WRITE(IOUT,61) I,(BUF(J,I),J=1,NCOL)
   61 FORMAT(1H0,I3,1X,15(1X,F7.4)/(5X,15(1X,F7.4)))
      GO TO 1000
C
C------------ FORMAT 20F5.0
   70 WRITE(IOUT,71) I,(BUF(J,I),J=1,NCOL)
   71 FORMAT(1H0,I3,1X,20(1X,F5.0)/(5X,20(1X,F5.0)))
      GO TO 1000
C
C------------ FORMAT 20F5.1
   80 WRITE(IOUT,81) I,(BUF(J,I),J=1,NCOL)
   81 FORMAT(1H0,I3,1X,20(1X,F5.1)/(5X,20(1X,F5.1)))
      GO TO 1000
C
C------------ FORMAT 20F5.2
   90 WRITE(IOUT,91) I,(BUF(J,I),J=1,NCOL)
   91 FORMAT(1H0,I3,1X,20(1X,F5.2)/(5X,20(1X,F5.2)))
      GO TO 1000
```

```
C
C------------ FORMAT 20F5.3
  100 WRITE(IOUT,101) I,(BUF(J,I),J=1,NCOL)
  101 FORMAT(1H0,I3,1X,20(1X,F5.3)/(5X,20(1X,F5.3)))
      GO TO 1000
C
C------------ FORMAT 20F5.4
  110 WRITE(IOUT,111) I,(BUF(J,I),J=1,NCOL)
  111 FORMAT(1H0,I3,1X,20(1X,F5.4)/(5X,20(1X,F5.4)))
      GO TO 1000
C
C------------ FORMAT 10G11.4
  120 WRITE(IOUT,121) I,(BUF(J,I),J=1,NCOL)
  121 FORMAT(1H0,I3,2X,1PG11.4,9(1X,G11.4)/(5X,10(1X,G11.4)))
C
 1000 CONTINUE
C
C5------RETURN
      RETURN
      END
      SUBROUTINE ULASAV(BUF,TEXT,KSTP,KPER,PERTIM,TOTIM,NCOL,
     1                   NROW,ILAY,ICHN)
C
C-----VERSION 1445 20APR1983 ULASAV
C     ******************************************************************
C     SAVE 1 LAYER ARRAY ON DISK
C     ******************************************************************
C
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
      DIMENSION BUF(NCOL,NROW)
CF66
C     DIMENSION TEXT(4)
CF66
CF77
      CHARACTER*4 TEXT(4)
CF77
C     ------------------------------------------------------------------
C
C1------WRITE AN UNFORMATTED RECORD CONTAINING IDENTIFYING
C1------INFORMATION.
      WRITE(ICHN) KSTP,KPER,PERTIM,TOTIM,TEXT,NCOL,NROW,ILAY
C
C2------WRITE AN UNFORMATTED RECORD CONTAINING ARRAY VALUES
C2------THE ARRAY IS DIMENSIONED (NCOL,NROW)
      WRITE(ICHN) ((BUF(IC,IR),IC=1,NCOL),IR=1,NROW)
C
C3------RETURN
      RETURN
      END
      SUBROUTINE U1DREL(A,ANAME,JJ,IN,IOUT)
C
C
C-----VERSION 1436 20MAY1983 U1DREL
```

```
C     ******************************************************************
C     ROUTINE TO INPUT 1-D REAL DATA MATRICES
C       A IS ARRAY TO INPUT
C       ANAME IS 24 CHARACTER DESCRIPTION OF A
C       JJ IS NO. OF ELEMENTS
C       IN IS INPUT UNIT
C       IOUT IS OUTPUT UNIT
C     ******************************************************************
C
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
      DIMENSION A(JJ)
CF66
C     DIMENSION ANAME(6),FMTIN(5)
CF66
CF77
      CHARACTER ANAME(6)*4,FMTIN*20
CF77
C     ------------------------------------------------------------------
C
C1------READ ARRAY CONTROL RECORD.
      READ (IN,1) LOCAT,CNSTNT,FMTIN,IPRN
CF66
C   1 FORMAT(I10,F10.0,5A4,I10)
CF66
CF77
    1 FORMAT(I10,F10.0,A20,I10)
CF77
C
C2------USE LOCAT TO SEE WHERE ARRAY VALUES COME FROM.
      IF(LOCAT.GT.0) GO TO 90
C
C3------IF LOCAT=0 THEN SET ALL ARRAY VALUES EQUAL TO CNSTNT. RETURN
      DO 80 J=1,JJ
   80 A(J)=CNSTNT
      WRITE(IOUT,3) ANAME,CNSTNT
    3 FORMAT(1H0,52X,6A4,' =',G15.7)
      RETURN
C
C4------IF LOCAT>0 THEN READ FORMATTED RECORDS USING FORMAT FMTIN.
   90 WRITE(IOUT,5) ANAME,LOCAT,FMTIN
CF66
C   5 FORMAT(1H0,///30X,6A4,' WILL BE READ ON UNIT',I3,
C    1         ' USING FORMAT: ',5A4/30X,79('-')/)
CF66
CF77
    5 FORMAT(1H0,///30X,6A4,' WILL BE READ ON UNIT',I3,
     1         ' USING FORMAT: ',A20/30X,79('-')/)
CF77
      READ (LOCAT,FMTIN) (A(J),J=1,JJ)
C
C5------IF CNSTNT NOT ZERO THEN MULTIPLY ARRAY VALUES BY CNSTNT.
      IF(CNSTNT.EQ.0.) GO TO 120
      DO 100 J=1,JJ
```

```
  100 A(J)=A(J)*CNSTNT
C
C6------IF PRINT CODE (IPRN) =>0 THEN PRINT ARRAY VALUES.
 120  IF(IPRN.LT.0) RETURN
      WRITE(IOUT,1001) (A(J),J=1,JJ)
 1001 FORMAT((1X,1PG12.5,9(1X,G12.5)))
      RETURN
C
C7------CONTINUE
      END
      SUBROUTINE U2DINT(IA,ANAME,II,JJ,K,IN,IOUT)
C
C
C-----VERSION 1442 20APR1983 U2DINT
C     ******************************************************************
C     ROUTINE TO INPUT 2-D INTEGER DATA MATRICES
C       IA IS ARRAY TO INPUT
C       ANAME IS 24 CHARACTER DESCRIPTION OF IA
C       II IS NO. OF ROWS
C       JJ IS NO. OF COLS
C       K IS LAYER NO. (USED WITH NAME TO TITLE PRINTOUT UNLESS K IS 0)
C       IN IS INPUT UNIT
C       IOUT IS OUTPUT UNIT
C     ******************************************************************
C
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
      DIMENSION IA(JJ,II)
CF66
C     DIMENSION ANAME(6),FMTIN(5)
CF66
CF77
      CHARACTER ANAME(6)*4,FMTIN*20
CF77
C     ------------------------------------------------------------------
C
C1------READ ARRAY CONTROL RECORD.
      READ (IN,1) LOCAT,ICONST,FMTIN,IPRN
CF66
C   1 FORMAT(I10,I10,5A4,I10)
CF66
CF77
    1 FORMAT(I10,I10,A20,I10)
       WRITE(IOUT,*)'  IN = ',IN
CF77
C
C2------USE LOCAT TO SEE WHERE ARRAY VALUES COME FROM.
      IF(LOCAT) 200,50,90
C
C3------IF LOCAT=0 THEN SET ALL ARRAY VALUES EQUAL TO ICONST. RETURN
   50 DO 80 I=1,II
      DO 80 J=1,JJ
   80 IA(J,I)=ICONST
      IF(K.GT.0) WRITE(IOUT,2) ANAME,ICONST,K
    2 FORMAT(1H0,52X,6A4,' =',I15,' FOR LAYER',I3)
      IF(K.LE.0) WRITE(IOUT,3) ANAME,ICONST
    3 FORMAT(1H0,52X,6A4,' =',I15)
      RETURN
C
C4------IF LOCAT>0 THEN READ FORMATTED RECORDS USING FORMAT FMTIN.
   90 IF(K.GT.0) WRITE(IOUT,4) ANAME,K,LOCAT,FMTIN
CF66
C   4 FORMAT(1H0,///30X,6A4,' FOR LAYER',I3,' WILL BE READ ON UNIT',
C    1       I3,' USING FORMAT: ',5A4/30X,96('-'))
CF66
CF77
    4 FORMAT(1H0,///30X,6A4,' FOR LAYER',I3,' WILL BE READ ON UNIT',
     1       I3,' USING FORMAT: ',A20/30X,96('-'))
CF77
      IF(K.LE.0) WRITE(IOUT,5) ANAME,LOCAT,FMTIN
CF66
C   5 FORMAT(1H0,///30X,6A4,' WILL BE READ ON UNIT',
C    1        I3,' USING FORMAT: ',5A4/30X,83('-'))
CF66
CF77
    5 FORMAT(1H0,///30X,6A4,' WILL BE READ ON UNIT',
     1        I3,' USING FORMAT: ',A20/30X,83('-'))
CF77
      DO 100 I=1,II
      READ (LOCAT,FMTIN) (IA(J,I),J=1,JJ)
  100 CONTINUE
      GO TO 300
C
C5------LOCAT<0 THEN READ UNFORMATTED RECORD CONTAINING ARRAY VALUES
  200 LOCAT=-LOCAT
      IF(K.GT.0) WRITE(IOUT,201) ANAME,K,LOCAT
  201 FORMAT(1H0,///30X,6A4,', LAYER',I3,
     1    ' WILL BE READ UNFORMATTED ON UNIT',I3/30X,73('-'))
      IF(K.LE.0) WRITE(IOUT,202) ANAME,LOCAT
  202 FORMAT(1H0,///30X,6A4,
     1    ' WILL BE READ UNFORMATTED ON UNIT',I3/30X,60('-'))
C
C5A------READ AN UNFORMATTED DUMMY RECORD FIRST.
      READ(LOCAT)
      READ(LOCAT) IA
C
C6------IF ICONST NOT ZERO THEN MULTIPLY ARRAY VALUES BY ICONST.
  300 IF(ICONST.EQ.0) GO TO 320
      DO 310 I=1,II
      DO 310 J=1,JJ
      IA(J,I)=IA(J,I)*ICONST
  310 CONTINUE
C
C7------IF PRINT CODE (IPRN) =>0 THEN PRINT ARRAY VALUES.
  320 IF(IPRN.LT.0) RETURN
      IF(IPRN.GT.5) IPRN=0
      IPRN=IPRN+1
C
```

```
C8------PRINT COLUMN NUMBERS AT TOP OF PAGE.
      IF(IPRN.EQ.1) CALL UCOLNO(1,JJ,0,10,12,IOUT)
      NL=125/IPRN/5*5
      IF(IPRN.GT.1) CALL UCOLNO(1,JJ,4,NL,IPRN,IOUT)
C
C9------PRINT EACH ROW IN THE ARRAY.
      DO 110 I=1,II
C
C10-----SELECT THE FORMAT
      GO TO(101,102,103,104,105,106), IPRN
C
C----------------FORMAT 10I11
  101 WRITE(IOUT,1001) I,(IA(J,I),J=1,JJ)
 1001 FORMAT(1H0,I3,2X,I11,9(1X,I11)/(5X,10(1X,I11)))
      GO TO 110
C
C----------------FORMAT 60I1
  102 WRITE(IOUT,1002) I,(IA(J,I),J=1,JJ)
 1002 FORMAT(1H0,I3,1X,60(1X,I1)/(5X,60(1X,I1)))
      GO TO 110
C
C----------------FORMAT 40I2
  103 WRITE(IOUT,1003) I,(IA(J,I),J=1,JJ)
 1003 FORMAT(1H0,I3,1X,40(1X,I2)/(5X,40(1X,I2)))
      GO TO 110
C
C----------------FORMAT 30I3
  104 WRITE(IOUT,1004) I,(IA(J,I),J=1,JJ)
 1004 FORMAT(1H0,I3,1X,30(1X,I3)/(5X,30(1X,I3)))
      GO TO 110
C
C----------------FORMAT 25I4
  105 WRITE(IOUT,1005) I,(IA(J,I),J=1,JJ)
 1005 FORMAT(1H0,I3,1X,25(1X,I4)/(5X,25(1X,I4)))
      GO TO 110
C
C----------------FORMAT 20I5
  106 WRITE(IOUT,1006) I,(IA(J,I),J=1,JJ)
 1006 FORMAT(1H0,I3,1X,20(1X,I5)/(5X,20(1X,I5)))
  110 CONTINUE
      RETURN
C
C11-----RETURN
      END
      SUBROUTINE U2DREL(A,ANAME,II,JJ,K,IN,IOUT)
C
C
C-----VERSION 1439 20APR1983 U2DREL
C     ******************************************************************
C     ROUTINE TO INPUT 2-D REAL DATA MATRICES
C       A IS ARRAY TO INPUT
C       ANAME IS 24 CHARACTER DESCRIPTION OF A
C       II IS NO. OF ROWS
C       JJ IS NO. OF COLS
```

```
C       K IS LAYER NO. (USED WITH NAME TO TITLE PRINTOUT UNLESS K IS 0)
C       IN IS INPUT UNIT
C       IOUT IS OUTPUT UNIT
C     *****************************************************************
C
C        SPECIFICATIONS:
C     -----------------------------------------------------------------
      DIMENSION A(JJ,II)
CF66
C     DIMENSION ANAME(6),FMTIN(5)
CF66
CF77
      CHARACTER ANAME(6)*4,FMTIN*20
CF77
C     -----------------------------------------------------------------
C
C1------READ ARRAY CONTROL RECORD.
      READ (IN,1) LOCAT,CNSTNT,FMTIN,IPRN
CF66
C   1 FORMAT(I10,F10.0,5A4,I10)
CF66
CF77
    1 FORMAT(I10,F10.0,A20,I10)
CF77
C
C2------USE LOCAT TO SEE WHERE ARRAY VALUES COME FROM.
      IF(LOCAT) 200,50,90
C
C3------IF LOCAT=0 THEN SET ALL ARRAY VALUES EQUAL TO CNSTNT. RETURN
   50 DO 80 I=1,II
      DO 80 J=1,JJ
   80 A(J,I)=CNSTNT
      IF(K.GT.0) WRITE(IOUT,2) ANAME,CNSTNT,K
    2 FORMAT(1H0,52X,6A4,' =',G15.7,' FOR LAYER',I3)
      IF(K.LE.0) WRITE(IOUT,3) ANAME,CNSTNT
    3 FORMAT(1H0,52X,6A4,' =',G15.7)
      RETURN
C
C4------IF LOCAT>0 THEN READ FORMATTED RECORDS USING FORMAT FMTIN.
   90 IF(K.GT.0) WRITE(IOUT,4) ANAME,K,LOCAT,FMTIN
CF66
C   4 FORMAT(1H0,///30X,6A4,' FOR LAYER',I3,' WILL BE READ ON UNIT',
C    1        I3,' USING FORMAT: ',5A4/30X,96('-'))
CF66
CF77
    4 FORMAT(1H0,///30X,6A4,' FOR LAYER',I3,' WILL BE READ ON UNIT',
     1        I3,' USING FORMAT: ',A20/30X,96('-'))
CF77
      IF(K.LE.0) WRITE(IOUT,5) ANAME,LOCAT,FMTIN
CF66
C   5 FORMAT(1H0,///30X,6A4,' WILL BE READ ON UNIT',
C    1        I3,' USING FORMAT: ',5A4/30X,83('-'))
CF66
CF77
```

```
    5 FORMAT(1H0,///30X,6A4,' WILL BE READ ON UNIT',
     1       I3,' USING FORMAT: ',A20/30X,83('-'))
CF77
      DO 100 I=1,II
      READ (LOCAT,FMTIN) (A(J,I),J=1,JJ)
  100 CONTINUE
      GO TO 300
C
C5------LOCAT<0 THEN READ UNFORMATTED RECORD CONTAINING ARRAY VALUES
  200 LOCAT=-LOCAT
      IF(K.GT.0) WRITE(IOUT,201) ANAME,K,LOCAT
  201 FORMAT(1H0,///30X,6A4,', LAYER',I3,
     1    ' WILL BE READ UNFORMATTED ON UNIT',I3/30X,73('-'))
      IF(K.LE.0) WRITE(IOUT,202) ANAME,LOCAT
  202 FORMAT(1H0,///30X,
     1    ' WILL BE READ UNFORMATTED ON UNIT',I3/30X,60('-'))
C
C5A------READ AN UNFORMATTED DUMMY RECORD FIRST.
      READ(LOCAT)
      READ(LOCAT) A
C
C6------IF CNSTNT NOT ZERO THEN MULTIPLY ARRAY VALUES BY CNSTNT.
  300 IF(CNSTNT.EQ.0.) GO TO 320
      DO 310 I=1,II
      DO 310 J=1,JJ
      A(J,I)=A(J,I)*CNSTNT
  310 CONTINUE
C
C7------IF PRINT CODE (IPRN) =>0 THEN PRINT ARRAY VALUES.
  320 IF(IPRN.LT.0) RETURN
      CALL ULAPRW(A,ANAME,0,0,JJ,II,0,IPRN,IOUT)
      RETURN
C
C8------RETURN
      END
      SUBROUTINE BAS1AD(DELT,TSMULT,TOTIM,PERTIM,HNEW,HOLD,KSTP,
     1                  NCOL,NROW,NLAY)
C
C-----VERSION 1412 22FEB1982 BAS1AD
C
C     ******************************************************************
C     ADVANCE TO NEXT TIME STEP
C     ******************************************************************
C
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
      DOUBLE PRECISION HNEW
C
      DIMENSION HNEW(NCOL,NROW,NLAY), HOLD(NCOL,NROW,NLAY)
C     ------------------------------------------------------------------
C
C1------IF NOT FIRST TIME STEP THEN CALCULATE TIME STEP LENGTH.
      IF(KSTP.NE.1) DELT=TSMULT*DELT
C
```

```
C2------ACCUMULATE ELAPSED TIME IN SIMULATION(TOTIM) AND IN THIS
C2------STRESS PERIOD(PERTIM).
      TOTIM=TOTIM+DELT
      PERTIM=PERTIM+DELT
C
C3------COPY HNEW TO HOLD.
      DO 10 K=1,NLAY
      DO 10 I=1,NROW
      DO 10 J=1,NCOL
   10 HOLD(J,I,K)=HNEW(J,I,K)
C
C4------RETURN
      RETURN
      END
      SUBROUTINE BAS1AL(ISUM,LENX,LCHNEW,LCHOLD,LCIBOU,LCCR,LCCC,LCCV,
     1            LCHCOF,LCRHS,LCDELR,LCDELC,LCSTRT,LCBUFF,LCIOFL,INBAS,
     1            ISTRT,NCOL,NROW,NLAY,IOUT)
C-----VERSION 0927 08DEC1983 BAS1AL
C     ******************************************************************
C     ALLOCATE SPACE FOR BASIC MODEL ARRAYS
C     ******************************************************************
C
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
      INTEGER*4 LENX
C     ------------------------------------------------------------------
C
C1------PRINT A MESSAGE IDENTIFYING THE PACKAGE.
      WRITE(IOUT,1)INBAS
    1 FORMAT(1H0,'BAS1 -- BASIC MODEL PACKAGE, VERSION 1, 12/08/83',
     2' INPUT READ FROM UNIT',I3)
C
C2------READ & PRINT FLAG IAPART (RHS & BUFFER SHARE SPACE?) AND
C2------FLAG ISTRT (SHOULD STARTING HEADS BE SAVED FOR DRAWDOWN?)
      READ(INBAS,2) IAPART,ISTRT
    2 FORMAT(2I10)
      IF(IAPART.EQ.0) WRITE(IOUT,3)
    3 FORMAT(1X,'ARRAYS RHS AND BUFF WILL SHARE MEMORY.')
      IF(ISTRT.NE.0) WRITE(IOUT,4)
    4 FORMAT(1X,'START HEAD WILL BE SAVED')
      IF(ISTRT.EQ.0) WRITE(IOUT,5)
    5 FORMAT(1X,'START HEAD WILL NOT BE SAVED',
     1      ' -- DRAWDOWN CANNOT BE CALCULATED')
C
C3------STORE,IN ISOLD, LOCATION OF FIRST UNALLOCATED SPACE IN X.
      ISOLD=ISUM
      NRCL=NROW*NCOL*NLAY
C
C4------ALLOCATE SPACE FOR ARRAYS.
      LCHNEW=ISUM
      ISUM=ISUM+2*NRCL
      LCHOLD=ISUM
      ISUM=ISUM+NRCL
      LCIBOU=ISUM
      ISUM=ISUM+NRCL
      LCCR=ISUM
      ISUM=ISUM+NRCL
      LCCC=ISUM
      ISUM=ISUM+NRCL
      LCCV=ISUM
      ISUM=ISUM+NROW*NCOL*(NLAY-1)
      LCHCOF=ISUM
      ISUM=ISUM+NRCL
      LCRHS=ISUM
      ISUM=ISUM+NRCL
      LCDELR=ISUM
      ISUM=ISUM+NCOL
      LCDELC=ISUM
      ISUM=ISUM+NROW
      LCIOFL=ISUM
      ISUM=ISUM+NLAY*4
C
C5------IF BUFFER AND RHS SHARE SPACE THEN LCBUFF=LCRHS.
      LCBUFF=LCRHS
      IF(IAPART.EQ.0) GO TO 50
      LCBUFF=ISUM
      ISUM=ISUM+NRCL
C
C6------IF STRT WILL BE SAVED THEN ALLOCATE SPACE.
   50 LCSTRT=ISUM
      IF(ISTRT.NE.0) ISUM=ISUM+NRCL
      ISP=ISUM-ISOLD
C
C7------PRINT AMOUNT OF SPACE USED.
      WRITE(IOUT,6) ISP
    6 FORMAT(1X,I6,' ELEMENTS IN X ARRAY ARE USED BY BAS')
      ISUM1=ISUM-1
      WRITE(IOUT,7) ISUM1,LENX
    7 FORMAT(1X,I6,' ELEMENTS OF X ARRAY USED OUT OF',I7)
      IF(ISUM1.GT.LENX) WRITE(IOUT,8)
    8 FORMAT(1X,'   ***X ARRAY MUST BE DIMENSIONED LARGER***')
C
C
C8------RETURN
      RETURN
C
      END
      SUBROUTINE BAS1DF(ISUM,HEADNG,NPER,ITMUNI,TOTIM,NCOL,NROW,
     1       NLAY,NODES,INBAS,IOUT,IUNIT)
C
C-----VERSION 1128 28DEC1983 BAS1DF
C     ******************************************************************
C     DEFINE KEY MODEL PARAMETERS
C     ******************************************************************
C
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
      DIMENSION HEADNG(32),IUNIT(24)
```

```
C     ------------------------------------------------------------------
C
C1------PRINT THE NAME OF THE PROGRAM.
      WRITE(IOUT,1)
    1 FORMAT(1H1,20X,'ETA PREPROCESSOR - PREPARES MODFLOW FILES AND',
     1      ' OUTPUT FOR INPUT TO RAND3D')
C
C2------READ AND PRINT A HEADING.
      READ(INBAS,2) HEADNG
    2 FORMAT(20A4)
      WRITE(IOUT,3) HEADNG
    3 FORMAT(1H0,32A4)
C
C3------READ NUMBER OF LAYERS,ROWS,COLUMNS,STRESS PERIODS AND
C3------UNITS OF TIME CODE.
      READ(INBAS,4) NLAY,NROW,NCOL,NPER,ITMUNI
    4 FORMAT(8I10)
C
C4------PRINT # OF LAYERS, ROWS, COLUMNS AND STRESS PERIODS.
      WRITE(IOUT,5) NLAY,NROW,NCOL
    5 FORMAT(1X,I4,' LAYERS',I10,' ROWS',I10,' COLUMNS')
      WRITE(IOUT,6) NPER
    6 FORMAT(1X,I3,' STRESS PERIOD(S) IN SIMULATION')
C
C5------SELECT AND PRINT A MESSAGE SHOWING TIME UNITS.
      IF(ITMUNI.LT.0 .OR. ITMUNI.GT.5) ITMUNI=0
      GO TO (10,20,30,40,50),ITMUNI
      WRITE(IOUT,9)
    9 FORMAT(1X,'MODEL TIME UNITS ARE UNDEFINED')
      GO TO 100
   10 WRITE(IOUT,11)
   11 FORMAT(1X,'MODEL TIME UNIT IS SECONDS')
      GO TO 100
   20 WRITE(IOUT,21)
   21 FORMAT(1X,'MODEL TIME UNIT IS MINUTES')
      GO TO 100
   30 WRITE(IOUT,31)
   31 FORMAT(1X,'MODEL TIME UNIT IS HOURS')
      GO TO 100
   40 WRITE(IOUT,41)
   41 FORMAT(1X,'MODEL TIME UNIT IS DAYS')
      GO TO 100
   50 WRITE(IOUT,51)
   51 FORMAT(1X,'MODEL TIME UNIT IS YEARS')
C
C6------READ & PRINT INPUT UNIT NUMBERS (IUNIT) FOR MAJOR OPTIONS.
  100 READ(INBAS,101) IUNIT
  101 FORMAT(24I3)
      WRITE(IOUT,102) (I,I=1,24),IUNIT
  102 FORMAT(1H0,'I/O UNITS:'/1X,'ELEMENT OF IUNIT:',24I3,
     1                       /1X,'        I/O UNIT:',24I3)
C
C7------INITIALIZE TOAL ELAPSED TIME COUNTER STORAGE ARRAY COUNTER
C7------AND CALCULATE NUMBER OF CELLS.
```

```
      TOTIM=0.
      ISUM=1
      NODES=NCOL*NROW*NLAY
C
C8------RETURN
      RETURN
      END
      SUBROUTINE BAS1FM(HCOF,RHS,NCOL,NROW,NLAY,NODES)
C
C
C-----VERSION 1408 22FEB1982 BAS1FM
C     ******************************************************************
C     SET HCOF=RHS=0.
C     ******************************************************************
C
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
      DIMENSION HCOF(NODES),RHS(NODES)
C     ------------------------------------------------------------------
C
C1------FOR EACH CELL INITIALIZE HCOF AND RHS ACCUMULATORS.
      DO 100 I=1,NODES
      HCOF(I)=0.
      RHS(I)=0.
  100 CONTINUE
C
C2------RETURN
      RETURN
      END
      SUBROUTINE BAS1OC(NSTP,KSTP,KPER,ISTRT,ICNVG,IOFLG,NLAY,
     1      IBUDFL,ICBCFL,IHDDFL,INOC,IOUT)
C
C-----VERSION 0949 03NOV1982 BAS1OC
C     ******************************************************************
C     OUTPUT CONTROLLER FOR HEAD, DRAWDOWN, AND BUDGET
C     ******************************************************************
C
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
      DIMENSION IOFLG(NLAY,4)
C     ------------------------------------------------------------------
C
C
C1------TEST UNIT NUMBER (INOC (INOC=IUNIT(12))) TO SEE IF
C1------OUTPUT CONTROL IS ACTIVE.
      IF(INOC.NE.0)GO TO 500
C
C2------IF OUTPUT CONTROL IS INACTIVE THEN SET DEFAULTS AND RETURN.
      IHDDFL=0
      IF(ICNVG.EQ.0 .OR. KSTP.EQ.NSTP)IHDDFL=1
      IBUDFL=0
      IF(ICNVG.EQ.0 .OR. KSTP.EQ.NSTP)IBUDFL=1
      ICBCFL=0
      GO TO 1000
```

```
C
C3-------READ AND PRINT OUTPUT FLAGS AND CODE FOR DEFINING IOFLG.
  500 READ(INOC,1) INCODE,IHDDFL,IBUDFL,ICBCFL
    1 FORMAT(4I10)
      WRITE(IOUT,3) IHDDFL,IBUDFL,ICBCFL
    3 FORMAT(1H0,'HEAD/DRAWDOWN PRINTOUT FLAG =',I2,
     1    5X,'TOTAL BUDGET PRINTOUT FLAG =',I2,
     2    5X,'CELL-BY-CELL FLOW TERM FLAG =',I2)
C
C4------DECODE INCODE TO DETERMINE HOW TO SET FLAGS IN IOFLG.
      IF(INCODE) 100,200,300
C
C5------USE IOFLG FROM LAST TIME STEP.
  100 WRITE(IOUT,101)
  101 FORMAT(1H ,'REUSING PREVIOUS VALUES OF IOFLG')
      GO TO 600
C
C6------READ IOFLG FOR LAYER 1 AND ASSIGN SAME TO ALL LAYERS
  200 READ(INOC,201) (IOFLG(1,M),M=1,4)
  201 FORMAT(4I10)
      DO 210 K=1,NLAY
      IOFLG(K,1)=IOFLG(1,1)
      IOFLG(K,2)=IOFLG(1,2)
      IOFLG(K,3)=IOFLG(1,3)
      IOFLG(K,4)=IOFLG(1,4)
  210 CONTINUE
      WRITE(IOUT,211) (IOFLG(1,M),M=1,4)
  211 FORMAT(1H0,'OUTPUT FLAGS FOR ALL LAYERS ARE THE SAME:'/
     1   1X,'  HEAD    DRAWDOWN  HEAD  DRAWDOWN'/
     2   1X,'PRINTOUT  PRINTOUT  SAVE    SAVE'/
     3   1X,34('-')/1X,I5,I10,I8,I8)
      GO TO 600
C
C7------READ IOFLG IN ENTIRETY
  300 READ(INOC,301) ((IOFLG(K,I),I=1,4),K=1,NLAY)
  301 FORMAT(4I10)
      WRITE(IOUT,302)
  302 FORMAT(1H0,'OUTPUT FLAGS FOR EACH LAYER:'/
     1   1X,'         HEAD    DRAWDOWN  HEAD  DRAWDOWN'/
     2   1X,'LAYER  PRINTOUT  PRINTOUT  SAVE    SAVE'/
     3   1X,41('-'))
      WRITE(IOUT,303) (K,(IOFLG(K,I),I=1,4),K=1,NLAY)
  303 FORMAT(1X,I4,I8,I10,I8,I8)
C
C8------THE LAST STEP IN A STRESS PERIOD AND STEPS WHERE ITERATIVE
C8------PROCEDURE FAILED TO CONVERGE GET A VOLUMETRIC BUDGET.
  600 IF(ICNVG.EQ.0 .OR. KSTP.EQ.NSTP) IBUDFL=1
C
C9------RETURN
 1000 RETURN
      END
      SUBROUTINE BAS1OT(HNEW,STRT,ISTRT,BUFF,IOFLG,MSUM,IBOUND,VBNM,
     1  VBVL,KSTP,KPER,DELT,PERTIM,TOTIM,ITMUNI,NCOL,NROW,NLAY,ICNVG,
     2  IHDDFL,IBUDFL,IHEDFM,IHEDUN,IDDNFM,IDDNUN,IOUT)
```

```
C-----VERSION 1154 29MAR1984 BAS1OT
C     ******************************************************************
C     OUTPUT TIME, VOLUMETRIC BUDGET, HEAD, AND DRAWDOWN
C     ******************************************************************
C
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
      DOUBLE PRECISION HNEW
C
      DIMENSION HNEW(NCOL,NROW,NLAY),STRT(NCOL,NROW,NLAY),
     1          VBVL(1),IOFLG(NLAY,4),
     2          IBOUND(NCOL,NROW,NLAY),BUFF(NCOL,NROW,NLAY)
CF66
C     DIMENSION VBNM(1)
CF66
CF77
      CHARACTER*4 VBNM(1)
CF77
C     ------------------------------------------------------------------
C
C1------CLEAR PRINTOUT FLAG (IPFLG)
      IPFLG=0
C
C2------IF ITERATIVE PROCEDURE FAILED TO CONVERGE PRINT MESSAGE
      IF(ICNVG.EQ.0) WRITE(IOUT,1) KSTP,KPER
    1 FORMAT(1H0,10X,'****FAILED TO CONVERGE IN TIME STEP',I3,
     1      ' OF STRESS PERIOD',I3,'****')
C
C3------IF HEAD AND DRAWDOWN FLAG (IHDDFL) IS SET WRITE HEAD AND
C3------DRAWDOWN IN ACCORDANCE WITH FLAGS IN IOFLG.
      IF(IHDDFL.EQ.0) GO TO 100
C
      CALL SBAS1H(HNEW,BUFF,IOFLG,KSTP,KPER,NCOL,NROW,
     1    NLAY,IOUT,IHEDFM,IHEDUN,IPFLG,PERTIM,TOTIM)
      CALL SBAS1D(HNEW,BUFF,IOFLG,KSTP,KPER,NCOL,NROW,NLAY,IOUT,
     1 IDDNFM,IDDNUN,STRT,ISTRT,IBOUND,IPFLG,PERTIM,TOTIM)
C
C4------PRINT TOTAL BUDGET IF REQUESTED
  100 IF(IBUDFL.EQ.0) GO TO 120
      CALL SBAS1V(MSUM,VBNM,VBVL,KSTP,KPER,IOUT)
      IPFLG=1
C
C5------END PRINTOUT WITH TIME SUMMARY AND FORM FEED IF ANY PRINTOUT
C5------WILL BE PRODUCED.
  120 IF(IPFLG.EQ.0) RETURN
      CALL SBAS1T(KSTP,KPER,DELT,PERTIM,TOTIM,ITMUNI,IOUT)
      WRITE(IOUT,101)
  101 FORMAT(1H1)
C
C6------RETURN
      RETURN
      END
      SUBROUTINE BAS1RP(IBOUND,HNEW,STRT,HOLD,ISTRT,INBAS,
     1    HEADNG,NCOL,NROW,NLAY,NODES,VBVL,IOFLG,INOC,IHEDFM,
     2    IDDNFM,IHEDUN,IDDNUN,IOUT)
C-----VERSION 0956 03NOV1982 BAS1RP
C     ******************************************************************
C     READ AND INITIALIZE BASIC MODEL ARRAYS
C     ******************************************************************
C
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
      DOUBLE PRECISION HNEW,HNOFLO
C
      DIMENSION HNEW(NODES),IBOUND(NODES),STRT(NODES),HOLD(NODES),
     1             VBVL(4,20),IOFLG(NLAY,4),HEADNG(32)
CF66
C     DIMENSION ANAME(6,2)
CF66
CF77
      CHARACTER*4 ANAME(6,2)
CF77
C
      DATA ANAME(1,1),ANAME(2,1),ANAME(3,1),ANAME(4,1),ANAME(5,1),
     1   ANAME(6,1) /'    ','    ','  BO','UNDA','RY A','RRAY'/
      DATA ANAME(1,2),ANAME(2,2),ANAME(3,2),ANAME(4,2),ANAME(5,2),
     1   ANAME(6,2) /'    ','    ','    ','INIT','IAL ','HEAD'/
C     ------------------------------------------------------------------
C
C1------PRINT SIMULATION TITLE, CALCULATE # OF CELLS IN A LAYER.
      WRITE(IOUT,1) HEADNG
    1 FORMAT(1H1,32A4)
      NCR=NCOL*NROW
C
C2------READ BOUNDARY ARRAY(IBOUND) ONE LAYER AT A TIME.
      DO 100 K=1,NLAY
      LOC=1+(K-1)*NCR
      CALL U2DINT(IBOUND(LOC),ANAME(1,1),NROW,NCOL,K,INBAS,IOUT)
  100 CONTINUE
C
C3------READ AND PRINT HEAD VALUE TO BE PRINTED FOR NO-FLOW CELLS.
      READ(INBAS,2) TMP
    2 FORMAT(F10.0)
      HNOFLO=TMP
      WRITE(IOUT,3) TMP
    3 FORMAT(1H0,'AQUIFER HEAD WILL BE SET TO ',1PG11.5,
     1       ' AT ALL NO-FLOW NODES (IBOUND=0).')
C
C4------READ STARTING HEADS.
      DO 300 K=1,NLAY
      LOC=1+(K-1)*NCR
      CALL U2DREL(HOLD(LOC),ANAME(1,2),NROW,NCOL,K,INBAS,IOUT)
  300 CONTINUE
C
C5------COPY INITIAL HEADS FROM HOLD TO HNEW.
      DO 400 I=1,NODES
      HNEW(I)=HOLD(I)
      IF(IBOUND(I).EQ.0) HNEW(I)=HNOFLO
```

```
C4------INITIALIZE PERTIM (ELAPSED TIME WITHIN STRESS PERIOD).
      PERTIM=0.
C
C5------RETURN
      RETURN
      END
      SUBROUTINE SBAS1D(HNEW,BUFF,IOFLG,KSTP,KPER,NCOL,NROW,
     1    NLAY,IOUT,IDDNFM,IDDNUN,STRT,ISTRT,IBOUND,IPFLG,
     2    PERTIM,TOTIM)
C-----VERSION 1147 29MAR1984 SBAS1D
C     *******************************************************
C     CALCULATE PRINT AND RECORD DRAWDOWNS
C     *******************************************************
C
C        SPECIFICATIONS
C     -------------------------------------------------------
      DOUBLE PRECISION HNEW
C
      DIMENSION HNEW(NCOL,NROW,NLAY),IOFLG(NLAY,4),
     1     BUFF(NCOL,NROW,NLAY),STRT(NCOL,NROW,NLAY),
     2     IBOUND(NCOL,NROW,NLAY)
CF66
C     DIMENSION TEXT(4)
CF66
CF77
      CHARACTER*4 TEXT(4)
CF77
C
      DATA TEXT(1),TEXT(2),TEXT(3),TEXT(4) /'    ','    ','DRAW',
     1       'DOWN'/
C     -------------------------------------------------------
C
C1------FOR EACH LAYER CALCULATE DRAWDOWN IF PRINT OR RECORD
C1------IS REQUESTED
      DO 59 K=1,NLAY
C
C2------IS DRAWDOWN NEEDED FRO THIS LAYER?
      IF(IOFLG(K,2).EQ.0 .AND. IOFLG(K,4).EQ.0) GO TO 59
C
C3------DRAWDOWN IS NEEDED. WERE STARTING HEADS SAVED?
      IF(ISTRT.NE.0) GO TO 53
C
C4------STARTING HEADS WERE NOT SAVED. PRINT MESSAGE AND STOP.
      WRITE(IOUT,52)
   52 FORMAT(1H0,'CANNOT CALCULATE DRAWDOWN BECAUSE START',
     1   ' HEADS WERE NOT SAVED')
      STOP
C
C5------CALCULATE DRAWDOWN FOR THE LAYER.
   53 DO 58 I=1,NROW
      DO 58 J=1,NCOL
      HSING=HNEW(J,I,K)
      BUFF(J,I,K)=HSING
      IF(IBOUND(J,I,K).NE.0) BUFF(J,I,K)=STRT(J,I,K)-HSING
```

```
  400 CONTINUE
C
C6------IF STARTING HEADS ARE TO BE SAVED THEN COPY HOLD TO STRT.
      IF(ISTRT.EQ.0) GO TO 590
      DO 500 I=1,NODES
      STRT(I)=HOLD(I)
  500 CONTINUE
C
C7------INITIALIZE VOLUMETRIC BUDGET ACCUMULATORS TO ZERO.
  590 DO 600 I=1,20
      DO 600 J=1,4
      VBVL(J,I)=0.
  600 CONTINUE
C
C8------SET UP OUTPUT CONTROL.
      CALL SBAS1I(NLAY,ISTRT,IOFLG,INOC,IOUT,IHEDFM,
     1         IDDNFM,IHEDUN,IDDNUN)
C
C9------RETURN
 1000 RETURN
      END
      SUBROUTINE BAS1ST(NSTP,DELT,TSMULT,PERTIM,KPER,INBAS,IOUT)
C
C
C-----VERSION 1614 08SEP1982 BAS1ST
C     ******************************************************************
C     SETUP TIME PARAMETERS FOR NEW TIME PERIOD
C     ******************************************************************
C
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
C     ------------------------------------------------------------------
C
C1------READ LENGTH OF STRESS PERIOD, NUMBER OF TIME STEPS AND.
C1------TIME STEP MULTIPLIER.
      READ (INBAS,1) PERLEN,NSTP,TSMULT
    1 FORMAT(F10.0,I10,F10.0)
C
C2------CALCULATE THE LENGTH OF THE FIRST TIME STEP.
C
C2A-----ASSUME TIME STEP MULTIPLIER IS EQUAL TO ONE.
      DELT=PERLEN/FLOAT(NSTP)
C
C2B-----IF TIME STEP MULTIPLIER IS NOT ONE THEN CALCULATE FIRST
C2B-----TERM OF GEOMETRIC PROGRESSION.
      IF(TSMULT.NE.1.) DELT=PERLEN*(1.-TSMULT)/(1.-TSMULT**NSTP)
C
C3------PRINT TIMING INFORMATION.
      WRITE (IOUT,2) KPER,PERLEN,NSTP,TSMULT,DELT
    2 FORMAT(1H1,51X,'STRESS PERIOD NO.',I4,', LENGTH =',G15.7/52X
     1,46('-')//52X,'NUMBER OF TIME STEPS =',I6
     2//53X,'MULTIPLIER FOR DELT =',F10.3
     3//50X,'INITIAL TIME STEP SIZE =',G15.7)
C
```

```
   58 CONTINUE
   59 CONTINUE
C
C6------FOR EACH LAYER: DETERMINE IF DRAWDOWN SHOULD BE PRINTED.
C6------IF SO THEN CALL ULAPRS OR ULAPRW TO PRINT DRAWDOWN.
      DO 69 K=1,NLAY
      IF(IOFLG(K,2).EQ.0) GO TO 69
      IF(IDDNFM.LT.0) CALL ULAPRS(BUFF(1,1,K),TEXT(1),KSTP,KPER,
     1             NCOL,NROW,K,-IDDNFM,IOUT)
      IF(IDDNFM.GE.0) CALL ULAPRW(BUFF(1,1,K),TEXT(1),KSTP,KPER,
     1             NCOL,NROW,K,IDDNFM,IOUT)
      IPFLG=1
   69 CONTINUE
C
C7------FOR EACH LAYER: DETERMINE IF DRAWDOWN SHOULD BE RECORDED.
C7------IF SO THEN CALL ULASAV TO RECORD DRAWDOWN.
      IFIRST=1
      IF(IDDNUN.LE.0) GO TO 80
      DO 79 K=1,NLAY
      IF(IOFLG(K,4).LE.0) GO TO 79
      IF(IFIRST.EQ.1) WRITE(IOUT,74) IDDNUN,KSTP,KPER
   74 FORMAT(1H0,'DRAWDOWN WILL BE SAVED ON UNIT',I3,
     1    ' AT END OF TIME STEP',I3,', STRESS PERIOD',I3)
      IFIRST=0
      CALL ULASAV(BUFF(1,1,K),TEXT(1),KSTP,KPER,PERTIM,TOTIM,NCOL,
     1              NROW,K,IDDNUN)
   79 CONTINUE
C
C8------RETURN
   80 RETURN
      END
      SUBROUTINE SBAS1H(HNEW,BUFF,IOFLG,KSTP,KPER,NCOL,NROW,
     1    NLAY,IOUT,IHEDFM,IHEDUN,IPFLG,PERTIM,TOTIM)
C
C-----VERSION 1138 29MAR1984 SBAS1H
C     ******************************************************************
C     PRINT AND RECORD HEADS
C     ******************************************************************
C
C        SPECIFICATIONS
C     ------------------------------------------------------------------
      DOUBLE PRECISION HNEW
C
      DIMENSION HNEW(NCOL,NROW,NLAY),IOFLG(NLAY,4),
     1     BUFF(NCOL,NROW,NLAY)
CF66
C     DIMENSION TEXT(4)
CF66
CF77
      CHARACTER*4 TEXT(4)
CF77
C
      DATA TEXT(1),TEXT(2),TEXT(3),TEXT(4) /'    ','    ','    ',
     1      'HEAD'/
```

```
C     ----------------------------------------------------
C
C1------FOR EACH LAYER: PRINT HEAD IF REQUESTED.
      DO 39 K=1,NLAY
C
C2------TEST IOFLG TO SEE IF HEAD SHOULD BE PRINTED.
      IF(IOFLG(K,1).EQ.0) GO TO 39
      IPFLG=1
C
C3------COPY HEADS FOR THIS LAYER INTO BUFFER.
      DO 32 I=1,NROW
      DO 32 J=1,NCOL
      BUFF(J,I,1)=HNEW(J,I,K)
   32 CONTINUE
C
C4------CALL UTILITY MODULE TO PRINT CONTENTS OF BUFFER.
      IF(IHEDFM.LT.0) CALL ULAPRS(BUFF,TEXT(1),KSTP,KPER,NCOL,NROW,K,
     1          -IHEDFM,IOUT)
      IF(IHEDFM.GE.0) CALL ULAPRW(BUFF,TEXT(1),KSTP,KPER,NCOL,NROW,K,
     1             IHEDFM,IOUT)
   39 CONTINUE
C
C5------IF UNIT FOR RECORDING HEADS <= 0: THEN RETURN.
      IF(IHEDUN.LE.0)GO TO 50
      IFIRST=1
C
C6------FOR EACH LAYER: RECORD HEAD IF REQUESTED.
      DO 49 K=1,NLAY
C
C7------CHECK IOFLG TO SEE IF HEAD FOR THIS LAYER SHOULD BE RECORDED.
      IF(IOFLG(K,3).LE.0) GO TO 49
      IF(IFIRST.EQ.1) WRITE(IOUT,41) IHEDUN,KSTP,KPER
   41 FORMAT(1H0,'HEAD WILL BE SAVED ON UNIT',I3,' AT END OF TIME STEP',
     1    I3,', STRESS PERIOD',I3)
      IFIRST=0
C
C8------COPY HEADS FOR THIS LAYER INTO BUFFER.
      DO 44 I=1,NROW
      DO 44 J=1,NCOL
      BUFF(J,I,1)=HNEW(J,I,K)
   44 CONTINUE
C
C9------RECORD CONTENTS OF BUFFER ON UNIT=IHEDUN
      CALL ULASAV(BUFF,TEXT(1),KSTP,KPER,PERTIM,TOTIM,NCOL,NROW,K,
     1             IHEDUN)
   49 CONTINUE
C
C10-----RETURN
   50 RETURN
      END
      SUBROUTINE SBAS1I(NLAY,ISTRT,IOFLG,INOC,IOUT,IHEDFM,
     1         IDDNFM,IHEDUN,IDDNUN)
C
C-----VERSION 1138 03NOV1982 SBAS1I
```

```
C     ***************************************************************
C     SET UP OUTPUT CONTROL
C     ***************************************************************
C
C         SPECIFICATIONS
C
      DIMENSION IOFLG(NLAY,4)
C     ---------------------------------------------------------------
C
C1------TEST UNIT NUMBER FROM IUNIT (INOC) TO SEE IF OUTPUT
C1------CONTROL IS ACTIVE.
      IF(INOC.LE.0) GO TO 600
C
C2------READ AND PRINT FORMATS FOR PRINTING AND UNIT NUMBERS FOR
C2------RECORDING HEADS AND DRAWDOWN. THEN RETURN.
  500 READ (INOC,1)IHEDFM,IDDNFM,IHEDUN,IDDNUN
    1 FORMAT (4I10)
      WRITE (IOUT,3)IHEDFM,IDDNFM
    3 FORMAT (1H0,'HEAD PRINT FORMAT IS FORMAT NUMBER',I4,
     1        '    DRAWDOWN PRINT FORMAT IS FORMAT NUMBER',I4)
      WRITE (IOUT,4)IHEDUN,IDDNUN
    4 FORMAT (1H0,'HEADS WILL BE SAVED ON UNIT',I3,
     1        '    DRAWDOWNS WILL BE SAVED ON UNIT',I3)
      WRITE(IOUT,561)
  561 FORMAT(1H0,'OUTPUT CONTROL IS SPECIFIED EVERY TIME STEP')
      GO TO 1000
C
C3------OUTPUT CONTROL IS INACTIVE. PRINT A MESSAGE LISTING DEFAULTS.
  600 WRITE(IOUT,641)
  641 FORMAT(1H0,'DEFAULT OUTPUT CONTROL -- THE FOLLOWING OUTPUT',
     1      ' COMES AT THE END OF EACH STRESS PERIOD:')
      WRITE(IOUT,642)
  642 FORMAT(1X,'TOTAL VOLUMETRIC BUDGET')
      WRITE(IOUT,643)
  643 FORMAT(1X,10X,'HEAD')
      IF(ISTRT.NE.0)WRITE(IOUT,644)
  644 FORMAT(1X,10X,'DRAWDOWN')
C
C4------SET THE FORMAT CODES EQUAL TO THE DEFAULT FORMAT.
      IHEDFM=0
      IDDNFM=0
C
C5------SET DEFAULT FLAGS IN IOFLG SO THAT HEAD (AND DRAWDOWN) IS
C5------PRINTED FOR EVERY LAYER.
      ID=0
      IF(ISTRT.NE.0) ID=1
  670 DO 680 K=1,NLAY
      IOFLG(K,1)=1
      IOFLG(K,2)=ID
      IOFLG(K,3)=0
      IOFLG(K,4)=0
  680 CONTINUE
      GO TO 1000
C
```

```
C6------RETURN
1000  RETURN
      END
      SUBROUTINE SBAS1T(KSTP,KPER,DELT,PERTIM,TOTIM,ITMUNI,IOUT)
C
C
C-----VERSION 0837 09APR1982 SBAS1T
C     ******************************************************************
C     PRINT SIMULATION TIME
C     ******************************************************************
C
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
C     ------------------------------------------------------------------
      WRITE(IOUT,199) KSTP,KPER
  199 FORMAT(1H0,///10X,'TIME SUMMARY AT END OF TIME STEP',I3,
     1     ' IN STRESS PERIOD',I3)
C
C1------USE TIME UNIT INDICATOR TO GET FACTOR TO CONVERT TO SECONDS.
      CNV=0.
      IF(ITMUNI.EQ.1) CNV=1.
      IF(ITMUNI.EQ.2) CNV=60.
      IF(ITMUNI.EQ.3) CNV=3600.
      IF(ITMUNI.EQ.4) CNV=86400.
      IF(ITMUNI.EQ.5) CNV=31557600.
C
C2------IF FACTOR=0 THEN TIME UNITS ARE NON-STANDARD.
      IF(CNV.NE.0.) GO TO 100
C
C2A-----PRINT TIMES IN NON-STANDARD TIME UNITS.
      WRITE(IOUT,301) DELT,PERTIM,TOTIM
  301 FORMAT(21X,'     TIME STEP LENGTH =',G15.6/
     1       21X,'   STRESS PERIOD TIME =',G15.6/
     2       21X,'TOTAL SIMULATION TIME =',G15.6)
C
C2B-----RETURN
      RETURN
C
C3------CALCULATE LENGTH OF TIME STEP & ELAPSED TIMES IN SECONDS.
  100 DELSEC=CNV*DELT
      TOTSEC=CNV*TOTIM
      PERSEC=CNV*PERTIM
C
C4------CALCULATE TIMES IN MINUTES,HOURS,DAYS AND YEARS.
      DELMN=DELSEC/60.
      DELHR=DELMN/60.
      DELDY=DELHR/24.
      DELYR=DELDY/365.25
      TOTMN=TOTSEC/60.
      TOTHR=TOTMN/60.
      TOTDY=TOTHR/24.
      TOTYR=TOTDY/365.25
      PERMN=PERSEC/60.
      PERHR=PERMN/60.
      PERDY=PERHR/24.
      PERYR=PERDY/365.25
C
C5------PRINT TIME STEP LENGTH AND ELAPSED TIMES IN ALL TIME UNITS.
      WRITE(IOUT,200)
  200 FORMAT(27X,'   SECONDS        MINUTES         HOURS',10X,
     1    'DAYS          YEARS'/27X,75('-'))
      WRITE (IOUT,201) DELSEC,DELMN,DELHR,DELDY,DELYR
  201 FORMAT(1X,'     TIME STEP LENGTH',5X,5G15.6)
      WRITE(IOUT,202) PERSEC,PERMN,PERHR,PERDY,PERYR
  202 FORMAT(1X,'   STRESS PERIOD TIME',5X,5G15.6)
      WRITE(IOUT,203) TOTSEC,TOTMN,TOTHR,TOTDY,TOTYR
  203 FORMAT(1X,'TOTAL SIMULATION TIME',5X,5G15.6)
C
C6------RETURN
      RETURN
      END
      SUBROUTINE SBAS1V(MSUM,VBNM,VBVL,KSTP,KPER,IOUT)
C
C
C-----VERSION 1153 03NOV1982 SBAS1V
C     ******************************************************************
C     PRINT VOLUMETRIC BUDGET
C     ******************************************************************
C
C     SPECIFICATIONS:
C     ------------------------------------------------------------------
      DIMENSION VBVL(4,20)
CF66
C     DIMENSION VBNM(4,20)
CF66
CF77
      CHARACTER*4 VBNM(4,20)
CF77
C     ------------------------------------------------------------------
C
C1------DETERMINE NUMBER OF INDIVIDUAL BUDGET ENTRIES.
      MSUM1=MSUM-1
      IF(MSUM1.LE.0) RETURN
C
C2------CLEAR RATE AND VOLUME ACCUMULATORS.
      TOTRIN=0.
      TOTROT=0.
      TOTVIN=0.
      TOTVOT=0.
C
C3------ADD RATES AND VOLUMES (IN AND OUT) TO ACCUMULATORS.
      DO 100 L=1,MSUM1
      TOTRIN=TOTRIN+VBVL(3,L)
      TOTROT=TOTROT+VBVL(4,L)
      TOTVIN=TOTVIN+VBVL(1,L)
      TOTVOT=TOTVOT+VBVL(2,L)
  100 CONTINUE
C
```

```
C4------PRINT TIME STEP NUMBER AND STRESS PERIOD NUMBER.
      WRITE(IOUT,260) KSTP,KPER
      WRITE(IOUT,265)
C
C5------PRINT INDIVIDUAL INFLOW RATES AND VOLUMES AND THEIR TOTALS.
      DO 200 L=1,MSUM1
      WRITE(IOUT,275) (VBNM(I,L),I=1,4),VBVL(1,L),(VBNM(I,L),I=1,4)
     1,VBVL(3,L)
  200 CONTINUE
      WRITE(IOUT,286) TOTVIN,TOTRIN
C
C6------PRINT INDIVIDUAL OUTFLOW RATES AND VOLUMES AND THEIR TOTALS.
      WRITE(IOUT,287)
      DO 250 L=1,MSUM1
      WRITE(IOUT,275) (VBNM(I,L),I=1,4),VBVL(2,L),(VBNM(I,L),I=1,4)
     1,VBVL(4,L)
  250 CONTINUE
      WRITE(IOUT,298) TOTVOT,TOTROT
C
C7------CALCULATE THE DIFFERENCE BETWEEN INFLOW AND OUTFLOW.
C
C7A-----CALCULATE DIFFERENCE BETWEEN RATE IN AND RATE OUT.
      DIFFR=TOTRIN-TOTROT
C
C7B-----CALCULATE PERCENT DIFFERENCE BETWEEN RATE IN AND RATE OUT.
      PDIFFR=100.*DIFFR/((TOTRIN+TOTROT)/2)
C
C7C-----CALCULATE DIFFERENCE BETWEEN VOLUME IN AND VOLUME OUT.
      DIFFV=TOTVIN-TOTVOT
C
C7D-----GET PERCENT DIFFERENCE BETWEEN VOLUME IN AND VOLUME OUT.
      PDIFFV=100.*DIFFV/((TOTVIN+TOTVOT)/2)
C
C8------PRINT DIFFERENCES AND PERCENT DIFFERENCES BETWEEN INPUT
C8------AND OUTPUT RATES AND VOLUMES.
      WRITE(IOUT,299) DIFFV,DIFFR
      WRITE(IOUT,300) PDIFFV,PDIFFR
C
C9------RETURN
      RETURN
C
C    ---FORMATS
C
  260 FORMAT(1H0,///30X,'VOLUMETRIC BUDGET FOR ENTIRE MODEL AT END OF'
     1,' TIME STEP',I3,' IN STRESS PERIOD',I3/30X,77('-'))
  265 FORMAT(1H0,19X,'CUMULATIVE VOLUMES',6X,'L**3',37X
     1,'RATES FOR THIS TIME STEP',6X,'L**3/T'/20X,18('-'),47X,24('-')
     2//26X,'IN:',68X,'IN:'/26X,'---',68X,'---')
  275 FORMAT(1X,18X,4A4,' =',G14.5,39X,4A4,' =',G14.5)
  286 FORMAT(1H0,26X,'TOTAL IN =',G14.5,47X,'TOTAL IN ='
     1,G14.5)
  287 FORMAT(1H0,24X,'OUT:',67X,'OUT:'/25X,4('-'),67X,4('-'))
  298 FORMAT(1H0,25X,'TOTAL OUT =',G14.5,46X,'TOTAL OUT ='
     1,G14.5)
  299 FORMAT(1H0,26X,'IN - OUT =',G14.5,47X,'IN - OUT =',G14.5)
  300 FORMAT(1H0,15X,'PERCENT DISCREPANCY =',F20.2
     1,30X,'PERCENT DISCREPANCY =',F20.2,///)
C
      END
```

```
$DEBUG
$LARGE
$NOFLOATCALLS
      SUBROUTINE WEL1AL(ISUM,LENX,LCWELL,MXWELL,NWELLS,IN,IOUT,
     1                       IWELCB)
C
C-----VERSION 0933 08DEC1983 WEL1AL
C     ******************************************************************
C     ALLOCATE ARRAY STORAGE FOR WELL PACKAGE
C     ******************************************************************
C
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
      INTEGER*4 LENX
C     ------------------------------------------------------------------
C
C1------IDENTIFY PACKAGE AND INITIALIZE NWELLS
      WRITE(IOUT,1)IN
    1 FORMAT(1H0,'WEL1 -- WELL PACKAGE, VERSION 1, 12/08/83',
     2' INPUT READ FROM',I3)
      NWELLS=0
C
C2------READ MAX NUMBER OF WELLS AND
C2------UNIT OR FLAG FOR CELL-BY-CELL FLOW TERMS.
      READ(IN,2) MXWELL,IWELCB
    2 FORMAT(2I10)
      WRITE(IOUT,3) MXWELL
    3 FORMAT(1H ,'MAXIMUM OF',I5,' WELLS')
      IF(IWELCB.GT.0) WRITE(IOUT,9) IWELCB
    9 FORMAT(1X,'CELL-BY-CELL FLOWS WILL BE RECORDED ON UNIT',I3)
      IF(IWELCB.LT.0) WRITE(IOUT,8)
    8 FORMAT(1X,'CELL-BY-CELL FLOWS WILL BE PRINTED WHEN ICBCFL NOT 0')
C
C3------SET LCWELL EQUAL TO LOCATION OF WELL LIST IN X ARRAY.
      LCWELL=ISUM
C
C4------ADD AMOUNT OF SPACE USED BY WELL LIST TO ISUM.
      ISP=4*MXWELL
      ISUM=ISUM+ISP
C
C5------PRINT NUMBER OF SPACES IN X ARRAY USED BY WELL PACKAGE.
      WRITE(IOUT,4) ISP
    4 FORMAT(1X,I6,' ELEMENTS IN X ARRAY ARE USED FOR WELLS')
      ISUM1=ISUM-1
      WRITE(IOUT,5) ISUM1,LENX
    5 FORMAT(1X,I6,' ELEMENTS OF X ARRAY USED OUT OF',I7)
C
C6------IF THERE ISN'T ENOUGH SPACE IN THE X ARRAY THEN PRINT
C6------A WARNING MESSAGE.
      IF(ISUM1.GT.LENX) WRITE(IOUT,6)
    6 FORMAT(1X,'   ***X ARRAY MUST BE DIMENSIONED LARGER***')
C7------RETURN
      RETURN
      END
      SUBROUTINE WEL1RP(WELL,NWELLS,MXWELL,IN,IOUT)
C
C
C-----VERSION 1544 22DEC1982 WEL1RP
C     ******************************************************************
C     READ NEW WELL LOCATIONS AND STRESS RATES
C     ******************************************************************
C
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
      DIMENSION WELL(4,MXWELL)
C     ------------------------------------------------------------------
C
C1------READ ITMP(NUMBER OF WELLS OR FLAG SAYING REUSE WELL DATA)
      READ (IN,1) ITMP
    1 FORMAT(I10)
      IF(ITMP.GE.0) GO TO 50
C
C1A-----IF ITMP LESS THAN ZERO REUSE DATA. PRINT MESSAGE AND RETURN.
      WRITE(IOUT,6)
    6 FORMAT(1H0,'REUSING WELLS FROM LAST STRESS PERIOD')
      RETURN
C
C1B-----ITMP=>0.  SET NWELLS EQUAL TO ITMP.
   50 NWELLS=ITMP
      IF(NWELLS.LE.MXWELL) GO TO 100
C
C2------NWELLS>MXWELL.  PRINT MESSAGE. STOP.
      WRITE(IOUT,99) NWELLS,MXWELL
   99 FORMAT(1H0,'NWELLS(',I4,') IS GREATER THAN MXWELL(',I4,')')
      STOP
C
C3------PRINT NUMBER OF WELLS IN CURRENT STRESS PERIOD.
  100 WRITE (IOUT,2) NWELLS
    2 FORMAT(1H0,10X,I4,' WELLS')
C
C4------IF THERE ARE NO ACTIVE WELLS IN THIS STRESS PERIOD THEN RETURN
      IF(NWELLS.EQ.0) GO TO 260
C
C5------READ AND PRINT LAYER,ROW,COLUMN AND RECHARGE RATE.
      WRITE(IOUT,3)
    3 FORMAT(1H ,47X,'LAYER    ROW    COL    STRESS RATE   WELL NO.'/
     1,48X,45('-'))
      DO 250 II=1,NWELLS
      READ (IN,4) K,I,J,Q
    4 FORMAT(3I10,F10.0)
      WRITE (IOUT,5) K,I,J,Q,II
    5 FORMAT(48X,I3,I8,I7,G16.5,I8)
      WELL(1,II)=K
      WELL(2,II)=I
      WELL(3,II)=J
      WELL(4,II)=Q
  250 CONTINUE
C
```

```
C6------RETURN
  260 RETURN
      END
      SUBROUTINE WELPRE(NWELLS,MXWELL,WELL,IBOUND,DELR,DELC,
     1        NCOL,NROW,NLAY,IVOUT,CONVRT,LLX,LUX,LLY,LUY)
C
C       ROUTINE TO ADD WELLS TO SINKS LIST MODIFED FROM
C----- USGS 3D MODFLOW MODEL, VERSION 1001 26AUG1982 WEL1FM
C
C     ******************************************************************
C     ADD WELLS TO LIST OF SINKS
C     ******************************************************************
C
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
      DIMENSION WELL(4,MXWELL),DELR(NCOL),DELC(NROW),
     1            IBOUND(NCOL,NROW,NLAY)
C     ------------------------------------------------------------------
C1------IF NUMBER OF WELLS <= 0 THEN RETURN.
      IF(NWELLS.LE.0) RETURN
C
C2------PROCESS EACH WELL IN THE WELL LIST.
      DO 100 L=1,NWELLS
      IR=WELL(2,L)
      IC=WELL(3,L)
      IL=WELL(1,L)
      Q=WELL(4,L)
C
C2A-----IF THE CELL IS INACTIVE THEN BYPASS PROCESSING.
      IF(IBOUND(IC,IR,IL).LE.0) GO TO 100
      IF(IC .LT. LLX .OR. IC .GT. LUX)GO TO 100
      IF(IR .LT. LLY .OR. IR .GT. LUY)GO TO 100
C
C2B-----IF THE CELL IS VARIABLE HEAD THEN ADD RECHARGE RATE
C       TO THE RHS ACCUMULATOR.
      CALL XYZPOS(IC,IR,IL,X,Y,Z,DELR,DELC,NROW,NCOL,NLAY)
      IF(Q .LT. 0)WRITE(IVOUT,1005)X,Y,Z,-Q*CONVRT
 1005 FORMAT(3F12.1,G12.4)
  100 CONTINUE
C
C3------RETURN
      RETURN
      END
```

```
$LARGE
$NOFLOATCALLS
      SUBROUTINE EVT1AL(ISUM,LENX,LCIEVT,LCEVTR,LCEXDP,LCSURF,
     1                  NCOL,NROW,NEVTOP,IN,IOUT,IEVTCB)
C
C-----VERSION 0943 08DEC1983 EVT1AL
C     ******************************************************************
C     ALLOCATE ARRAY STORAGE FOR EVAPOTRANSPIRATION
C     ******************************************************************
C
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
      INTEGER*4 LENX
C     ------------------------------------------------------------------
C
C1------IDENTIFY PACKAGE.
      WRITE(IOUT,1)IN
    1 FORMAT(1H0,'EVT1 -- EVAPOTRANSPIRATION PACKAGE, VERSION 1,',
     1     ' 12/08/83',' INPUT READ FROM UNIT',I3)
C
C2------READ NEVTOP AND IEVTCB.
      READ(IN,3)NEVTOP,IEVTCB
    3 FORMAT(2I10)
C
C3------CHECK TO SEE THAT ET OPTION IS LEGAL.
      IF(NEVTOP.GE.1.AND.NEVTOP.LE.2)GO TO 200
C
C3A-----IF ILLEGAL PRINT A MESSAGE & ABORT SIMULATION.
      WRITE(IOUT,8)
    8 FORMAT(1X,'ILLEGAL ET OPTION CODE. SIMULATION ABORTING')
      STOP
C
C4------IF THE OPTION IS LEGAL THEN PRINT THE OPTION CODE.
  200 IF(NEVTOP.EQ.1) WRITE(IOUT,201)
  201 FORMAT(1X,'OPTION 1 -- EVAPOTRANSPIRATION FROM TOP LAYER')
      IF(NEVTOP.EQ.2) WRITE(IOUT,202)
  202 FORMAT(1X,'OPTION 2 -- EVAPOTRANSPIRATION FROM ONE SPECIFIED',
     1      ' NODE IN EACH VERTICAL COLUMN')
      IRK=ISUM
C
C5------IF CELL-BY-CELL TERMS TO BE SAVED THEN PRINT UNIT NUMBER.
      IF(IEVTCB.GT.0) WRITE(IOUT,203) IEVTCB
  203 FORMAT(1X,'CELL-BY-CELL FLOW TERMS WILL BE SAVED ON UNIT',I3)
C
C6------ALLOCATE SPACE FOR THE ARRAYS EVTR, EXDP AND SURF.
      LCEVTR=ISUM
      ISUM=ISUM+NCOL*NROW
      LCEXDP=ISUM
      ISUM=ISUM+NCOL*NROW
      LCSURF=ISUM
      ISUM=ISUM+NCOL*NROW
C
C7------IF OPTION 2 THEN ALLOCATE SPACE FOR THE INDICATOR ARRAY(IEVT)
      IF(NEVTOP.NE.2)GO TO 300
      LCIEVT=ISUM
      ISUM=ISUM+NCOL*NROW
C
C8------CALCULATE & PRINT AMOUNT OF SPACE USED BY ET PACKAGE.
  300 IRK=ISUM-IRK
      WRITE(IOUT,4)IRK
    4 FORMAT(1X,I6,' ELEMENTS OF X ARRAY USED FOR EVAPOTRANSPIRATION')
      ISUM1=ISUM-1
      WRITE(IOUT,5)ISUM1,LENX
    5 FORMAT(1X,I6,' ELEMENTS OF X ARRAY USED OUT OF',I7)
      IF(ISUM1.GT.LENX)WRITE(IOUT,6)
    6 FORMAT(1X,'   ***X ARRAY MUST BE MADE LARGER***')
C
C9------RETURN.
      RETURN
      END
      SUBROUTINE EVT1RP(NEVTOP,IEVT,EVTR,EXDP,SURF,DELR,DELC,
     1                  NCOL,NROW,NLAY,IN,IOUT)
C
C-----VERSION 1631 08FEB1983 EVT1RP
C     ******************************************************************
C     READ EVAPOTRANSPIRATION DATA
C     ******************************************************************
C
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
      DIMENSION IEVT(NCOL,NROW),EVTR(NCOL,NROW),EXDP(NCOL,NROW),
     1          SURF(NCOL,NROW),DELR(NCOL),DELC(NROW)
CF66
C     DIMENSION ANAME(6,4)
CF66
CF77
      CHARACTER*4 ANAME(6,4)
CF77
C
      DATA ANAME(1,1),ANAME(2,1),ANAME(3,1),ANAME(4,1),ANAME(5,1),
     1  ANAME(6,1) /'    ','    ','  ET',' LAY','ER I','NDEX'/
      DATA ANAME(1,2),ANAME(2,2),ANAME(3,2),ANAME(4,2),ANAME(5,2),
     1  ANAME(6,2) /'    ','    ','    ','  ET',' SUR','FACE'/
      DATA ANAME(1,3),ANAME(2,3),ANAME(3,3),ANAME(4,3),ANAME(5,3),
     1  ANAME(6,3) /' EVA','POTR','ANSP','IRAT','ION ','RATE'/
      DATA ANAME(1,4),ANAME(2,4),ANAME(3,4),ANAME(4,4),ANAME(5,4),
     1  ANAME(6,4) /'    ','    ','EXTI','NCTI','ON D','EPTH'/
C     ------------------------------------------------------------------
C
C1------READ FLAGS SHOWING WHETHER DATA IS TO BE REUSED.
      READ(IN,6)INSURF,INEVTR,INEXDP,INIEVT
    6 FORMAT(4I10)
C
C2------TEST INSURF TO SEE WHERE SURFACE ELEVATION COMES FROM.
      IF(INSURF.GE.0)GO TO 32
C
C2A------IF INSURF<0 THEN REUSE SURFACE ARRAY FROM LAST STRESS PERIOD
      WRITE(IOUT,3)
    3 FORMAT(1H0,'REUSING SURF FROM LAST STRESS PERIOD')
      GO TO 35
C
C3-------IF INSURF=>0 THEN CALL MODULE U2DREL TO READ SURFACE.
   32 CALL U2DREL(SURF,ANAME(1,2),NROW,NCOL,0,IN,IOUT)
C
C4------TEST INEVTR TO SEE WHERE MAX ET RATE COMES FROM.
   35 IF(INEVTR.GE.0)GO TO 37
C
C4A-----IF INEVTR<0 THEN REUSE MAX ET RATE.
      WRITE(IOUT,4)
    4 FORMAT(1H0,'REUSING EVTR FROM LAST STRESS PERIOD')
      GO TO 45
C
C5------IF INEVTR=>0 CALL MODULE U2DREL TO READ MAX ET RATE.
   37 CALL U2DREL(EVTR,ANAME(1,3),NROW,NCOL,0,IN,IOUT)
C
C6------MULTIPLY MAX ET RATE BY CELL AREA TO GET VOLUMETRIC RATE
C            STATEMENT MODIFIED FOR PREMOD3D
      DO 40 IR=1,NROW
      DO 40 IC=1,NCOL
      EVTR(IC,IR)=EVTR(IC,IR)
   40 CONTINUE
C
C7------TEST INEXDP TO SEE WHERE EXTINCTION DEPTH COMES FROM
   45 IF(INEXDP.GE.0)GO TO 47
C
C7A------IF INEXDP<0 REUSE EXTINCTION DEPTH FROM LAST STRESS PERIOD
      WRITE(IOUT,5)
    5 FORMAT(1H0,'REUSING EXDP FROM LAST STRESS PERIOD')
      GO TO 48
C
C8-------IF INEXDP=>0 CALL MODULE U2DREL TO READ EXTINCTION DEPTH
   47 CALL U2DREL(EXDP,ANAME(1,4),NROW,NCOL,0,IN,IOUT)
C
C9------IF OPTION(NEVTOP) IS 2 THEN WE NEED AN INDICATOR ARRAY.
   48 IF(NEVTOP.NE.2)GO TO 50
C
C10------IF INIEVT<0 THEN REUSE LAYER INDICATOR ARRAY.
      IF(INIEVT.GE.0)GO TO 49
      WRITE(IOUT,2)
    2 FORMAT(1H0,'REUSING IEVT FROM LAST STRESS PERIOD')
      GO TO 50
C
C11------IF INIEVT=>0 THEN CALL MODULE U2DINT TO READ INDICATOR ARRAY.
   49 CALL U2DINT(IEVT,ANAME(1,1),NROW,NCOL,0,IN,IOUT)
C
C12-----RETURN
   50 RETURN
      END
      SUBROUTINE PREEVT(NEVTOP,IEVT,EVTR,EXDP,SURF,ZVEL,
     1                 IBOUND,HNEW,NCOL,NROW,NLAY)
C
C-----MODIFIED FOR WATER TABLE VELOCITY CALC FROM
```

```
C        VERSION 0835 10FEB1983 EVT1FM OF USGS3D MODEL
C     ******************************************************************
C        ADD EVAPOTRANSPIRATION TO RHS AND HCOF
C     ******************************************************************
C
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
      DOUBLE PRECISION HNEW
      DIMENSION IEVT(NCOL,NROW),EVTR(NCOL,NROW),EXDP(NCOL,NROW),
     1          SURF(NCOL,NROW),ZVEL(NCOL,NROW),
     2          IBOUND(NCOL,NROW,NLAY),
     3          HNEW(NCOL,NROW,NLAY)
C     ------------------------------------------------------------------
C
C1------PROCESS EACH HORIZONTAL CELL LOCATION
      DO 10 IR=1,NROW
      DO 10 IC=1,NCOL
C
C2------SET THE LAYER INDEX EQUAL TO 1
      IL=1
C
C        THIS OPTION IS INOPERATIVE IN RAND3D MODEL
C3------IF OPTION 2 IS SPECIFIED THEN GET LAYER INDEX FROM IEVT ARRAY
      IF(NEVTOP.EQ.2)IL=IEVT(IC,IR)
C
C4------IF THE CELL IS EXTERNAL IGNORE IT.
      IF(IBOUND(IC,IR,IL).LE.0)GO TO 10
      IF(IC .LT. LLX .OR. IC .GT. LUX)GO TO 10
      IF(IR .LT. LLY .OR. IR .GT. LUY)GO TO 10
      C=EVTR(IC,IR)
      S=SURF(IC,IR)
      H=HNEW(IC,IR,IL)
C
C5------IF AQUIFER HEAD IS GREATER THAN OR EQUAL TO SURF, ET IS CONSTANT
      IF(H.LT.S) GO TO 5
C
C5A-----SUBTRACT -EVTR FROM RHS
      ZVEL(IC,IR)=ZVEL(IC,IR) + C
      GO TO 10
C
C6------IF DEPTH TO WATER>=EXTINCTION DEPTH THEN ET IS 0
    5 D=S-H
      X=EXDP(IC,IR)
      IF(D.GE.X)GO TO 10
C
C7------LINEAR RANGE. ADD ET TERMS TO BOTH RHS AND HCOF  Q=-EVTR(H-EXEL)/EXDP
      ZVEL(IC,IR)=ZVEL(IC,IR)-C*D/X+C

   10 CONTINUE
C
C8------RETURN
      RETURN
      END
```

```
$LARGE
$NOFLCATCALLS
      SUBROUTINE GHB1AL(ISUM,LENX,LCBNDS,NBOUND,MXBND,IN,IOUT,
     1                       IGHBCB)
C
C-----VERSION 0940 08DEC1983 GHB1AL
C     ******************************************************************
C     ALLOCATE ARRAY STORAGE FOR HEAD-DEPENDENT BOUNDARIES
C     ******************************************************************
C
C     SPECIFICATIONS:
C     ------------------------------------------------------------------
      INTEGER*4 LENX
C     ------------------------------------------------------------------
C
C1------IDENTIFY PACKAGE AND INITIALIZE # OF GENERAL HEAD BOUNDS
      WRITE(IOUT,1)IN
    1 FORMAT(1H0,'GHB1 -- GHB PACKAGE, VERSION 1, 12/08/83',
     2' INPUT READ FROM UNIT',I3)
      NBOUND=0
C
C2------READ AND PRINT MXBND AND IGHBCB (MAX # OF BOUNDS AND UNIT
C2------FOR CELL-BY-CELL FLOW TERMS FOR GHB)
      READ(IN,2) MXBND,IGHBCB
    2 FORMAT(2I10)
      WRITE(IOUT,3) MXBND
    3 FORMAT(1H ,'MAXIMUM OF',I5,' HEAD-DEPENDENT BOUNDARY NODES')
      IF(IGHBCB.GT.0) WRITE(IOUT,9) IGHBCB
    9 FORMAT(1X,'CELL-BY-CELL FLOW WILL BE RECORDED ON UNIT',I3)
      IF(IGHBCB.LT.0) WRITE(IOUT,8)
    8 FORMAT(1X,'CELL-BY-CELL FLOW WILL BE PRINTED WHEN ICBCFL NOT 0')
C
C3------SET LCBNDS EQUAL TO ADDRESS OF FIRST UNUSED SPACE IN X.
      LCBNDS=ISUM
C
C4------CALCULATE AMOUNT OF SPACE USED BY THE GENERAL HEAD LIST.
      ISP=5*MXBND
      ISUM=ISUM+ISP
C
C5------PRINT AMOUNT OF SPACE USED BY THE GHB PACKAGE
      WRITE(IOUT,4) ISP
    4 FORMAT(1X,I6,' ELEMENTS IN X ARRAY ARE USED FOR HEAD',
     1      '-DEPENDENT BOUNDARIES')
      ISUM1=ISUM-1
      WRITE(IOUT,5) ISUM1,LENX
    5 FORMAT(1X,I6,' ELEMENTS OF X ARRAY USED OUT OF',I7)
      IF(ISUM1.GT.LENX) WRITE(IOUT,6)
    6 FORMAT(1X,'   ***X ARRAY MUST BE DIMENSIONED LARGER***')
C
C6------RETURN
      RETURN
      END
      SUBROUTINE GHB1RP(BNDS,NBOUND,MXBND,IN,IOUT)
C
```

```
C
C-----VERSION 1651 02FEB1983 GHB1RP
C     ******************************************************************
C     READ DATA FOR GHB
C     ******************************************************************
C
C     SPECIFICATIONS:
C     ------------------------------------------------------------------
      DIMENSION BNDS(5,MXBND)
C     ------------------------------------------------------------------
C
C1------READ ITMP(# OF GENERAL HEAD BOUNDS OR FLAG TO REUSE DATA.)
      READ(IN,8) ITMP
    8 FORMAT(I10)
C
C2------TEST ITMP
      IF(ITMP.GE.0) GO TO 50
C
C2A-----IF ITMP<0 THEN REUSE DATA FROM LAST STRESS PERIOD
      WRITE(IOUT,7)
    7 FORMAT(1H0,'REUSING HEAD-DEPENDENT BOUNDS FROM LAST STRESS',
     1      ' PERIOD')
      GO TO 260
C
C3------IF ITMP=>0 THEN IT IS THE # OF GENERAL HEAD BOUNDS.
   50 NBOUND=ITMP
C
C4------IF MAX NUMBER OF BOUNDS IS EXCEEDED THEN STOP
      IF(NBOUND.LE.MXBND) GO TO 100
      WRITE(IOUT,99) NBOUND,MXBND
   99 FORMAT(1H0,'NBOUND(',I4,') IS GREATER THAN MXBND(',I4,')')
C
C4A-----ABNORMAL STOP
      STOP
C
C5------PRINT # OF GENERAL HEAD BOUNDS THIS STRESS PERIOD
  100 WRITE(IOUT,1) NBOUND
    1 FORMAT(1H0,//1X,I5,' HEAD-DEPENDENT BOUNDARY NODES')
C
C6------IF THERE ARE NO GENERAL HEAD BOUNDS THEN RETURN.
      IF(NBOUND.EQ.0) GO TO 260
C
C7------READ & PRINT DATA FOR EACH GENERAL HEAD BOUNDARY.
      WRITE(IOUT,3)
    3 FORMAT(1H0,15X,'LAYER',5X,'ROW',5X
     1,'COL   ELEVATION   CONDUCTANCE   BOUND NO.'/1X,15X,60('-'))
      DO 250 II=1,NBOUND
      READ (IN,4) K,I,J,BNDS(4,II),BNDS(5,II)
    4 FORMAT(3I10,2F10.0)
      WRITE (IOUT,5) K,I,J,BNDS(4,II),BNDS(5,II),II
    5 FORMAT(1X,15X,I4,I9,I8,G13.4,G14.4,I8)
      BNDS(1,II)=K
      BNDS(2,II)=I
      BNDS(3,II)=J
  250 CONTINUE
C
C8------RETURN
  260 RETURN
      END
      SUBROUTINE GHBPRE(NBOUND,MXBND,BNDS,IBOUND,HNEW,
     1           DELR,DELC,NCOL,NROW,NLAY,IVOUT,CONVRT,
     2           LLX,LUX,LLY,LUY)
C
C       CREATE SINKS AT GENERAL HEAD BOUNDARIES MODIFIED FROM USGS 3D
C-----VERSION 1605 02FEB1983 GHB1FM
C     ******************************************************************
C     ADD GHB BOUNDARIES TO SINK LIST
C     ******************************************************************
C
C     SPECIFICATIONS:
C     ------------------------------------------------------------------
      DOUBLE PRECISION HNEW
      DIMENSION BNDS(5,MXBND),HNEW(NCOL,NROW,NLAY),
     1         DELR(NCOL),DELC(NROW),IBOUND(NCOL,NROW,NLAY)
C     ------------------------------------------------------------------
C
C1------IF NBOUND<=0 THEN THERE ARE NO GENERAL HEAD BOUNDS. RETURN.
      IF(NBOUND.LE.0) RETURN
C
C2------PROCESS EACH ENTRY IN THE GENERAL HEAD BOUND LIST (BNDS)
      DO 100 L=1,NBOUND
C
C3------GET COLUMN, ROW AND LAYER OF CELL CONTAINING BOUNDARY
      IL=BNDS(1,L)
      IR=BNDS(2,L)
      IC=BNDS(3,L)
C
C4------IF THE CELL IS EXTERNAL THEN SKIP IT.
      IF(IBOUND(IC,IR,IL).LE.0) GO TO 100
      IF(IC .LT. LLX .OR. IC .GT. LUX)GO TO 100
      IF(IR .LT. LLY .OR. IR .GT. LUY)GO TO 100
C
C5------SINCE THE CELL IS INTERNAL GET THE BOUNDARY DATA.
      HB=BNDS(4,L)
      C=BNDS(5,L)
C
C6------COMPUTE RATE OF FLOW
      RATE=-C*(HNEW(IC,IR,IL)-HB)
      CALL XYZPOS(IC,IR,IL,X,Y,Z,DELR,DELC,NROW,NCOL,NLAY)
      IF(RATE .LT. O)WRITE(6,1005)X,Y,Z,-RATE*CONVRT
 1005 FORMAT(3F12.1,G12.4)
  100 CONTINUE
C
C7------RETURN
      RETURN
      END
```

```
$DEBUG
$LARGE
$NOFLOATCALLS
      SUBROUTINE RIV1AL(ISUM,LENX,LCRIVR,MXRIVR,NRIVER,IN,IOUT,
     1                      IRIVCB)
C
C-----VERSION 0935 08DEC1983 RIV1AL
C     *****************************************************************
C     ALLOCATE ARRAY STORAGE FOR RIVERS
C     *****************************************************************
C
C     SPECIFICATIONS:
C     -----------------------------------------------------------------
      INTEGER*4 LENX
C     -----------------------------------------------------------------
C
C1------IDENTIFY PACKAGE AND INITIALIZE NRIVER.
      WRITE(IOUT,1)IN
    1 FORMAT(1H0,'RIV1 -- RIVER PACKAGE, VERSION 1,  12/08/83',
     2' INPUT READ FROM UNIT',I3)
      NRIVER=0
C
C2------READ & PRINT MXRIVR & IRIVCB(UNIT OR FLAG FOR C-B-C FLOWS)
      READ(IN,2)MXRIVR,IRIVCB
    2 FORMAT(2I10)
      WRITE(IOUT,3)MXRIVR
    3 FORMAT(1H ,'MAXIMUM OF',I5,' RIVER NODES')
      IF(IRIVCB.GT.0) WRITE(IOUT,9) IRIVCB
    9 FORMAT(1X,'CELL-BY-CELL FLOWS WILL BE RECORDED ON UNIT',I3)
      IF(IRIVCB.LT.0) WRITE(IOUT,8)
    8 FORMAT(1X,'CELL-BY-CELL FLOWS WILL BE PRINTED')
C
C3------SET LCRIVR EQUAL TO ADDRESS OF FIRST UNUSED SPACE IN X.
      LCRIVR=ISUM
C
C4------CALCULATE AMOUNT OF SPACE USED BY RIVER LIST.
      ISP=6*MXRIVR
      ISUM=ISUM+ISP
C
C5------PRINT AMOUNT OF SPACE USED BY RIVER PACKAGE.
      WRITE (IOUT,4)ISP
    4 FORMAT(1X,I6,' ELEMENTS IN X ARRAY ARE USED FOR RIVERS')
      ISUM1=ISUM-1
      WRITE(IOUT,5)ISUM1,LENX
    5 FORMAT(1X,I6,' ELEMENTS OF X ARRAY USED OUT OF',I7)
      IF(ISUM1.GT.LENX) WRITE(IOUT,6)
    6 FORMAT(1X,'   ***X ARRAY MUST BE DIMENSIONED LARGER***')
C
C7------RETURN
      RETURN
      END
      SUBROUTINE RIV1RP(RIVR,NRIVER,MXRIVR,IN,IOUT)
C
C
```

```
C-----VERSION 1319 25AUG1982 RIV1RP
C     ******************************************************************
C     READ RIVER HEAD, CONDUCTANCE AND BOTTOM ELEVATION
C     ******************************************************************
C
C     SPECIFICATIONS:
C     ------------------------------------------------------------------
      DIMENSION RIVR(6,MXRIVR)
C     ------------------------------------------------------------------
C
C
C1------READ ITMP(NUMBER OF RIVER REACHES OR FLAG TO REUSE DATA)
      READ(IN,8)ITMP
    8 FORMAT(I10)
C
C2------TEST ITMP.
      IF(ITMP.GE.0)GO TO 50
C
C2A-----IF ITMP <0 THEN REUSE DATA FROM LAST STRESS PERIOD.
      WRITE(IOUT,7)
    7 FORMAT(1H0,'REUSING RIVER REACHES FROM LAST STRESS PERIOD')
      GO TO 260
C
C3------IF ITMP=> ZERO THEN IT IS THE NUMBER OF RIVER REACHES
   50 NRIVER=ITMP
C
C4------IF NRIVER>MXRIVR THEN STOP.
      IF(NRIVER.LE.MXRIVR)GO TO 100
      WRITE(IOUT,99)NRIVER,MXRIVR
   99 FORMAT(1H0,'NRIVER(',I4,') IS GREATER THAN MXRIVR(',I4,')')
C
C4A-----ABNORMAL STOP.
      STOP
C
C5------PRINT NUMBER OF RIVER REACHES IN THIS STRESS PERIOD.
  100 WRITE(IOUT,1)NRIVER
    1 FORMAT(1H0,//1X,I5,' RIVER REACHES')
C
C6------IF THERE ARE NO RIVER REACHES THEN RETURN.
      IF(NRIVER.EQ.0) GO TO 260
C
C7------READ AND PRINT DATA FOR EACH RIVER REACH.
      WRITE(IOUT,3)
    3 FORMAT(1H0,15X,'LAYER',5X,'ROW',5X,'COL    '
     1,'  STAGE     CONDUCTANCE   BOTTOM ELEVATION   RIVER REACH'
     2/1X,15X,80('-'))
      DO 250 II=1,NRIVER
      READ(IN,4)K,I,J,RIVR(4,II),RIVR(5,II),RIVR(6,II)
    4 FORMAT(3I10,3F10.0)
      WRITE(IOUT,5)K,I,J,RIVR(4,II),RIVR(5,II),RIVR(6,II),II
    5 FORMAT(1X,15X,I4,I9,I8,G13.4,G14.4,G19.4,I10)
      RIVR(1,II)=K
      RIVR(2,II)=I
      RIVR(3,II)=J
  250 CONTINUE
C
C8------RETURN
  260 RETURN
      END
      SUBROUTINE RIVPRE(NRIVER,MXRIVR,RIVR,HNEW,IBOUND,
     1           DELR,DELC,NCOL,NROW,NLAY,IVOUT,CONVRT,
     2           LLX,LUX,LLY,LUY)
C
C       ROUTINE TO ADD RIVER NODES TO SINK LIST MODIFED FROM USGS 3D
C-----VERSION 0915 27AUG1982 RIV1FM
C     ******************************************************************
C     ADD RIVER TERMS TO SINK LIST
C     ******************************************************************
C
C     SPECIFICATIONS:
C     ------------------------------------------------------------------
C
      DOUBLE PRECISION HNEW
      DIMENSION RIVR(6,MXRIVR),HNEW(NCOL,NROW,NLAY),
     1          DELR(NCOL),DELC(NROW),
     2          IBOUND(NCOL,NROW,NLAY)
C     ------------------------------------------------------------------
C
C
C1------IF NRIVER<=0 THERE ARE NO RIVERS. RETURN.
      IF(NRIVER.LE.0)RETURN
C
C2------PROCESS EACH CELL IN THE RIVER LIST.
      DO 100 L=1,NRIVER
C
C3------GET COLUMN, ROW, AND LAYER OF CELL CONTAINING REACH
      IL=RIVR(1,L)
      IR=RIVR(2,L)
      IC=RIVR(3,L)
C
C4------IF THE CELL IS EXTERNAL SKIP IT.
      IF(IBOUND(IC,IR,IL).LE.0)GO TO 100
      IF(IC .LT. LLX .OR. IC .GT. LUX)GO TO 100
      IF(IR .LT. LLY .OR. IR .GT. LUY)GO TO 100
C
C5------SINCE THE CELL IS INTERNAL GET THE RIVER DATA.
      HRIV=RIVR(4,L)
      CRIV=RIVR(5,L)
      RBOT=RIVR(6,L)
      HHNEW=HNEW(IC,IR,IL)
C
C6------COMPARE AQUIFER HEAD TO BOTTOM OF STREAM BED.
      IF(HHNEW.LE.RBOT)GO TO 96
C
C7------SINCE HEAD>BOTTOM CREATE SINK.
      RATE=-CRIV*(HHNEW-HRIV)
      GO TO 98
C
```

```
C8------SINCE HEAD<BOTTOM NO SINK.
   96 RATE=CRIV*(HRIV-RBOT)
   98 CALL XYZPOS(IC,IR,IL,X,Y,Z,DELR,DELC,NROW,NCOL,NLAY)
      IF(RATE .LT. 0)WRITE(IVOUT,1005)X,Y,Z,-RATE*CONVRT
 1005 FORMAT(3F12.1,G12.4)
  100 CONTINUE
C
C9------RETURN
      RETURN
      END
      SUBROUTINE PRERIV(NRIVER,MXRIVR,RIVR,HNEW,IBOUND,RBOT,
     1           DELR,DELC,ZVEL,NCOL,NROW,NLAY,IVOUT,CONVRT)
C
C      ROUTINE TO COMPUTE VELOCITY OF WATER TABLE IN RIVER NODES
C        MODIFED FROM USGS 3D
C-----VERSION 0915 27AUG1982 RIV1FM
C     ******************************************************************
C     COMPUTE WATER TABLE VELOCITY UNDER RIVERS AND FILL RBOT ARRAY
C     ******************************************************************
C
C     SPECIFICATIONS:
C     ------------------------------------------------------------------
C
      DOUBLE PRECISION HNEW
      DIMENSION RIVR(6,MXRIVR),HNEW(NCOL,NROW,NLAY),
     1          DELR(NCOL),DELC(NROW),ZVEL(NCOL,NROW),
     2          IBOUND(NCOL,NROW,NLAY),RBOT(NCOL,NROW)
C     ------------------------------------------------------------------
C
C
C1------IF NRIVER<=0 THERE ARE NO RIVERS. RETURN.
      IF(NRIVER.LE.0)RETURN
C
C2------PROCESS EACH CELL IN THE RIVER LIST.
      DO 100 L=1,NRIVER
C
C3------GET COLUMN, ROW, AND LAYER OF CELL CONTAINING REACH
      IL=RIVR(1,L)
      IR=RIVR(2,L)
      IC=RIVR(3,L)
C
C4------IF THE CELL IS EXTERNAL SKIP IT.
      IF(IBOUND(IC,IR,IL).LE.0)GO TO 100
C
C5------SINCE THE CELL IS INTERNAL GET THE RIVER DATA.
      HRIV=RIVR(4,L)
      CRIV=RIVR(5,L)
      RBOT(IC,IR)=RIVR(6,L)
      HHNEW=HNEW(IC,IR,IL)
C
C6------COMPARE AQUIFER HEAD TO BOTTOM OF STREAM BED.
      IF(HHNEW.LE.RBOT(IC,IR))GO TO 96
C
C7------SINCE HEAD>BOTTOM ADD TERMS TO RHS AND HCOF.
```

```
      RATE=-CRIV*(HHNEW-HRIV)
      GO TO 98
C
C8------SINCE HEAD<BOTTOM ADD TERM ONLY TO RHS.
   96 RATE=CRIV*(HRIV-RBOT(IC,IR))
   98 ZVEL(IC,IR)=ZVEL(IC,IR)-RATE/DELR(IC)/DELC(IR)
  100 CONTINUE
C
C9------RETURN
      RETURN
      END
```

```
$LARGE
$NOFLOATCALLS
      SUBROUTINE RCH1AL(ISUM,LENX,LCIRCH,LCRECH,NRCHOP,
     C                  NCOL,NROW,IN,IOUT,IRCHCB)
C
C-----VERSION 0939 08DEC1983 RCH1AL
C     ******************************************************************
C     ALLOCATE ARRAY STORAGE FOR RECHARGE
C     ******************************************************************
C
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
      INTEGER*4 LENX
C     ------------------------------------------------------------------
C
C1------IDENTIFY PACKAGE.
      WRITE(IOUT,1)IN
    1 FORMAT(1H0,'RCH1 -- RECHARGE PACKAGE, VERSION 1, 12/08/83',
     2' INPUT READ FROM UNIT',I3)
C
C2------READ NRCHOP AND IRCHCB.
      READ(IN,2)NRCHOP,IRCHCB
    2 FORMAT(2I10)
C
C3------CHECK TO SEE THAT OPTION IS LEGAL.
      IF(NRCHOP.GE.1.AND.NRCHOP.LE.3)GO TO 200
C
C3A-----IF ILLEGAL PRINT A MESSAGE AND ABORT SIMULATION
      WRITE(IOUT,8)
    8 FORMAT(1X,'ILLEGAL OPTION CODE. SIMULATION ABORTING')
      STOP
C
C4------IF OPTION IS LEGAL.PRINT OPTION CODE.
  200 IRK=ISUM
      IF(NRCHOP.EQ.1) WRITE(IOUT,201)
  201 FORMAT(1X,'OPTION 1 -- RECHARGE TO TOP LAYER')
      IF(NRCHOP.EQ.2) WRITE(IOUT,202)
  202 FORMAT(1X,'OPTION 2 -- RECHARGE TO ONE SPECIFIED NODE IN EACH',
     1       ' VERTICAL COLUMN')
      IF(NRCHOP.EQ.3) WRITE(IOUT,203)
  203 FORMAT(1X,'OPTION 3 -- RECHARGE TO HIGHEST ACTIVE NODE IN EACH',
     1       ' VERTICAL COLUMN')
C
C5------IF CELL-BY-CELL FLOW  TERMS TO BE SAVED THEN PRINT UNIT #
      IF(IRCHCB.GT.0) WRITE(IOUT,204) IRCHCB
  204 FORMAT(1X,'CELL-BY-CELL FLOW TERMS WILL BE RECORDED ON UNIT',I3)
C
C6------ALLOCATE SPACE FOR THE RECHARGE ARRAY(RECH).
      LCRECH=ISUM
      ISUM=ISUM+NCOL*NROW
C
C7------IF OPTION 2 THEN ALLOCATE SPACE FOR INDICATOR ARRAY(IRCH)
      IF(NRCHOP.NE.2)GO TO 300
      LCIRCH=ISUM
      ISUM=ISUM+NCOL*NROW
C
C8------CALCULATE AND PRINT AMOUNT OF SPACE USED BY RECHARGE.
  300 IRK=ISUM-IRK
      WRITE(IOUT,4)IRK
    4 FORMAT(1X,I6,' ELEMENTS OF X ARRAY USED FOR RECHARGE')
      ISUM1=ISUM-1
      WRITE(IOUT,5)ISUM1,LENX
    5 FORMAT(1X,I6,' ELEMENTS OF X ARRAY USED OUT OF',I7)
      IF(ISUM1.GT.LENX)WRITE(IOUT,6)
    6 FORMAT(1X,'      ***X ARRAY MUST BE MADE LARGER***')
C
C9------RETURN
      RETURN
      END
      SUBROUTINE RCH1RP(NRCHOP,IRCH,RECH,DELR,DELC,NROW,NCOL,
     C                  NLAY,IN,IOUT)
C
C-----VERSION 1513 22DEC1982 RCH1RP
C     ******************************************************************
C     READ RECHARGE RATES
C     ******************************************************************
C
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
      DIMENSION IRCH(NCOL,NROW),RECH(NCOL,NROW),DELR(NCOL),DELC(NROW)
C
CF66
C     DIMENSION ANAME(6,2)
CF66
CF77
      CHARACTER*4 ANAME(6,2)
CF77
      DATA ANAME(1,1),ANAME(2,1),ANAME(3,1),ANAME(4,1),ANAME(5,1),
     1  ANAME(6,1) /'    ','RECH','ARGE',' LAY','ER I','NDEX'/
      DATA ANAME(1,2),ANAME(2,2),ANAME(3,2),ANAME(4,2),ANAME(5,2),
     1  ANAME(6,2) /'    ','    ','    ','    ','RECH','ARGE'/
C     ------------------------------------------------------------------
C
C1------READ FLAGS SHOWING WHETHER DATA IS TO BE REUSED.
      READ(IN,4)INRECH,INIRCH
    4 FORMAT(2I10)
C
C2------TEST INRECH TO SEE WHERE RECH IS COMING FROM.
      IF(INRECH.GE.0)GO TO 32
C
C2A-----IF INRECH<0 THEN REUSE RECHARGE ARRAY FROM LAST STRESS PERIOD
      WRITE(IOUT,3)
    3 FORMAT(1H0,'REUSING RECH FROM LAST STRESS PERIOD')
      GO TO 55
C
C3------IF INRECH=>0 THEN CALL U2DREL TO READ RECHARGE RATE.
   32 CALL U2DREL(RECH,ANAME(1,2),NROW,NCOL,0,IN,IOUT)
C
```

```
C4------MULTIPLY RECHARGE RATE BY CELL AREA TO GET VOLUMETRIC RATE.
      DO 50 IR=1,NROW
      DO 50 IC=1,NCOL
      RECH(IC,IR)=RECH(IC,IR)
   50 CONTINUE
C
C5------IF NRCHOP=2 THEN A LAYER INDICATOR ARRAY IS NEEDED.
  55  IF (NRCHOP.NE.2)GO TO 60
C
C6------IF INIRCH<0 THEN REUSE LAYER INDICATOR ARRAY.
      IF(INIRCH.GE.0)GO TO 58
      WRITE(IOUT,2)
    2 FORMAT(1H0,'REUSING IRCH FROM LAST STRESS PERIOD')
      GO TO 60
C
C7------IF INIRCH=>0 CALL U2DINT TO READ LAYER IND ARRAY(IRCH)
   58 CALL U2DINT(IRCH,ANAME(1,1),NROW,NCOL,0,IN,IOUT)
C
C8------RETURN
   60 RETURN
      END
      SUBROUTINE PRERCH(NRCHOP,IRCH,RECH,ZVEL,IBOUND,NCOL,
     1                     NROW,NLAY)
C
C-----MODIFIED FOR WATER TABLE VELOCITY CALC FROM
C       VERSION 1518 22DEC1982 RCH1FM
C     *****************************************************************
C     ADD RECHARGE TO RHS
C     *****************************************************************
C
C        SPECIFICATIONS:
C     -----------------------------------------------------------------
      DIMENSION IRCH(NCOL,NROW),RECH(NCOL,NROW),
     1          ZVEL(NCOL,NROW),IBOUND(NCOL,NROW,NLAY)
C     -----------------------------------------------------------------
C
C1------IF NRCHOP IS 1 RECHARGE IS IN TOP LAYER. LAYER INDEX IS 1.
      IF(NRCHOP.NE.1) GO TO 15
C
      DO 10 IR=1,NROW
      DO 10 IC=1,NCOL
C
C1A-----IF CELL IS EXTERNAL THERE IS NO RECHARGE INTO IT.
      IF(IBOUND(IC,IR,1).LE.0)GO TO 10
C
C1B-----SUBTRACT RECHARGE RATE FROM RIGHT-HAND-SIDE.
      ZVEL(IC,IR)=-RECH(IC,IR)
C     WRITE(*,1001)IC,IR,ZVEL(IC,IR)
C1001 FORMAT(2I3,G12.3)
   10 CONTINUE
      GO TO 100
C
C   THE FOLLOWING OPTION DOES NOT WORK WITH THE CURRENT EDITION
C   OF PREMOD3D AND RAND3D BUT IS LEFT IN FOR FUTURE CONSIDERATION
```

```
C2------IF OPTION IS 2 THEN RECHARGE IS INTO LAYER IN INDICATOR ARRAY
   15 IF(NRCHOP.NE.2)GO TO 25
      DO 20 IR=1,NROW
      DO 20 IC=1,NCOL
C
C2A-----LAYER INDEX IS IN INDICATOR ARRAY.
      IL=IRCH(IC,IR)
C
C2B-----IF THE CELL IS EXTERNAL THERE IS NO RECHARGE INTO IT.
      IF(IBOUND(IC,IR,IL).LE.0)GO TO 20
C
C2C-----SUBTRACT RECHARGE FROM RIGHT-HAND-SIDE.
      ZVEL(IC,IR)=-RECH(IC,IR)
   20 CONTINUE
      GO TO 100
C
C3------IF OPTION IS 3 RECHARGE IS INTO HIGHEST INTERNAL CELL.
   25 IF(NRCHOP.NE.3)GO TO 100
C        CANNOT PASS THROUGH CONSTANT HEAD NODE
      DO 30 IR=1,NROW
      DO 30 IC=1,NCOL
      DO 28 IL=1,NLAY
C
C3A-----IF CELL IS CONSTANT HEAD MOVE ON TO NEXT HORIZONTAL LOCATION.
      IF(IBOUND(IC,IR,IL).LT.0) GO TO 30
C
C3B-----IF CELL IS INACTIVE MOVE DOWN A LAYER.
      IF (IBOUND(IC,IR,IL).EQ.0)GO TO 28
C
C3C-----SUBTRACT RECHARGE FROM RIGHT-HAND-SIDE.
      ZVEL(IC,IR)=-RECH(IC,IR)
      GO TO 30
   28 CONTINUE
   30 CONTINUE
  100 CONTINUE
C
C4------RETURN
      RETURN
      END
```

```
$LARGE
$NOFLOATCALLS
      SUBROUTINE DRN1AL(ISUM,LENX,LCDRAI,NDRAIN,MXDRN,IN,IOUT,
     1                           IDRNCB)
C
C-----VERSION 0956 08DEC1983 DRN1AL
C     ******************************************************************
C     ALLOCATE ARRAY STORAGE FOR DRAIN PACKAGE
C     ******************************************************************
C
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
      INTEGER*4 LENX
C     ------------------------------------------------------------------
C
C1------IDENTIFY PACKAGE AND INITIALIZE NDRAIN.
      WRITE(IOUT,1)IN
    1 FORMAT(1H0,'DRN1 -- DRAIN PACKAGE, VERSION 1, 12/08/83',
     2' INPUT READ FROM UNIT',I3)
      NDRAIN=0
C
C2------READ & PRINT MXDRN & IDRNCB(UNIT & FLAG FOR CELL-BY-CELL FLOW)
      READ(IN,2) MXDRN,IDRNCB
    2 FORMAT(2I10)
      WRITE(IOUT,3) MXDRN
    3 FORMAT(1H ,'MAXIMUM OF',I5,' DRAINS')
      IF(IDRNCB.GT.0) WRITE(IOUT,9) IDRNCB
    9 FORMAT(1X,'CELL-BY-CELL FLOWS WILL BE RECORDED ON UNIT',I3)
      IF(IDRNCB.LT.0) WRITE(IOUT,8)
    8 FORMAT(1X,'CELL-BY-CELL FLOWS WILL BE PRINTED WHEN ICBCFL NOT 0')
C
C3------SET LCDRAI EQUAL TO ADDRESS OF FIRST UNUSED SPACE IN X.
      LCDRAI=ISUM
C
C4------CALCULATE AMOUNT OF SPACE USED BY THE DRAIN PACKAGE.
      ISP=5*MXDRN
      ISUM=ISUM+ISP
C
C5------PRINT AMOUNT OF SPACE USED BY DRAIN PACKAGE.
      WRITE(IOUT,4) ISP
    4 FORMAT(1X,I6,' ELEMENTS IN X ARRAY ARE USED FOR DRAINS')
      ISUM1=ISUM-1
      WRITE(IOUT,5) ISUM1,LENX
    5 FORMAT(1X,I6,' ELEMENTS OF X ARRAY USED OUT OF',I7)
      IF(ISUM1.GT.LENX) WRITE(IOUT,6)
    6 FORMAT(1X,'   ***X ARRAY MUST BE DIMENSIONED LARGER***')
C
C6------RETURN
      RETURN
      END
      SUBROUTINE DRN1RP(DRAI,NDRAIN,MXDRN,IN,IOUT)
C
C
```

```
C-----VERSION 1603 25APR1983 DRN1RP
C     ******************************************************************
C     READ DRAIN LOCATIONS, ELEVATIONS, AND CONDUCTANCES
C     ******************************************************************
C
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
      DIMENSION DRAI(5,MXDRN)
C     ------------------------------------------------------------------
C
C1------READ ITMP(NUMBER OF DRAIN CELLS OR FLAG TO REUSE DATA)
      READ(IN,8) ITMP
    8 FORMAT(I10)
C
C2------TEST ITMP
      IF(ITMP.GE.0) GO TO 50
C
C2A-----IF ITMP<0 THEN REUSE DATA FROM LAST STRESS PERIOD.
      WRITE(IOUT,7)
    7 FORMAT(1H0,'REUSING DRAINS FROM LAST STRESS PERIOD')
      RETURN
C
C3------IF ITMP=>0 THEN IT IS THE NUMBER OF DRAINS.
   50 NDRAIN=ITMP
      IF(NDRAIN.LE.MXDRN) GO TO 100
C
C4------IF NDRAIN>MXDRN THEN STOP
      WRITE(IOUT,99) NDRAIN,MXDRN
   99 FORMAT(1H0,'NDRAIN(',I4,') IS GREATER THAN MXDRN(',I4,')')
      STOP
C
C5------PRINT NUMBER OF DRAINS IN THIS STRESS PERIOD.
  100 WRITE(IOUT,1) NDRAIN
    1 FORMAT(1H0,//1X,I5,' DRAINS')
C
C6------IF THERE ARE NO DRAINS THEN RETURN.
      IF(NDRAIN.EQ.0) GO TO 260
C
C7------READ AND PRINT DATA FOR EACH DRAIN.
      WRITE(IOUT,3)
    3 FORMAT(1H0,15X,'LAYER',5X,'ROW',5X
     1,'COL   ELEVATION   CONDUCTANCE   DRAIN NO.'/1X,15X,60('-'))
      DO 250 II=1,NDRAIN
      READ (IN,4) K,I,J,DRAI(4,II),DRAI(5,II)
    4 FORMAT(3I10,2F10.0)
      WRITE (IOUT,5) K,I,J,DRAI(4,II),DRAI(5,II),II
    5 FORMAT(1X,15X,I4,I9,I8,G13.4,G14.4,I8)
      DRAI(1,II)=K
      DRAI(2,II)=I
      DRAI(3,II)=J
  250 CONTINUE
C
C8------RETURN
  260 RETURN
```

```
      END
      SUBROUTINE DRNPRE(NDRAIN,MXDRN,DRAI,HNEW,IBOUND,
     1              DELR,DELC,NCOL,NROW,NLAY,IVOUT,CONVRT,
     2              LLX,LUX,LLY,LUY)
C
C      ROUTINE TO ADD DRAINS TO SINK LIST MODIFIED FROM USGS 3D
C-----VERSION 1638 25APR1983 DRN1FM
C
C     ******************************************************************
C     ADD DRAINS TO SINKS
C     ******************************************************************
C
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
      DOUBLE PRECISION HNEW
C
      DIMENSION DRAI(5,MXDRN),HNEW(NCOL,NROW,NLAY),
     1         DELR(NCOL),DELC(NROW),IBOUND(NCOL,NROW,NLAY)
C     ------------------------------------------------------------------
C
C1------IF NDRAIN<=0 THERE ARE NO DRAINS. RETURN
      IF(NDRAIN.LE.0) RETURN
C
C2------PROCESS EACH CELL IN THE DRAIN LIST
      DO 100 L=1,NDRAIN
C
C3------GET COLUMN, ROW AND LAYER OF CELL CONTAINING DRAIN.
      IL=DRAI(1,L)
      IR=DRAI(2,L)
      IC=DRAI(3,L)
C
C4-------IF THE CELL IS EXTERNAL SKIP IT.
      IF(IBOUND(IC,IR,IL).LE.0) GO TO 100
      IF(IC .LT. LLX .OR. IC .GT. LUX)GO TO 100
      IF(IR .LT. LLY .OR. IR .GT. LUY)GO TO 100
C
C5-------IF THE CELL IS INTERNAL GET THE DRAIN DATA.
      EL=DRAI(4,L)
      HHNEW=HNEW(IC,IR,IL)
C
C6------IF HEAD IS LOWER THAN DRAIN THEN SKIP THIS CELL.
      IF(HHNEW.LE.EL) GO TO 100
C
C7------HEAD IS HIGHER THAN DRAIN. ADD TERMS TO RHS AND HCOF.
      C=DRAI(5,L)
      RATE=-C*(HHNEW-EL)
      CALL XYZPOS(IC,IR,IL,X,Y,Z,DELR,DELC,NROW,NCOL,NLAY)
      IF(RATE .LT. 0)WRITE(IVOUT,1005)X,Y,Z,-RATE*CONVRT
 1005 FORMAT(3F12.1,G12.4)
  100 CONTINUE
C
C8------RETURN
      RETURN
      END
```

Appendix C - User's Manual for RAND3D

I. Introduction

The RAND3D program is a three dimensional version of the random walk algorithm developed by Thomas Prickett at the Illinois Water Survey as an efficient algorithm for solving ground water solute transport problems (Prickett, Naymik, and Lonnquist, 1981). The model was originally developed for two dimensional solute transport. Thomas A. Prickett and Associates developed a three dimensional version of the model. Further modifications and improvements were made to the model as part of this project.

The random-walk technique is based on the concept that dispersion in porous media is a random process. A particle, representing the mass of a specific chemical constituent contained in a defined volume of water, moves through an aquifer with two types of motion. One motion is with the mean flow (along streamlines determined by finite differences), and the other is random motion, governed by scaled probability curves related to flow length and the longitudinal and transverse dispersion coefficients. Enough particles are included in simulations so that their locations and density, as they move through a flow model, are adequate to describe the distribution of the dissolved constituent. Each particle represents a fixed mass of solute. As more particles, with correspondingly smaller masses, are used in a given simulation, accuracy improves.

One of the major features of the RAND3D model is its interactive operation on an IBM PC or compatible microcomputer. After velocity files are prepared using PREMOD3D or some other suitable procedure, the user may use this program to simulate solute transport and watch the results on the monitor. The program operates from a menu. The user is prompted for all data inputs. A major feature of the model is the ability to display geographic features on the computer screen and superimpose the plume simulation. The user may zoom in on any area of the model to see a more detailed simulation. The geographic features are input by the user in any convenient right-handed (x-y) coordinate system in feet (such as a State Plane coordinate system). These features may then be displayed on the screen as background reference for the plume simulation.

The RAND3D model includes the following features:

o calculation of horizontal advective transport based on a four point interpolation of the input velocity vectors;

- calculation of vertical advective transport based on linear interpolation between the input vertical velocity vectors at the top and bottom of each layer;
- calculation of dispersion using constant dispersivities, longitudinal, transverse, and vertical;
- calculation of first-order decay;
- calculation of linear, reversible adsorption (retardation);
- the ability to originate solute (particles) in the model as sequences of prisms, cylinders, or lines;
- calculation of solute concentrations exiting the model at sinks (wells or gaining streams);
- mapping of solute concentration in user selected areas of the model, either plan view or cross-section concentration maps may be prepared;
- output of gridded solute concentrations by layer for plotting;
- interactive operation;
- on-screen display of plume (particle) movement in user selected area;
- on-screen display of user input geographic features at user selected scale as background for the plume display;
- saving and viewing of screen slides;
- saving and restart of model parameters at any time;
- transient flow simulations may be simulated by inputting a series of velocity files.

The RAND3D model was designed for an IBM PC or compatible microcomputer with 640K, a numeric coprocessor, a hard drive, a line printer, and a color monitor with a color graphics adapter. The program is written in Microsoft Quick Basic Version 3.0. Current limits in the program are:

- maximum input grid of 45 columns, 45 rows, three aquifer layers, and two confining layers;
- maximum number of particles is 10000;

- maximum number of sinks (wells or gaining streams) is 99;

- maximum number of special feature files is 20;

- at least two layers must be simulated.

This user's manual is divided in two sections, theory and user instructions. Under theory, first the basic theory of the random walk algorithm is presented. This material has been copied with a few minor changes from Prickett, Naymik, and Lonnquist, A Random-Walk Solute Transport Model for Selected Groundwater Quality Evaluations, Illinois Water Survey Bulletin 65, 1981. Then a more detailed presentation of the same material is made showing the actual equations used in the model along with some comments on what is happening and why. The user instructions shows how all the menu commands work and what the proper inputs are.

II. Theory

A. Overview

The RAND3D model solves the solute transport equation in three dimensions. One form of the governing equation for solute transport in one dimension is

$$@(D/R_d\ @C/@x)/@x - v/R_d\ @C/@x + C_sQ = @C/@t \qquad (1)$$

dispersion - advection + sources/sinks = concentration

where

V = interstitial (seepage) velocity
D = coefficient of hydrodynamic dispersion
D = dxV , where dx = longitudinal dispersivity
x = space dimension
R_d = retardation factor
C_sQ = source or sink function having a concentration C_s
C = concentration
t = time

Problems including solute transport in groundwater involve solving the above in one, two or three dimensions.

The two other commonly used algorithms in ground water solute transport are the method of characteristics, and the direct finite element solution of the differential equation of solute transport. The method of characteristics, or particle-in-a-cell algorithm, treats the solute transport equation in two parts. First, the convective term containing the velocity is solved by tracking the movement of uniformly placed particles within each cell. Then the dispersive term is solved by using a finite difference grid associated with the concentration distribution. A large number of computer-generated particles move about by the velocity vectors and which carry the concentration information between the convection and dispersion terms during the solution of the equation. The description of the method of characteristics is straightforward, but the computer code is highly involved, and it requires a large computer to effect a solution.

The finite element solution of the solute transport equation is a direct numerical solution. Numerical dispersion is typically a problem with this type of solution. Very small nodal spacings are necessary on the leading edge of a contaminant plume for stable solutions. Weighting the upgradient term in the finite element equation (up-wind weighting) improves stability at the expense of accuracy. The random-walk algorithm avoids all problems with numerical stability by approaching the problem differently.

The random-walk technique is based on the concept that dispersion in porous media is a random process. On a microscopic basis, dispersion may occur as shown in Figure 1. As indicated in Figure 1C, dispersion can take place in two directions even though the mean flow is in one direction to the right. A particle, representing the mass of a specific chemical constituent contained in a defined volume of water, moves through an aquifer with two types of motion. One motion is with the mean flow (along streamlines determined by finite differences), and the other is random motion, governed by scaled probability curves related to flow length and the longitudinal and transverse dispersion coefficients. Finally, in the computer code, enough particles are included so that their locations and density, as they move through a flow model, are adequate to describe the distribution of the dissolved constituent.

The advantages of this random-walk technique over the MOC and finite element method, are:

1. There is no dispersion equation to solve. The dispersion part of the solute transport equation is solved in the computer code with less than twenty statements.

2. There is only one finite difference grid involved in solving the convective portion of the solute transport equation. The particle movement takes place in continuous space.

3. Concentrations are calculated only when they are of interest.

4. Particles are needed only where water quality is of interest. This is an distinct advantage in the typical ground water contamination problem where a pollutant plume is moving through an aquifer. It is only necessary to perform calculations where the plume exists; other algorithms solve for concentrations in all model grids or nodes.

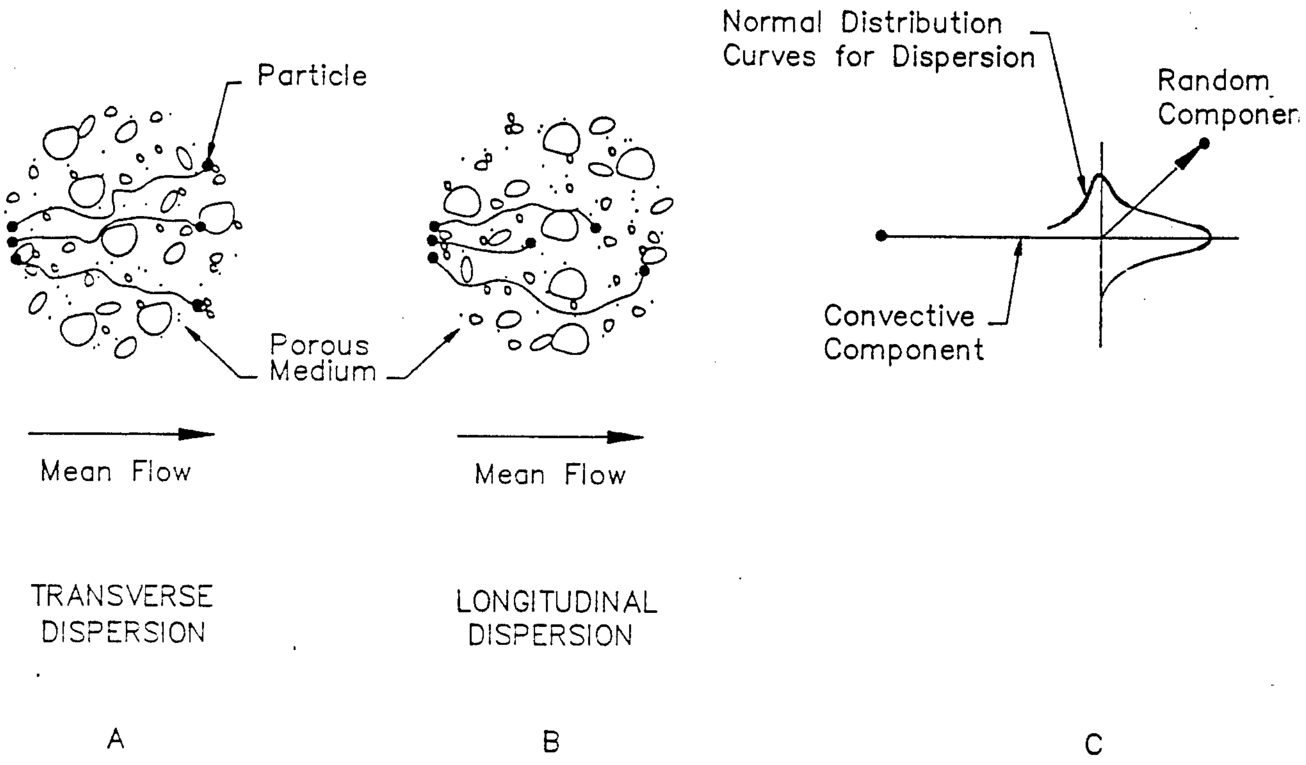
5. Solutions are additive. If not enough particles are included for adequate definition in one run, a second run can be done and the results accumulated.

6. In the traditional sense of the words, finite differencing phenomena associated with "overshoot" and "numerical dispersion" are eliminated.

Figure 1

Basic Concept of Random Walk

(from Prickett, Naymik, and Lonnquist, 1981)



Although there are numerous advantages to this technique, there are also some disadvantages:

1. Concentrations greater than initial conditions are possible, especially when coarse discretizing is used.

2. A printout of concentrations may be misleading when the number of particles is small.

3. The method may take a large number of particles to produce an acceptable solution for some problems.

4. Engineering judgement is an absolute requirement in arriving at an acceptable solution. This is because of the "lumpy" character of the output. Therefore, experience with this technique is needed before one can apply the code successfully to a field situation.

B. Basic Theory

1. Introduction

The basis for the transport calculations of dissolved constituents in this computer code is that the chemical concentration of constituents in the water in an aquifer can be represented by the distribution of a finite number of discrete particles. Each of these particles is moved by groundwater flow and is assigned a mass which represents a fraction of the total mass of chemical constituent involved. In the limit, as the number of particles gets extremely large and approaches the molecular level, an exact solution to the actual situation is obtained. Experience indicates that relatively few particles are needed to arrive at a solution that will suffice for many engineering applications.

There are two prime mechanisms which can change contaminant concentration in groundwater: dispersion, and dilution and mixing. The effects of mechanical dispersion as the fluid spreads through the pore space of the porous medium are described by the first and second terms on the left side of the solute transport equation (Equation 1). The effects of dilution and mixing are expressed in the second and third terms on the left side of solute transport equation.

2. Dispersion

To illustrate the details of the random-walk technique as it relates to dispersion, consider the progress of a unit slug of tracer-marked fluid, placed initially at x=0, in an infinite column of porous medium with steady flow in the x direction. With C_SQ equal to zero, equation 1 describes the concentration of the slug as it moves downstream. Bear (1972) describes the solution as

$$C(x,t) = [1/(4\pi d_L vt)^{0.5}]\exp[-(x-vt)^2/4d_L vt] \qquad (2)$$

where

C = concentrations
d_L = longitudinal dispersivity
v = interstitial velocity
t = time
x = distance along the x axis

The shapes of the curves $C(x',t)$ are shown in Figure 2 where $x' = x-vt$.

A random variable x is said to be normally distributed if its density function, $n(x)$, is given by

$$n(x) = [1/\sqrt{2\pi}\sigma]\exp[-(x-u)^2/2\sigma^2)] \qquad (3)$$

where
σ = standard deviation of the distribution
u = mean of the distribution

Now, let us equate the following terms of equations 2 and 3 as

$$\sigma = \sqrt{2d_L vt} \qquad (4)$$

$$u = vt \qquad (5)$$

$$n(x) = C(x,t) \qquad (6)$$

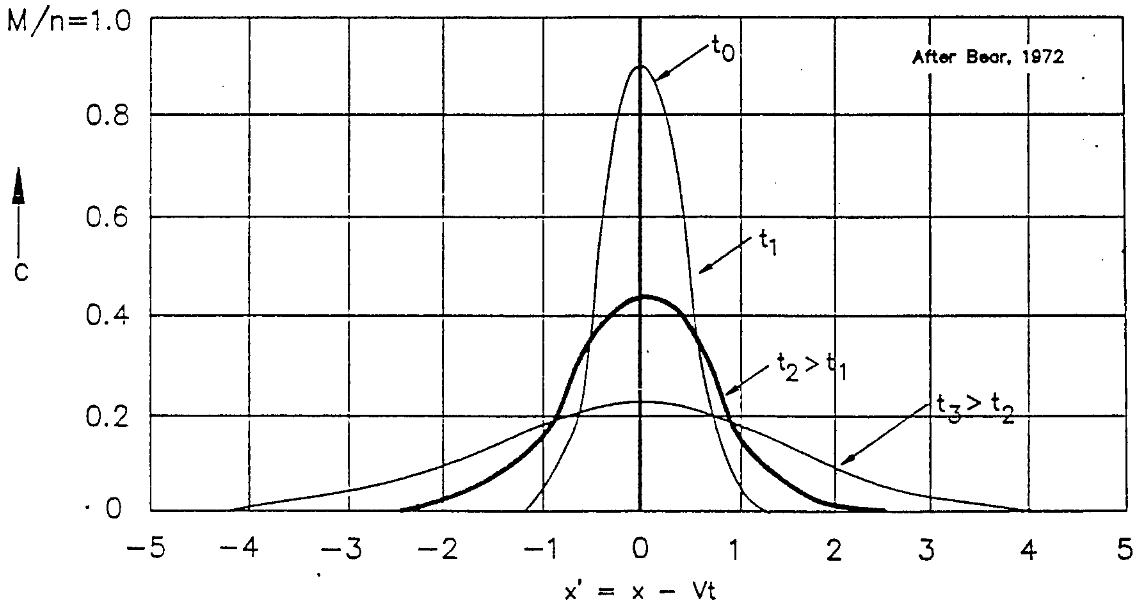
With the identities of equations 4 through 6 taken into account, equations 2 and 3 are equivalent.

Figure 3A represents the way particles are moved in the computer code when the flow is in the x direction and one considers only longitudinal dispersion. During a time increment, DELP, a particle with coordinates xx,yy is first moved from an old to a new position in the aquifer by convection according to its velocity at the old position V_x. Then, a random movement in the +x or -x direction is added to represent the effects of dispersion. This random movement is given the magnitude

Figure 2

Progress of Slug Around Meanflow

(from Prickett, Naymik, and Lonnquist, 1981)



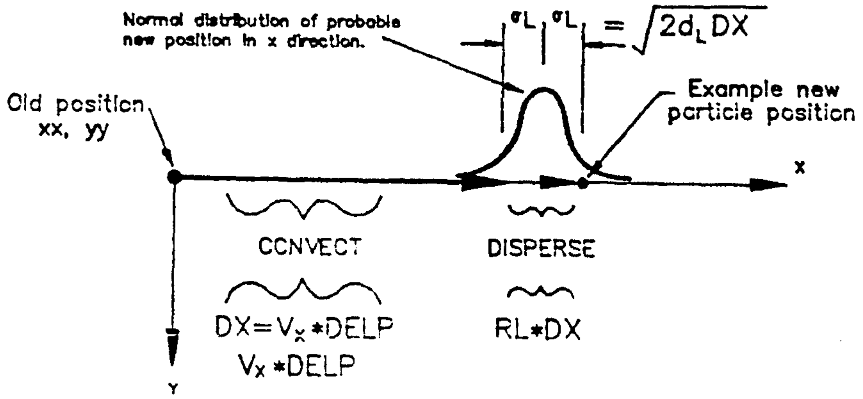
After Bear, 1972

Figure 3

Computer Code Scheme for Convection and Longitudinal (A) and Transverse (B) Dispersion along X axis

(from Prickett, Naymik, and Lonnquist, 1981)

(A) LONGITUDINAL DISPERSION



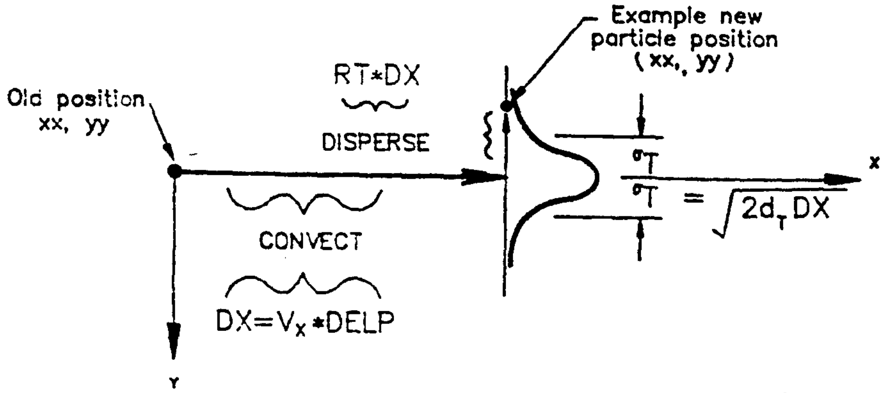
Normal distribution of probable new position in x direction.
$\sigma_L \, \sigma_L = \sqrt{2d_L DX}$
Old position xx, yy
Example new particle position
x
CONVECT
DISPERSE
DX=V_x*DELP
V_x*DELP
RL*DX
Y

Where:

$$RL*DX=\sqrt{2d_L DX}*ANORM\ (0)$$

New position	=	Old position	+	Convection	+	Dispersion
xx	=	xx	+	DX	+	RL*DX
yy	=	yy				

(B) TRANSVERSE DISPERSION



Example new particle position (xx, yy)
Old position xx, yy
RT*DX
DISPERSE
σ_T
$\sigma_T = \sqrt{2d_T DX}$
x
CONVECT
DX=V_x*DELP
Y

Where :

$$RT*DX = \sqrt{2d_T DX}\ ANORM\ (0)$$

New position	=	Old position	+	Convection	+	Dispersion
xx	=	xx	+	DX	+	0
yy	=	yy	+	0	−	RT*DX

$$(\sqrt{2d_L DX})\,ANORM(0) \qquad (7)$$

where

$\sqrt{2d_L DX} = \sigma = \sqrt{2d_L vt}$
v = velocity in x direction
DX = vt = advective move distance
t = time step
ANORM(0) = a number between -6 and +6, drawn from a normal distribution of numbers having a standard deviation of 1 and a mean of zero.

The new position of the particle in Figure 3A is the old position plus a convective term (vt) plus the effect of the dispersion term

$$(\sqrt{2d_L DX})\,ANORM(0)$$

If the above process is repeated for numerous particles, all having the same initial position and convective term, a map of the new positions of the particles can be created having the discrete density distribution

$$C(x,t) - n(x) - N/dx = N_o/(\sqrt{4\pi d_L DX})\ \exp[-(x-vt)^2/4d_L vt] \qquad (8)$$

where

N_o = Total number of particles in the experiment

Equations 2, 3, and 8 are equivalent, with the exception that equations 2 and 3 are continuous distributions and equation 8 is discrete. As illustrated in Figure 4A, the distribution of particles around the mean position, vt, is made to be normally distributed via the function ANORM(0). The function ANORM(O) is generated in the computer code as a simple function involving a summation of random numbers. Probable locations of particles, however, are considered only out to 6 standard deviations either side of the mean. On a practical basis, the probability is low of a particle moving beyond that distance.

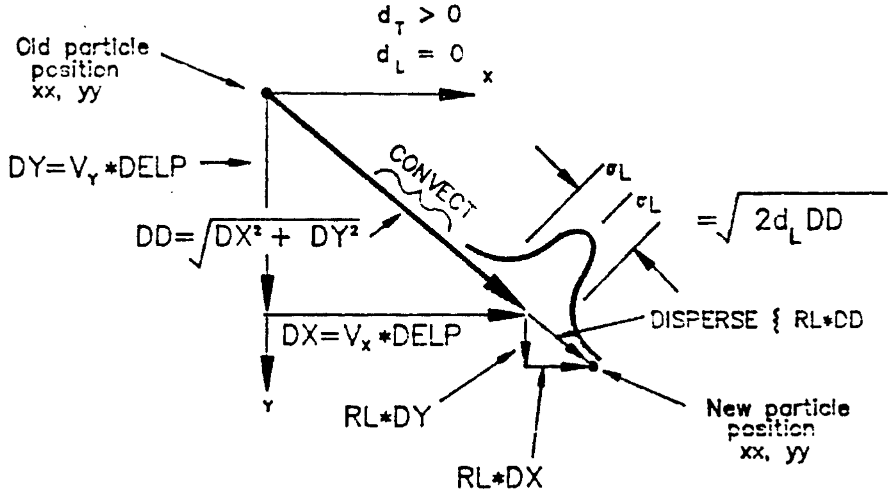
One further emphasis is appropriate concerning the so-called "density function" of equations 2, 3, and 8. The equivalent density functions C(x,t) and N/dx provided the means for relating the concentration of a contaminant in a field problem to the concentration of particles found in portions of a finite difference model. Various density functions will be defined later, by example, as they are needed for application purposes.

Figure 4

General scheme for convection and longitudinal (A) and transverse (B) dispersion

(from Prickett, Naymik, and Lonnquist, 1981)

(A) LONGITUDINAL DISPERSION

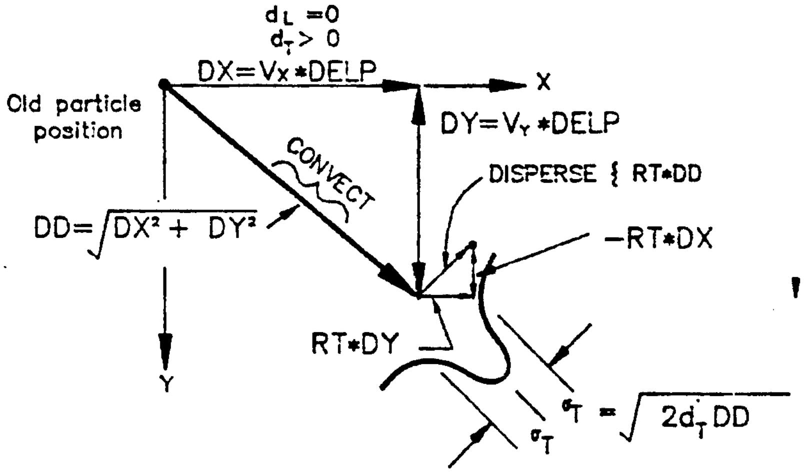


Where :

$RL*DD = \sqrt{2d_L DD}$ ANORM (0)

New position	=	Old position	+	Convection	+	Dispersion
xx	=	xx	+	DX	+	RL*DX
yy	=	yy	+	DY	+	RL*DY

TRANSVERSE DISPERSION



Where :

$RT*DD = \sqrt{2d_T DD}$ ANORM (0)

New position	=	Old position	+	Convection	+	Dispersion
xx	=	xx	+	DX	+	RT*DY
yy	=	yy	+	DY	−	RT*DX

Figure 3B illustrates the extension of the random-walk method to account for dispersion in a direction transverse to the mean flow. Figures 4A and 4B illustrate the algebra involved when the flow isn't aligned with the x-y coordinate system. Finally, Figure 5 shows both longitudinal and transverse dispersion taking place simultaneously, and the appropriate vector algebra.

3. Dilution, Mixing, Retardation, and Radioactive Decay

Consider the one-dimensional flow problem in Figure 6A in which the flow and concentrations of the sources are given. With dispersion set to zero and retardation set to one, the distribution of concentrations in the system is simply a result of pure mixing as illustrated in Figure 6B.

Third, let us assume that in the computer model one particle represents 10 mg/l. Figure 6C shows the time density of particles that would be used for input data in the computer model. Figure 6D shows the space density of particles in the computer model that would be simulated. Once the space density of particles is known, a multiplication by the particle mass yields the concentration of the flowing water.

In equation 1 the retardation factor (R_d) is used to represent the change in the solute concentration in the fluid caused by chemical reactions with the medium. These reactions include adsorption, organic fixation, etc. Chemical reactions between the dissolved constituent and the medium tend to retard the movement of the constituent relative to the groundwater velocity. The retardation of a concentration front in groundwater relative to the bulk mass of water is described by the relation

$$v/v_c = 1 + K_d p/n = R_d \qquad (9)$$

where

v = interstitial velocity of the groundwater
v_c = velocity of the C/Co = 0.5 in the concentration front
p = bulk mass density
n = effective porosity
K_d = distribution coefficient
R_d = retardation factor

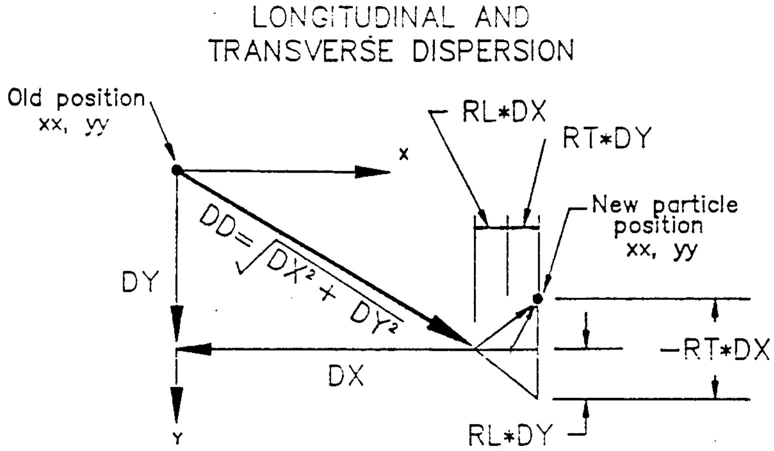
Radioactive decay is also included in the model. Radioactive decay is a first-order process where:

$$@c/@t = Kc$$

Figure 5

General scheme for convection and dispersion

(from Prickett, Naymik, and Lonnquist, 1981)



$$RL*DD = \sqrt{2d_L DD}\ \text{ANORM}\ (0)$$

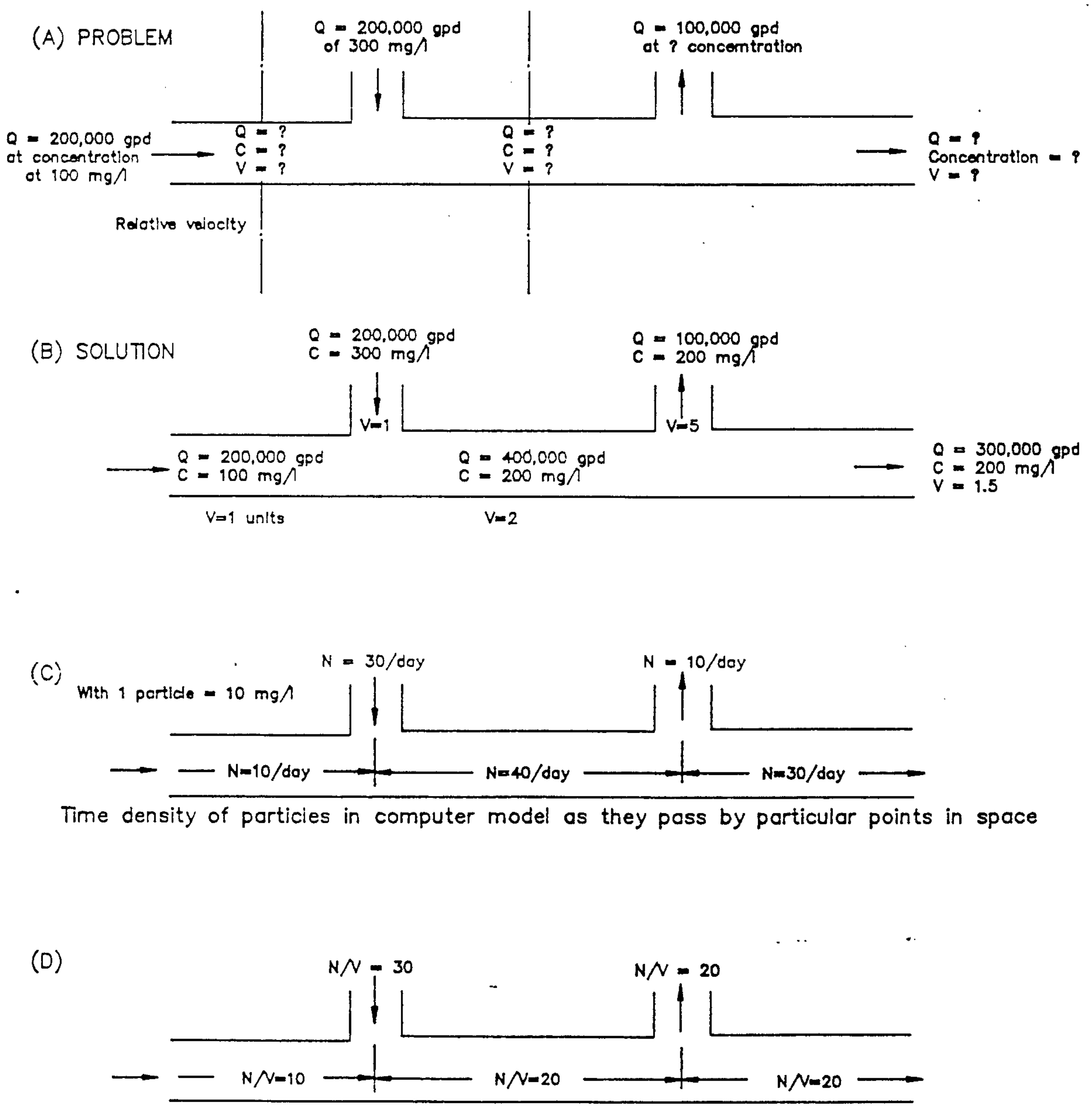
$$RT*DD = \sqrt{2d_T DD}\ \text{ANORM}\ (0)$$

						Longitudinal		Transverse
New position	=	Old position	+	Convection	+	Dispersion	+	Dispersion
xx	=	xx	+	DX	+	RL*DX	+	RT*DY
yy	=	yy	+	DY	+	RL*DY	−	RT*DX

Figure 6

Mixing and dilution effects in water quality problems.

(from Prickett, Naymik, and Lonnquist, 1981)



Time density of particles in computer model as they pass by particular points in space

Space density of particles in computer model on any particular day where

c = concentration
t = time
K = decay coefficient

The relationship between the decay coefficient and the commonly used measure of first order decay, half-life, is:

$$\text{half-life} = \ln(2)/K$$

where

ln = natural log function

There are two methods of implementing first order decay in the model. The first is to calculate the decay of each particle using the integrated form of the decay equation.

$$M = M_o\exp(-Kt)$$

where

M = particle mass at end of time step
M_o = particle mass at beginning of time step
t = time step

The second, and the one implemented in this model, is to compute the fraction of each particle decaying during the time step with an equation of the form:

$$\text{fraction} = [1-.5Kt/\ln(2)]$$

The fractions are summed and when the total decay fraction is greater than one, a particle is removed from the model. Thus, the total number of particles is used to track the effects of decay.

C. Solute Transport (MOVE routine)

1. Introduction

The MOVE subroutine of the RAND3D model is the basic routine of the simulation. Particles are moved, dispersed, decayed, and retarded. The user specifies the time step and the maximum move in the horizontal and vertical directions before velocities are reinterpolated. The user selects the graphic display area. Each particle is then moved for the time step. The particle's path is shown on the display screen. After each move, the distance between the particle and each sink is calculated. If the particle would reach the sink during the time step, the particle is removed from the simulation. The move includes the effects of dispersion and adsorption (retardation). First order decay is calculated by removing particles from the model.

Particles are tracked through all routines in the RAND3D model by a real world based, foot-denominated, right handed coordinate system (x,y,z). This feature makes it easy for the user to enter model input in terms of State Plane coordinates, or some other relevant coordinate system. The input velocity vectors must be aligned with the coordinate system chosen. The position of the velocity vectors is fixed by giving the coordinates of the lower left corner of the velocity vector grid. Aquifers in the RAND3D model are numbered from bottom (layer 1) to top (layer NL). Confining layers are numbered from bottom to top also with numbers above the top aquifer number (NL+1 to NL+NL-1). Row numbering increases with increasing y. Column number numbering increases with increasing x. Figure 7 shows the numbering scheme used in the model.

2. Advection

Advection is the movement of solute (particles) along the streamlines or path lines of ground water flow. The RAND3D model calculates advective movement by interpolating a velocity from the velocity vectors that are read into the model. The particle is moved either the time step, or the user specified maximum distance. Velocity is then reinterpolated at the new particle location. Figure 8 shows a schematic of the velocity approximation.

Horizontal velocity is calculated from four velocity vectors. Particle position is normally accounted for in real world cartesian coordinates (x,y). To do the velocity interpolation, the cartesian coordinates must be mapped to the grid system on which the input velocity vectors are based. Then linear interpolation between the two nearest velocity vectors is used to calculate the x and y velocity components. This interpolation scheme has been proven to be adequate for most situations. More complex interpolation schemes may yield better advective movement in some circumstances. A more sophisticated horizontal velocity interpolation is given in A Random-Walk Solute Transport Model for Selected Groundwater Quality Evaluations (Prickett, Naymik, and Lonnquist, 1981).

Figure 7

RAND3D Coordinate and Grid System

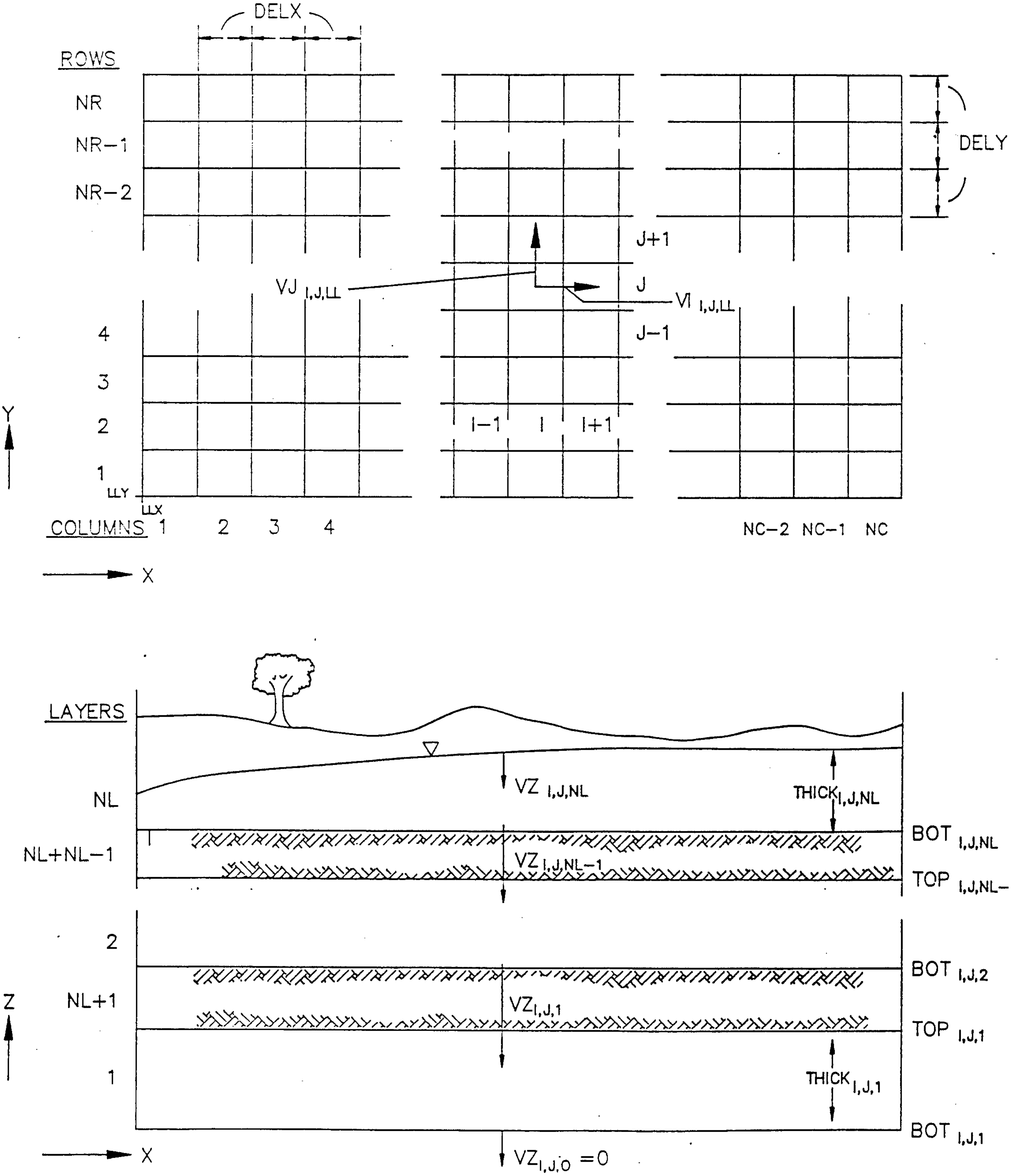
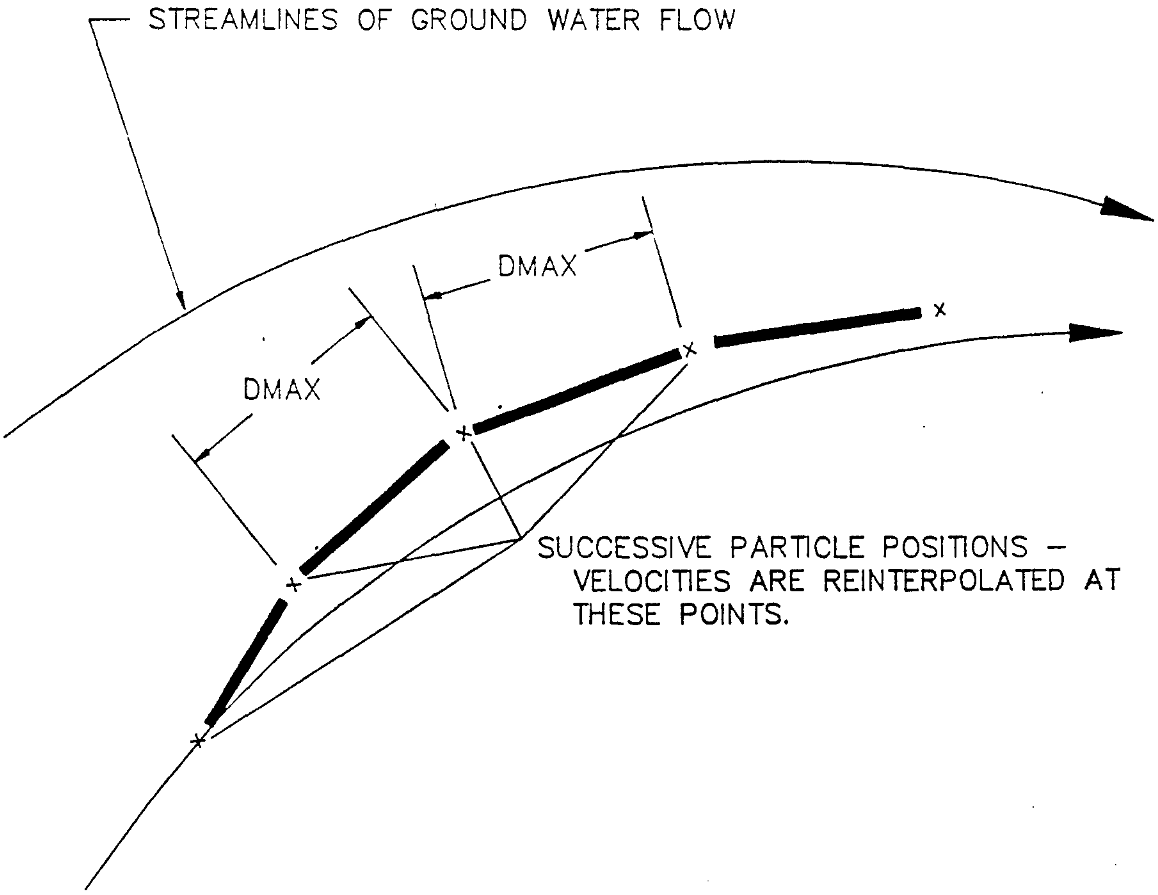


Figure 8

Advective Movement of Particle


STREAMLINES OF GROUND WATER FLOW
DMAX
DMAX
SUCCESSIVE PARTICLE POSITIONS – VELOCITIES ARE REINTERPOLATED AT THESE POINTS.

$$XP=(X_K-LLX)/DELX+.5$$
$$YP=(Y_K-LLY)/DELY+.5$$

$$I=INT(XP)$$
$$J=INT(YP)$$

$$AX=XP-INT(XP)$$
$$AY=YP-INT(YP)$$

$$VX=((1-AY)*VI_{I,J,LL}+AY*VI_{I,J+1,LL})/E$$
$$VY=((1-AX)*VJ_{I,J,LL}+AX*VJ_{I+1,J,LL})/E$$

where

X_K,Y_K = horizontal position of particle K in cartesian coordinates (ft)
LLX = x coordinate of lower left corner of grid (ft)
LLY = y coordinate of lower left corner of grid (ft)
DELX = grid spacing in x direction (ft)
DELY = grid spacing in y direction (ft)
AX = fraction of distance from center of grid I,J to I+1,J
AY = fraction of distance from center of grid I,J to I,J+1
VX = velocity in x direction at particle (ft/day)
VY = velocity in y direction at particle (ft/day)
$VI_{I,J,LL}$ = velocity vector from center of grid I,J to I+1,J (ft/day)
$VJ_{I,J,LL}$ = velocity vector from center of grid I,J to I,J+1 (ft/day)
LL = layer of particle
E = effective porosity

The above equations are used in the aquifer layers of the model. Figure 9 shows the horizontal velocity interpolation scheme. If the particle is in a confining layer, then the horizontal velocity is zero.

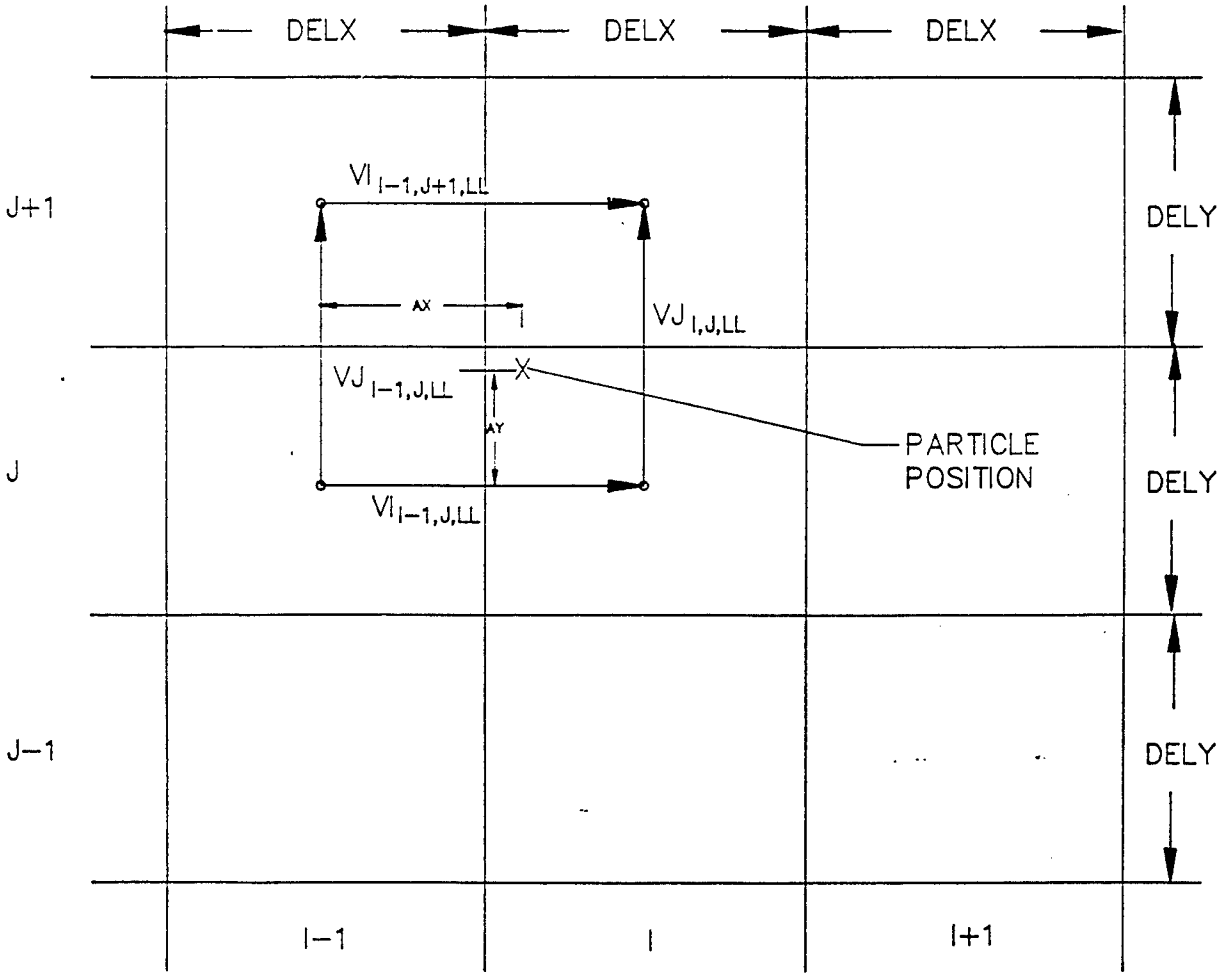
The vertical advective movement of the particle is calculated by linear interpolation between the overlying and underlying vertical velocity vectors. The following equations apply to the vertical movement of particles in aquifer layers.

$$I=INT(XP+.5)$$
$$J=INT(YP+.5)$$

$$AZ=(Z_K-BOT_{I,J,LL})/(TOP_{I,J,LL}-BOT_{I,J,LL})$$
$$VZ=(AZ*VK_{I,J,LL}+(1-AZ)*VK_{I,J,LL-1})/E$$

Figure 9

Horizontal Velocity Interpolation



where

Z_K = z coordinate of particle (ft)
$BOT_{I,J,LL}$ = elevation of bottom of aquifer LL in grid I,J (ft)
$TOP_{I,J,LL}$ = elevation of top of aquifer LL in grid I,J (ft)
AZ = fractional distance of particle above bottom of aquifer
VZ = vertical velocity of particle (ft/day)
$VK_{I,J,LL}$ = vertical velocity vector from layer LL to LL+1 in grid I,J (ft/day)
E = effective porosity

When the particle is in the confining bed, the vertical velocity is assigned as:

$$VZ = VK_{I,J,LL}/E$$

when

$$Z_K > TOP_{I,J,LL} \text{ and } Z_K < BOT_{I,J,LL+1}$$

Figure 10 shows the vertical velocity interpolation scheme.

The particle is then moved according to each of the velocity vectors (ignoring dispersion and retardation which will be explained later).

$$DX = VX * F$$
$$DY = VY * F$$
$$DZ = VZ * F$$

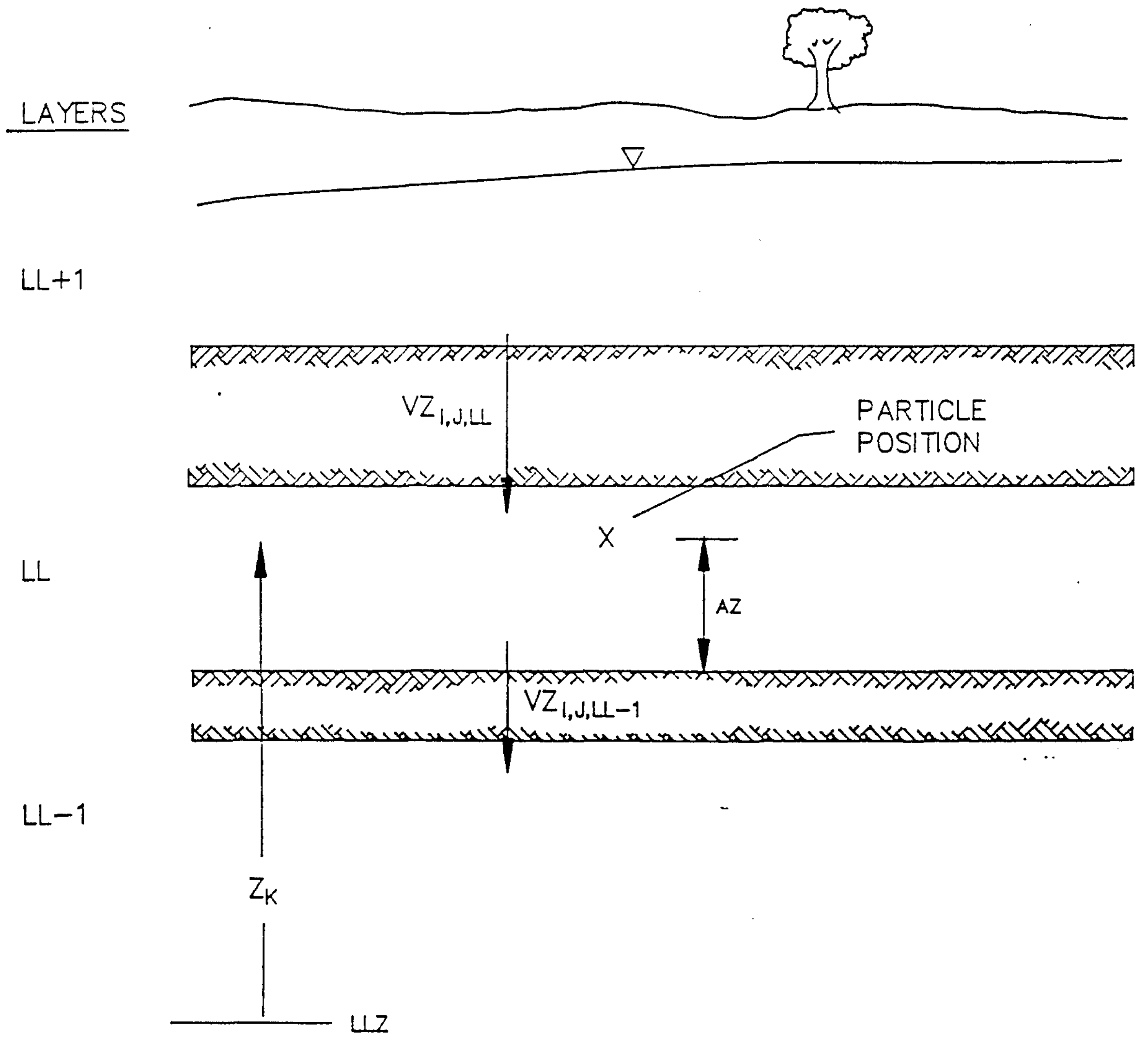
where

F = min(|DMAX/VX|,|DMAX/VY|,|ZMAX,VZ|,DELTA)
DX = distance moved in x direction (ft)
DY = distance moved in y direction (ft)
DZ = distance moved in z direction (ft)
F = move time (days)
DMAX = maximum horizontal move distance before reinterpolating velocity (ft)
ZMAX = maximum vertical move distance before reinterpolating velocity (ft)
DELTA = user specified time step for move (days)

The sequence of interpolation, move, interpolation, move is repeated for each particle until the total move time (sum of the F's) is equal to DELTA, the user entered time step in days. At each move, the position of the particle is updated.

Figure 10

Vertical Velocity Interpolation



The maximum move distances, DMAX and ZMAX, should be selected carefully by the user. If they are too large, particles will move incorrectly. If they are too small, the computations take a very long time. A value of 0.2 times the smallest grid dimension has been found to be suitable for DMAX (Prickett, Naymik, and Lonnquist, 1981). The best value for ZMAX is typically quite a bit smaller than that, good results have been achieved with ZMAX=DMAX/10.

3. Dispersion

Dispersion is simulated in the RAND3D model by letting the particle move stochastically around the advective flow path. Dispersion is assumed to be a linear function of velocity. The constant of proportionality is the dispersivity. There are other forms of the dispersion equation that could be implemented in the model. Constant dispersion is suitable for problems where molecular diffusion is significant. Recent research indicates that dispersivity is not a constant but rather a function of the distance traveled, with an asymptotic maximum (Molz, Guven and Melville, 1983). Either of these functions could be added to the model code relatively easily. This version of the code relys, however, on the most common dispersion equation. Vertical dispersion is calculated separately from the horizontal dispersion. Horizontal dispersion is given by a longitudinal dispersivity, which is in the direction of flow, and a transverse dispersivity, which is perpendicular to the direction of flow. In an isotropic aquifer, these terms would adequately characterize the dispersion process. In a typical aquifer, however, there is severe anisotropy between the horizontal permeability and the vertical permeability. Experience indicates that a separate and smaller dispersivity should be used for vertical movement. Thus a third dispersivity term, vertical dispersivity is an input to the model.

First the direction and distance of the advective move are determined. Notice that the distances are determined both for horizontal vectors only, and with all three advective move vectors.

$$PHI=\tan^{-1}(DY/DX)$$

$$DD=\sqrt{DX*DX+DY*DY}$$

$$DDX=\sqrt{DX*DX+DY*DY+DZ*DZ}$$

where

PHI = angle of horizontal move vector
DD = distance of horizontal move (ft)
DDX = total move distance (ft)

Then the dispersion moves are calculated. Each dispersion move is determined by multiplying the dispersion by a normally distributed random number with a mean of zero and a standard deviation of one. The program computes this normally distributed random number by summing together 12 uniformly distributed random numbers between zero and one which are generated by the BASIC random number function and adding -6.

$$RL=\sqrt{(2*DL*DD)}/DD*RN$$

$$RT=\sqrt{(2*DT*DD)}/DD*RN$$

$$RV=\sqrt{(2*DV*DDX)}*RN$$

where

RL = the one standard deviation longitudinal dispersion distance, i.e. the distance that the particle will be moved by dispersion assuming the normally distributed random number is one - one standard deviation from the mean.

RT = the one standard deviation transverse dispersion distance (ft)

RV = the one standard deviation vertical dispersion distance (ft)

RN = a normally distributed random number with a mean of zero and standard deviation of one.

DL = longitudinal dispersivity (ft)

DT = transverse dispersivity (ft)

DV = vertical dispersivity (ft)

The particles are then moved to their new locations. The following statements include both the advective and dispersive moves. The geometry of the move distances is explained in the basic theory section and Figures 3, 4, and 5.

$$X_K=X_K+(DX+RL*DX+RT*DY)$$

$$Y_K=Y_K+(DY+RL*DY-RT*DX)$$

$$Z_K=Z_K+DZ+RV$$

where

X_K = x coordinate of particle (ft)

Y_K = y coordinate of particle (ft)

Z_K = z coordinate of particle (ft)

4. Adsorption

The RAND3D model simulates adsorption by using a retardation coefficient. A separate retardation coefficient may be entered for each aquifer layer and each confining layer. The use of a constant retardation coefficient assumes a linear and reversible adsorption isotherm.

The retardation coefficient is used in the MOVE subroutine to modify the velocity vectors.

```
$VX=VX/RETARD_{LL}$
$VY=VY/RETARD_{LL}$
$VZ=VZ/RETARD_{LL}$
```

where

VX = velocity in x direction (ft/day)
VY = velocity in y direction (ft/day)
VZ = velocity in z direction (ft/day)
$RETARD_{LL}$ = retardation coefficient in layer LL

5. Decay

The RAND3D model simulates first order decay by removing particles from the simulation at the beginning of each time step. The program loops through each particle calculating the decay fraction as a function of the half-life and the time step. When the decay fraction is greater and or equal to one, a particle is removed from the simulation and the last particle is moved to the position of the removed particle and the number of particles decremented by one.

```
$ZZZ=ZZZ+(1-.5^{(DELTA/LAMDBA/365)})$
IF ZZZ=> 1 THEN

remove a particle
    $X_K=X_{NP}$
    $Y_K=Y_{NP}$
    $Z_K=Z_{NP}$
    NP=NP-1
    ZZZ=ZZZ-1.0
```

where

ZZZ = fraction of particle removed
LAMDBA = half-life (years)
DELTA = time step (days)
X_K,Y_K,Z_K = particle position (ft)
NP = number of particles in simulation

6. Sinks

The only way particles leave the simulation is by decay (described above) and exiting at sinks. Sinks are where ground water leaves the model system. Typical sinks are wells and gaining streams, but may include constant head boundary nodes, or leakage type boundaries. Sink locations and flow rates are input to the model in the velocity vector inputs.

Each time a particle moves, the program calculates how far the particle is from each sink in the model. If the particle is within the capture radius of the sink, then the particle is removed from the simulation. The total number of particles removed by each sink is accumulated, and the concentration of the solute in the water being removed is calculated and printed at the end of each time step.

For each sink, the cartesian coordinates (x,y), layer, and flow rate are input to the model. If a sink is not input for a place where ground water leaves the model system, then particles will move to this location and bounce around. The model will not recognize and diagnose the user's failure to enter the correct sink locations.

At the end of each particle move increment (defined by DMAX, ZMAX, and DELTA above), the horizontal distance from each sink to the particle is calculated. Which layer the particle is in is also computed.

$$R3=SQR((X_K-X1_L)*(X_K-X1_L)+(Y_K-Y1_L)*(Y_K-Y1_L))$$

where

$R3$ = distance from sink L to particle K (ft)
X_K, Y_K = particle position (ft)
$X1_L, Y1_L$ = sink position (ft)

Then the capture radius of the sink is computed. Different formula are used depending if the sink is a well, or a gaining stream. Gaining streams are denoted in the model by a sink in a grid node that contains an aquifer top elevation in the top layer. Stream sinks are only allowed in the top layer of the model, which is always assumed to be a water table layer. The capture radius of a well or nonstream sink is the minimum of the maximum horizontal move (DMAX) and the analytical distance that a particle could move toward the well during the time step.

$$R4 = \min[DMAX, SQR(QSUM_L*DELTA/3.14/THICK_{II,JJ,KK}*E*7.48)]$$

where

R4 = capture radius (ft)
DMAX = maximum horizontal move before recomputing the time step (ft)
$QSUM_L$ = flow rate for sink L (gpd)
DELTA = time step (days)
$THICK_{II,JJ,KK}$ = thickness of aquifer at column II, row JJ, and layer KK (ft)
E = effective porosity

If particle is closer to the well than the capture radius (R3 < R4) and the particle is in the same layer as the well sink (thus the sink is assumed to be fully penetrating), then the particle is removed from the simulation and the concentration at the sink is updated. Notice that the well capture radius is computed as the entire time step and is constant, rather than the actual time of the move (F). This causes a slight inaccuracy; some particles are captured sooner than they should be. This impact is offset by restricting the capture radius to the maximum move distance. The algorithm is designed to present a realistic picture of contamination approaching a well. If the actual move time is used as the time parameter in the equation above rather than DELTA, then wells with large pumpage rates will capture all the particles in the model at the beginning of the time step, and nothing appears on the screen. This problem is solved by restricting the capture radius to DMAX, the maximum move distance. If one then calculated the capture radius with the time F, then particles would have difficulty reaching the well as F became small. The problem is that the particle move algorithm is a forward difference algorithm. Velocities change rapidly near a pumping well, so the error in the forward difference procedure is accentuated near a well. For most real world problems, however, this is not a problem. It is important to know that the particle reaches the well, not to know the exact time of the arrival. The program has been written with this objective in mind.

$CONC_L=CONC_L+(PM*119872)/(QSUM_L*DELTA)$
$X_K=X_{NP}$
$Y_K=Y_{NP}$
$Z_K=Z_{NP}$
$NP=NP-1$

where
$CONC_L$ = concentration of solute in discharge at sink L (ppm)
PM = particle mass (lbs)
NP = number of particles in simulation

The capture radius of a gaining stream is computed differently. A gaining stream is indicated in the model by having an entry for the top elevation of the aquifer where there is a sink in the top (water table) layer. Normally, this entry is blank. The top elevation is the elevation of the bottom of the river. The river is not fully penetrating, but considered areally extensive. The particle is captured when it gets close enough to the bottom of the stream. The particle capture radius is calculated as the vertical extent of capture.

$R4=QSUM_L*F/DELX/DELY/7.48/E$
$BOTCAPTURE=TOP_{II,JJ,NL}-R4$

CAPTUREMAX=SQR(DELX*DELY)

where

F = move increment until reinterpolation (days)
DELX = column spacing (ft)
DELY = row spacing (ft)
NL = number of layers, denotes top aquifer of model

If R3<CAPTUREMAX and Z_K>=BOTCAPTURE then the particle is removed from the model (using the same algorithm shown above for removal by a well). The assumptions made for gaining streams are that the solute will move up under the stream and into the stream bed.

D. Mapping (MAP and PLT routines)

The MAP and PLT subroutines give the user the ability to prepare concentration maps at a user selected scale of any section of the aquifers being simulated. Data grid (the grid used for calculating velocity vectors) is not used in either of these routines. The user may select the grid spacing independently. The MAP routine prepares a particle map or concentrations map on the screen on a 10 row by 14 column grid. The PLT routine outputs concentrations with x,y coordinates for contour plotting (using other software).

At the start of the MAP routine, the user is asked to select the area to be shown on the screen map graphically. Section II.F. of this report discusses the implementation of the screen graphics and special feature presentation. The user may zoom in or out on the drawing to select the area to be displayed. Either a top view (MAPTYPE$="A"), column cross-section (MAPTYPE$="B"), or row cross-section (MAPTYPE$="C") may be selected.

Once the rectangular area of the screen to be displayed is selected, the RAND3D model creates a 10 by 14 grid.

CDX=(I2!-I1!)/WD
CDY=(J2!-J1!)/LN

where

CDX = column spacing of MAP grid (ft)
CDY = row spacing of MAP grid (ft)
I1!,J1! = x,y coordinates of lower left corner of area to be mapped (ft)
I2!,J2! = x,y coordinates of upper right corner of area to be mapped (ft)
WD = 14 = number of columns
LN = 10 = number of rows

Then the user selects whether a particle map or concentration map is desired. A particle map shows the number of particles in each map grid of the model. A concentration map shows the average concentration in each grid. Either a single layer, row, or column may be viewed or all layers, rows, or columns may be viewed. The program then loops through each particle and determines whether or not the particle is in the user selected layer, row, or column.

A second loop is started to determine where in the MAP grid the particle falls. The number of particles in each grid is tabulated.

$$I = INT(1.5 + ((SCREENX-I1!)/CDX))$$
$$J = INT(1.5 + ((J2!-SCREENY)/CDY))$$
$$NMAP_{I,J} = NMAP_{I,J}+1$$

where

I,J = column and row of map grid in which particle falls

SCREENX,SCREENY - x,y position of particle (ft)

$NMAP_{I,J}$ = counter for tabulating number of particles in each MAP grid

If a particle map was selected, the particle map is output. The values of NMAPI,J are output to the screen. If a concentration map was selected, the coordinates of each grid intersection in the MAP grid is calculated and the relevant thickness at that location found. If the map is a top view of a single layer then the relevant thickness is the aquifer thickness. If the map is a top view of all layers, then the relevant thickness is the total thickness of all aquifers and confining layers. If the map is a row or column cross-section view of a single row or column, then the relevant thickness is the row or column thickness. If the map is a row or column view of all rows or columns, then the relevant thickness is the total dimension of the model in that direction. The relevant thickness is then multiplied by the retardation coefficient of the layer. Concentration is calculated as:

$$NMAP_{I,J} = 16030*NMAP_{I,J}*PM/(E*CDX*CDY*TEMPTHCK)*PPP$$

where

PM = particle mass (lbs)

E = effective porosity

TEMPTHCK = relevant thickness times retardation coefficient

PPP = user selected factor, = .001 for parts per thousand; =1 for parts per million; =1000 for parts per billion

The concentration map is then output to the screen.

The PLT routine operates in a similar fashion to the MAP routine, except the user selects the grid spacing. The user is prompted for the lower left corner of the area to be gridded in model coordinates (x,y). Then the user is prompted for the grid spacing and the number of rows or columns. The user may select a layer of the model to grid, or all aquifer layers may be gridded and output files created for each aquifer layer. The logic and algorithm are then identical to those described for the MAP routine with the exception that cross-section gridding is not allowed in the PLT routine; only top view plots may be created.

E. Initializing Particles (P routine)

The previously discussed routines show how the RAND3D model moves and maps particles. There must be a method of putting particles (solute) into the model. The model provides a routine for initiating the number of particles in the model at each time step. These routines directly input the number of particles. It is up to the user to make sure that the number of particles input are appropriate given the particle mass and/or input concentrations.

Particles may be input in three different geometric forms: rectangular prism, line, or cylinder. By combining these shapes, almost any initial distribution of solute may be simulated. There are no limits on the number of times the user cycles through the particle input routine.

The rectangular prism must be aligned with model axes. The user enters the lower left x,y coordinates, the upper right x,y coordinates, and the top and bottom elevations, the number of particles, and whether or not this source is continuous. A continuous source emits particles constantly throughout the time step. A slug source emits all the particles at the beginning of the time step. Particles are created in the model randomly throughout the rectangular prism. A uniformly distributed random number is used to generate the particle locations.

```
DX = X6  - X5
DY = Y6 - Y5
DZ = Z6 - Z5
FOR I = 1 TO M
  X = X5 + DX*RND
  Y = Y5 + DY*RND
  Z = Z5 +DZ*RND
  NP=NP+1
  $X_{NP}$=X
  $Y_{NP}$=Y
  $Z_{NP}$=Z
  IF A$="C" OR A$="c" THEN SWITCH%$_{NP}$=1 ELSE SWITCH%$_{NP}$=0
NEXT M
```

X5,Y5,Z5 = coordinates of lower left corner of rectangular prism (ft)
X6,Y6,Z6 = coordinates of upper right corner of rectangular prism (ft)
RND = uniformly distributed random number between 0 and 1
NP = number of particles
X_{NP}, Y_{NP}, Z_{NP} = particle position in x,y,z coordinates (ft)
SWITCH$\%_{NP}$ = flag indicating whether or not particle is continuous, 1=continuous, 0=slug
M = loop index

A particle from a continuous source is simulated in the model by making its move a uniformly distributed random time between 0 and DELTA. The actual code implementing this algorithm is at the beginning of the MOVE subroutine. For each particle, the SWITCH% variable is checked, it set equal to one, then the total time step for that particle is set equal to DELTA*RND, where RND is a uniformly distributed random number between zero and one. If SWITCH% is equal to zero (denoting a slug source) then DELTA retains its original value. At the end of the MOVE subroutine, all SWITCH% are set equal to zero. Thus to simulate a continuous source of contamination over multiple time steps it is necessary to reenter the source at the beginning of each time step.

A line of particles in the model is originated in a similar fashion. The user enters the coordinates (x,y,z) of the ends of line, the number of particles, and whether or not it is a continuous source. Particles are then generated uniformly along the line.

Starting the particles in a cylinder is a good option for simulating an injection well. Particles may be originated in the shape of a vertical cylinder. The subroutine prompts for the center coordinates, the top and bottom elevation of the cylinder, the number of particles, and whether or not it is a continuous source. Particles are originated uniformly around the circumference of the cylinder and vertically in a uniform, random distribution.

F. Geographic Special features

One of the unique features of the RAND3D model is its ability to display special features on the screen so the progress of the plume may be monitored with background geographic features as reference points. The user is able to enter the coordinates of up to 20 different special features, selectively draw them on the screen, zoom in or out on these features, and then view the particle paths directly on the screen using the MOVE subroutine. A series of subroutines have been written for the model that implements the special geographic feature option of the model.

The key to understanding the special geographic features subroutine is the WINDOW statement of BASIC. The WINDOW statement allows the user to map the screen directly in real world coordinates, instead of screen pixels. By redefining the screen in real world coordinates, and redrawing the objects, zooming may be implemented. Defining the screen uses the following BASIC statements.

```
VIEW (G1,G2)-(G3,G4),0
WINDOW (I1!,J1!)-(I2!,J2!)
```

where

G1,G2 = upper left corner of screen area to be defined in real world coordinates in screen pixels, =0,0 for color graphics adaptor

G3,G4 = lower right corner of screen area to be defined in real world coordinates in screen pixels, =319,199 for color graphics adaptor

I1!,J1! = real world coordinates of lower left corner of screen area (ft)

I2!,J2! = real world coordinates of upper right corner of screen area (ft)

Any background geographic feature (rivers, buildings, streets, etc) that may be represented as a series of straight lines may be drawn on the screen. The user prepares a special feature file containing graphic commands and x,y coordinates. The graphic commands are similar to those used for driving a Hewlett-Packard pen plotter and the Golden Software PLOT program.

MA x,y - move absolute, move to position x,y with the pen up

PA x,y - plot absolute, plot a line from the present position to x,y

MR x,y - move relative, move x units to the right and y units up with the pen up

PR x,y - plot relative, plot a line from the present position to a position x units to the right and y units up

SP ipen - select pen, use ipen as current pen color

A simple geographic special feature file is shown below. It defines a box.

```
MA 0,0
PA 1000,0
PA 1000,1500
PA 0,1500
PA 0,0
```

If the screen had been defined to include the area from 0,0 to 1000,714 (a typical screen aspect ratio of 1.4) using the statement WINDOW (0,0)-(1000,714), the above special feature file would produce a plot of the lower half of the box. If the screen was then redefined using the statement WINDOW (0,0)-(2100,1500), the entire box would be visible.

The program allows the user to define different special feature files (up to 20) and to assign each file to a key. When the MAP or MOVE subroutines are entered for the first time, the user is presented with a blank screen. This screen has been defined in real world coordinates also entered by the user. By hitting the special feature keys, the different special feature files are opened, read, and plotted on the screen. Some keys are reserved for the purpose of clearing the screen, returning to the default screen coordinates, plotting the grid, plotting sink locations, and zooming.

Zooming is performed by redefining the screen graphically. The zoom routine presents a small box on the screen. The box may be expanded, shrunk, and moved around. When the area to be shown is circumscribed by the box, the program resets the screen coordinates according to the corners of the box and all previously plotted special features are redrawn on the screen.

When the desired area of the screen and special features have been displayed, the user continues with the chosen subroutine. In MOVE, the paths of each particle are plotted on the screen as a red line. The final position of the particle is shown as a white dot. All particles in all layers are visible in the MOVE screen display. In MAP, the corners of the screen area selected are used to define the grid area (I1!,J1! and I2!,J2!).

Cross-section views may also be shown with the special feature routines. The user enters the subroutine in the top view. After displaying selected features, the row or column cross-section mode is entered. The user may graphically select a row or cross-section to profile on the screen. The rows and columns are selected from the rows and columns of the original data grid. The program plots the tops and bottoms of each aquifer layer using a user selected vertical exaggeration ratio. The horizontal extent of the screen will match the displayed extent of the model in the top view in that dimension. For example, if the screen is defined as (0,0) to (1000,741) in the top view, a column cross-section will display 0 to 1000 in the x direction. A row cross-section would display from 0 to 741. The vertical scale is calculated as the default x scale for the screen (SCALEDEF) divided by the user selected vertical exaggeration ratio (ZASPECT). The vertical screen will be defined as from the lower left coordinate of the entered velocity data (LLZ) to lower left coordinate of the entered velocity data (LLZ) to SCALEDEF/ZASPECT. This convention allows for consistent viewing of cross-sections in spite of having different horizontal views. Vertical zooming may be performed by changing the vertical exaggeration of the screen (ZASPECT). Wells are plotted on the screen if they are in the selected row or column. Rivers are plotted also. In the MOVE routine, all particles are displayed as they move through the aquifer. In the MAP routine, the corners of the screen define the area to be gridded.

When the user exits the graphic view in either the MOVE or MAP routines, the view and the position of the screen is saved. If a top view, the entire screen is saved as pixels in file TEMP.SCR on the default drive. The next time the MOVE or MAP routines are executed, the image in TEMP.SCR is restored to the screen. If a cross-section view, only the coordinates of the screen are saved. When the MOVE or MAP routines are executed, the appropriate cross-section is redrawn.

The special feature routines have a help menu to help users remember what key performs which functions. Files of special features and default values may be stored and loaded. The screen aspect ratio may be entered by user. This is necessary because all color screens are different sizes. In order to have geographic features represented without distortion, it is necessary for the user to know the aspect ratio of their screen. The aspect ratio may be rapidly found using the following program.

```
10 REM  FIND ASPECT RATIO OF SCREEN
20 REM  SET SCREEN COORDINATES
30 G1=0
40 G2=199
50 G3=639
60 G4=0
70 REM ENTER ASPECT RATIO
80 INPUT "  ENTER ASPECT RATIO ";G5
90 ASPECT=G5*ABS((G2-G4)/(G3-G1))
100 XC=(G3+G1)/2!
110 YC=(G4+G2)/2!
115 R=ABS(G4-G2)/2
120 REM  SET HI-RES GRAPHICS MODE
130 SCREEN 2:CLS
140 CIRCLE (XC,YC),R,,,,ASPECT
150 LOCATE 24,1
160 PRINT "IF THIS IS A CIRCLE ASPECT RATIO IS ";G5
170 END
```

Another program feature that is grouped with the special feature routines is the ability to predefine the time intervals for transient flow simulations. Nonsteady input files. Each file represents a short period of time. The user may predefine the time periods to be used. A series (up to 20) of times are entered. Each time represents the total time where a velocity file ends. As the user performs subsequent MOVE's, the time remaining in each velocity file is tracked, so that the wrong velocity file is not used. This feature may be ignored; it is not necessary to use this feature or any entries required.

G. Miscellaneous

1. Velocity Input files

The basic and essential input to the RAND3D model is a velocity file. The velocity file contains the grid size and spacing data, the velocity vectors for each grid node in the model, and a list of sinks.

The velocity file is typically created from the output of a finite difference ground water flow model. PREMOD3D is the program written to prepare velocity files from the output of the U.S.G.S three dimensional finite difference model (MODFLOW). Velocity files may be prepared in other ways, as long as they follow the specified format.

The velocity file is read from an ASCII file as a stream of variables. The following list shows the input variables in order.

NC - number of columns
NR - number of rows
NL - number of aquifer layers
DELX - column grid spacing (ft)
DELY - row grid spacing (ft)
LLX - x coordinate of lower left corner of grid (ft)
LLY - y coordinate of lower left corner of grid (ft)
LLZ - z coordinate of lower left corner of grid (ft)

followed by a series of inputs from 1 to NC, 1 to NR, and 1 to NL; for each grid node the following data appears

I - column
J - row
K - aquifer layer
$THICK_{I,J,K}$ - saturated thickness (ft)
$VI_{I,J,K}$ - Darcy velocity from I to I+1 (ft/day)
$VJ_{I,J,K}$ - Darcy velocity from J to J+1 (ft/day)
$VK_{I,J,K}$ - Darcy velocity from K to K+1 (ft/day) in the top layer (NL) it is the apparent velocity of the water table
$BOT_{I,J,K}$ - bottom elevation of layer (ft)
$TOP_{I,J,K}$ - top elevation of layer (ft)

in the top layer (NL) this ordinarily is zero, unless it is a river node in which case it is the elevation of the bottom of the river

followed by a list of sinks, for each sink the following data appears

$X1_L$ - x coordinate of sink (ft)
$Y1_L$ - y coordinate of sink (ft)
$Z1_L$ - layer of sink
$Q1_L$ - discharge of sink (gpd), positive is flow out of model

Example Velocity file (abc110.rnd)

```
20 20  3    100.0     100.0       .0         .0    490.0
 1 20  3   40.0      -.984E-01  .000       .413E-01  660.       695.
 2 20  3   42.0      -.869E-01  .000      -.288E-02  660.       .000
 3 20  3   43.7      -.771E-01  .000      -.339E-02  660.       .000
 4 20  3   45.3      -.686E-01  .000      -.386E-02  660.       .000
 5 20  3   46.7      -.611E-01  .000      -.427E-02  660.       .000
 6 20  3   47.9      -.546E-01  .000      -.464E-02  660.       .000
 7 20  3   49.0      -.487E-01  .000      -.498E-02  660.       .000
 8 20  3   50.0      -.435E-01  .000      -.527E-02  660.       .000
 9 20  3   50.8      -.388E-01  .000      -.552E-02  660.       .000
10 20  3   51.6      -.346E-01  .000      -.573E-02  660.       .000
11 20  3   52.3      -.307E-01  .000      -.591E-02  660.       .000
12 20  3   52.9      -.271E-01  .000      -.606E-02  660.       .000
13 20  3   53.4      -.236E-01  .000      -.617E-02  660.       .000
14 20  3   53.9      -.202E-01  .000      -.626E-02  660.       .000
15 20  3   54.3      -.169E-01  .000      -.633E-02  660.       .000
16 20  3   54.7      -.135E-01  .000      -.637E-02  660.       .000
17 20  3   54.9      -.102E-01  .000      -.640E-02  660.       .000
18 20  3   55.1      -.680E-02  .000      -.642E-02  660.       .000
19 20  3   55.3      -.340E-02  .000      -.643E-02  660.       .000
20 20  3   55.3       .000      .000      -.644E-02  660.       .000
 1 19  3   40.0      -.983E-01  .000       .413E-01  660.       695.
 2 19  3   42.0      -.868E-01 -.110E-03  -.288E-02  660.       .000
 3 19  3   43.7      -.770E-01 -.220E-03  -.340E-02  660.       .000
                .
                .
                .
                .
                .
                .
12  1  1   100.      -.174E-02  .183E-04  -.115E-03  500.       600.
13  1  1   100.      -.160E-02  .183E-04  -.148E-03  500.       600.
14  1  1   100.      -.142E-02  .366E-04  -.176E-03  500.       600.
15  1  1   100.      -.123E-02  .183E-04  -.201E-03  500.       600.
16  1  1   100.      -.101E-02  .183E-04  -.221E-03  500.       600.
17  1  1   100.      -.769E-03  .183E-04  -.238E-03  500.       600.
18  1  1   100.      -.522E-03  .000      -.250E-03  500.       600.
19  1  1   100.      -.266E-03  .183E-04  -.259E-03  500.       600.
20  1  1   100.       .000      .183E-04  -.262E-03  500.       600.
     1150.0       950.0         3.0   .1440E+05
       50.0      1950.0         3.0   3260.
       50.0      1850.0         3.0   3260.
       50.0      1750.0         3.0   3255.
       50.0      1650.0         3.0   3246.
       50.0      1550.0         3.0   3232.
       50.0      1450.0         3.0   3219.
       50.0      1350.0         3.0   3205.
       50.0      1250.0         3.0   3191.
       50.0      1150.0         3.0   3182.
       50.0      1050.0         3.0   3173.
       50.0       950.0         3.0   3173.
```

2. Saving and restoring a run

The RAND3D program permits the user to save all data at any point in the simulation and restart the model from the same point at some later date. ASCII files containing the name of the velocity file, the name of the special feature file, the mass transport coefficients, and the location of each particle in the simulation may be saved and loaded.

The data are saved in an ASCII file in a continuous stream of variables. The following format is used. It is possible to create a file in this format for starting the model.

line 1
    "name:"FILE$ - file name (for reference only)
line 2
    R$ - name of velocity file
line 3
    SPFSAVE$ - name of file of special features and defaults
line 4 and on
    T2 - current time (days)
    E - effective porosity
    DL - longitudinal dispersivity (ft)
    DT - transverse dispersivity (ft)
    DZ - vertical dispersivity (ft)
    PM - particle mass (lbs)
    LAMDBA - half-life (yrs)
    I1! - x coordinate of lower left corner of current view (ft)
    J1! - y coordinate of lower left corner of current view (ft)
    CDX - column grid spacing of current MAP grid (ft)
    CDY - row grid spacing of current MAP grid (ft)
    NP - number of particles
    I2! - x coordinate of upper right corner of current view (ft)
    J2! - y coordinate of upper right corner of current view (ft)
    O - number of slides generated
    SCALEX = I2! - I1!, current x dimension of plan view
    NL - number of layers
    TEMPSCR - temporary screen flag, if =1, then there must be file TEMP.SCR on the default drive and path that contains the saved screen image
    "MAPTYPE$" - flag for type of view, if =
        "A" - plan view
        "B" - row cross-section view
        "C" - column cross-section view
    JROWXS - row number for current row cross-section
    ICOLXS - column number for current column cross-section

for each aquifer layer (1 to NL) and each confining layer (NL+1 to NL+NL-1), the following variable appears

RETARD - retardation factor for layer

for each particle (1 to NP) the following variables appear

X - x coordinate of particle (ft)
Y - y coordinate of particle (ft)
Z - z coordinate of particle (ft)
SWITCH% - flag indicating if particle is from a continuous source or not,
=0 for slug source,
=1 for continuous source

3. Setting Coefficients

RAND3D has a separate routine for inputting and editing mass transport coefficients (SET). The program prompts the user for effective porosity, particle mass, longitudinal dispersivity, transverse dispersivity, vertical dispersivity, half-life for radioactive decay, and retardation coefficients for each aquifer layer and each confining layer.

4. Saving screens

RAND3D permits the user to save the screen views generated with the model. At the end of the MOVE routine, the user may press "s" to save the screen as a file on the default drive. The file is saved using the BSAVE command of BASIC. The first file saved is named SLIDE0.DAT. The second file saved is named SLIDE1.DAT, and so on.

There is a separate routine in the RAND3D model for viewing previously saved screens. This routine goes from 0 (zero) to the current value of variable O (oh), the current number of screens saved, restoring the screen views using the BLOAD command of BASIC.

The special feature routines also save screens for the purpose of speed. After the user selects the plan view area on the screen at the beginning of the MOVE or MAP routines, and proceeds to the rest of the routines, the program saves the current screen to file TEMP.SCR using the BSAVE command of BASIC. The next time the MOVE or MAP routines are entered, file TEMP.SCR is restored to the screen. File TEMP.SCR is saved and restored from the default drive and path.

5. Batch Operation

RAND3D is designed for user friendly, interactive use. There are many real world problems, however, that will take hours to run. An alternative method of running this program is in batch mode. The program is able to run unattended, assuming sufficient printer paper and adequate storage on the default drive.

Unattended operation is performed using the MS-DOS redirection feature. Redirection permits any program using MS-DOS standard input and output to redirect the input and/or output to/from a file. RAND3D uses standard input and output devices. Standard input is the keyboard. Standard output is the screen (monitor). Batch operation of the model is performed by redirecting standard input to be from a file rather than the keyboard. By creating ASCII files of the keystrokes one would enter with the keyboard, RAND3D may operate unattended.

A redirection file is an ASCII file with the same keystrokes as one would enter from the keyboard interactively to make a model run. The file must contain exactly the same keystrokes including "enter"s. A typical redirection file is shown below. The left column shows the file, annotations have been added in the right column. The actual file cannot contain any remarks or annotations.

```
RE                          read in previously
PICCOL.DAT                  saved file PICCOL.DAT
MOVE
60.8333                     move particles for 60 days
10                          with max moves of 10' and 1'
1
GS                          go to move and save screen
INPUT
COL02                       input a new velocity file
MOVE
60.8333                     move again
10
1
GS
INPUT                       input a new velocity file
COL03
MOVE
60.8333                     move again
10
1
GS
WR                          save restart file
COLTEST.DAT
MAP                         map current screen area
GC                          go - concentration map
B                           ppb
```

```
3                        map 3rd layer only
                         end of map (blank line)
PLT                      plot concentrations to file
2090,840                 lower left corner of plot
60                       grid spacing
32,21                    rows, columns
0                        create files for all aquifers
CL1                      seed for plot file names
Q                        quit
```

The redirection file is used by entering in DOS the following command:

```
rand3d <col.inp
```

where

rand3d - command for executing program assuming
RAND3D.EXE on default drive and path
< - MS-DOS redirection of standard input operator
col.inp - name of ASCII input redirection file
containing keystrokes.

Redirection only works with standard character input (as far as this author knows); it is not possible to use arrow keys to redefine the screen window. Thus, it is necessary to interactively run the model to set the screen to the correct area. A typical session consists of running the model interactively to input initial velocity file, inputting mass transport coefficients, selecting the proper screen view, and saving this simulation data. A file (redirection file) of keystrokes is then created using an appropriate editor. The first step of this redirection file is the reading of the previously saved simulation data file. Typically slides are saved throughout the simulation, as well as periodic plots generated. The user is advised to save data at intervals throughout the unattended operation, so in case of error, the run may be restarted.

IV. User Instructions

A. Preliminary

Before running the RAND3D program, several preliminary steps must be performed. First, check to see if the RAND3D model will adequate simulate your problem. Current program limits are:

- maximum input grid of 45 columns, 45 rows, and three layers;
- maximum number of particles is 10000;
- maximum number of sinks (wells or gaining streams) is 99;
- maximum number of special feature files is 20;.
- at least two aquifer layers must be simulated.

These limits may be easily changed by changing the dimension statements at the beginning of the program and recompiling. The RAND3D program was originally compiled with Microsoft QuickBasic 3.0.

Second, a suitable computer system is necessary. The computer must be an IBM PC compatible system with at least 640K of RAM (random access memory). For most real world problems, a hard drive will be required, although small problems may be run from a floppy disk. A line printer must be attached to the computer; the program prints a record of the actions taken as the simulation proceeds. There must be a numeric coprocessor (80x87 chip) in the computer, although it is possible to recompile the program using Microsoft QuickBasic to run without the numeric coprocessor. The program requires a color graphics adaptor (CGA video card) and color monitor or compatible (EGA, VGA, or MCGA).

Third, a velocity file must be prepared. A velocity file is typically prepared from the output of a numerical ground water flow model. A program, PREMOD3D, has been written to prepare velocity files from the output of the U.S.G.S. three dimensional finite difference model (MODFLOW). The velocity file contains the velocity vectors from grid node to grid node. The grid must be regularly spaced in the horizontal plane. Layers may be of variable thickness. Section III.G.1. of this user's manual describes the format for the velocity file.

Fourth, the mass transport parameters of the problem must be determined. These include effective porosity, the dispersivities in the longitudinal, transverse, and vertical directions, retardation coefficients (to simulate linear reversible adsorption), and half-life (to simulate first order decay). The strength (lbs), location (x,y,z coordinates), and timing of sources of contamination must be determined. Pollutant sources are entered into the model as rectangular prisms, vertical cylinders, or as lines. The location of each source must be tabulated in x,y,z coordinates. The coordinate system used by the RAND3D model is based on the flow model grid that was used to prepare a velocity file, although any coordinate system could be used as long as the grid is rectilinear with the coordinate system (aligned). For the typical case, the lower left corner of the flow model grid is assigned coordinate 0,0,0. The x coordinate increases to the right (east). The y coordinate increases to the top of the grid (north), and the z coordinate increases from the bottom of layer 1 to the water table layer on top. Figure 7 in Section III of this user's manual defines the coordinate system. The final determination necessary to run the model is the particle mass. The user must select the total number of particles to be used in the simulation. The maximum number is 10,000 in this version of the program, but this limit could be changed with recompilation. Typically, several thousand are necessary for adequate precision and resolution. The maximum total mass of contamination that is to be in the model at any given moment (lbs) divided by the chosen number of particles gives the particle mass. This number should be chosen carefully and conservatively. It is not possible to increase the number of particles above the limit compiled into the program, and one should not change the particle mass in the middle of a simulation.

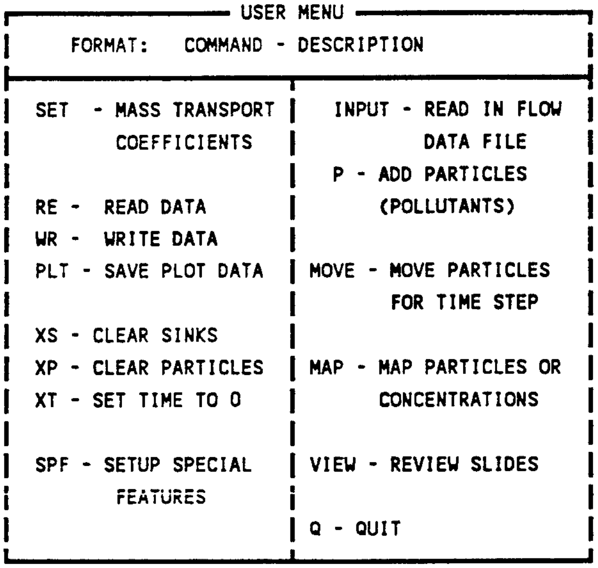
The source parameters for a problem should be calçulated before running the model. For each source, tabulate the shape (rectangular, cylinder,line), the location (x,y,z), the time (what step(s) of the simulation) of particle insertion, the type (slug or continuous), and the number of particles. The total number of particles is calculated by dividing the total strength of the pollutant source (lbs) by the chosen particle mass (lbs) for the simulation.

Fifth, special feature files may be prepared. Special feature files are ASCII files containing the x,y coordinates of geographic features to be displayed on the screen while the RAND3D program is running. The structure of these files is described in Section III.F.

Finally, the program is ready to run. Run the RAND3D program by copying the program (RAND3D.EXE) to the default drive (here it is C) and typing in

```
C>rand3d
```

The program prints the program title on the printer. The user will see the following menu displayed on the screen.



```
┌──────────────── USER MENU ──────────────────┐
│   FORMAT:   COMMAND - DESCRIPTION            │
├──────────────────────┬──────────────────────┤
│ SET  - MASS TRANSPORT │  INPUT - READ IN FLOW │
│        COEFFICIENTS   │          DATA FILE    │
│                       │  P - ADD PARTICLES    │
│ RE -  READ DATA       │      (POLLUTANTS)     │
│ WR -  WRITE DATA      │                       │
│ PLT - SAVE PLOT DATA  │ MOVE - MOVE PARTICLES │
│                       │        FOR TIME STEP  │
│ XS - CLEAR SINKS      │                       │
│ XP - CLEAR PARTICLES  │ MAP - MAP PARTICLES OR│
│ XT - SET TIME TO 0    │       CONCENTRATIONS  │
│                       │                       │
│ SPF - SETUP SPECIAL   │ VIEW - REVIEW SLIDES  │
│       FEATURES        │                       │
│                       │ Q - QUIT              │
└───────────────────────┴───────────────────────┘

COMMAND ?
```

The menu options may be executed in any logical order.

B. INPUT

The typical first option to execute is to read in a velocity file (INPUT). As discussed above, the velocity file must be previously prepared, typically from the output of a ground water flow model. The velocity file must have a file name extension of ".RND" and should reside on the default drive of the computer system.

```
┌─────────────────── USER MENU ──────────────────┐
|    FORMAT:   COMMAND - DESCRIPTION             |
├────────────────────────┬───────────────────────┤
| SET  - MASS TRANSPORT  |   INPUT - READ IN FLOW |
|        COEFFICIENTS    |           DATA FILE    |
|                        |   P - ADD PARTICLES    |
| RE -  READ DATA        |       (POLLUTANTS)     |
| WR -  WRITE DATA       |                        |
| PLT - SAVE PLOT DATA   | MOVE - MOVE PARTICLES  |
|                        |        FOR TIME STEP   |
| XS - CLEAR SINKS       |                        |
| XP - CLEAR PARTICLES   | MAP - MAP PARTICLES OR |
| XT - SET TIME TO 0     |       CONCENTRATIONS   |
|                        |                        |
| SPF - SETUP SPECIAL    | VIEW - REVIEW SLIDES   |
|       FEATURES         |                        |
|                        | Q - QUIT               |
└────────────────────────┴───────────────────────┘

COMMAND ? input
```

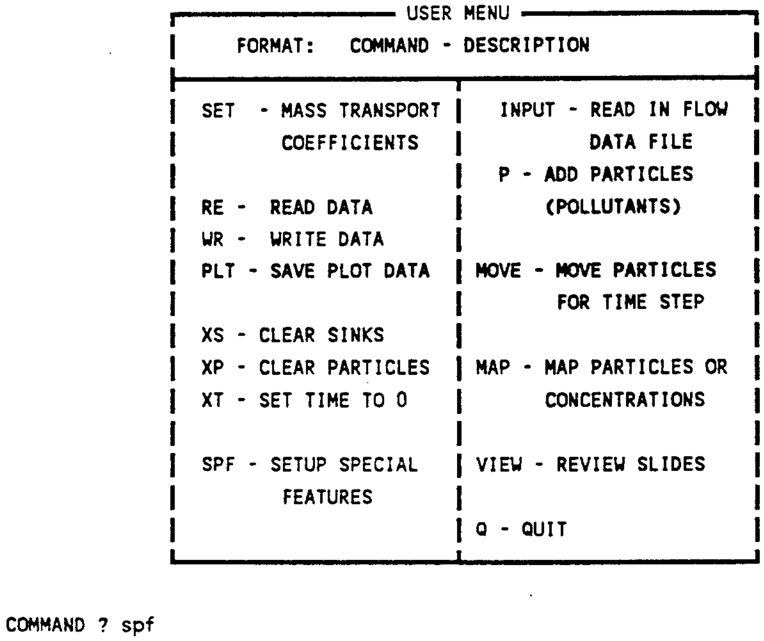
The program lists the "*.RND" files that reside on the default drive and prompts the user for the file name to be read in. Only the first part of the file name is entered; the program automatically adds the extension ".RND"

```
                  VELOCITY FILE INPUT DATA MENU

D:\DON
TST101   .RND      TST102   .RND      TST103   .RND      TST104   .RND
TST105   .RND      TST106   .RND      TST107   .RND      TST108   .RND
TST109   .RND      TST110   .RND      TST111   .RND      TST112   .RND
TST113   .RND      TST114   .RND      TST115   .RND      TST116   .RND
TST117   .RND      TST118   .RND      TST119   .RND      TST120   .RND
COL18    .RND
 4.350091371478808D-311 Bytes free

Enter the name of the external VELOCITY file from above--DO NOT
use any extensions or drive designation nor more than
eight (8) characters.  You MUST choose from the above list--
-------------------are you ready??

Enter the name of the external file for input (Example HEAD1) col18
```

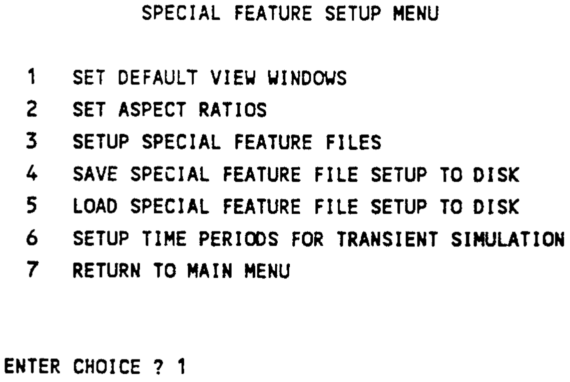
The name of the velocity file that is read is printed on the line printer.

C. SPF

The next option that is typically executed, and must be executed before the MOVE and/or MAP routines are run is the special features routine (SPF). Even if there are no special features, the user must select a default screen viewing window and aspect ratio for the screen and the cross-section view.



```
┌──────────────── USER MENU ─────────────────┐
|    FORMAT:   COMMAND - DESCRIPTION         |
├─────────────────────┬──────────────────────┤
| SET  - MASS TRANSPORT |  INPUT - READ IN FLOW |
|        COEFFICIENTS   |          DATA FILE    |
|                       |    P - ADD PARTICLES  |
| RE -   READ DATA      |        (POLLUTANTS)   |
| WR -   WRITE DATA     |                       |
| PLT - SAVE PLOT DATA  | MOVE - MOVE PARTICLES |
|                       |        FOR TIME STEP  |
| XS - CLEAR SINKS      |                       |
| XP - CLEAR PARTICLES  | MAP - MAP PARTICLES OR|
| XT - SET TIME TO 0    |       CONCENTRATIONS  |
|                       |                       |
| SPF - SETUP SPECIAL   | VIEW - REVIEW SLIDES  |
|       FEATURES        |                       |
|                       | Q - QUIT              |
└───────────────────────┴───────────────────────┘

COMMAND ? spf
```



```
              SPECIAL FEATURE SETUP MENU

    1   SET DEFAULT VIEW WINDOWS
    2   SET ASPECT RATIOS
    3   SETUP SPECIAL FEATURE FILES
    4   SAVE SPECIAL FEATURE FILE SETUP TO DISK
    5   LOAD SPECIAL FEATURE FILE SETUP TO DISK
    6   SETUP TIME PERIODS FOR TRANSIENT SIMULATION
    7   RETURN TO MAIN MENU


 ENTER CHOICE ? 1
```

The first special feature routine that should be executed is to set the default window size. When the MOVE or MAP routines are first run, the screen is mapped to these default coordinates. The plan view section of the aquifer that the user wishes to have appear on the screen as a default should be defined. The program prompts for the lower left corner (x,y) coordinates and the distance across the bottom of the screen.

```
ENTER LOWER LEFT CORNER OF DEFAULT VIEW WINDOW [ 2090 , 840 ]
ENTER HORIZONTAL WIDTH OF DEFAULT VIEW WINDOW [ 1260 ] ? 1500
```

The coordinates displayed as defaults for these prompts (in the brackets) are based on the input velocity file. The lower left corner coordinates are the LLX and LLY coordinates of the input velocity file (see Section III.G.1. for a description of the velocity file). The default for the horizontal width of the view window is the distance across the columns of the input model grid (DELX*NC). If the user accepts these default values, the input grid will fill the width of the screen. Default values are accepted by pressing "enter".

The next special feature routine (Option 2) is the specification of aspect ratios. The user may specify an aspect ratio for the screen and select a vertical exaggeration for cross-section plots. These inputs are required to run the program and <u>must</u> be entered before the MOVE or MAP routines are executed. The screen aspect ratio is the width of the screen divided by the height of the screen. Most IBM PC's have a value between 1.2 and 1.45. The default value is 1.4. The user should check the proper value for their equipment (see Section III.F.). The vertical exaggeration ratio is the factor that defines the vertical scale for cross-section plots. Vertical exaggeration allows the cross-section plots to be legibly displayed. The default vertical exaggeration is calculated as the default screen width divided by the distance from the water table in column 1, row 1 to the bottom of layer 1 in column 1, row 1. This default is frequently too large ; the cross-section plot fills the screen making the water table layer hard to see. A smaller vertical exaggeration ratio frequently permits better cross-section views.

```
ENTER ASPECT RATIO (HORIZONTAL/VERTICAL) OF SCREEN [ 1.33 ] ?
ENTER VERTICAL EXAGGERATION OF CROSS-SECTION PLOT [ 5 ]? 6
```

The next special feature routine (Option 3) is optional. The user may plot background geographic features on the screen for reference. Anything may be plotted by preparing an ASCII file that details the pen movements (up or down), and the x,y coordinates of the geographic feature. A full description of the format of special feature files is given in Section III.F. Up to twenty different special feature files may be prepared and used. The program permits each special feature file to be assigned to a key. When this key is pressed while in the screen plotting routine, the file is plotted on the screen. This feature permits the user to select the geographic features that are displayed. Some or all of the features may be displayed. The program prompts the user for a file name and the key to assign this feature. When all features have been entered, press "enter" in response to the file prompt to end the routine.

```
ASSIGN SPECIAL FEATURE FILE NAMES TO KEYS, ENTER EACH FILE NAME
AND KEY ASSIGNMENT IN ORDER, ENTER A CARRIAGE RETURN FOR A BLANK
FILE NAME TO EXIT THIS ROUTINE

ENTER SPECIAL FEATURE FILE NAME [] rivers.dat
ENTER ONE KEY FOR THIS SPECIAL FEATURE [] ? a
ENTER SPECIAL FEATURE FILE NAME [] bldgs.dat
ENTER ONE KEY FOR THIS SPECIAL FEATURE [] ? b
ENTER SPECIAL FEATURE FILE NAME [] streets.dat
ENTER ONE KEY FOR THIS SPECIAL FEATURE [] ? c

THAT KEY IS ALREADY ASSIGNED TO ANOTHER SPECIAL FEATURE, CHOOSE ANOTHER
ENTER SPECIAL FEATURE FILE NAME [streets.dat]
ENTER ONE KEY FOR THIS SPECIAL FEATURE [c] ? f
```

Some keys are reserved for other uses. The program will not permit you to assign two special feature files to the same key and will not permit assignment to a key with some other function. The reserved keys are C, D, M, N, W, X, Y, and Z.

When the above special feature routines have been executed, the user may save the entries (option 4). This saves time in future simulations of the same problem. Saving the special feature setup data also permits batch operation of the program. Any file name may be assigned to the special feature setup file. This file contains the default screen size, the aspect ratios, and the special feature file key assignments.

```
ENTER FILE SPECIFICATION TO SAVE SPECIAL FEATURE DATA ON [] xxx.spf
```

There is another routine (option 5) for reloading a previously saved special feature file.

```
ENTER FILE SPECIFICATION TO LOAD SPECIAL FEATURE DATA FROM [] xxx.spf
```

Option 6 of the special feature submenu is the a setup routine for transient simulation. This routine allows the user to specify end times for each velocity file when performing a transient simulation. The program tracks the time that each velocity file is to be used so the user does not run transient simulations with an invalid velocity file. This option is rarely used. Typical entries are shown below. There are five velocity files each representing a duration of two months. The user enters the end time of each velocity file. The user may then run time steps of any length (MOVE) and the program will prompt the user when a new velocity file is to be entered. These data are not saved in the special feature setup file; they must be reentered each time the program is run.

```
IF MULTIPLE VELOCITY FILES ARE TO BE USED (TRANSIENT SIMULATION)
THE USER MUST ENTER THE NUMBER OF VELOCITY FILES THAT WILL BE USED
AND THE ENDING TIME FOR EACH VELOCITY FILE

ENTER THE NUMBER OF VELOCITY FILES (TIME PERIODS) [ 0 ] ? 5
ENTER THE ENDING TIME (DAYS) FOR VELOCITY FILE # 1  [ 0 ] ? 60.833
ENTER THE ENDING TIME (DAYS) FOR VELOCITY FILE # 2  [ 0 ] ? 121.666
ENTER THE ENDING TIME (DAYS) FOR VELOCITY FILE # 3  [ 0 ] ? 181.5
ENTER THE ENDING TIME (DAYS) FOR VELOCITY FILE # 4  [ 0 ] ? 242.333
ENTER THE ENDING TIME (DAYS) FOR VELOCITY FILE # 5  [ 0 ] ? 365
```

Option 7 of the special feature submenu returns to the main menu.

```
                 SPECIAL FEATURE SETUP MENU

   1   SET DEFAULT VIEW WINDOWS
   2   SET ASPECT RATIOS
   3   SETUP SPECIAL FEATURE FILES
   4   SAVE SPECIAL FEATURE FILE SETUP TO DISK
   5   LOAD SPECIAL FEATURE FILE SETUP TO DISK
   6   SETUP TIME PERIODS FOR TRANSIENT SIMULATION
   7   RETURN TO MAIN MENU


ENTER CHOICE ? 7
```

D. SET

The next typical routine to execute is the setting of mass transport coefficients (SET). The user is prompted for effective porosity, particle mass (lbs), longitudinal dispersivity (ft), transverse dispersivity (ft), and vertical dispersivity (ft), half-life (years), and retardation coefficients for each aquifer and each confining layer of the model. Porosity must be between zero and one. It is used to calculate the seepage velocity from the input velocity as well as calculate the volume of ground water in a section of the aquifer for concentration calculations. Particle mass must have been previously calculated by the user before starting the simulation (see the beginning of the user instructions). Dispersivities determine the dispersion coefficients. The dispersion coefficient is assumed to be a linear function of velocity. There are other functions for dispersion, such as asymptotic with distance of travel, but the constant dispersivity function is the most widely used. Vertical dispersivity is typically a small number, less than one foot. Zero may be suitable for many simulations. Excessive vertical dispersivity leads to unrealistic results. Half-life is entered if the user wishes to have the solute undergo first order decay. An entry of zero will default to a half-life of 1E10 years, which prevents any decay. Retardation simulates linear, reversible adsorption. The assumption of linear, reversible adsorption is quite suitable for many chemicals at low concentrations. Section II.B.3. discusses the calculation of retardation based on aquifer properties and the chemical distribution coefficient. A separate retardation coefficient may be entered for each layer of the simulation, including the confining layers. The layers are numbered through the aquifers from bottom to top and then through the confining layers. Thus in a three aquifer system, with two confining layers, the aquifers would be numbered 1-3, from bottom to top, and the confining layers would be 4 and 5 from bottom to top. Figure 1 shows the layer number scheme used in the model. A retardation of 1 (the default) indicates no adsorption.

```
   COMMAND ? set

///////////////BASIC TRANSPORT COEFFICIENTS\\\\\\\\\\\\\\
     ENTER POROSITY [ 0 ] ? .1
     ENTER PARTICLE MASS (LBS/PARTICLE) [ 0 ] ? .04
     ENTER LONGITUDINAL DISPERSIVITY (FT) [ 0 ] ? 30
     ENTER TRANSVERSE DISPERSIVITY (FT) [ 0 ] ? 5
     ENTER VERTICAL DISPERSIVITY (FT) [ 0 ] ? .1
     ENTER HALF-LIFE FOR FIRST ORDER DECAY (YEARS)(ZERO FOR NONE) [ 0 ] ?
      ENTER RETARDATION COEFFICIENT FOR LAYER  1  [ 1 ] ?
      ENTER RETARDATION COEFFICIENT FOR LAYER  2  [ 1 ] ?
      ENTER RETARDATION COEFFICIENT FOR LAYER  3  [ 1 ] ?
      ENTER RETARDATION COEFFICIENT FOR LAYER  4  [ 1 ] ? 1.5
      ENTER RETARDATION COEFFICIENT FOR LAYER  5  [ 1 ] ? 1.5
```

This routine may be used at any time in the simulation to change the values of the mass transport parameters,however, do not change the particle mass. The previously entered values will appear in brackets at the end of the prompt. This value may be retained by pressing the "enter" key. Whenever the SET routine is executed, the chosen parameter values are printed to the line printer.

```
 COMMAND ? set

////////////////BASIC TRANSPORT COEFFICIENTS\\\\\\\\\\\\\\
    ENTER POROSITY [ .1 ] ?
    ENTER PARTICLE MASS (LBS/PARTICLE) [ .04 ] ?
    ENTER LONGITUDINAL DISPERSIVITY (FT) [ 30 ] ?
    ENTER TRANSVERSE DISPERSIVITY (FT) [ 5 ] ?
    ENTER VERTICAL DISPERSIVITY (FT) [ 0.1 ] ?
    ENTER HALF-LIFE FOR FIRST ORDER DECAY (YEARS)(ZERO FOR NONE) [ 1E+10 ] ?
     ENTER RETARDATION COEFFICIENT FOR LAYER  1  [ 1 ] ?
     ENTER RETARDATION COEFFICIENT FOR LAYER  2  [ 1 ] ?
     ENTER RETARDATION COEFFICIENT FOR LAYER  3  [ 1 ] ?
     ENTER RETARDATION COEFFICIENT FOR LAYER  4  [ 1.5 ] ?
     ENTER RETARDATION COEFFICIENT FOR LAYER  5  [ 1.5 ] ?
```

E. P(ollutants)

At this point, the RAND3D model is ready to run, except there are no particles (solute) in the system. Particles are entered into the model using the Add Pollutant subroutine (P). Particles may be entered in any of three configurations: rectangular prism, vertical cylinder, or line. Particles may be assumed to be a continuous source, or slug source. A slug source is a source that is assumed instantaneous at a single point in time. A continuous source generates particles continuously over the next time step. This is an important point; the number of particles entered for a continuous source is a function of both the strength of the source and the time step to be used. The program does not check the time step against strength (it cannot, it only knows how many particles there are); the user must do this properly. The formula for the number of particles from a continuous source is:

# = (strength -lbs/day)(DELTA)/PM

where

# - number of particles at continuous source
DELTA - time step entered in MOVE routine (days)
PM - particle mass entered in SET routine (lbs)

For a rectangular source, the user enters the lower left plan view coordinates of the rectangular prism (x,y), the upper right plan view coordinates of the rectangular prism (x,y), and the bottom and top elevations. All coordinates are in feet in the previously described coordinate system. The program checks that all entered coordinates are within the model grid. The program randomly distributes particles within this space.

```
 COMMAND ? p
///////////////////////////PARTICLES\\\\\\\\\\\\\\\\\\\\\\\\\\\


      ADD PARTICLES TO SYSTEM? (Y-YES, <RETURN>-NO)? y

      CYLINDER, LINE, OR RECTANGULAR (C, L, R) ? r
      ENTER COORDINATES OF LOWER LEFT CORNER (X,Y)? 2500,2600
      ENTER COORDINATES OF UPPER RIGHT CORNER (X,Y)? 2500,2600
      ENTER Z COORDINATES (LOWER,UPPER) ? 650,690

      Enter the letter C for CONTINUOUS POLLUTION--otherwise a <RETURN>
      or any other character will assume a SLUG input? c
      YOU ARE NOW IN THE CONTINUOUS MODE FOR THIS SOURCE

      ENTER NUMBER OF PARTICLES ? 100

      DO YOU WANT TO REDO THIS SCREEN? (Y-YES, <RETURN>-NO)?
```

For a cylindrical source, the user enters the plan view coordinates of the center of the cylinder (x,y), the radius, and the bottom and top elevations. All coordinates are in feet in the previously described coordinate system. The program checks that all entered coordinates are within the model grid. The program distributes particles around the circumference of the cylinder evenly in the plan view and randomly in the vertical dimension.

```
      ADD PARTICLES TO SYSTEM? (Y-YES, <RETURN>-NO)? y

      CYLINDER, LINE, OR RECTANGULAR (C, L, R) ? c
      CYLINDER NUMBER  1

      ENTER CYLINDER CENTER COORDINATES (X,Y)? 2700,2000
      ENTER CYLINDER RADIUS (FT) ? 20
      ENTER BOTTOM AND TOP  ELEVATION OF CYLINDER (FT) ? 500,550
      Enter the letter C for CONTINUOUS POLLUTION--otherwise a <RETURN>
      or any other character will assume a SLUG input?

      ENTER NUMBER OF PARTICLES ? 20

DO YOU WANT TO REDO THIS SCREEN? (Y-YES, <RETURN>-NO)?
        CYLINDER DONE

      ADD PARTICLES TO SYSTEM? (Y-YES, <RETURN>-NO)?
```

For a line source, the user enters the x,y,z coordinates of both ends of the line. All coordinates are in feet in the previously described coordinate system. The program checks that all entered coordinates are within the model grid. Particles are uniformly spaced along this line.

```
 COMMAND ? p
//////////////////////////PARTICLES\\\\\\\\\\\\\\\\\\\\\\\\\\


       ADD PARTICLES TO SYSTEM? (Y-YES, <RETURN>-NO)? y

       CYLINDER, LINE, OR RECTANGULAR (C, L, R) ? l
       LINE NUMBER  1

         COORDINATES OF LINE BEGINNING (X,Y,Z) ? 2300,1000,600
         COORDINATES OF END OF LINE (X,Y,Z) ? 2400,1200,650
       Enter the letter C for CONTINUOUS POLLUTION--otherwise a <RETURN>
       or any other character will assume a SLUG input?

         NUMBER OF PARTICLES ? 43

DO YOU WANT TO REDO THIS SCREEN? (Y-YES, <RETURN>-NO)?

       ADD PARTICLES TO SYSTEM? (Y-YES, <RETURN>-NO)?
```

The Add Pollutant subroutine (P) prints all entries to the line printer. One may enter as many sources as needed; there is no limit in the program. The program continuously places and initiates particles during this routine. At the end of each entry, the total number of particles in the model is printed to the screen and line printer.

F. RE(ad) and WR(ite)

At this point, the model is ready to run. This is also an appropriate time to discuss the saving and restoring of model data. By using the WRITE and READ routines, all data inputs may be saved to an ASCII file and later restored. The program prompts for a file name and then reads/writes the data. There are no default extensions assumed for either of these routines; they must be entered by the user. The format of the files is described in Section III.G.2. If these routines have been previously used during a model session, the default file names will appear in brackets. The default name may be used by pressing "enter".

(writing a file)

```
COMMAND ? wr
Enter the name of the external file to be opened, including ext. ? xxx.dat
```

(reading a file)

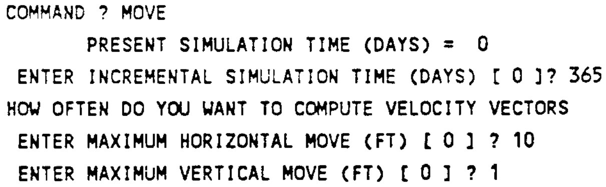
```
COMMAND ? re

Enter filename, including extension (example PEORIA.DAT) [xxx.dat] ?
NOW PROCESSING FILE col18.RND
```

When either of these commands are executed, the name of the file written or read is printed on the line printer.

G. MOVE

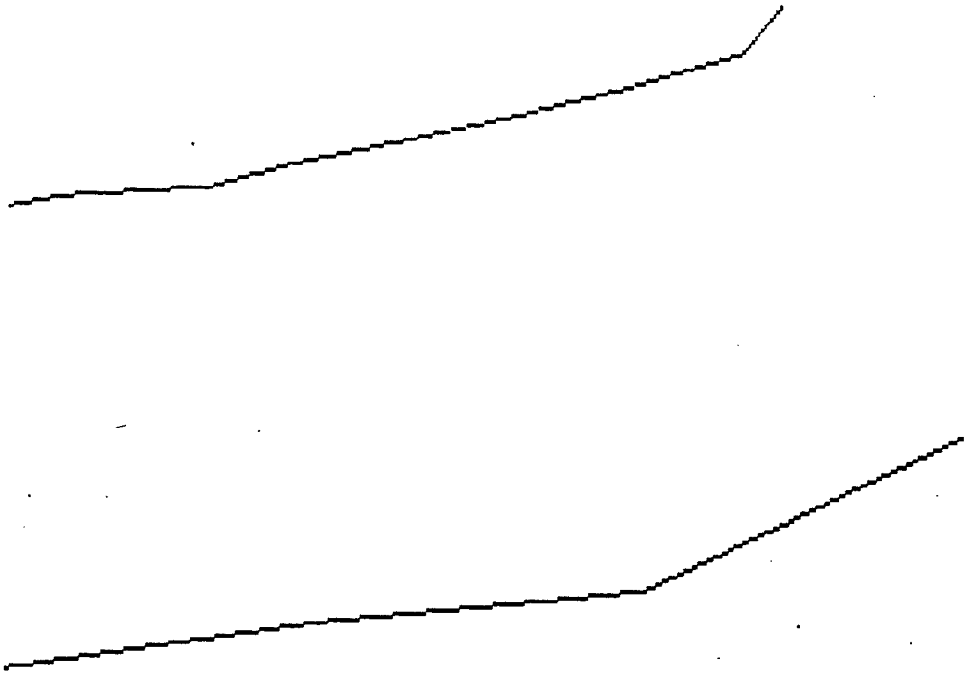
The MOVE routine is the heart of the program. The MOVE routine advances the particles for a user specified time step by advection, and dispersion. The effects of adsorption (retardation) and first order decay are included in this routine.

At the beginning of the MOVE routine, the user is prompted for the time step (days), the maximum horizontal move distance (ft), and the maximum vertical move distance (ft). The maximum horizontal move distance is the plan view distance that a particle is allowed to move before the velocity vectors are reinterpolated. Smaller distances give more accurate results (more frequent reinterpolation). Larger distances permit more rapid calculation. An appropriate value is 0.2 times the minimum horizontal grid dimension (Prickett, Naymik, and Lonnquist, 1981). An appropriate value for the maximum vertical move distance is typically much less than the maximum horizontal move because vertical velocities are typically much smaller than horizontal velocities. The ratio of maximum horizontal move to maximum vertical move should be about the same as the typical ratio of horizontal to vertical velocity.

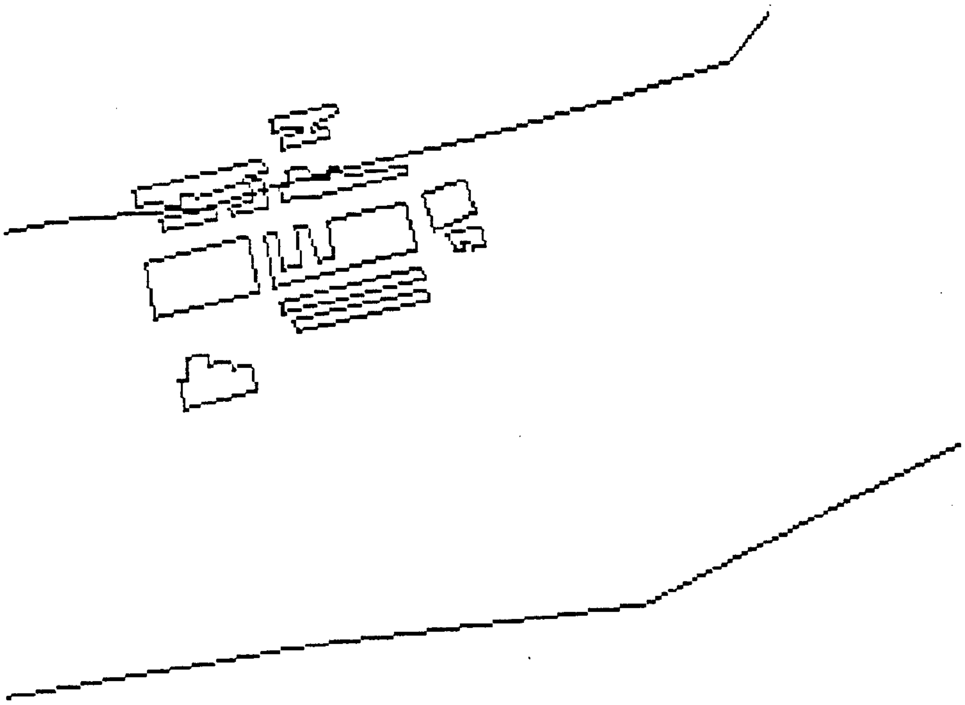


```
COMMAND ? MOVE
       PRESENT SIMULATION TIME (DAYS) =  0
 ENTER INCREMENTAL SIMULATION TIME (DAYS) [ 0 ]? 365
HOW OFTEN DO YOU WANT TO COMPUTE VELOCITY VECTORS
 ENTER MAXIMUM HORIZONTAL MOVE (FT) [ 0 ] ? 10
 ENTER MAXIMUM VERTICAL MOVE (FT) [ 0 ] ? 1
```

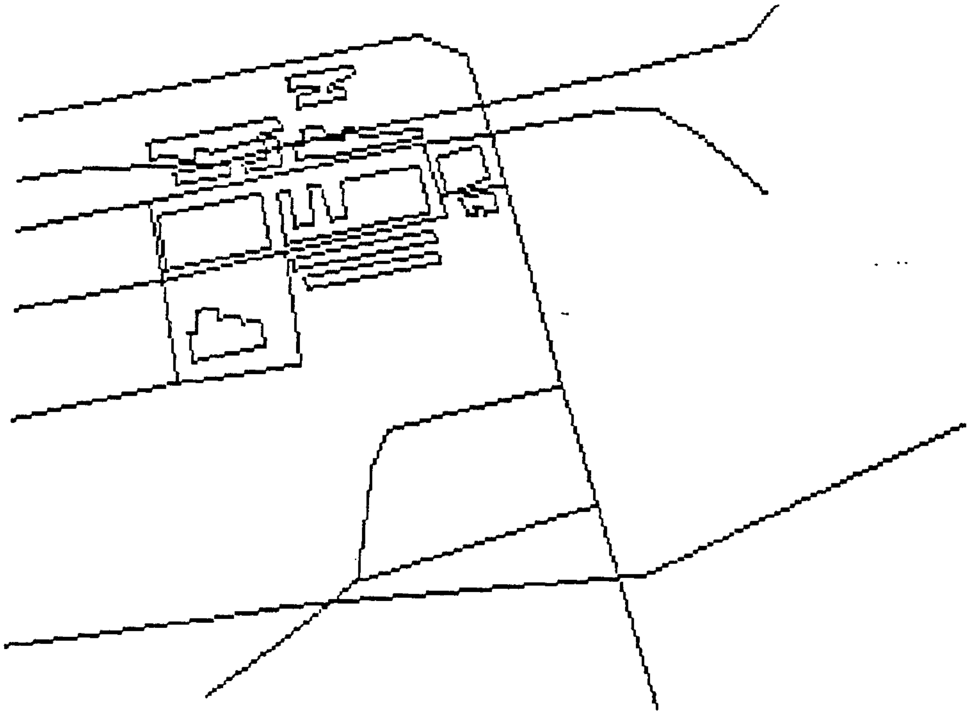
The next screen the user sees is blank the first time the program is run. This screen is the special features graphic screen. Predefined special features may be plotted on the screen (see instructions under the SPF routine above). The user may select the area to be displayed by zooming in or out. The progress of the particles may then be seen on the screen as the program runs. A special feature file is plotted on the screen by pressing the key associated with the file (see SPF instructions above). Assuming that the "a" key has been predefined as the key for an ASCII file "rivers.dat" that contains the plotting directions to plot two stream courses on the screen, and that the default screen window as been appropriately defined, pressing the "a" key yields the following screen



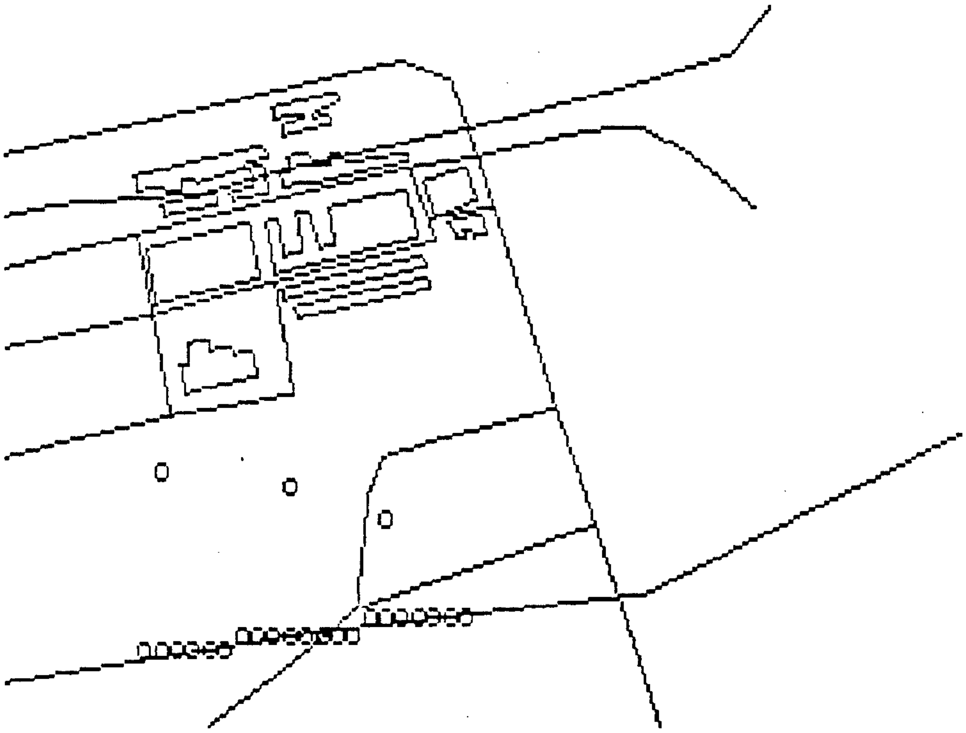
Pressing the "b" key assuming that "b" has been previously associated with "bldgs.dat" yields:



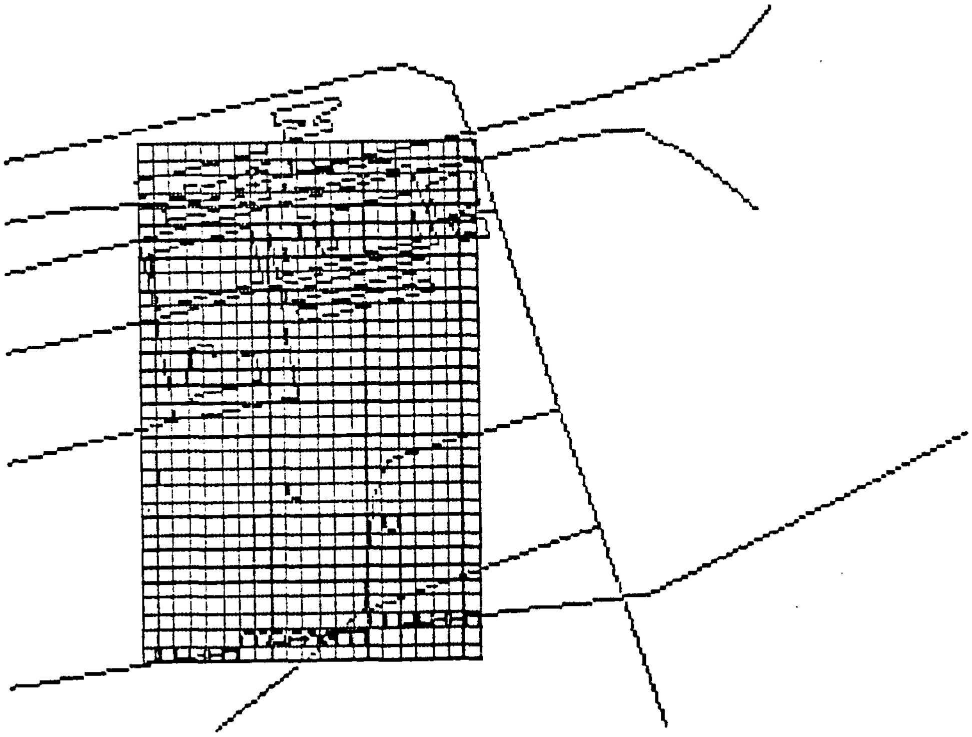
Pressing the "f" key assuming that "f" has been previously associated with "streets.dat" yields:



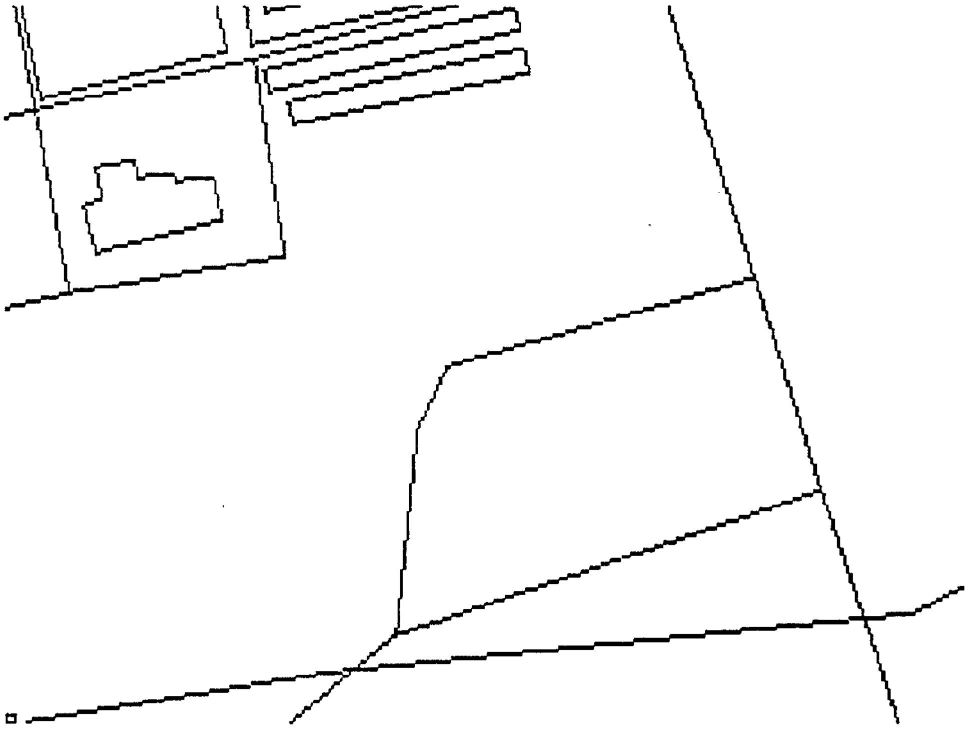
The above three keys were user created and specified geographic feature files. Up to 20 of these files may be specified and drawn to the screen by pressing a key. The files defining the coordinates may contain color commands (see Section III.F.). There are also some predefined keys. Pressing "w" draws all the sinks (wells, gaining streams) on the screen as small circles in white. The size of the circles is calculated by the program and is a constant fraction of the screen width.



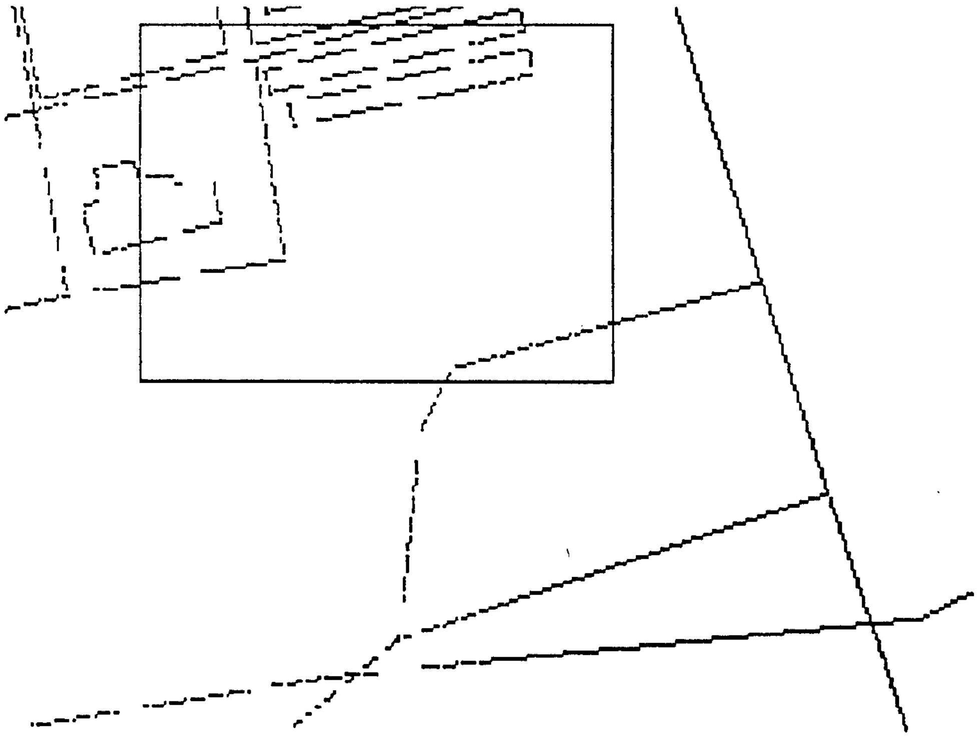
Pressing "d" plots the input velocity file grid on the screen. The grid is a block centered convention. The lower left corner of the grid is at LLX,LLY (see Section III.G.1.). It is shown in light blue. Figure 7 shows the grid definition.



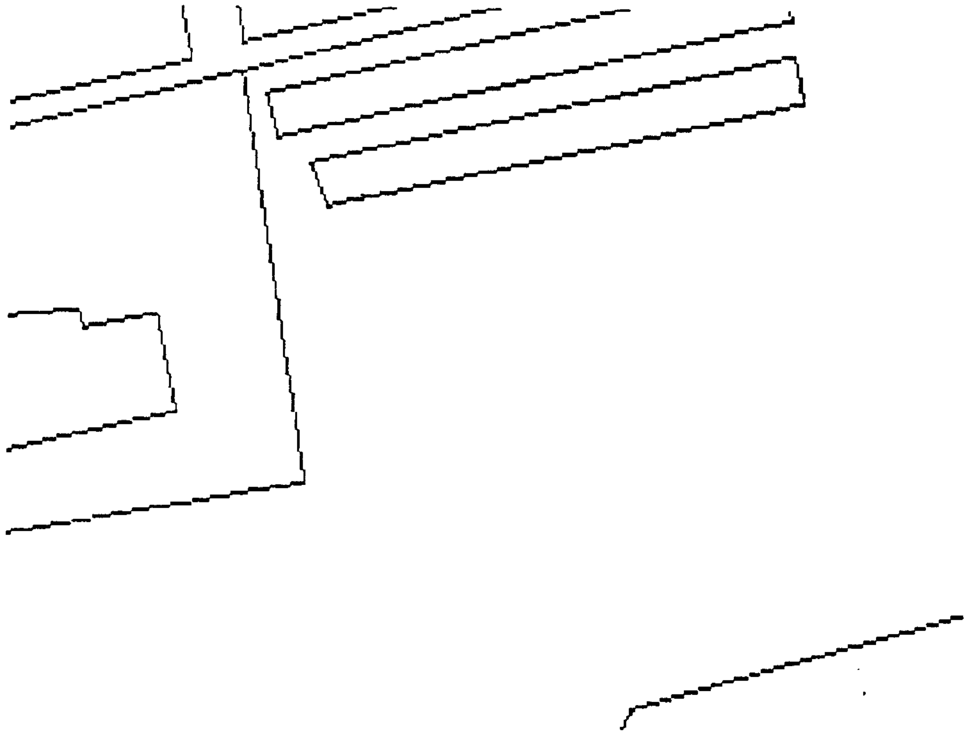
The ability of the user to customize the screen display and to graphically select the area for display is one of the major features of the RAND3D model. Zooming is performed by pressing the "z" key to bring the light blue zoom box to the lower left corner of the screen, moving, expanding the shrinking the zoom box until the proper area has been selected, and then pressing "enter" to actually perform the zoom. On the following picture, notice the zoom box in the lower left corner of the screen.



The zoom box is moved using the four arrow keys. It may be moved off the screen, and work, but the user will not know exactly where it is. The zoom box is expanded by pressing "e" and shrunk by pressing "s". The amount of expansion and shrinkage that occurs with each pressing or "e" or "s" is a constant ratio of the current screen window size. The following picture shows the zoom box positioned.

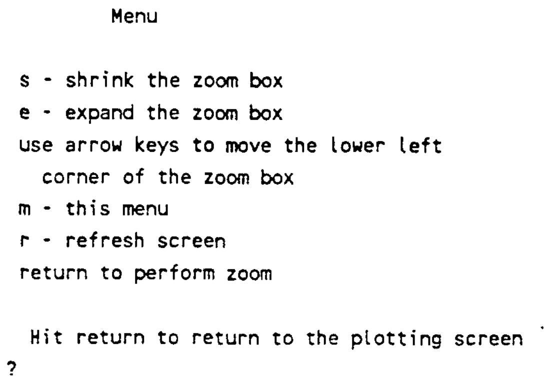


The zoom is performed by pressing the "enter" key.



If the zoom routine is reentered (by pressing "z" again), it will appear on the screen with the box around the same coordinates as previously (the full screen inscribed by a light blue box).

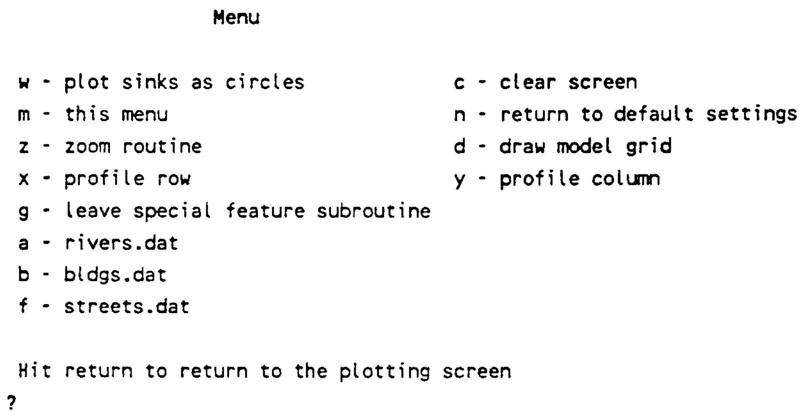
There are several other keys that perform useful functions in the zoom routine. One of these is the "r" key. The "r" key refreshes the screen. As the zoom box is moved around, parts of the screen image are obliterated; pressing "r" replots the screen. Another useful key is the menu. Pressing "m" shows all the valid keys in the zoom mode and what they do.



```
         Menu

  s - shrink the zoom box
  e - expand the zoom box
  use arrow keys to move the lower left
    corner of the zoom box
  m - this menu
  r - refresh screen
  return to perform zoom

   Hit return to return to the plotting screen
 ?
```

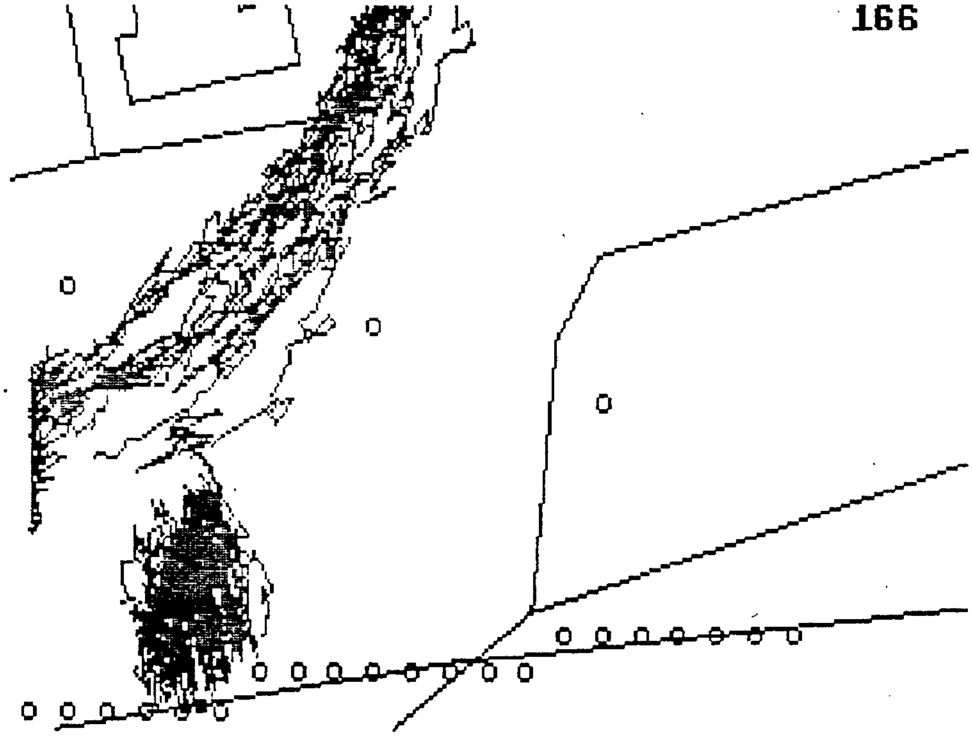
There are several utility commands when viewing the special features graphics screen. Pressing "c" will clear the screen while retaining the current screen coordinates (same area view). Pressing "n" will clear the screen and return to the default screen view (screen coordinates selected as defaults during the execution of the SPF routine).

There is also a menu for the graphics commands. The menu shows the functions of each valid key. The menu is displayed by pressing "m". Return to the main graphics screen from the menu by pressing "enter".



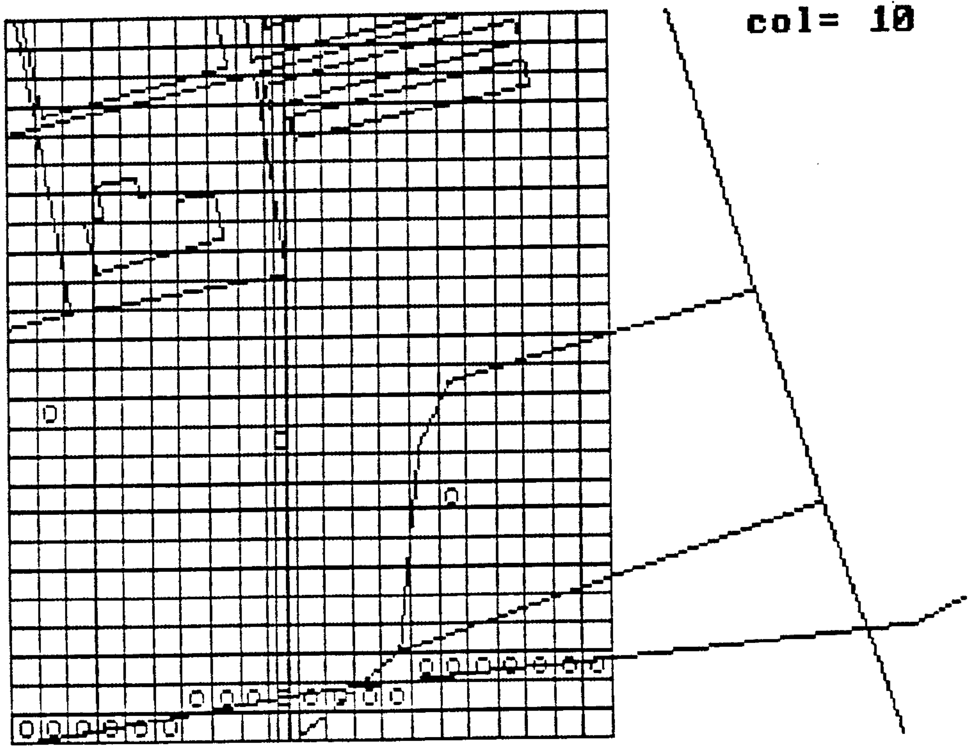
```
                    Menu


 w - plot sinks as circles          c - clear screen
 m - this menu                      n - return to default settings
 z - zoom routine                   d - draw model grid
 x - profile row                    y - profile column
 g - leave special feature subroutine
 a - rivers.dat
 b - bldgs.dat
 f - streets.dat

 Hit return to return to the plotting screen
?
```

At this point in the MOVE routine, the user has selected the proper area of the screen for viewing. By pressing "g" the simulation will begin. Each particle is checked for decay (if the decay fraction is greater than one, then the particle is removed from the simulation), the advective part of the particles movement calculated (see Section III.C.2.), the dispersion part of the move calculated (see Section III.C.3.), and whether or not a particle enters sink during the time step is checked (see Section III.C.6.). The particle is moved to its new location and the program continues to the next particle. The path that the particle takes across the screen is shown as a red line. The final position of the particle at the end of the step is shown as a white dot. All particles are moved during a time step even if they are not visible on the screen. Particles in all layers are visible. The upper right corner of the screen displays the number of the particles moved.

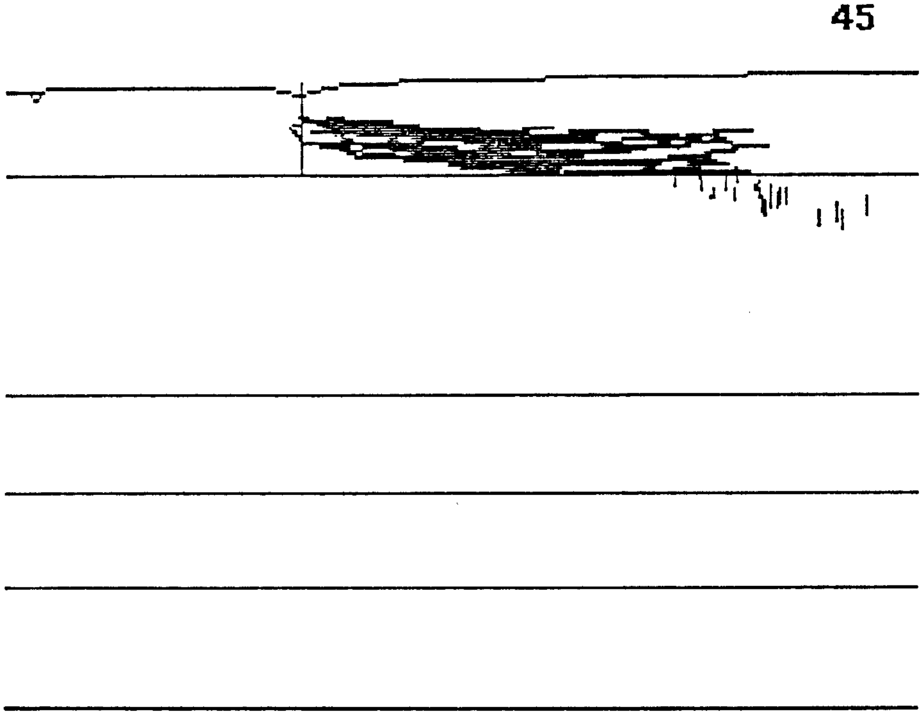


When all the particles have been moved, the screen is static. The user may exit the MOVE routine by pressing "enter" or "s". Pressing "s" will save a screen image as a slide. The screen image is written to a file named SLIDE?.DAT where ? is the number of the slide. Slides are numbered starting with zero for each simulation.

There is another type of special feature that may be displayed during a MOVE, a cross-section view. When the user is selecting what part of the screen is to be viewed, it is possible to select any row or column of the model and plot a cross-section view on the screen. The movement of the particles through the different layers may then be viewed. Pressing "y" starts the column cross-section process. A red rectangle is displayed on the screen over column 1 of the model grid. The left and right arrow keys may be used to move the red rectangle to different columns. The current column selection is shown in the upper right corner of the screen. When the chosen column has been selected, pressing "enter" will show a cross-section view of that column.



The cross-section view shows the selected column aquifer tops and bottoms plotted on the screen. The vertical scale of the screen plot is determined by the vertical exaggeration ratio selected when the SPF routine was previously executed. The horizontal scale of the column plot is determined by the current plan view coordinates. The coordinates of the screen display from bottom to top in the plan view become the left to right coordinates of the column cross-section view. The lower left z coordinate of the screen is LLZ, the value read in from the velocity file (see Section III.G.1.). Wells are shown on the screen from above the water table to the bottom of the layer the well is completed in. Rivers are shown as small triangles plotted at the water table (rivers are only allowed in the top layer). The movement of particles on the cross-section view is initiated by pressing "g". If one wishes to return to the plan view from the cross-section view, press "enter". Pressing enter will return the user to a clear screen with the default view (same action as pressing "n"). There is no help screen for the cross-section view, because the only valid commands are "g" to simulate particle movement and "enter" to return to plan view.



The row cross-section view operates similarly. Pressing "x" shows the row to be cross-sectioned on the screen. The arrow keys may be used to move the red row selection rectangle to a different row. The cross-section screen will have the the left to right coordinates that are identical to the current plan view left to right coordinates (x coordinates).

If a cross-section view simulation was performed, the MOVE routine is ended by pressing the same keys that are used in exiting the plan view, "enter" to exit, or "s" to save a slide and exit.

H. MAP

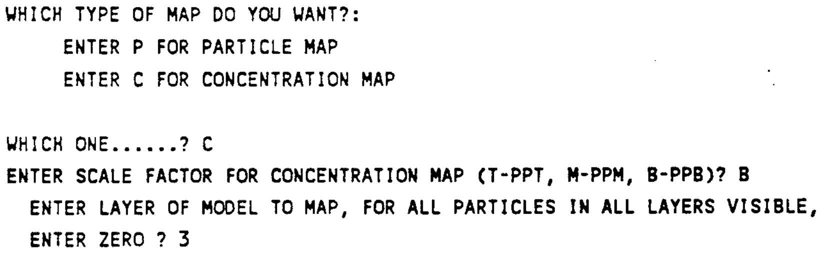
The MAP routine allows the user to plot a map of particle locations or aquifer concentrations on the screen. Any layer of the model may be plotted or all layers simultaneously. Cross-section plots may also be prepared showing all columns or rows or a selected column or row. On entering "map" the user is immediately presented with the special features graphics screen. The area to be mapped is selected using the procedures described above under the MOVE routine. Either a plan view, row cross-section, or column cross-section may be selected.

After selecting the view and area, the user is prompted for whether a particle map or concentration map is desired. The particle map shows particle locations within the user selected area. The concentration map actually calculates concentrations of solute. If a concentration map is selected, the user is prompted for the units to be used in calculated concentration, "t" for parts per thousand (g/l), "m" for parts per million (mg/l), or "b" for parts per billion (microg/l). The user is also prompted for the depth of the map. Either a single layer, row or column may be mapped or all layers, rows, or columns may be mapped. For a particle map, it is sometimes useful to see particle locations through the model (all). Concentration maps have little meaning unless a single layer, row or column is selected (s). Aquifers are numbered from the bottom of the aquifer starting with one. Confining layers are numbered from lowest to highest starting with the water table layer number plus one. If a column or row cross-section has been selected the user will be prompted if particles in all rows/columns are to be visible, or only ones in the selected row or column.

(for a particle map)

```
WHICH TYPE OF MAP DO YOU WANT?:
     ENTER P FOR PARTICLE MAP
     ENTER C FOR CONCENTRATION MAP

WHICH ONE......? P
   ENTER LAYER OF MODEL TO MAP, FOR ALL PARTICLES IN ALL LAYERS VISIBLE,
   ENTER ZERO ? 3
```

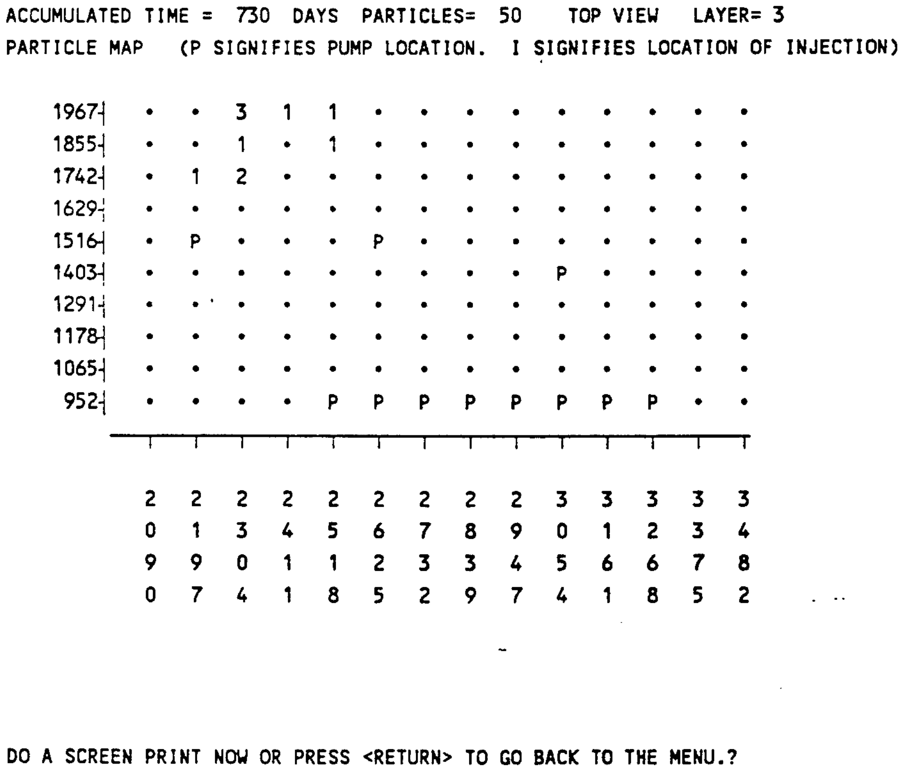
(for a concentration map)



```
WHICH TYPE OF MAP DO YOU WANT?:
     ENTER P FOR PARTICLE MAP
     ENTER C FOR CONCENTRATION MAP

WHICH ONE......? C
ENTER SCALE FACTOR FOR CONCENTRATION MAP (T-PPT, M-PPM, B-PPB)? B
  ENTER LAYER OF MODEL TO MAP, FOR ALL PARTICLES IN ALL LAYERS VISIBLE,
  ENTER ZERO ? 3
```

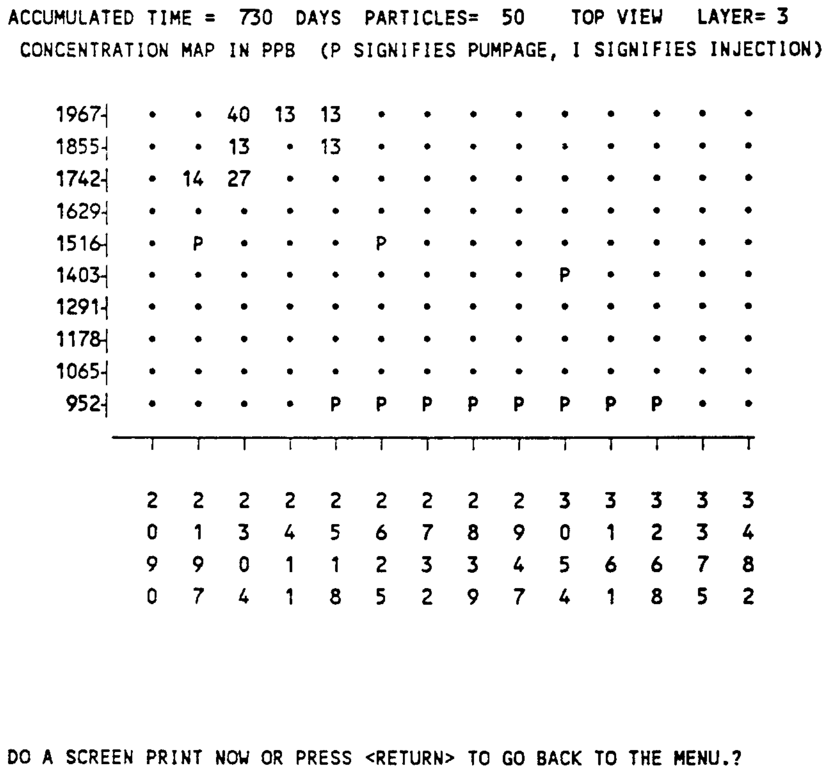
Following this command, the map is displayed. The map is a 14 column by 10 row grid that corresponds to the last viewed screen area. The program computes the number of particles that fall within each grid of the map. If a particle map was chosen, these are displayed. If a concentration map was chosen, average concentrations in each map grid are calculated and displayed.

(for a particle map)



```
ACCUMULATED TIME =  730  DAYS  PARTICLES=  50    TOP VIEW   LAYER= 3
PARTICLE MAP   (P SIGNIFIES PUMP LOCATION.  I SIGNIFIES LOCATION OF INJECTION)

    1967-|  .   .   3   1   1   .   .   .   .   .   .   .   .   .
    1855-|  .   .   1   .   1   .   .   .   .   .   .   .   .   .
    1742-|  .   1   2   .   .   .   .   .   .   .   .   .   .   .
    1629-|  .   .   .   .   .   .   .   .   .   .   .   .   .   .
    1516-|  .   P   .   .   .   P   .   .   .   .   .   .   .   .
    1403-|  .   .   .   .   .   .   .   .   .   P   .   .   .   .
    1291-|  .   .   .   .   .   .   .   .   .   .   .   .   .   .
    1178-|  .   .   .   .   .   .   .   .   .   .   .   .   .   .
    1065-|  .   .   .   .   .   .   .   .   .   .   .   .   .   .
     952-|  .   .   .   .   P   P   P   P   P   P   P   P   .   .
         ---------------------------------------------------------
            |   |   |   |   |   |   |   |   |   |   |   |   |   |

            2   2   2   2   2   2   2   2   2   3   3   3   3   3
            0   1   3   4   5   6   7   8   9   0   1   2   3   4
            9   9   0   1   1   2   3   3   4   5   6   6   7   8
            0   7   4   1   8   5   2   9   7   4   1   8   5   2




DO A SCREEN PRINT NOW OR PRESS <RETURN> TO GO BACK TO THE MENU.?
```

(for a concentration map)



```
ACCUMULATED TIME =  730  DAYS  PARTICLES=  50    TOP VIEW   LAYER= 3
 CONCENTRATION MAP IN PPB  (P SIGNIFIES PUMPAGE, I SIGNIFIES INJECTION)

 1967-|   .   .  40  13  13   .   .   .   .   .   .   .   .   .
 1855-|   .   .  13   .  13   .   .   .   .   .   .   .   .   .
 1742-|   .  14  27   .   .   .   .   .   .   .   .   .   .   .
 1629-|   .   .   .   .   .   .   .   .   .   .   .   .   .   .
 1516-|   .   P   .   .   .   P   .   .   .   .   .   .   .   .
 1403-|   .   .   .   .   .   .   .   .   .   P   .   .   .   .
 1291-|   .   .   .   .   .   .   .   .   .   .   .   .   .   .
 1178-|   .   .   .   .   .   .   .   .   .   .   .   .   .   .
 1065-|   .   .   .   .   .   .   .   .   .   .   .   .   .   .
  952-|   .   .   .   .   P   P   P   P   P   P   P   P   .   .
        ------------------------------------------------------
          T   T   T   T   T   T   T   T   T   T   T   T   T   T

          2   2   2   2   2   2   2   2   2   3   3   3   3   3
          0   1   3   4   5   6   7   8   9   0   1   2   3   4
          9   9   0   1   1   2   3   3   4   5   6   6   7   8
          0   7   4   1   8   5   2   9   7   4   1   8   5   2



DO A SCREEN PRINT NOW OR PRESS <RETURN> TO GO BACK TO THE MENU.?
```

The coordinates shown on the axes of the above maps are the coordinates of the centers of the map grids.

I. PLT

The MAP routine only generates screen displays. To generate a contour plot of concentrations, the PLT routine may be used to create a file of x, y, and concentration for a user selected aquifer layer. Next, the user is prompted for the lower left corner of the area to be plotted (ft), the grid spacing (ft), the number of rows and columns, and the layer of the model to create a file for. The user is then prompted for three characters to be used to create a file name from. The program calculates the concentration in each grid and writes the x coordinate of the center of the grid, the y coordinate of the center of the grid, and the average concentration to a file on the default drive. The file name will be the three letters entered by the user plus the layer number plus the extension ".DAT". If the user selects all layers (entering "0" (zero)), then a separate file for each aquifer layer will be created on the default drive.

```
ENTER LOWER LEFT CORNER COORDINATES OF PLOT MAP (X,Y)? 2100,900
ENTER DESIRED GRID SIZE OF MAP, IN FEET ? 50
ENTER NUMBER OF ROWS & COLUMNS OF PLOT MAP (ROWS, COLUMNS) ? 30,40
ENTER LAYER OF MODEL TO GENERATE PLOT FILE FOR, FOR ALL LAYERS ENTER ZERO ? 3


ENTER A 3-LETTER CODE FOR THE PLOT FILE
File will be stored on default disk (Example abc01.dat)? ABC
```

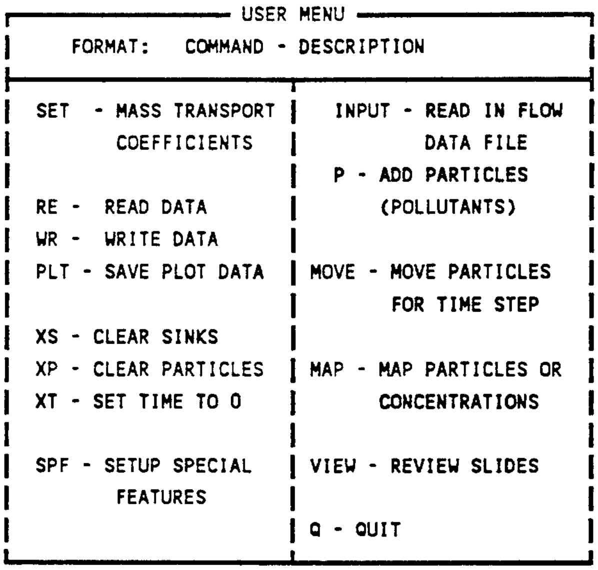
J. VIEW

Previously saved slides from the MOVE routine may be viewed in forward or reverse sequence.

```
COMMAND ? VIEW


                         Which mode do you want to operate in?
                         FORWARD in time?  Or BACKWARD in time?

            Enter the mode that you want.  F - FORWARD or B - BACKWARD? F
```

K. Miscellaneous

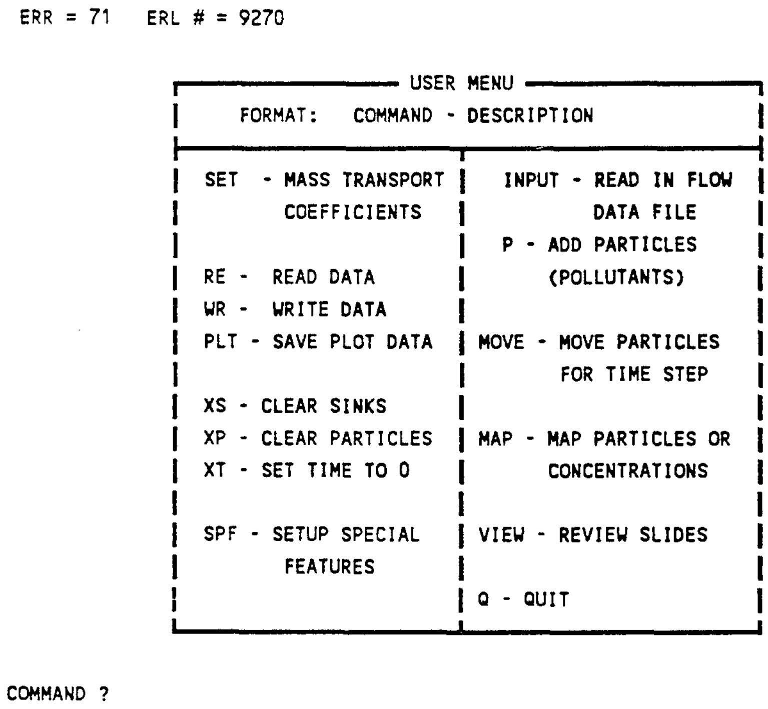
There are three utility routines that are of use in operating the RAND3D model. The XS routine enables the user to erase all sinks from the model. The XT routine enables the user to reinitialize time at zero. The XP routine enables the user to remove all particles. The XT and XP routines are frequently used to restart a simulation without exiting from the model.



```
┌──────────────────── USER MENU ─────────────────────┐
│   FORMAT:   COMMAND - DESCRIPTION                  │
├─────────────────────────┬──────────────────────────┤
│ SET  - MASS TRANSPORT   │  INPUT - READ IN FLOW    │
│        COEFFICIENTS     │          DATA FILE       │
│                         │  P - ADD PARTICLES       │
│ RE -  READ DATA         │      (POLLUTANTS)        │
│ WR -  WRITE DATA        │                          │
│ PLT - SAVE PLOT DATA    │ MOVE - MOVE PARTICLES    │
│                         │        FOR TIME STEP     │
│ XS - CLEAR SINKS        │                          │
│ XP - CLEAR PARTICLES    │ MAP - MAP PARTICLES OR   │
│ XT - SET TIME TO 0      │       CONCENTRATIONS     │
│                         │                          │
│ SPF - SETUP SPECIAL     │ VIEW - REVIEW SLIDES     │
│       FEATURES          │                          │
│                         │ Q - QUIT                 │
└─────────────────────────┴──────────────────────────┘
```

```
COMMAND ? XS
     NUMBER OF SINKS INITIATED



COMMAND ? XT
     SIMULATION TIME INITIATED



COMMAND ? XP
     NUMBER OF PARTICLES INITIATED
```

The RAND3D program contains an error recovery routine. For any error that may be trapped by QuickBasic, the program prints the error number the line number on which the error occurred, and returns the user to the menu



```
ERR = 71   ERL # = 9270

┌──────────────────── USER MENU ───────────────────┐
│   FORMAT:   COMMAND - DESCRIPTION                │
├────────────────────────┬─────────────────────────┤
│ SET  - MASS TRANSPORT  │  INPUT - READ IN FLOW   │
│        COEFFICIENTS    │          DATA FILE      │
│                        │  P - ADD PARTICLES      │
│ RE -  READ DATA        │     (POLLUTANTS)        │
│ WR -  WRITE DATA       │                         │
│ PLT - SAVE PLOT DATA   │ MOVE - MOVE PARTICLES   │
│                        │        FOR TIME STEP    │
│ XS - CLEAR SINKS       │                         │
│ XP - CLEAR PARTICLES   │ MAP - MAP PARTICLES OR  │
│ XT - SET TIME TO 0     │       CONCENTRATIONS    │
│                        │                         │
│ SPF - SETUP SPECIAL    │ VIEW - REVIEW SLIDES    │
│       FEATURES         │                         │
│                        │ Q - QUIT                │
└────────────────────────┴─────────────────────────┘

COMMAND ?
```

The error shown above was caused by the program being unable to access the selected drive (error 71 is disk not ready) at line number 9270 which is part of the velocity file input routine.

One error that may occur is error 6, which is an overflow error. The usual cause of this error is dividing by zero or generating a number larger than QuickBasic can handle (integer larger than 32767 or real number larger than 1.7E38). One other cause of this error, that has been noticed in the running of the program, is the repeated use of the FILES statement, which is in the velocity file input routine (INPUT). Evidently, repeatedly executing the FILES statement with many velocity files (.RND) to choose from causes some sort of string space corruption that causes overflow errors. The program traps this error (#6) and resumes execution. Another answer to this problem is to remove the FILES statement from the program and recompile.

To quit the RAND3D program, enter "q". The program ends and returns the user to DOS.

V. References

Prickett,T.A., Naymik,T.G., Lonnquist,C.G., A Random-Walk Solute Transport Model for Selected Groundwater Quality Evaluations, Illinois Water Survey Bulletin 65, 1981.

Molz,F.J., Guven,O., and Melville,J.,"An Examination of Scale-Dependant Dispersion Coefficients", **Ground Water**, Vol. 21, No. 8, p. 715-725.

Bear, Jacob. 1972. Dynamics of fluids in porous media, American Elsevier Publishing Company, New York, New York, 764 pages.

Freeze, R.A., and J.A. Cherry. 1979. **Groundwater**. Prentice-Hall, Inc., 604 p.

Hunt, B.,"Dispersive Sources in Uniform Ground-Water Flow", American Society of Civil Engineers, Journal of the Hydraulics Division, Vol. 104, No. HY1, January 1978, pp. 75-85.

VI. Variables

The following list defines the important variables used in the RAND3D program.

ASPECT - ASPECT RATIO OF SCREEN

BOT(I,J,K) - ELEVATION OF BOTTOM OF NODE I,J,K (FT)

BOT(I,J,K) - ELEVATION OF BOTTOM OF NODE I,J,K (FT)

BOTCAPTURE - BOTTOM ELEVATION AT WHICH PARTICLE MAY BE CAPTURED BY SINK

CAPTUREMAX - RIVER MAXIMUM CAPTURE DISTANCE

CDX - COLUMN SPACING OF MAP ARRAY

CDY - ROW SPACING OF MAP ARRAY

COLUMNS - NUMBER OF COLUMNS IN PLOT ARRAY

CONC(L) - SINK CONCENTRATION FOR SINK L(MG/L)

D - TIME STEP FOR MOVE, =DELTA IF SLUG SOURCE PARTICLE
= RANDOM FRACTION OF DELTA FOR CONTINUOUS SOURCE PARTICLE (DAYS)

DD - HORIZONTAL DISTANCE TO MOVE (FT)

DDX - DISTANCE OF MOVE (FT)

DELTA - TIME STEP OF MOVE SUBROUTINE (DAYS)

DELX - COLUMN SPACING (FT)

DELY - ROW SPACING (FT)

DL - LONGITUDINAL DISPERSIVITY (FT)

DMAX - MAXIMUM HORIZONTAL MOVE DISTANCE BEFORE RECOMPUTING VELOCITY (DAYS)

DT - TRANSVERSE DISPERSIVITY (FT)

DV - VERTICAL DISPERSIVITY (FT)

E - POROSITY

ENDTIME(M) - ENDTIMES FOR EACH MODEL VELOCITY FILE (DAYS)

F$ - MAP FLAG, IF="A" THEN MAP HAS BEEN RUN

G$ - FLAG INDICATING COEFFICIENTS HAVE BEEN ENTERED IF ="B"

I1! - X COORDINATE OF LOWER LEFT CORNER OF SCREEN WINDOW (FT)

I2! - X COORDINATE OF UPPER LEFT CORNER OF SCREEN WINDOW (FT)

ICOLXS - COLUMN DISPLAYED IN COLUMN CROSS-SECTION VIEW

IPEN - COLOR OF SPECIAL FEATURES ON SCREEN

J1! - Y COORDINATE OF LOWER LEFT CORNER OF SCREEN WINDOW (FT)

J2! - Y COORDINATE OF UPPER LEFT CORNER OF SCREEN WINDOW (FT)

JROWXS - ROW DISPLAYED IN ROW CROSS-SECTION VIEW

LAMDBA - HALF LIFE FOR FIRST ORDER DECAY (YEARS)

LL - LAYER OF PARTICLE

LLX - X COORDINATE OF LOWER LEFT CORNER OF MODEL GRID (FT)

LLY - Y COORDINATE OF LOWER LEFT CORNER OF MODEL GRID (FT)

LLZ - Z COORDINATE OF LOWER LEFT CORNER OF MODEL GRID (FT)

LN - LINES OF MAP

LOWERLX - X COORDINATE OF LOWER LEFT CORNER OF PLOT AREA (FT)

LOWERLY - Y COORDINATE OF LOWER LEFT CORNER OF PLOT AREA (FT)

MAPTYPE$ - FLAG INDICATING WHAT VIEW OF THE PROBLEM WAS LAST DISPLAYED ON THE SCREEN , A - TOP VIEW, B - ROW CROSS-SECTION VIEW, C - COLUMN CROSS-SECTION VIEW

NC - NUMBER OF COLUMNS

NENDTIME - NUMBER OF TIME PERIODS

NEWX - X COORDINATE OF PARTICLE AFTER MOVE (FT)

NEWY - Y COORDINATE OF PARTICLE AFTER MOVE (FT)

NL - NUMBER OF LAYERS

NMAP(I,J) - MAP AND PLOT STORAGE FOR # OF PARTICLES OR CONCENTRATIONS

NP - NUMBER OF PARTICLES

NR - NUMBER OF ROWS

NS - NUMBER OF SINKS

NSPF - NUMBER OF SPECIAL FEATURE FILES (WITH KEYS AND FILE NAMES)

O - SLIDE COUNTER

PM - PARTICLE MASS (LBS)

PPP - CONCENTRATION FACTOR, =0.001 FOR PPB, =1 FOR PPM =1000 FOR PPT

QSUM(L) - DISCHARGE OF SINK L (GPD)

R$ - NAME OF VELOCITY FILE (WITH EXTENSION .RND)

R3 - DISTANCE FROM SINK TO PARTICLE (FT)

R4 - VERTICAL DISTANCE FROM PARTICLE TO BOTTOM OF RIVER

RETARD(K) - RETARDATION FACTORS FOR EACH LAYER AND CONFINING LAYER

RL - LONGITUDINAL DISPERSION (FT)

ROWS - NUMBER OF ROWS IN PLOT ARRAY

RT - TRANSVERSE DISPERSION (FT)

RZ - VERTICAL DISPERSION (FT)

SCALEDEF - DEFAULT X DISTANCE FOR SCREEN WINDOW

SCALEX - X DIMENSION OF SCREEN WINDOW (I2!-I1!) (FT)

SCREENX - X COORDINATE TO PLOT ON SCREEN (FT)

SCREENY - Y COORDINATE TO PLOT ON SCREEN (FT)

SIZE - GRID SPACING FOR PLOT AREA (FT)

SPFFILE$(I) - FILE NAMES OF SPECIAL FEATURE FILES

SPFKEY$(I,J) - KEYS USED TO CALL SPECIAL FEATURE FILE PLOTS, I IS SPECIAL FEATURE COUNTER, J IS 1 OR 2 CONTAINING BOTH UPPER AND LOWER CASE LETTERS

SPFSAVE$ - NAME OF FILE WITH SPECIAL FEATURE DATA

SVIEWX - X DIMENSION OF ZOOM BOX

SWITCH%(I) - ARRAY OF FLAGS INDICATING WHETHER IT IS A SLUG OR CONTINUOUS PARTICLE, 1= CONTINUOUS, 0=SLUG

T2 - TIME (DAYS)

TEMPSCR - FLAG, IF=1 IT INDICATES A SCREEN IMAGE HAS BEEN SAVED AS "TEMP.SCR"

THICK(I,J,K) - THICKNESS OF AQUIFER OF COLUMN I, ROW J, AND LAYER K

TOP(I,J,K) - ELEVATION OF TOP OF NODE I,J,K, IF THE TOP WATER TABLE LAYER THEN =0 UNLESS A RIVER NODE, THEN = BOTTOM OF RIVER ELEVATION (FT)

TOPCAPTURE - TOP ELEVATION AT WHICH PARTICLE MAY BE BE CAPTURED BY SINK

VI(I,J,K) - VELOCITY IN X DIRECTION FROM I,J,K TO I+1,J,K (FT/DAY)

VJ(I,J,K) - VELOCITY IN Y DIRECTION FROM I,J,K TO I,J+1,K (FT/DAY)

VK(I,J,K) - VELOCITY IN Z DIRECTION FROM I,J,K TO I,J,K+1 (FT/DAY)

VX - INTERPOLATED X VELOCITY (FT/DAY)

VY - INTERPOLATED Y VELOCITY (FT/DAY)

VZ - INTERPOLATED Z VELOCITY (FT/DAY)

WD - COLUMNS OF MAP

X(I) - ARRAY OF X COORDINATES FOR PARTICLES (FT)

X1(L) - X COORDINATE OF SINK L (FT)

XP - X COORDINATE OF PARTICLE POSITION IN GRID UNITS

XX$ - FLAG INDICATING A COLOR MONITOR IS PRESENT, ALWAYS="Y" IN THIS VERSION

Y(I) - ARRAY OF Y COORDINATES FOR PARTICLES (FT)

Y1(L) - Y COORDINATE OF SINK L (FT)

YASPECT - ASPECT RATIO OF SCREEN FOR TOP VIEW

YP - Y COORDINATE OF PARTICLE POSITION IN GRID UNITS

Z(I) - ARRAY OF Z COORDINATES FOR PARTICLES (FT)

Z1(L) - LAYER OF SINK L

Z2$ - STRING OF KEYS PRESSED TO PUT SPECIAL FEATURES ON SCREEN

Z3$ - STRING OF KEYS PRESSED TO PUT SPECIAL FEATURES ON SCREEN

ZASPECT - ASPECT RATIO OF SCREEN FOR CROSS-SECTION VIEW

ZMAX - MAXIMUM VERTICAL MOVE DISTANCE BEFORE RECOMPUTING VELOCITY (DAYS)

ZZZ - VARIABLE USED TO IMPLEMENT FIRST ORDER DECAY

VII. Examples

Example #1

The RAND3D model was tested by comparing the results of the model to a simple three dimensional analytical solution.

There are relatively few three dimensional analytical solutions for solute transport. One of the simplest is the problem of a single instantaneous, point source of solute in a uniform flow field, with dispersion in all three dimensions. The analytical equation describing this problem is

$$C = [M/(8n(\pi^3 t^3 D_x D_y D_z)^{.5}]\exp(-(x-vt)^2/4D_x t - y^2/4D_y t - z^2/4D_z t)$$

(Hunt, 1978)

where

C = concentration
M = initial mass entering aquifer as a slug
x,y,z = cartesian coordinates, slug source is at 0,0,0 and direction of flow is along x axis
v = seepage velocity in x direction
D_x = dispersion in x direction = $\text{alpha}_x v$
D_y = dispersion in y direction = $\text{alpha}_y v$
D_z = dispersion in z direction = $\text{alpha}_z v$
n = porosity
t = time since addition of slug
exp = exponential function (inverse natural log)

This equation was used to calculate the plume resulting from the instantaneous addition of 50 lbs of solute to an aquifer with a uniform seepage velocity of one ft/day. The theoretical aquifer porosity was assumed to be 0.1. Dispersivities were assumed to be 10 feet in the longitudinal direction, 3 feet in the transverse direction, and 1 foot in the vertical. With the seepage velocity of one foot/day, the resulting dispersion coefficients were 10, 3 and 1 ft^2/day. Concentration plumes were calculated at a time of ten days for a ten foot thick horizontal section (layer 2) centered on z=0 and a vertical cross-section centered on y=0.

In order to run the RAND3D model for the same situation, it was first necessary to create a velocity file for input. The velocity file was created with a simple Basic program. The program created the velocity file with a uniform Darcy velocity in the x direction (column direction) of 0.1 ft/day. A 14 column by 11 row by 3 layer grid was designed. Column and row widths were ten feet. Layer two was ten feet thick, layers one and three were twenty feet thick.

```
10 REM
```

```
20 REM  PROGRAM TO CREATE UNIFORM VELOCITY FILE FOR
30 REM    INPUT TO RAND3D
40 REM
50 DEFINT I,J,K,N
60 REM
70 LINE INPUT "  ENTER FILE NAME FOR OUTPUT VELOCITY
     FILE - NO EXTENSION ";A$
80 A$=A$+".RND"
90 OPEN "O",1,A$
100 PRINT "  FILE ";A$;" OPENED FOR OUTPUT"
110 INPUT "  ENTER # OF COLUMNS ";NC
120 INPUT "  ENTER # OF ROWS ";NR
130 INPUT "  ENTER # OF LAYERS ";NL
140 INPUT "  ENTER GRID SPACING FOR COLUMNS ";DELX
150 INPUT "  ENTER GRID SPACING FOR ROWS ";DELY
160 INPUT "  ENTER X,Y,Z COORDINATES OF LOWER LEFT
     CORNER OF MODEL ";LLX,LLY,LLZ
170 PRINT#1,NC;NR;NL;DELX;DELY;LLX;LLY;LLZ
180 FOR K=1 TO NL
190   PRINT "  FOR LAYER #";K
200   INPUT "  ENTER THICKNESS (FT) ";THICK
210   INPUT "  ENTER VX (FT/DAY) ";VX
220   INPUT "  ENTER VY (FT/DAY) ";VY
230   INPUT "  ENTER VZ (FT/DAY) ";VZ
240   INPUT "  ENTER TOP ELEVATION (FT)";TOP
250   INPUT "  ENTER BOTTOM ELEVATION (FT)";BOT
260   FOR J=1 TO NR
270     FOR I=1 TO NC
280        PRINT#1,I;J;K;THICK;VX;VY;VZ;BOT;TOP
290     NEXT I
300   NEXT J
310 NEXT K
320 CLOSE 1
330 END
```

The velocity file which was created is printed in attachment 1 for this example.

There was no retardation or decay. Five thousand particles were placed in the middle of row 6, column 5, layer 2 (model coordinates 50,55,25) as a slug source. Each particle weighed 0.01 lbs. The model was run for 10 days and maps and plots were prepared. Attachment 2 to this example shows the printout produced for this example.

Figure 11 shows the concentrations in layer 2 after ten days from the RAND3D model. Figure 12 shows the corresponding analytical solution. They match extremely well. There are some differences due to the stochastic nature of the RAND3D algorithm. Some distortion and averaging resulted from the gridding algorithm used to prepare the contour plots. The peak concentration in the analytical solution is 797 mg/l at x=60, y=55. The peak concentrations in RAND3D are 722 mg/l at x=55, y=55 and 770 mg/l at x=65 and y=55.

Figure 13 shows the concentrations in row 6 of the model from the RAND3D model. Figure 14 shows the corresponding analytical solution. The extent of the plume matches well. The gridding algorithm smoothed the peak in the middle of the model plume. The peak concentration in the middle of the plume in the analytical solution is 941 mg/l at x=60, z=25. The peak concentration in the middle of the plume in the RAND3D output is 936 mg/l at x=60, z=25. The differences are due to the random-walk nature of the algorithm.

Figure 11

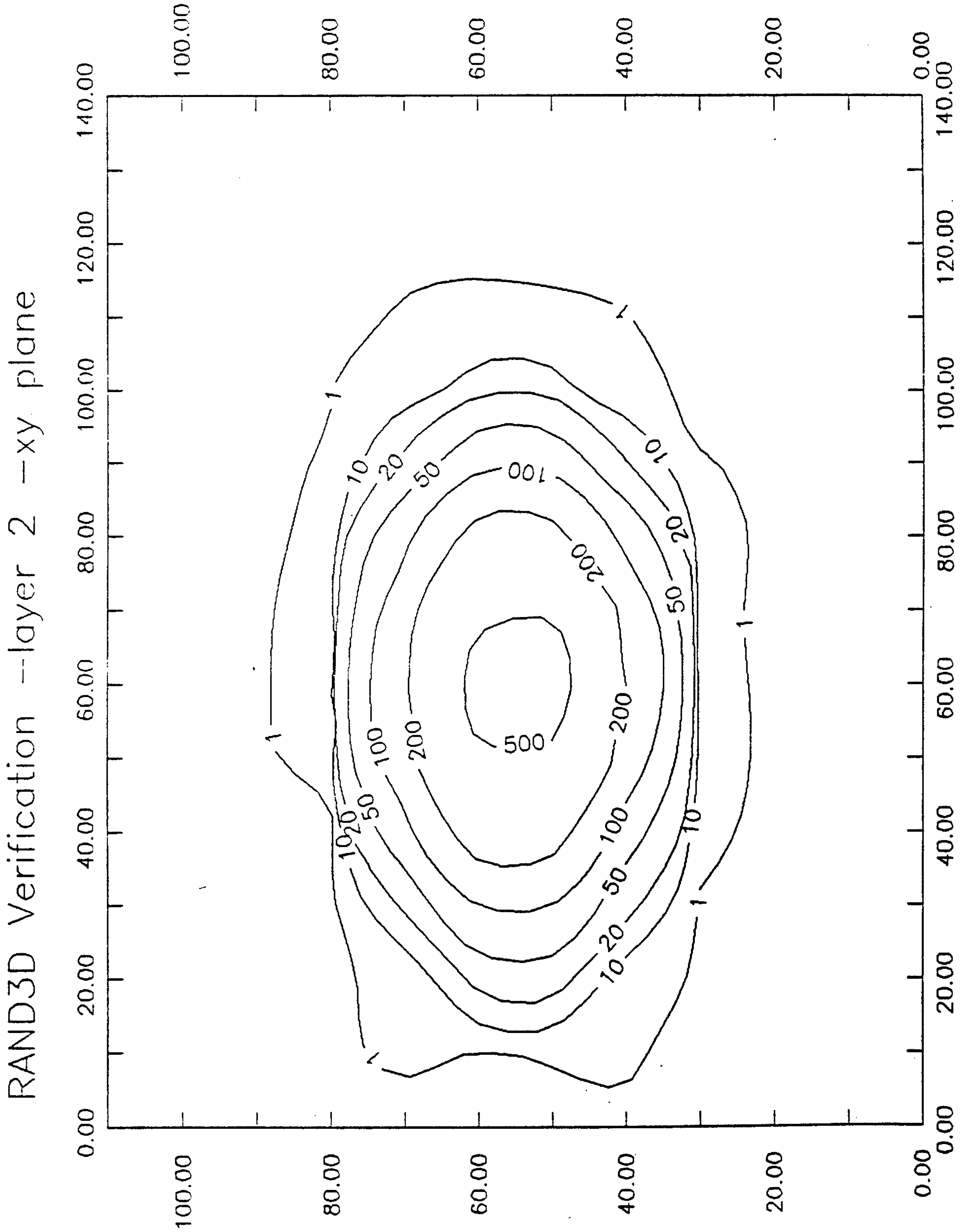

RAND3D Verification --layer 2 --xy plane
0.00
20.00
40.00
60.00
80.00
100.00
120.00
140.00
100.00
80.00
60.00
40.00
20.00
0.00
1
10
20
50
100
200
500

Figure 12

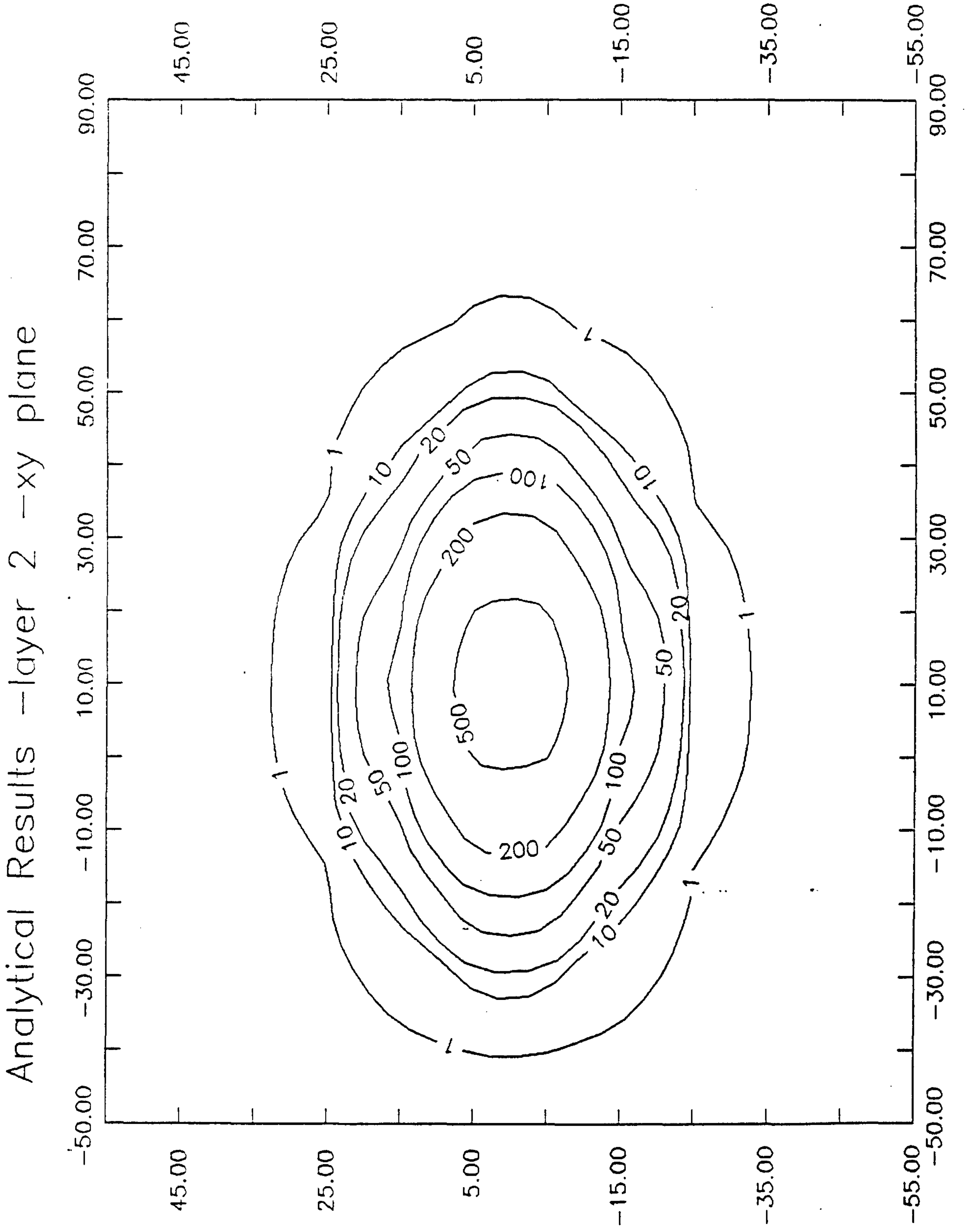

Analytical Results -layer 2 -xy plane
-50.00
-30.00
-10.00
10.00
30.00
50.00
70.00
90.00
45.00
25.00
5.00
-15.00
-35.00
-55.00
1
10
20
50
100
200
500

Figure 13

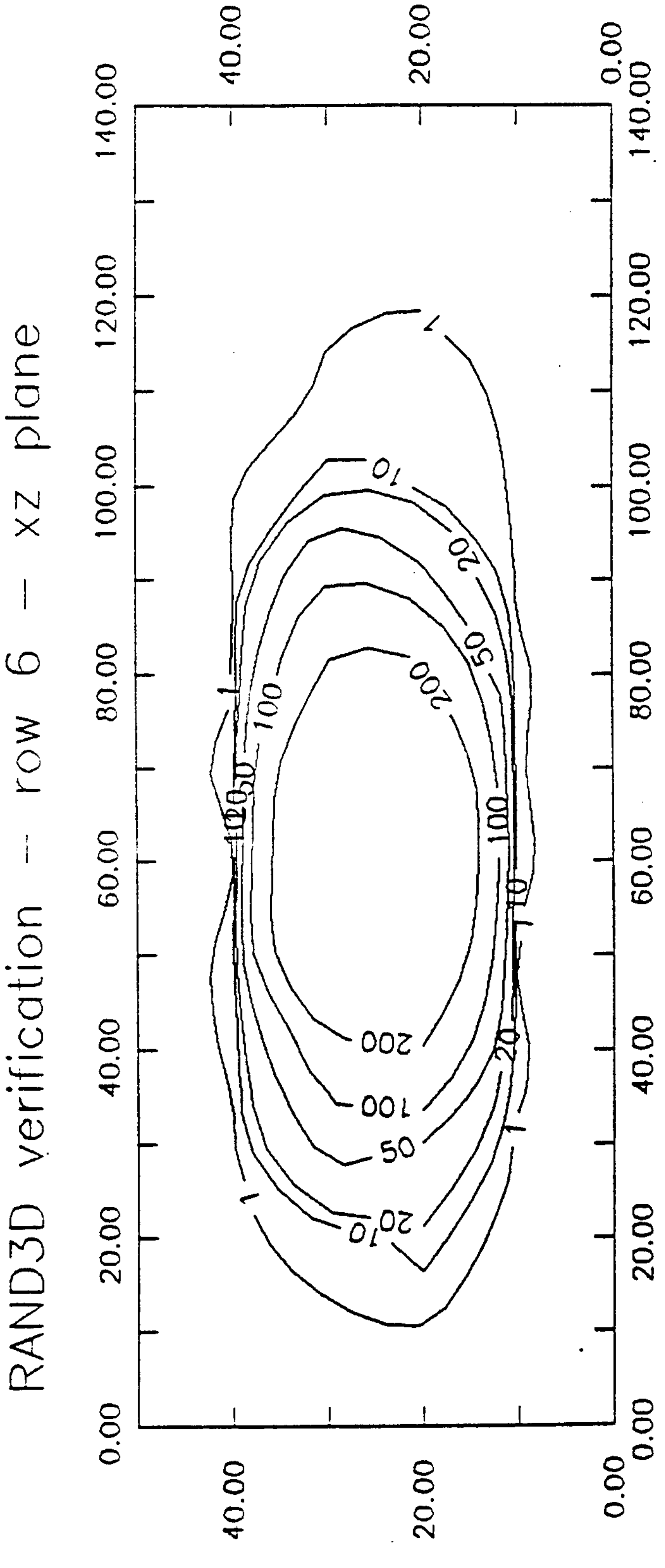
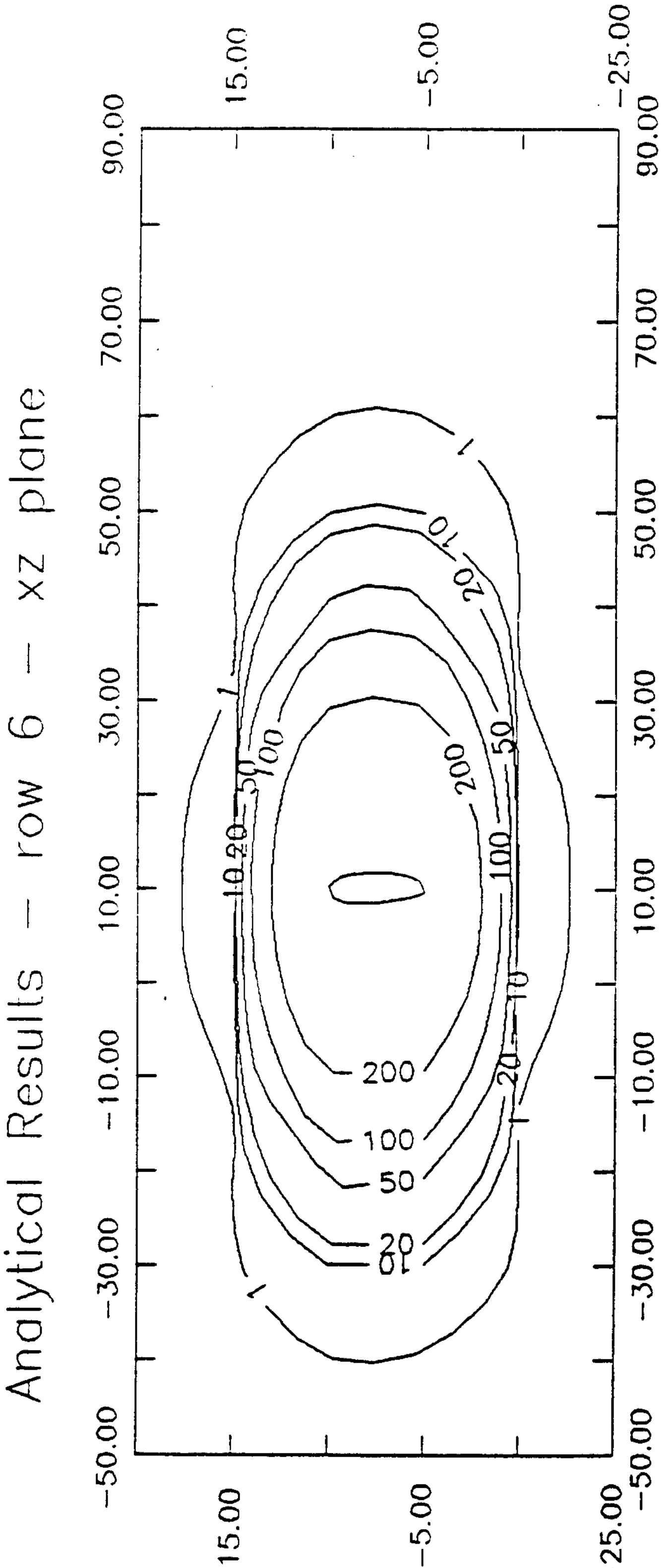

RAND3D verification - row 6 - xz plane
0.00
20.00
40.00
60.00
80.00
100.00
120.00
140.00
0.00
20.00
40.00
1
10
20
50
100
200

Figure 14


Analytical Results – row 6 – xz plane
-50.00
-30.00
-10.00
10.00
30.00
50.00
70.00
90.00
15.00
-5.00
-25.00
1
10
20
50
100
200

Example 1 - Attachment 1
Velocity File

```
14  11  3  10  10  0  0  0
1  1  1  20  .1  0  0  0  20
2  1  1  20  .1  0  0  0  20
3  1  1  20  .1  0  0  0  20
4  1  1  20  .1  0  0  0  20
5  1  1  20  .1  0  0  0  20
6  1  1  20  .1  0  0  0  20
7  1  1  20  .1  0  0  0  20
8  1  1  20  .1  0  0  0  20
9  1  1  20  .1  0  0  0  20
10  1  1  20  .1  0  0  0  20
11  1  1  20  .1  0  0  0  20
12  1  1  20  .1  0  0  0  20
13  1  1  20  .1  0  0  0  20
14  1  1  20  .1  0  0  0  20
1  2  1  20  .1  0  0  0  20
2  2  1  20  .1  0  0  0  20
3  2  1  20  .1  0  0  0  20
4  2  1  20  .1  0  0  0  20
5  2  1  20  .1  0  0  0  20
6  2  1  20  .1  0  0  0  20
7  2  1  20  .1  0  0  0  20
8  2  1  20  .1  0  0  0  20
9  2  1  20  .1  0  0  0  20
10  2  1  20  .1  0  0  0  20
11  2  1  20  .1  0  0  0  20
12  2  1  20  .1  0  0  0  20
13  2  1  20  .1  0  0  0  20
14  2  1  20  .1  0  0  0  20
1  3  1  20  .1  0  0  0  20
2  3  1  20  .1  0  0  0  20
3  3  1  20  .1  0  0  0  20
4  3  1  20  .1  0  0  0  20
5  3  1  20  .1  0  0  0  20
6  3  1  20  .1  0  0  0  20
7  3  1  20  .1  0  0  0  20
8  3  1  20  .1  0  0  0  20
9  3  1  20  .1  0  0  0  20
10  3  1  20  .1  0  0  0  20
11  3  1  20  .1  0  0  0  20
12  3  1  20  .1  0  0  0  20
13  3  1  20  .1  0  0  0  20
14  3  1  20  .1  0  0  0  20
1  4  1  20  .1  0  0  0  20
2  4  1  20  .1  0  0  0  20
3  4  1  20  .1  0  0  0  20
4  4  1  20  .1  0  0  0  20
5  4  1  20  .1  0  0  0  20
6  4  1  20  .1  0  0  0  20
7  4  1  20  .1  0  0  0  20
8  4  1  20  .1  0  0  0  20
```

```
9   4   1   20   .1   0   0   0   20
10   4   1   20   .1   0   0   0   20
11   4   1   20   .1   0   0   0   20
12   4   1   20   .1   0   0   0   20
13   4   1   20   .1   0   0   0   20
14   4   1   20   .1   0   0   0   20
1   5   1   20   .1   0   0   0   20
2   5   1   20   .1   0   0   0   20
3   5   1   20   .1   0   0   0   20
4   5   1   20   .1   0   0   0   20
5   5   1   20   .1   0   0   0   20
6   5   1   20   .1   0   0   0   20
7   5   1   20   .1   0   0   0   20
8   5   1   20   .1   0   0   0   20
9   5   1   20   .1   0   0   0   20
10   5   1   20   .1   0   0   0   20
11   5   1   20   .1   0   0   0   20
12   5   1   20   .1   0   0   0   20
13   5   1   20   .1   0   0   0   20
14   5   1   20   .1   0   0   0   20
1   6   1   20   .1   0   0   0   20
2   6   1   20   .1   0   0   0   20
3   6   1   20   .1   0   0   0   20
4   6   1   20   .1   0   0   0   20
5   6   1   20   .1   0   0   0   20
6   6   1   20   .1   0   0   0   20
7   6   1   20   .1   0   0   0   20
8   6   1   20   .1   0   0   0   20
9   6   1   20   .1   0   0   0   20
10   6   1   20   .1   0   0   0   20
11   6   1   20   .1   0   0   0   20
12   6   1   20   .1   0   0   0   20
13   6   1   20   .1   0   0   0   20
14   6   1   20   .1   0   0   0   20
1   7   1   20   .1   0   0   0   20
2   7   1   20   .1   0   0   0   20
3   7   1   20   .1   0   0   0   20
4   7   1   20   .1   0   0   0   20
5   7   1   20   .1   0   0   0   20
6   7   1   20   .1   0   0   0   20
7   7   1   20   .1   0   0   0   20
8   7   1   20   .1   0   0   0   20
9   7   1   20   .1   0   0   0   20
10   7   1   20   .1   0   0   0   20
11   7   1   20   .1   0   0   0   20
12   7   1   20   .1   0   0   0   20
13   7   1   20   .1   0   0   0   20
14   7   1   20   .1   0   0   0   20
1   8   1   20   .1   0   0   0   20
2   8   1   20   .1   0   0   0   20
3   8   1   20   .1   0   0   0   20
4   8   1   20   .1   0   0   0   20
5   8   1   20   .1   0   0   0   20
6   8   1   20   .1   0   0   0   20
```

```
7  8  1  20  .1  0  0  0  20
8  8  1  20  .1  0  0  0  20
9  8  1  20  .1  0  0  0  20
10  8  1  20  .1  0  0  0  20
11  8  1  20  .1  0  0  0  20
12  8  1  20  .1  0  0  0  20
13  8  1  20  .1  0  0  0  20
14  8  1  20  .1  0  0  0  20
1  9  1  20  .1  0  0  0  20
2  9  1  20  .1  0  0  0  20
3  9  1  20  .1  0  0  0  20
4  9  1  20  .1  0  0  0  20
5  9  1  20  .1  0  0  0  20
6  9  1  20  .1  0  0  0  20
7  9  1  20  .1  0  0  0  20
8  9  1  20  .1  0  0  0  20
9  9  1  20  .1  0  0  0  20
10  9  1  20  .1  0  0  0  20
11  9  1  20  .1  0  0  0  20
12  9  1  20  .1  0  0  0  20
13  9  1  20  .1  0  0  0  20
14  9  1  20  .1  0  0  0  20
1  10  1  20  .1  0  0  0  20
2  10  1  20  .1  0  0  0  20
3  10  1  20  .1  0  0  0  20
4  10  1  20  .1  0  0  0  20
5  10  1  20  .1  0  0  0  20
6  10  1  20  .1  0  0  0  20
7  10  1  20  .1  0  0  0  20
8  10  1  20  .1  0  0  0  20
9  10  1  20  .1  0  0  0  20
10  10  1  20  .1  0  0  0  20
11  10  1  20  .1  0  0  0  20
12  10  1  20  .1  0  0  0  20
13  10  1  20  .1  0  0  0  20
14  10  1  20  .1  0  0  0  20
1  11  1  20  .1  0  0  0  20
2  11  1  20  .1  0  0  0  20
3  11  1  20  .1  0  0  0  20
4  11  1  20  .1  0  0  0  20
5  11  1  20  .1  0  0  0  20
6  11  1  20  .1  0  0  0  20
7  11  1  20  .1  0  0  0  20
8  11  1  20  .1  0  0  0  20
9  11  1  20  .1  0  0  0  20
10  11  1  20  .1  0  0  0  20
11  11  1  20  .1  0  0  0  20
12  11  1  20  .1  0  0  0  20
13  11  1  20  .1  0  0  0  20
14  11  1  20  .1  0  0  0  20
1  1  2  10  .1  0  0  20  30
2  1  2  10  .1  0  0  20  30
3  1  2  10  .1  0  0  20  30
4  1  2  10  .1  0  0  20  30
```

```
5  1  2  10  .1  0  0  20  30
6  1  2  10  .1  0  0  20  30
7  1  2  10  .1  0  0  20  30
8  1  2  10  .1  0  0  20  30
9  1  2  10  .1  0  0  20  30
10  1  2  10  .1  0  0  20  30
11  1  2  10  .1  0  0  20  30
12  1  2  10  .1  0  0  20  30
13  1  2  10  .1  0  0  20  30
14  1  2  10  .1  0  0  20  30
1  2  2  10  .1  0  0  20  30
2  2  2  10  .1  0  0  20  30
3  2  2  10  .1  0  0  20  30
4  2  2  10  .1  0  0  20  30
5  2  2  10  .1  0  0  20  30
6  2  2  10  .1  0  0  20  30
7  2  2  10  .1  0  0  20  30
8  2  2  10  .1  0  0  20  30
9  2  2  10  .1  0  0  20  30
10  2  2  10  .1  0  0  20  30
11  2  2  10  .1  0  0  20  30
12  2  2  10  .1  0  0  20  30
13  2  2  10  .1  0  0  20  30
14  2  2  10  .1  0  0  20  30
1  3  2  10  .1  0  0  20  30
2  3  2  10  .1  0  0  20  30
3  3  2  10  .1  0  0  20  30
4  3  2  10  .1  0  0  20  30
5  3  2  10  .1  0  0  20  30
6  3  2  10  .1  0  0  20  30
7  3  2  10  .1  0  0  20  30
8  3  2  10  .1  0  0  20  30
9  3  2  10  .1  0  0  20  30
10  3  2  10  .1  0  0  20  30
11  3  2  10  .1  0  0  20  30
12  3  2  10  .1  0  0  20  30
13  3  2  10  .1  0  0  20  30
14  3  2  10  .1  0  0  20  30
1  4  2  10  .1  0  0  20  30
2  4  2  10  .1  0  0  20  30
3  4  2  10  .1  0  0  20  30
4  4  2  10  .1  0  0  20  30
5  4  2  10  .1  0  0  20  30
6  4  2  10  .1  0  0  20  30
7  4  2  10  .1  0  0  20  30
8  4  2  10  .1  0  0  20  30
9  4  2  10  .1  0  0  20  30
10  4  2  10  .1  0  0  20  30
11  4  2  10  .1  0  0  20  30
12  4  2  10  .1  0  0  20  30
13  4  2  10  .1  0  0  20  30
14  4  2  10  .1  0  0  20  30
1  5  2  10  .1  0  0  20  30
2  5  2  10  .1  0  0  20  30
```

```
3   5   2   10   .1   0   0   20   30
4   5   2   10   .1   0   0   20   30
5   5   2   10   .1   0   0   20   30
6   5   2   10   .1   0   0   20   30
7   5   2   10   .1   0   0   20   30
8   5   2   10   .1   0   0   20   30
9   5   2   10   .1   0   0   20   30
10   5   2   10   .1   0   0   20   30
11   5   2   10   .1   0   0   20   30
12   5   2   10   .1   0   0   20   30
13   5   2   10   .1   0   0   20   30
14   5   2   10   .1   0   0   20   30
1   6   2   10   .1   0   0   20   30
2   6   2   10   .1   0   0   20   30
3   6   2   10   .1   0   0   20   30
4   6   2   10   .1   0   0   20   30
5   6   2   10   .1   0   0   20   30
6   6   2   10   .1   0   0   20   30
7   6   2   10   .1   0   0   20   30
8   6   2   10   .1   0   0   20   30
9   6   2   10   .1   0   0   20   30
10   6   2   10   .1   0   0   20   30
11   6   2   10   .1   0   0   20   30
12   6   2   10   .1   0   0   20   30
13   6   2   10   .1   0   0   20   30
14   6   2   10   .1   0   0   20   30
1   7   2   10   .1   0   0   20   30
2   7   2   10   .1   0   0   20   30
3   7   2   10   .1   0   0   20   30
4   7   2   10   .1   0   0   20   30
5   7   2   10   .1   0   0   20   30
6   7   2   10   .1   0   0   20   30
7   7   2   10   .1   0   0   20   30
8   7   2   10   .1   0   0   20   30
9   7   2   10   .1   0   0   20   30
10   7   2   10   .1   0   0   20   30
11   7   2   10   .1   0   0   20   30
12   7   2   10   .1   0   0   20   30
13   7   2   10   .1   0   0   20   30
14   7   2   10   .1   0   0   20   30
1   8   2   10   .1   0   0   20   30
2   8   2   10   .1   0   0   20   30
3   8   2   10   .1   0   0   20   30
4   8   2   10   .1   0   0   20   30
5   8   2   10   .1   0   0   20   30
6   8   2   10   .1   0   0   20   30
7   8   2   10   .1   0   0   20   30
8   8   2   10   .1   0   0   20   30
9   8   2   10   .1   0   0   20   30
10   8   2   10   .1   0   0   20   30
11   8   2   10   .1   0   0   20   30
12   8   2   10   .1   0   0   20   30
13   8   2   10   .1   0   0   20   30
14   8   2   10   .1   0   0   20   30
```

```
1  9  2  10  .1  0  0  20  30
2  9  2  10  .1  0  0  20  30
3  9  2  10  .1  0  0  20  30
4  9  2  10  .1  0  0  20  30
5  9  2  10  .1  0  0  20  30
6  9  2  10  .1  0  0  20  30
7  9  2  10  .1  0  0  20  30
8  9  2  10  .1  0  0  20  30
9  9  2  10  .1  0  0  20  30
10  9  2  10  .1  0  0  20  30
11  9  2  10  .1  0  0  20  30
12  9  2  10  .1  0  0  20  30
13  9  2  10  .1  0  0  20  30
14  9  2  10  .1  0  0  20  30
1  10  2  10  .1  0  0  20  30
2  10  2  10  .1  0  0  20  30
3  10  2  10  .1  0  0  20  30
4  10  2  10  .1  0  0  20  30
5  10  2  10  .1  0  0  20  30
6  10  2  10  .1  0  0  20  30
7  10  2  10  .1  0  0  20  30
8  10  2  10  .1  0  0  20  30
9  10  2  10  .1  0  0  20  30
10  10  2  10  .1  0  0  20  30
11  10  2  10  .1  0  0  20  30
12  10  2  10  .1  0  0  20  30
13  10  2  10  .1  0  0  20  30
14  10  2  10  .1  0  0  20  30
1  11  2  10  .1  0  0  20  30
2  11  2  10  .1  0  0  20  30
3  11  2  10  .1  0  0  20  30
4  11  2  10  .1  0  0  20  30
5  11  2  10  .1  0  0  20  30
6  11  2  10  .1  0  0  20  30
7  11  2  10  .1  0  0  20  30
8  11  2  10  .1  0  0  20  30
9  11  2  10  .1  0  0  20  30
10  11  2  10  .1  0  0  20  30
11  11  2  10  .1  0  0  20  30
12  11  2  10  .1  0  0  20  30
13  11  2  10  .1  0  0  20  30
14  11  2  10  .1  0  0  20  30
1  1  3  20  .1  0  0  30  50
2  1  3  20  .1  0  0  30  50
3  1  3  20  .1  0  0  30  50
4  1  3  20  .1  0  0  30  50
5  1  3  20  .1  0  0  30  50
6  1  3  20  .1  0  0  30  50
7  1  3  20  .1  0  0  30  50
8  1  3  20  .1  0  0  30  50
9  1  3  20  .1  0  0  30  50
10  1  3  20  .1  0  0  30  50
11  1  3  20  .1  0  0  30  50
12  1  3  20  .1  0  0  30  50
```

```
13  1  3  20  .1  0  0  30  50
14  1  3  20  .1  0  0  30  50
1  2  3  20  .1  0  0  30  50
2  2  3  20  .1  0  0  30  50
3  2  3  20  .1  0  0  30  50
4  2  3  20  .1  0  0  30  50
5  2  3  20  .1  0  0  30  50
6  2  3  20  .1  0  0  30  50
7  2  3  20  .1  0  0  30  50
8  2  3  20  .1  0  0  30  50
9  2  3  20  .1  0  0  30  50
10  2  3  20  .1  0  0  30  50
11  2  3  20  .1  0  0  30  50
12  2  3  20  .1  0  0  30  50
13  2  3  20  .1  0  0  30  50
14  2  3  20  .1  0  0  30  50
1  3  3  20  .1  0  0  30  50
2  3  3  20  .1  0  0  30  50
3  3  3  20  .1  0  0  30  50
4  3  3  20  .1  0  0  30  50
5  3  3  20  .1  0  0  30  50
6  3  3  20  .1  0  0  30  50
7  3  3  20  .1  0  0  30  50
8  3  3  20  .1  0  0  30  50
9  3  3  20  .1  0  0  30  50
10  3  3  20  .1  0  0  30  50
11  3  3  20  .1  0  0  30  50
12  3  3  20  .1  0  0  30  50
13  3  3  20  .1  0  0  30  50
14  3  3  20  .1  0  0  30  50
1  4  3  20  .1  0  0  30  50
2  4  3  20  .1  0  0  30  50
3  4  3  20  .1  0  0  30  50
4  4  3  20  .1  0  0  30  50
5  4  3  20  .1  0  0  30  50
6  4  3  20  .1  0  0  30  50
7  4  3  20  .1  0  0  30  50
8  4  3  20  .1  0  0  30  50
9  4  3  20  .1  0  0  30  50
10  4  3  20  .1  0  0  30  50
11  4  3  20  .1  0  0  30  50
12  4  3  20  .1  0  0  30  50
13  4  3  20  .1  0  0  30  50
14  4  3  20  .1  0  0  30  50
1  5  3  20  .1  0  0  30  50
2  5  3  20  .1  0  0  30  50
3  5  3  20  .1  0  0  30  50
4  5  3  20  .1  0  0  30  50
5  5  3  20  .1  0  0  30  50
6  5  3  20  .1  0  0  30  50
7  5  3  20  .1  0  0  30  50
8  5  3  20  .1  0  0  30  50
9  5  3  20  .1  0  0  30  50
10  5  3  20  .1  0  0  30  50
```

```
11   5   3   20   .1   0   0   30   50
12   5   3   20   .1   0   0   30   50
13   5   3   20   .1   0   0   30   50
14   5   3   20   .1   0   0   30   50
1   6   3   20   .1   0   0   30   50
2   6   3   20   .1   0   0   30   50
3   6   3   20   .1   0   0   30   50
4   6   3   20   .1   0   0   30   50
5   6   3   20   .1   0   0   30   50
6   6   3   20   .1   0   0   30   50
7   6   3   20   .1   0   0   30   50
8   6   3   20   .1   0   0   30   50
9   6   3   20   .1   0   0   30   50
10   6   3   20   .1   0   0   30   50
11   6   3   20   .1   0   0   30   50
12   6   3   20   .1   0   0   30   50
13   6   3   20   .1   0   0   30   50
14   6   3   20   .1   0   0   30   50
1   7   3   20   .1   0   0   30   50
2   7   3   20   .1   0   0   30   50
3   7   3   20   .1   0   0   30   50
4   7   3   20   .1   0   0   30   50
5   7   3   20   .1   0   0   30   50
6   7   3   20   .1   0   0   30   50
7   7   3   20   .1   0   0   30   50
8   7   3   20   .1   0   0   30   50
9   7   3   20   .1   0   0   30   50
10   7   3   20   .1   0   0   30   50
11   7   3   20   .1   0   0   30   50
12   7   3   20   .1   0   0   30   50
13   7   3   20   .1   0   0   30   50
14   7   3   20   .1   0   0   30   50
1   8   3   20   .1   0   0   30   50
2   8   3   20   .1   0   0   30   50
3   8   3   20   .1   0   0   30   50
4   8   3   20   .1   0   0   30   50
5   8   3   20   .1   0   0   30   50
6   8   3   20   .1   0   0   30   50
7   8   3   20   .1   0   0   30   50
8   8   3   20   .1   0   0   30   50
9   8   3   20   .1   0   0   30   50
10   8   3   20   .1   0   0   30   50
11   8   3   20   .1   0   0   30   50
12   8   3   20   .1   0   0   30   50
13   8   3   20   .1   0   0   30   50
14   8   3   20   .1   0   0   30   50
1   9   3   20   .1   0   0   30   50
2   9   3   20   .1   0   0   30   50
3   9   3   20   .1   0   0   30   50
4   9   3   20   .1   0   0   30   50
5   9   3   20   .1   0   0   30   50
6   9   3   20   .1   0   0   30   50
7   9   3   20   .1   0   0   30   50
8   9   3   20   .1   0   0   30   50
```

```
9  9  3  20  .1  0  0  30  50
10  9  3  20  .1  0  0  30  50
11  9  3  20  .1  0  0  30  50
12  9  3  20  .1  0  0  30  50
13  9  3  20  .1  0  0  30  50
14  9  3  20  .1  0  0  30  50
1  10  3  20  .1  0  0  30  50
2  10  3  20  .1  0  0  30  50
3  10  3  20  .1  0  0  30  50
4  10  3  20  .1  0  0  30  50
5  10  3  20  .1  0  0  30  50
6  10  3  20  .1  0  0  30  50
7  10  3  20  .1  0  0  30  50
8  10  3  20  .1  0  0  30  50
9  10  3  20  .1  0  0  30  50
10  10  3  20  .1  0  0  30  50
11  10  3  20  .1  0  0  30  50
12  10  3  20  .1  0  0  30  50
13  10  3  20  .1  0  0  30  50
14  10  3  20  .1  0  0  30  50
1  11  3  20  .1  0  0  30  50
2  11  3  20  .1  0  0  30  50
3  11  3  20  .1  0  0  30  50
4  11  3  20  .1  0  0  30  50
5  11  3  20  .1  0  0  30  50
6  11  3  20  .1  0  0  30  50
7  11  3  20  .1  0  0  30  50
8  11  3  20  .1  0  0  30  50
9  11  3  20  .1  0  0  30  50
10  11  3  20  .1  0  0  30  50
11  11  3  20  .1  0  0  30  50
12  11  3  20  .1  0  0  30  50
13  11  3  20  .1  0  0  30  50
14  11  3  20  .1  0  0  30  50
```

Attachment 2 - Example 1

```
:ANDOM WALK 3-D MASS TRANSPORT MODEL WITH GEOGRAPHIC FEATURE OVERLAYS
                              BY
                          DONALD KOCH
               ENGINEERING TECHNOLOGIES ASSOCIATES
                3458 ELLICOTT CENTER DRIVE
                ELLICOTT CITY, MD,  21043

                PHONE: 301-461-9920


  INPUT VELOCITY FILE ver3d.RND
           POROSITY =  .1
           PARTICLE MASS (LBS/PARTICLE) =  .01
           LONGITUDINAL DISPERSIVITY (FT)= 10
           TRANSVERSE DISPERSIVITY (FT)= 3
           VERTICAL DISPERSIVITY (FT)= 1
           HALF-LIFE FOR FIRST ORDER DECAY (YEARS)= 1E+10
       RETARDATION COEFFICIENT FOR LAYER  1  IS  1
       RETARDATION COEFFICIENT FOR LAYER  2  IS  1
       RETARDATION COEFFICIENT FOR LAYER  3  IS  1
       RETARDATION COEFFICIENT FOR CONFINING LAYER  1  IS  1
       RETARDATION COEFFICIENT FOR CONFINING LAYER  2  IS  1
///////////////////////////PARTICLES\\\\\\\\\\\\\\\\\\\\\\\\


            PARTICLES IN A RECTANGLAR PRISM
               COORDINATES:
               LOWER LEFT CORNER (X,Y) =  50 , 55  FT
               UPPER RIGHT CORNER (X,Y) =  50 , 55  FT
               Z COORDINATES (LOWER,UPPER) =  25 , 25  FT
               NUMBER OF PARTICLES =  5000

         TOTAL SYSTEM PARTICLES =  5000
\ \\\\\\\\\\\\\\\\\\\\\\\\\\\///////////////////////////////
  MAP AT TIME = 0  DAYS OF TOP VIEW   LAYER=ALL

A^CUMULATED TIME =  0  DAYS  PARTICLES=  5000    TOP VIEW   LAYER=ALL
E RTICLE MAP   (P SIGNIFIES PUMP LOCATION.  I SIGNIFIES LOCATION OF INJECTION)

     100Ç   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª
      90Ç   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª
      80Ç   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª
      70Ç   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª
      60Ç   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª
      50Ç   ª   ª   ª   ª   ª5000   ª   ª   ª   ª   ª   ª   ª   ª
      40Ç   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª
      30Ç   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª
      20Ç   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª
      10Ç   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª
        áááôáááôáááôáááôáááôáááôáááôáááôáááôáááôáááôáááôáááôáááô
```

```
            0   1   2   3   4   5   6   7   8   9   1   1   1   1
                0   0   0   0   0   0   0   0   0   0   1   2   3
                                                    0   0   0   0



DO A SCREEN PRINT NOW OR PRESS <RETURN> TO GO BACK TO THE MENU.?

  MAP AT TIME = 0  DAYS OF    COL X-SECTION COL= 8
  MAP AT TIME = 0  DAYS OF    COL X-SECTION COL=ALL
  MAP AT TIME = 0  DAYS OF    ROW X-SECTION ROW= 6

ACCUMULATED TIME =   0  DAYS  PARTICLES=  5000    ROW X-SECTION ROW= 6
PARTICLE MAP   (P SIGNIFIES PUMP LOCATION.  I SIGNIFIES LOCATION OF INJECTION)

      70Ç   ▲   ▲   ▲   ▲   ▲   ▲   ▲   ▲   ▲   ▲   ▲   ▲   ▲   ▲
      63Ç   ▲   ▲   ▲   ▲   ▲   ▲   ▲   ▲   ▲   ▲   ▲   ▲   ▲   ▲
      56Ç   ▲   ▲   ▲   ▲   ▲   ▲   ▲   ▲   ▲   ▲   ▲   ▲   ▲   ▲
      49Ç   ▲   ▲   ▲   ▲   ▲   ▲   ▲   ▲   ▲   ▲   ▲   ▲   ▲   ▲
      42Ç   ▲   ▲   ▲   ▲   ▲   ▲   ▲   ▲   ▲   ▲   ▲   ▲   ▲   ▲
      35Ç   ▲   ▲   ▲   ▲   ▲   ▲   ▲   ▲   ▲   ▲   ▲   ▲   ▲   ▲
      28Ç   ▲   ▲   ▲   ▲   ▲5000   ▲   ▲   ▲   ▲   ▲   ▲   ▲   ▲
      21Ç   ▲   ▲   ▲   ▲   ▲   ▲   ▲   ▲   ▲   ▲   ▲   ▲   ▲   ▲
      14Ç   ▲   ▲   ▲   ▲   ▲   ▲   ▲   ▲   ▲   ▲   ▲   ▲   ▲   ▲
       7Ç   ▲   ▲   ▲   ▲   ▲   ▲   ▲   ▲   ▲   ▲   ▲   ▲   ▲   ▲
         áááôáááôáááôáááôáááôáááôáááôáááôáááôáááôáááôáááôáááôáááô

            0   1   2   3   4   5   6   7   8   9   1   1   1   1
                0   0   0   0   0   0   0   0   0   0   1   2   3
                                                    0   0   0   0



DO A SCREEN PRINT NOW OR PRESS <RETURN> TO GO BACK TO THE MENU.?


  WRITING PLUME DATA TO EXTERNAL FILE ver3d1.dat

      PRESENT SIMULATION TIME =  0   DAYS
       INCREMENTAL SIMULATION TIME =  10  DAYS
XMAX =  2  FT   ZMAX =  1  FT

NP= 5000



  MAP AT TIME = 10  DAYS OF TOP VIEW   LAYER= 2
CONCENTRATION MAP IN PPm  (P SIGNIFIES PUMPAGE, I SIGNIFIES INJECTION)

ACCUMULATED TIME =  10  DAYS  PARTICLES=  5000    TOP VIEW   LAYER= 2
CONCENTRATION MAP IN PPm  (P SIGNIFIES PUMPAGE, I SIGNIFIES INJECTION)

     105Ç   ▲   ▲   ▲   ▲   ▲   ▲   ▲   ▲   ▲   ▲   ▲   ▲   ▲   ▲
      95Ç   ▲   ▲   ▲   ▲   ▲   ▲   ▲   ▲   ▲   ▲   ▲   ▲   ▲   ▲
      85Ç   ▲   ▲   ▲   ▲   3   ▲   ▲   ▲   ▲   ▲   ▲   ▲   ▲   ▲
      76Ç   ▲   ▲   ▲   3   3  33  21  30   8   2   2   ▲   ▲   ▲
  56Ç   ª   2  18 116 311 633 795 574 320  92  18   ª   ª   ª
  46Ç   ª   2  13  69 166 316 495 385 177  67   7   3   ª   ª
  36Ç   ª   ª   ª   8  21  54  69  48  30   3   ª   ª   ª   ª
  26Ç   ª   ª   ª   2   ª   2   ª   2   ª   ª   ª   ª   ª   ª
  16Ç   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª
     áááôáááôáááôáááôáááôáááôáááôáááôáááôáááôáááôáááôáááôáááô

        0   1   2   3   3   4   5   6   7   8   9   1   1   1
            0   0   0   9   9   9   9   9   9   9   0   1   2
                                                    9   9   8
```

```
DO A SCREEN PRINT NOW OR PRESS <RETURN> TO GO BACK TO THE MENU.?

 MAP AT TIME = 10  DAYS OF TOP VIEW   LAYER= 2
CONCENTRATION MAP IN PPm  (P SIGNIFIES PUMPAGE, I SIGNIFIES INJECTION)

:CUMULATED TIME =  10  DAYS  PARTICLES=  5000   TOP VIEW   LAYER= 2
:ONCENTRATION MAP IN PPm  (P SIGNIFIES PUMPAGE, I SIGNIFIES INJECTION)

 105Ç   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª
  95Ç   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª
  85Ç   ª   ª   ª   ª   3   ª   ª   ª   ª   ª   ª   ª   ª   ª
  76Ç   ª   ª   ª   3   3  33  21  30   8   2   2   ª   ª   ª
  66Ç   ª   2   5  30 151 218 315 251 144  38   5   2   ª   ª
  56Ç   ª   2  18 116 311 633 795 574 320  92  18   ª   ª   ª
  46Ç   ª   2  13  69 166 316 495 385 177  67   7   3   ª   ª
  36Ç   ª   ª   ª   8  21  54  69  48  30   3   ª   ª   ª   ª
  26Ç   ª   ª   ª   2   ª   2   ª   2   ª   ª   ª   ª   ª   ª
  16Ç   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª
     áááôáááôáááôáááôáááôáááôáááôáááôáááôáááôáááôáááôáááôáááô

        0   1   2   3   3   4   5   6   7   8   9   1   1   1
            0   0   0   9   9   9   9   9   9   9   0   1   2
                                                    9   9   8


DO A SCREEN PRINT NOW OR PRESS <RETURN> TO GO BACK TO THE MENU.?

I WER LEFT-HAND CORNER OF PLOT POSITION =  0              0
GRID SIZE OF PLOT MAP, IN FEET =  10
N MBER OF ROWS AND COLUMNS OF PLOT MAP =  11              14

Plot file name is =vxy2.DAT
```

Example #2

This example tests the decay routine of the RAND3D model. The simplest test is to set the model simulation time to the half-life decay value. This causes exactly half of the initial mass to decay at the end of the simulation.

A two layer aquifer was modeled with two identical 10 foot thick aquifers with a porosity of 0.1, a uniform seepage velocity of 1 ft/day, and a vertical seepage velocity of 0.1 ft/day. Dispersivities were assumed to be 10 feet in longitudinal direction, 3 feet in the transverse, and 1 foot in the vertical. A 15 by 15 grid with 10 feet spacings was used with a slug source at node (8,8). A cylindrical source was modeled at the bottom elevation of 10 feet and top at 15 feet with a 10 foot diameter. The model was initiated with 5000 particles at 0.0125 particle weight.

The half-life decay was set at 10-days (0.0274 years) thus the model simulation time was set at 10 days. The results of the simulation is included as an attachment. The results showed that 2500 particles decayed and 2500 particles remained after 10 days which verified the decay routine.

Attachment 1 - Example 2

```
RANDOM WALK 3-D MASS TRANSPORT MODEL WITH GEOGRAPHIC FEATURE OVERLAYS
                            BY
                       DONALD KOCH
              ENGINEERING TECHNOLOGIES ASSOCIATES
               3458 ELLICOTT CENTER DRIVE
               ELLICOTT CITY, MD,  21043

               PHONE: 301-461-9920


 READING IN PLUME DATA FROM EXTERNAL FILE test0.dat

 INPUT VELOCITY FILE test.RND
          POROSITY =  .1
          PARTICLE MASS (LBS/PARTICLE) =  .0125
          LONGITUDINAL DISPERSIVITY (FT)= 10
          TRANSVERSE DISPERSIVITY (FT)= 5
          VERTICAL DISPERSIVITY (FT)= 1
          HALF-LIFE FOR FIRST ORDER DECAY (YEARS)= .0274
     RETARDATION COEFFICIENT FOR LAYER  1  IS  1
     RETARDATION COEFFICIENT FOR LAYER  2  IS  1
     RETARDATION COEFFICIENT FOR CONFINING LAYER  1  IS  1
 MAP AT TIME = 0  DAYS OF TOP VIEW   LAYER=ALL

ACCUMULATED TIME =  0  DAYS  PARTICLES= 5000   TOP VIEW   LAYER=ALL
PARTICLE MAP   (P SIGNIFIES PUMP LOCATION.  I SIGNIFIES LOCATION OF INJECTION)

     142Ç   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª
     128Ç   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª
     114Ç   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª
     100Ç   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª
      85Ç   ª   ª   ª   ª   ª12501711   ª   ª   ª   ª   ª   ª   ª
      71Ç   ª   ª   ª   ª   ª 7891250   ª   ª   ª   ª   ª   ª   ª
      57Ç   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª
      42Ç   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª
      28Ç   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª
      14Ç   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª   ª
         áááôáááôáááôáááôáááôáááôáááôáááôáááôáááôáááôáááôáááôáááô

            0   1   2   4   5   7   8   1   1   1   1   1   1   1
                4   8   2   7   1   5   0   1   2   4   5   7   8
                                        0   4   8   2   7   1   5


.. A SCREEN PRINT NOW OR PRESS <RETURN> TO GO BACK TO THE MENU.?

      PRESENT SIMULATION TIME =   0   DAYS
       INCREMENTAL SIMULATION TIME =  10  DAYS
)MAX =  10  FT   ZMAX =  1  FT

 = 2500
 MAP AT TIME = 0  DAYS OF TOP VIEW   LAYER=ALL

ACCUMULATED TIME =   0  DAYS  PARTICLES= 2500    TOP VIEW   LAYER=ALL
PARTICLE MAP   (P SIGNIFIES PUMP LOCATION.  I SIGNIFIES LOCATION OF INJECTION)

     142Ç   ▴   ▴   ▴   ▴   ▴   ▴   ▴   ▴   ▴   ▴   ▴   ▴   ▴   ▴
     128Ç   ▴   ▴   ▴   ▴   ▴   ▴   ▴   ▴   ▴   ▴   ▴   ▴   ▴   ▴
     114Ç   ▴   ▴   ▴   ▴   ▴   2   4   8   1   ▴   ▴   ▴   ▴   ▴
     100Ç   ▴   ▴   ▴   ▴  12  53 120  76  23   2   ▴   ▴   ▴   ▴
      85Ç   ▴   ▴   ▴   3  50 247 427 264  87   5   ▴   ▴   ▴   ▴
      71Ç   ▴   ▴   ▴   4  64 213 366 243  59   3   1   ▴   ▴   ▴
      57Ç   ▴   ▴   ▴   ▴   8  44  62  34   7   4   ▴   ▴   ▴   ▴
      42Ç   ▴   ▴   ▴   ▴   ▴   2   1   1   ▴   ▴   ▴   ▴   ▴   ▴
      28Ç   ▴   ▴   ▴   ▴   ▴   ▴   ▴   ▴   ▴   ▴   ▴   ▴   ▴   ▴
      14Ç   ▴   ▴   ▴   ▴   ▴   ▴   ▴   ▴   ▴   ▴   ▴   ▴   ▴   ▴
        áááôáááôáááôáááôáááôáááôáááôáááôáááôáááôáááôáááôáááôáááô

            0   1   2   4   5   7   8   1   1   1   1   1   1   1
                4   8   2   7   1   5   0   1   2   4   5   7   8
                                        0   4   8   2   7   1   5


 A SCREEN PRINT NOW OR PRESS <RETURN> TO GO BACK TO THE MENU.?
```

Appendix D - RAND3D Model Source Code

```
REM NEW RANDOM WALK 3-D MODEL
REM   5/9/89
REM
REM     VARIABLES
REM     ---------
REM ASPECT - ASPECT RATIO OF SCREEN
REM BOT(I,J,K) - ELEVATION OF BOTTOM OF NODE I,J,K (FT)
REM BOT(I,J,K) - ELEVATION OF BOTTOM OF NODE I,J,K (FT)
REM BOTCAPTURE - BOTTOM ELEVATION AT WHICH PARTICLE MAY BE
REM              CAPTURED BY SINK
REM CAPTUREMAX - RIVER MAXIUM CAPTURE DISTANCE
REM CDX - COLUMN SPACING OF MAP ARRAY
REM CDY - ROW SPACING OF MAP ARRAY
REM COLUMNS - NUMBER OF COLUMNS IN PLOT ARRAY
REM CONC(L) - SINK CONCENTRATION FOR SINK L(MG/L)
REM D - TIME STEP FOR MOVE, =DELTA IF SLUG SOURCE PARTICLE
REM     = RANDOM FRACTION OF DELTA FOR CONTINUOUS SOURCE PARTICLE
REM     (DAYS)
REM DD - HORIZONTAL DISTANCE TO MOVE (FT)
REM DDX - DISTANCE OF MOVE (FT)
REM DELTA - TIME STEP OF MOVE SUBROUTINE (DAYS)
REM DELX - COLUMN SPACING (FT)
REM DELY - ROW SPACING (FT)
REM DL  - LONGITUDINAL DISPERSIVITY (FT)
REM DMAX - MAXIMUM HORIZONTAL MOVE DISTANCE BEFORE RECOMPUTING
REM        VELOCITY (DAYS)
REM DT  - TRANSVERSE DISPERSIVITY (FT)
REM DV  - VERTICAL DISPERSIVITY (FT)
REM E   - POROSITY
REM ENDTIME(M) - ENDTIMES FOR EACH MODEL VELOCITY FILE (DAYS)
REM F$ - MAP FLAG, IF="A" THEN MAP HAS BEEN RUN
REM G$ - FLAG INDICATING COEFFICIENTS HAVE BEEN ENTERED IF
REM      ="B"
REM I1! - X COORDINATE OF LOWER LEFT CORNER OF SCREEN WINDOW
REM       (FT)
REM I2! - X COORDINATE OF UPPER LEFT CORNER OF SCREEN WINDOW
REM       (FT)
REM ICOLXS - COLUMN DISPLAYED IN COLUMN CROSS-SECTION VIEW
REM IPEN - COLOR OF SPECIAL FEATURES ON SCRREN
REM J1! - Y COORDINATE OF LOWER LEFT CORNER OF SCREEN WINDOW
REM       (FT)
REM J2! - Y COORDINATE OF UPPER LEFT CORNER OF SCREEN WINDOW
REM       (FT)
REM JROWXS - ROW DISPLAYED IN ROW CROSS-SECTION VIEW
REM LAMDBA - HALF LIFE FOR FIRST ORDER DECAY (YEARS)
REM LL - LAYER OF PARTICLE
REM LLX  - X COORDINATE OF LOWER LEFT CORNER OF MODEL GRID (FT)
REM LLY  - Y COORDINATE OF LOWER LEFT CORNER OF MODEL GRID (FT)
REM LLZ  - Z COORDINATE OF LOWER LEFT CORNER OF MODEL GRID (FT)
REM LN - LINES OF MAP
REM LOWERLX - X COORDINATE OF LOWER LEFT CORNER OF PLOT AREA (FT)
REM LOWERLY - Y COORDINATE OF LOWER LEFT CORNER OF PLOT AREA (FT)
REM MAPTYPE$ - FLAG INDICATING WHAT VIEW OF THE PROBLEM WAS LAST
REM            DISPLAYED ON THE SCREEN , A - TOP VIEW, B - ROW
REM            CROSS-SECTION VIEW, C - COLUMN CROSS-SECTION VIEW
REM NC  - NUMBER OF COLUMNS
REM NENDTIME - NUMBER OF TIME PERIODS
REM NEWX - X COORDINATE OF PARTICLE AFTER MOVE (FT)
REM NEWY - Y COORDINATE OF PARTICLE AFTER MOVE (FT)
REM NL  - NUMBER OF LAYERS
REM NMAP(I,J) - MAP AND PLOT STORAGE FOR # OF PARTICLES OR
REM             CONCENTRATIONS
REM NP  - NUMBER OF PARTICLES
REM NR  - NUMBER OF ROWS
REM NS - NUMBER OF SINKS
REM NSPF - NUMBER OF SPECIAL FEATURE FILES (WITH KEYS AND FILE
REM        NAMES)
REM O   - SLIDE COUNTER
REM PM  - PARTICLE MASS (LBS)
REM PPP - CONCENTRATION FACTOR, =0.001 FOR PPB, =1 FOR PPM
REM       =1000 FOR PPT
REM QSUM(L) - DISCHARGE OF SINK L (GPD)
REM R$ - NAME OF VELOCITY FILE (WITH EXTENSION .RND)
REM R3 - DISTANCE FROM SINK TO PARTICLE (FT)
REM R4 - VERTICAL DISTANCE FROM PARTICLE TO BOTTOM OF RIVER
REM RETARD(K) - RETARDATION FACTORS FOR EACH LAYER AND CONFINING LAYER
REM RL - LONGITUDINAL DISPERSION (FT)
REM ROWS - NUMBER OF ROWS IN PLOT ARRAY
REM RT - TRANSVERSE DISPERSION (FT)
REM RZ - VERTICAL DISPERSION (FT)
REM SCALEDEF - DEFAULT X DISTANCE FOR SCREEN WINDOW
REM SCALEX - X DIMENSION OF SCREEN WINDOW (I2!-I1!) (FT)
REM SCREENX - X COORDINATE TO PLOT ON SCREEN (FT)
REM SCREENY - Y COORDINATE TO PLOT ON SCREEN (FT)
REM SIZE - GRID SPACING FOR PLOT AREA (FT)
REM SPFFILE$(I) - FILE NAMES OF SPECIAL FEATURE FILES
REM SPFKEY$(I,J) - KEYS USED TO CALL SPECIAL FEATURE FILE PLOTS, I
REM                IS SPECIAL FEATURE COUNTER, J IS 1 OR 2 CONTAINING
REM                BOTH UPPER AND LOWER CASE LETTERS
REM SPFSAVE$ - NAME OF FILE WITH SPECIAL FEATURE DATA
REM SVIEWX - X DIMENSION OF ZOOM BOX
REM SWITCH%(I) - ARRAY OF FLAGS INDICATING WHETHER IT IS A
REM               SLUG OR CONTINUOUS PARTICLE, 1= CONTINUOUS,
REM               0=SLUG
REM T2  - TIME (DAYS)
REM TEMPSCR - FLAG, IF=1 IT INDICATES A SCREEN IMAGE HAS BEEN
REM           SAVED AS "TEMP.SCR"
REM THICK(I,J,K) - THICKNESS OF AQUIFER OF COLUMN I, ROW J, AND LAYER K
REM TOP(I,J,K) - ELEVATION OF TOP OF NODE I,J,K, IF THE TOP WATER TABLE
REM                  LAYER THEN =0 UNLESS A RIVER NODE, THEN = BOTTOM OF
REM                  RIVER ELEVATION (FT)
REM TOPCAPTURE - TOP ELEVATION AT WHICH PARTICLE MAY BE
REM               BE CAPTURED BY SINK
REM VI(I,J,K) - VELOCITY IN X DIRECTION FROM I,J,K TO I+1,J,K (FT/DAY)
REM VJ(I,J,K) - VELCOITY IN Y DIRECTION FROM I,J,K TO I,J+1,K (FT/DAY)
REM VK(I,J,K) - VELOCITY IN Z DIRECTION FROM I,J,K TO I,J,K+1 (FT/DAY)
```

```
   REM VX - INTERPOLATED X VELOCITY (FT/DAY)
   REM VY - INTERPOLATED Y VELOCITY (FT/DAY)
   REM VZ - INTERPOLATED Z VELCOITY (FT/DAY)
   REM WD - COLUMNS OF MAP
   REM X(I) - ARRAY OF X COORDINATES FOR PARTICLES (FT)
   REM X1(L) - X COORDINATE OF SINK L (FT)
   REM XP - X COORDINATE OF PARTICLE POSITION IN GRID UNITS
   REM XX$ - FLAG INDICATING A COLOR MONITOR IS PRESENT, ALWAYS="Y"
   REM       IN THIS VERSION
   REM Y(I) - ARRAY OF Y COORDINATES FOR PARTICLES (FT)
   REM Y1(L) - Y COORDINATE OF SINK L (FT)
   REM YASPECT - ASPECT RATIO OF SCREEN FOR TOP VIEW
   REM YP - Y COORDINATE OF PARTICLE POSITION IN GRID UNITS
   REM Z(I) - ARRAY OF Z COORDINATS FOR PARTICLES (FT)
   REM Z1(L) - LAYER OF SINK L
   REM Z2$ - STRING OF KEYS PRESSED TO PUT SPECIAL FEATURES ON SCREEN
   REM Z3$ - STRING OF KEYS PRESSED TO PUT SPECIAL FEATURES ON SCREEN
   REM ZASPECT - ASPECT RATIO OF SCREEN FOR CROSS-SECTION VIEW
   REM ZMAX - MAXIUM VERTICAL MOVE DISTANCE BEFORE RECOMPUTING
   REM        VELOCITY (DAYS)
   REM ZZZ - VARIABLE USED TO IMPLEMENT FIRST ORDER DECAY
   REM
   REM
30 REM RECOVER FROM ALL ERRORS WITHOUT BOMBING OUT.
40 ON ERROR GOTO 5430
290 O=O:REM INITIALIZE SLIDE COUNTER
300 FOR I=1 TO 60
310 NEXT I
320 SCREEN 0:WIDTH 80
330 REM NAME: THREE DIMENSIONAL RANDOM WALK (MICROCOMPUTER VERSION)
350 REM
360 REM PURPOSE: TO SIMULATE 3-DIMENSIONAL, STEADY OR NONSTEADY
370 REM          MASS TRANSPORT PROBLEMS IN HETEROGENEOUS AQUIFERS
380 REM          WITH THE RANDOM WALK PARTICLE THEORY.
390 REM
400 REM WRITTEN BY:  DONALD KOCH
402 REM              ENGINEERING TECHNOLOGIES ASSOCIATES, INC.
404 REM              3458 ELLICOTT CENTER DRIVE
406 REM              ELLICOTT CITY, MD, 21043
408 REM              (301) 461-9920
409 REM     BASED ON CODE WRITTEN BY
    REM              THOMAS A. PRICKETT & ASSOCIATES
410 REM              URBANA, ILLINOIS 61801
420 REM              (217)384-0615
430 DEFINT I,J,K
440 REM ALLOW ARRAYS FOR 10000 PARTICLES AND 99 SINKS AND 45 BY 45 VELOCITY ARRAY
445 REM $DYNAMIC
450  DIM NMAP(45,45),CONC(99),THICK(45,45,3),VI(45,45,3),VJ(45,45,3),VK(45,45,3)
460  DIM X1(99),Y1(99),Z1(99),QSUM(99),TOP(45,45,3),BOT(45,45,3)
470 DIM X(10001),Y(10001),Z(10001),SWITCH%(10001)
471 REM $STATIC
472 DIM SPFKEY$(20,2),SPFFILE$(20),ENDTIME(20),RETARD(5)
480 REM SET UP MONITOR FOR COLOR OR WITHOUT COLOR.
490 REM LOCATE 8,30:PRINT "MONITOR SETUP PARAMETERS":PRINT:PRINT:PRINT
```

```
500 REM INPUT "              Do you have a color monitor?? (Y-Yes or <RETURN>-No) ",XX$
    XX$="Y":rem assume color monitor
510 IF XX$="Y" OR XX$="y" THEN COLOR 15,1,1:CLS ELSE CLS
535 NPER=0:REM INITIALIZE VELOCITY FILE (TIME PERIODS) COUNTER
537 NENDTIME=0:REM INITIALIZE NUMBER OF VELOCITY FILES
570 PRINT "    RANDOM WALK 3-D MASS TRANSPORT MODEL WITH GEOGRAPHIC FEATURE OVERLAYS"
580 LPRINT:LPRINT
590 LPRINT " RANDOM WALK 3-D MASS TRANSPORT MODEL WITH GEOGRAPHIC FEATURE OVERLAYS"
600 LPRINT "                              BY"
610 LPRINT "                         DONALD KOCH"
612 LPRINT "             ENGINEERING TECHNOLOGIES ASSOCIATES"
630 LPRINT "              3458 ELLICOTT CENTER DRIVE"
640 LPRINT "              ELLICOTT CITY, MD,  21043"
650 LPRINT ""
660 LPRINT "              PHONE: 301-461-9920":LPRINT ""
670 LPRINT "":LPRINT ""
680 PRINT"  "
690 PRINT ""
700 PRINT " "
705 G$="F":REM SET COEFFICIENT FLAG
720 T2=0:REM INITIALIZE VARIABLES, TIME=0
730 NS = 0:REM NUMBER OF SINKS=0
740 NP=0:REM NUMBER OF PARTICLES=0
750 REM HERE IS THE MAIN MENU.  REFER TO ASCII CODES FOR FRAME CHARACTERS.
760 PRINT:PRINT:SCREEN 0:WIDTH 80:IF XX$="Y" OR XX$="y" THEN COLOR 15,1,1
770 PRINT "             èëëëëëëëëëëëëëëëëë USER MENU ëëëëëëëëëëëëëëëëë£"
780 PRINT "             ¤     FORMAT:   COMMAND - DESCRIPTION             ¤"
790 PRINT "             äëëëëëëëëëëëëëëëëëëëëëëëëëÙëëëëëëëëëëëëëëëëëëëëëëëëë¿"
800 PRINT "             ¤  SET   - MASS TRANSPORT ¤  INPUT - READ IN FLOW  ¤"
810 PRINT "             ¤         COEFFICIENTS    ¤          DATA FILE     ¤"
820 PRINT "             ¤                         ¤  P - ADD PARTICLES     ¤"
830 PRINT "             ¤  RE -  READ DATA        ¤      (POLLUTANTS)      ¤"
840 PRINT "             ¤  WR -  WRITE DATA       ¤                        ¤
850 PRINT "             ¤  PLT - SAVE PLOT DATA   ¤ MOVE - MOVE PARTICLES  ¤"
860 PRINT "             ¤                         ¤         FOR TIME STEP  ¤"
870 PRINT "             ¤  XS - CLEAR SINKS       ¤                        ¤"
880 PRINT "             ¤  XP - CLEAR PARTICLES   ¤ MAP - MAP PARTICLES OR ¤"
890 PRINT "             ¤  XT - SET TIME TO 0     ¤       CONCENTRATIONS   ¤"
900 PRINT "             ¤                         ¤                        ¤"
910 PRINT "             ¤  SPF - SETUP SPECIAL    ¤ VIEW - REVIEW SLIDES   ¤"
911 PRINT "             ¤         FEATURES        ¤                        ¤"
912 PRINT "             ¤                         ¤ Q - QUIT               ¤"
920 PRINT "             àëëëëëëëëëëëëëëëëëëëëëëëëëòëëëëëëëëëëëëëëëëëëëëëëëëë¥"
930 PRINT:BEEP
940 INPUT " COMMAND ";A$
950 REM JUNCTION POINT FOR MENU COMMAND SELECTION.
960 IF A$="P" OR A$="p" THEN GOSUB 1270
970 IF A$="XP" OR A$="xp" THEN GOSUB 1180
980 IF A$ = "RE" OR A$ = "re" THEN GOSUB 5110
990 IF A$="XS" OR A$="xs" THEN GOSUB 1150
1000 IF A$ = "WR" OR A$ = "wr" THEN GOSUB 5280
1010 IF A$="XT" OR A$="xt" THEN GOSUB 1120
1020 IF A$="PLT" OR A$="plt" THEN GOSUB 6080
1030 IF A$="MOVE" OR A$="move" OR A$="Move" THEN GOSUB 2510
```

```
1040 IF A$="Q" OR A$="q" THEN END
1050 IF A$="MAP" OR A$="map" OR A$="Map" THEN GOSUB 3940
1060 IF A$="SET" OR A$="set" THEN GOSUB 5480
1070 IF A$="EDIT" OR A$="edit" THEN GOSUB 5480
1080 IF A$="VIEW" OR A$="view" THEN GOSUB 7270
1090 IF A$="INPUT" OR A$="input" THEN GOSUB 9140
1092 IF A$="SPF" OR A$="spf" THEN GOSUB 11000
1100 GOTO 760
1110 REM INITIATE OR ZERO OUT TIME, SINKS, OR PARTICLE POINTERS.
1120 T2=0:REM ZERO TIME
1130 PRINT "      SIMULATION TIME INITIATED  "
1140 RETURN
1150 NS=0:REM ZERO SINKS
1160 PRINT "      NUMBER OF SINKS INITIATED  "
1170 RETURN
1180 FOR I = 1 TO NP:REM ZERO PARTICLES
1190 SWITCH%(I)=0
1200 NEXT I
1210 PRINT "      NUMBER OF PARTICLES INITIATED  "
1220 NP=0
1230 RETURN
1240 REM  ++++++++++++++++++++++++++++++++++++
1250 REM  +  BEGIN PARTICLE INPUT SUBROUTINE  +
1260 REM  ++++++++++++++++++++++++++++++++++++
1270 PRINT "//////////////////////////PARTICLES\\\\\\\\\\\\\\\\\\\\\\\\\"
1280 PRINT
1290 LPRINT "////////////////////////////PARTICLES\\\\\\\\\\\\\\\\\\\\\\\\"
1300 LPRINT ""
1310 L=0:REM INITIALIZE LINE COUNTER
1320 C=0:REM INITIALIZE CYLINDER COUNTER
     IF DELX<=0 OR DELY<=0 THEN PRINT "  YOU MUST ENTER A VELOCITY FILE BEFORE INITIATING
PARTICLES":GOSUB 13100:RETURN
1330 PRINT
1340 INPUT "      ADD PARTICLES TO SYSTEM? (Y-YES, <RETURN>-NO)";A$
1350 PRINT
1360 IF A$ <> "Y" AND A$<>"y" THEN 2420
1370 INPUT "      CYLINDER, LINE, OR RECTANGULAR (C, L, R) ";A$
1380 IF A$="c" OR A$="C" THEN 2400
1390 IF A$="l" OR A$="L" THEN 1750
1400 IF A$="R" OR A$="r" THEN 1410 ELSE 1370
1410 LPRINT:REM                                           INITIATE PARTICLES IN RECTANGULAR PRISM
1420 LPRINT "            PARTICLES IN A RECTANGLAR PRISM"
1430 INPUT "      ENTER COORDINATES OF LOWER LEFT CORNER (X,Y)";X5,Y5
1440 LPRINT "               COORDINATES:"
1450 LPRINT "               LOWER LEFT CORNER (X,Y) = ";X5;",";Y5;" FT"
1460 INPUT "      ENTER COORDINATES OF UPPER RIGHT CORNER (X,Y)";X6,Y6
1470 LPRINT "               UPPER RIGHT CORNER (X,Y) = ";X6;",";Y6;" FT"
1472 INPUT "      ENTER Z COORDINATES (LOWER,UPPER) ";Z5,Z6
1474 LPRINT "               Z COORDINATES (LOWER,UPPER) = ";Z5;",";Z6;" FT"
1480 PRINT "":IF (X6-X5)<0 OR (Y6-Y5)<0 THEN PRINT "CAN'T FORM RECTANGLE--REDO":GOTO 1430
1490 IF X5<=(LLX) OR X5>=(LLX+DELX*(NC)) THEN PRINT "  LOWER LEFT CORNER X COORDINATE IS OUTSIDE THE
MODEL GRID":GOTO 1430
1500 IF Y5<=(LLY) OR Y5>=(LLY+DELY*(NR)) THEN PRINT "  LOWER LEFT CORNER Y COORDINATE IS OUTSIDE THE
MODEL GRID":GOTO 1430
```

```
1510 IF Y6<=(LLY) OR Y6>=(LLY+DELY*(NR)) THEN PRINT "  UPPER RIGHT CORNER Y COORDINATE IS OUTSIDE
THE MODEL GRID":GOTO 1430
1520 IF X6<=(LLX) OR X6>=(LLX+DELX*(NC)) THEN PRINT "  UPPER RIGHT CORNER X COORDINATE IS OUTSIDE
THE MODEL GRID":GOTO 1430
     IZ=INT((X5-LLX)/DELX+1.0)
     JZ=INT((Y5-LLY)/DELY+1.0)
     IF Z5 < BOT(IZ,JZ,1) OR Z5 > (BOT(IZ,JZ,NL)+THICK(IZ,JZ,NL)) THEN PRINT "  1ST Z COORDINATE
OUTSIDE MODEL":GOTO 1430
     IF Z6 < BOT(IZ,JZ,1) OR Z6 > (BOT(IZ,JZ,NL)+THICK(IZ,JZ,NL)) THEN PRINT "  2ND Z COORDINATE
OUTSIDE MODEL":GOTO 1430
1530 PRINT "      Enter the letter C for CONTINUOUS POLLUTION--otherwise a <RETURN>"
1540 INPUT "      or any other character will assume a SLUG input";A$
1550 IF A$="C" OR A$="c" THEN PRINT "      YOU ARE NOW IN THE CONTINUOUS MODE FOR THIS SOURCE"
1560 PRINT:IF A$="C" OR A$="c" THEN LPRINT "YOU ARE NOW IN THE CONTINUOUS MODE FOR THIS SOURCE"
1570 INPUT "      ENTER NUMBER OF PARTICLES ";M
1580 LPRINT "                NUMBER OF PARTICLES = ";M
1590 PRINT ""
1600 INPUT "      DO YOU WANT TO REDO THIS SCREEN? (Y-YES, <RETURN>-NO)";OK$
1610 IF OK$="Y" OR OK$="y" THEN 1330
1620 DX = X6  - X5
1630 DY = Y6 - Y5
     DZ = Z6 - Z5
1640 FOR I = 1 TO M
1650 X = X5 + DX*RND
1660 Y = Y5 + DY*RND
     Z = Z5 +DZ*RND
1670 GOSUB 2100:REM ADD A PARTICLE
1690 NEXT I
1700 PRINT
1710 PRINT "      SYSTEM PARTICLES = ";NP
1720 GOTO 1330:REM RETURN TO BEGINNING OF PARTICLE SUBROUTINE
1730 PRINT:REM ****************************************
1740 LPRINT:REM   INITIATE PARTICLES ON 3D LINE
1750 PRINT "      LINE NUMBER ";L+1
1760 LPRINT "          PARTICLES ON A LINE"
1770 LPRINT "             LINE NUMBER";L+1
1780 PRINT
1790 INPUT "        COORDINATES OF LINE BEGINNING (X,Y,Z) ";X2,Y2,Z2
1800 LPRINT "             LINE COORDINATES:"
1810 LPRINT "             BEGINNING POINT (X,Y,Z) = ";X2;",";Y2;",";Z2;" FT"
1820 INPUT "        COORDINATES OF END OF LINE (X,Y,Z) ";X3,Y3,Z3
1830 LPRINT "             END POINT OF LINE (X,Y,Z) = ";X3;",";Y3;",";Z3;" FT"
1840 IF X2<=(LLX) OR X2>=(LLX+DELX*(NC)) THEN PRINT "  BEGINNING POINT X COORDINATE IS OUTSIDE THE
MODEL GRID":GOTO 1790
1850 IF Y2<=(LLY) OR Y2>=(LLY+DELY*(NR)) THEN PRINT "  BEGINNING POINT Y COORDINATE IS OUTSIDE THE
MODEL GRID":GOTO 1790
1860 IF Y3<=(LLY) OR Y3>=(LLY+DELY*(NR)) THEN PRINT "  END POINT Y COORDINATE IS OUTSIDE THE MODEL
GRID":GOTO 1790
1870 IF X3<=(LLX) OR X3>=(LLX+DELX*(NC)) THEN PRINT "  END POINT X COORDINATE IS OUTSIDE THE MODEL
GRID":GOTO 1790
     IZ=INT((X2-LLX)/DELX+1.0)
     JZ=INT((Y2-LLY)/DELY+1.0)
     IF Z2 < BOT(IZ,JZ,1) OR Z2 > (BOT(IZ,JZ,NL)+THICK(IZ,JZ,NL)) THEN PRINT "  1ST Z COORDINATE
OUTSIDE MODEL":GOTO 1790
```

```
     IZ=INT((X3-LLX)/DELX+1.0)
     JZ=INT((Y3-LLY)/DELY+1.0)
     IF Z3 < BOT(IZ,JZ,1) OR Z3 > (BOT(IZ,JZ,NL)+THICK(IZ,JZ,NL)) THEN PRINT "  2ND Z COORDINATE
OUTSIDE MODEL":GOTO 1790
     PRINT "      Enter the letter C for CONTINUOUS POLLUTION--otherwise a <RETURN>"
     INPUT "      or any other character will assume a SLUG input";A$
     IF A$="C" OR A$="c" THEN PRINT "      YOU ARE NOW IN THE CONTINUOUS MODE FOR THIS SOURCE"
     PRINT:IF A$="C" OR A$="c" THEN LPRINT "YOU ARE NOW IN THE CONTINUOUS MODE FOR THIS SOURCE"
1880 INPUT "        NUMBER OF PARTICLES ";M
1890 PRINT ""
1900 LPRINT "              NUMBER OF PARTICLES = ";M
1910 INPUT "DO YOU WANT TO REDO THIS SCREEN? (Y-YES, <RETURN>-NO)";OK$
1920 IF OK$="Y" OR OK$="y" THEN 1330
1930 IF M<=0 THEN 1330
1940 X = X2:Y=Y2:Z=Z2
1950 REM THE NEXT GOSUB IS REQUIRED TO INSERT THE PARTICLE.
1960 GOSUB 2100
1970 IF M>1 THEN X = X3:Y=Y3:Z=Z3:GOSUB 2100
2000 IF M=2 THEN 1330
2010 FOR I = 3 TO M
2020 X = X2+(I-2)*(X3-X2)/(M-1)
2030 Y = Y2+(I-2)*(Y3-Y2)/(M-1)
     Z = Z2+(I-2)*(Z3-Z2)/(M-1)
2040 GOSUB 2100:REM ADD A PARTICLE
2050 NEXT I
2060 LPRINT "       SYSTEM PARTICLES = ";NP
2070 L=L+1
2080 GOTO 1330:REM RETURN TO BEGINNING OF PARTICLE SUBROUTINE
     REM *************************************
2090 REM SUBROUTINE TO ADD A PARTICLES
     REM ************************************
2100 NP=NP+1
2110 X(NP)=X
2120 Y(NP)=Y
     Z(NP)=Z
     IF A$="C" OR A$="c" THEN SWITCH%(NP)=1 ELSE SWITCH%(NP)=0
2130 RETURN
2140 REM END OF ADD A PARTICLE SECTION
     REM *****************************************
2150 PRINT:REM                                       INITIATE PARTICLES IN SPHERE
2160 LPRINT "        PARTICLES ON A SPHERE"
2170 PRINT "      SPHERE NUMBER ";C+1
2180 PRINT
2190 LPRINT "            SPHERE NUMBER ";C+1
2200 INPUT "      ENTER SPHERE CENTER COORDINATES (X,Y,Z)";X4,Y4,Z4
2210 LPRINT "            SPHERE CENTER COORDINATES (X,Y,Z) = ";X4;",";Y4;",";Z4;" FT"
2230 INPUT "      ENTER SPHERE RADIUS (FT) ";R
2240 LPRINT "            SPHERE RADIUS = ";R;" FT"
2242 IF (X4-R)<(LLX) OR (X4+R)>(LLX+DELX*(NC)) THEN PRINT "  DO NOT ENTER PARTICLES OUTSIDE FLOW
MODEL GRID - BAD X":GOTO 2200
2244 IF (Y4-R)<(LLY) OR (Y4+R)>(LLY+DELY*(NR)) THEN PRINT "  DO NOT ENTER PARTICLES OUTSIDE FLOW
MODEL GRID - BAD Y":GOTO 2200
     IZ=INT((X4-LLX)/DELX+1.0)
     JZ=INT((Y4-LLY)/DELY+1.0)
     IF Z4-R < BOT(IZ,JZ,1) OR Z4+R > (BOT(IZ,JZ,NL)+THICK(IZ,JZ,NL)) THEN PRINT "  DO NOT ENTER
PARTICLES OUTSIDE FLOW MODEL GRID - BAD Z ":GOTO 2200
     PRINT "      Enter the letter C for CONTINUOUS POLLUTION--otherwise a <RETURN>"
     INPUT "      or any other character will assume a SLUG input";A$
     IF A$="C" OR A$="c" THEN PRINT "      YOU ARE NOW IN THE CONTINUOUS MODE FOR THIS SOURCE"
     PRINT:IF A$="C" OR A$="c" THEN LPRINT "YOU ARE NOW IN THE CONTINUOUS MODE FOR THIS SOURCE"
2250 INPUT "      ENTER NUMBER OF PARTICLES ";M
2260 LPRINT "          NUMBER OF PARTICLES = ";M
2270 PRINT ""
2280 INPUT "DO YOU WANT TO REDO THIS SCREEN? (Y-YES, <RETURN>-NO)";OK$
2290 IF OK$="Y" OR OK$="y" THEN 1330
2300 ANG=0
2310 AG=2*3.14159/SQR(M)
2320 FOR I=1 TO INT(SQR(M))
2330 TH=AG*I+ANG
       FOR II=1 TO M/INT(SQR(M))
       PHI=2*3.14159-TH
       X = X4+R*SIN(PHI)*COS(TH)
       Y = Y4+R*SIN(TH)*SIN(PHI)
       Z = Z4+R*COS(PHI)
2370   GOSUB 2100:REM GO TO SUBROUTINE TO INSERT A PARTICLE.
       NEXT II
2380 NEXT I
2390 C = C+1
     PRINT "       SPHERE DONE"
     REM ********************************************
2400 REM   ROUTINE TO GENERATE PARTICLES FROM A VERTICALLY ORIENTED CYLINDER
     REM *******************************************
     LPRINT "       PARTICLES ON A CYLINDER"
     PRINT "      CYLINDER NUMBER ";C+1
     PRINT
     LPRINT "          CYLINDER NUMBER ";C+1
2410 INPUT "      ENTER CYLINDER CENTER COORDINATES (X,Y)";X4,Y4
     LPRINT "          CYLINDER CENTER COORDINATES (X,Y) = ";X4;",";Y4;"  FT"
     INPUT "      ENTER CYLINDER RADIUS (FT) ";R
     LPRINT "          CYLINDER RADIUS = ";R;" FT"
     INPUT "      ENTER BOTTOM AND TOP  ELEVATION OF CYLINDER (FT) ";Z3,Z4
     LPRINT "           BOTTOM AND TOP ELEVATION OF CYLINDER (FT) ";Z3;",";Z4;" FT"
     IF (X4-R)<(LLX) OR (X4+R)>(LLX+DELX*(NC)) THEN PRINT "  DO NOT ENTER PARTICLES OUTSIDE FLOW
MODEL GRID - BAD X":GOTO 2410
     IF (Y4-R)<(LLY) OR (Y4+R)>(LLY+DELY*(NR)) THEN PRINT "  DO NOT ENTER PARTICLES OUTSIDE FLOW
MODEL GRID - BAD Y":GOTO 2410
     IZ=INT((X4-LLX)/DELX+1.0)
     JZ=INT((Y4-LLY)/DELY+1.0)
     IF Z4 < BOT(IZ,JZ,1) OR Z4 > (BOT(IZ,JZ,NL)+THICK(IZ,JZ,NL)) THEN PRINT "  DO NOT ENTER
PARTICLES OUTSIDE FLOW MODEL GRID - BAD TOP ":GOTO 2410
     IF Z3 < BOT(IZ,JZ,1) OR Z3 > (BOT(IZ,JZ,NL)+THICK(IZ,JZ,NL)) THEN PRINT "  DO NOT ENTER
PARTICLES OUTSIDE FLOW MODEL GRID - BAD BOT ":GOTO 2410
     IF Z4<Z3 THEN PRINT "  TOP OF CYLINDER MUST BE HIGHER THAN BOTTOM":GOTO 2410
     PRINT "      Enter the letter C for CONTINUOUS POLLUTION--otherwise a <RETURN>"
     INPUT "      or any other character will assume a SLUG input";A$
     IF A$="C" OR A$="c" THEN PRINT "      YOU ARE NOW IN THE CONTINUOUS MODE FOR THIS SOURCE"
     PRINT:IF A$="C" OR A$="c" THEN LPRINT "YOU ARE NOW IN THE CONTINUOUS MODE FOR THIS SOURCE"
     INPUT "      ENTER NUMBER OF PARTICLES ";M
     LPRINT "          NUMBER OF PARTICLES = ";M
     PRINT ""
     INPUT "DO YOU WANT TO REDO THIS SCREEN? (Y-YES, <RETURN>-NO)";OK$
     IF OK$="Y" OR OK$="y" THEN 2410
     ANG=0
     AG=2*3.14159/M
     FOR I=1 TO M
        TH=AG*I+ANG
        X = X4+R*SIN(TH)
        Y = Y4+R*COS(TH)
        Z = Z3+(Z4-Z3)*RND
        GOSUB 2100:REM GO TO SUBROUTINE TO INSERT A PARTICLE.
     NEXT I
     C = C+1
     PRINT "       CYLINDER DONE"
     GOTO 1330:REM RETURN TO BEGINNING OF PARTICLE SUBROUTINE
2420 PRINT
2430 LPRINT
2440 LPRINT "         TOTAL SYSTEM PARTICLES = ";NP
2450 LPRINT "\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\///////////////////////////////"
2460 PRINT "       TOTAL SYSTEM PARTICLES = ";NP
2470 PRINT "\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\///////////////////////////////"
2480 RETURN
2490 REM MAKE SURE ALL DATA NECESSARY TO MAKE A MOVE HAS BEEN SPECIFIED.
2510 IF G$ <> "B" THEN PRINT "PLEASE CHOOSE COEFFICIENTS MENU ITEM NOW!!!":GOSUB 13100:RETURN
2515 IF NENDTIME>0 AND NPER=NENDTIME AND T2=ENDTIME(NPER) THEN PRINT "  ALL VELOCITY FILES USED,
SIMULATION COMPLETE":GOSUB 13100:RETURN
2520 IF NENDTIME>0 AND T2=ENDTIME(NPER) THEN PRINT "  READ IN THE NEXT VELOCITY VECTOR FILE
NOW!":GOSUB 13100:RETURN
2530 IF NP=0 THEN PRINT "DO SOMETHING ELSE---THERE ARE NO PARTICLES TO MOVE!!":GOSUB 13100:RETURN
2542 IF ASPECT<=0 THEN PRINT "  YOU DID NOT ENTER AN ASPECT RATIO FOR THE SCREEN ":GOSUB
13100:RETURN
2544 IF (SCALEDEF<=0 AND TEMPSCR=0) OR (SCALEX<=0 AND TEMPSCR=1) THEN PRINT "  YOU DID NOT ENTER A
DEFAULT SCREEN WIDTH ":GOSUB 13100:RETURN
2550 PRINT "        PRESENT SIMULATION TIME (DAYS) = ";T2
2560 LPRINT "      PRESENT SIMULATION TIME = ";T2;"  DAYS"
2570 IF NENDTIME>0 THEN PRINT "  TIME REMAINING FOR THIS VELOCITY FILE IS ";(ENDTIME(NPER)-T2);"
DAYS"
2580 PRINT "  ENTER INCREMENTAL SIMULATION TIME (DAYS) [";DELTA;"]";
2582 INPUT TEMP$
2584 IF LEN(TEMP$)>0 THEN DELTA=VAL(TEMP$)
2590 IF DELTA =<0 THEN PRINT "  DELTA MUST BE GREATER THAN ZERO!!!---REDO THIS INPUT NOW PLEASE."
2600 IF DELTA =<0 THEN GOTO 2580
2602 IF NENDTIME<=0 THEN GOTO 2630:REM ASSUME STEADY STATE IF NUMBER OF PERIODS HAVE NOT BEEN
ENTERED
2604 IF (DELTA+T2)>ENDTIME(NPER) THEN PRINT "  TIME STEP TOO LARGE, ENTER A SMALLER TIME
STEP":DELTA=ENDTIME(NPER)-T2:GOTO 2580
2630 LPRINT "        INCREMENTAL SIMULATION TIME = ";DELTA;" DAYS"
2640 PRINT " HOW OFTEN DO YOU WANT TO COMPUTE VELOCITY VECTORS"
     PRINT "  ENTER MAXIMUM HORIZONTAL MOVE (FT) [";DMAX;"] ";
2642 INPUT TEMP$
2644 IF LEN(TEMP$)>0 THEN DMAX=VAL(TEMP$)
     PRINT "  ENTER MAXIMUM VERTICAL MOVE (FT) [";ZMAX;"] ";
     INPUT TEMP$
     IF LEN(TEMP$)>0 THEN ZMAX=VAL(TEMP$)
2650 PRINT " DMAX = ";DMAX;" FT   ZMAX = ";ZMAX;" FT"
2660 LPRINT "DMAX = ";DMAX;" FT   ZMAX = ";ZMAX;" FT"
2670 PRINT
2680 LPRINT
2690 REM         GO TO SUBROUTINE TO SET UP SCREEN AND SPECIAL FEATURES ON SCREEN
2700 GOSUB 7520:REM CALL SPECIAL FEATURES GRAPHICS SUBROUTINE
2710 IF NP=0 THEN RETURN
2720 REM LOOP TO ZERO OUT PURGE WELL CONCENTRATION ARRAY.
2730 FOR K = 1 TO NS
2740 CONC(K)=0!
2750 NEXT K
2770 K=0:REM INITIALIZE PARTICLE COUNTER
     ZZZ=0:REM INITIALIZE 1ST ORDER DECAY CHECK
2780 REM START LOOP ON PARTICLE MOVE HERE. (SEE LOOP TERMINUS AT 3800 OR 3860)
2790 K=K+1
2800 LOCATE 1,35:PRINT K
2810 REM LOCATE 1,1
2820 REM EXAMINE SWITCH TO DETECT IF CONTINUOUS POLLUTION OR NOT.
2830 IF SWITCH%(K)=1 THEN D=DELTA*RND ELSE D=DELTA
     IF LAMDBA>=1E10 THEN GOTO 2880:REM SKIP FIRST ORDER DECAY TO SAVE TIME IF DECAY IF CONSERVATIVE
POLLUTION
     ZZZ=ZZZ+(1-.5^(D/LAMDBA/365)):REM FIRST ORDER DECAY IMPLEMENTED IN THESE STATEMENTS
     IF ZZZ=> 1 THEN GOTO 2832 ELSE GOTO 2880
2832 X(K)=X(NP)
     Y(K)=Y(NP)
     Z(K)=Z(NP)
     SWITCH%(K)=SWITCH%(NP)
     NP=NP-1
     ZZZ=ZZZ-1.0
     IF K>NP THEN 3660
     REM IF LAST PARTICLE DECAYS JUMP TO END OF ROUTINE
     GOTO 2830:REM LOOP BACK FOR DECAY ON NEXT PARTICLE
2840 REM +++++++++++++++++++++++++++++++++++++++++++++
2850 REM +  WHEN ALL PARTICLES ARE PROCESSED, THIS  +
2860 REM +  NEXT STATEMENT RETURNS YOU TO THE MENU  +
2870 REM +++++++++++++++++++++++++++++++++++++++++++++
2880 IF K>NP THEN T2=T2+DELTA:GOTO 3660
2890 XP=(X(K)-LLX)/DELX+.5
2900 YP=(Y(K)-LLY)/DELY+.5
2910 VX=.00001
2920 VY=.00001
     VZ=1E-10
2930 I=INT(XP)
2940 J=INT(YP)
     IZ=INT(XP+.5)
     JZ=INT(YP+.5)
     FOR KLK=1 TO NL-1:REM FIND LAYER OF PARTICLE AND INTERPOLATE THE VERTICAL VELOCITY
        IF Z(K)>= BOT(IZ,JZ,KLK) AND Z(K)<TOP(IZ,JZ,KLK) THEN
             LL=KLK
             AZ=(Z(K)-BOT(IZ,JZ,KLK))/(TOP(IZ,JZ,KLK)-BOT(IZ,JZ,KLK))
             V3=(AZ*VK(IZ,JZ,KLK)+(1-AZ)*VK(IZ,JZ,KLK-1))/E
             GOTO 2948
        END IF
        IF Z(K)>= TOP(IZ,JZ,KLK) AND Z(K)<BOT(IZ,JZ,KLK+1) THEN
             LL=KLK+NL
             V3=VK(IZ,JZ,KLK)/E
             GOTO 2948
        END IF
     NEXT KLK
     IF Z(K)>=BOT(IZ,JZ,NL) THEN
        LL=NL
        AZ=(Z(K)-BOT(IZ,JZ,NL))/THICK(IZ,JZ,NL)
        V3=(AZ*VK(IZ,JZ,NL)+(1-AZ)*VK(IZ,JZ,NL-1))/E
     END IF
2948 IF LL>NL THEN V1=0:V2=0:GOTO 2990:REM IF IN CONFINING LAYER, THEN SKIP HORIZONTAL INTERPOLATION
2950 AX=XP-INT(XP)
2960 AY=YP-INT(YP)
2970 V1=((1-AY)*VI(I,J,LL)+AY*VI(I,J+1,LL))/E:REM HORIZONTAL VELOCITY INTERPOLATION
2980 V2=((1-AX)*VJ(I,J,LL)+AX*VJ(I+1,J,LL))/E
2990 VX=VX+V1
3000 VY=VY+V2
     VX=VX/RETARD(LL):REM APPLY RETARDATION COEFFICIENTS TO VELOCITY
     VY=VY/RETARD(LL)
     VZ=(VZ+V3)/RETARD(LL)
3040 IF(D-ABS(DMAX/VX))<0 THEN 3070
3050 F=ABS(DMAX/VX)
3060 GOTO 3080
3070 F=D
3080 IF F-ABS(DMAX/VY)<0 THEN 3092
3090 F=ABS(DMAX/VY)
3092 IF(F-ABS(ZMAX/VZ))<0 THEN 3100
     F=ABS(ZMAX/VZ)
3100 D=D-F:REM CALCULATE REMAINING TIME IN MOVE
3110 DX=VX*F
3120 DY=VY*F
     DZ=VZ*F
3130 PHI=ATN(DY/DX)
3140 DD=SQR(DX*DX+DY*DY)
     DDX=SQR(DX*DX+DY*DY+DZ*DZ)
3210 RN=0
3220 GOTO 3280
     REM ********************************
     REM  NORMALLY DISTRIBUTED RANDOM NUMBER GENERATOR
     REM ********************************
3230 RN=-6
3240 FOR U%=1 TO 12
3250     RN=RN+RND
3260 NEXT U%
3270 RETURN
     REM *********************************
3280 IF DL>0 THEN GOSUB 3230
3290 RL=(SQR(2*DL*DD)/DD)*RN
3300 IF DT>0 THEN GOSUB 3230
3310 RT=(SQR(2*DT*DD)/DD)*RN
3312 IF DV>0 THEN GOSUB 3230
3314 RV=(SQR(2*DV*DDX))*RN
3340 OLDX = X(K)
```

```
3350 OLDY = Y(K)
     OLDZ = Z(K)
3360 IF I=>NC THEN XP=NC+.5:GOTO 3380
3370 XP=XP+(DX+RL*DX+RT*DY)/DELX
3380 IF J=>NR THEN YP=NR+.5:GOTO 3400
3390 YP=YP+(DY+RL*DY-RT*DX)/DELY
3400 IF XP<=1! THEN XP=1.01
3410 IF XP>=NC-.01 THEN XP=NC-.01
3420 IF YP>=NR-.01 THEN YP=NR-.01
3430 IF YP<=1! THEN YP=1.01
     Z(K)=Z(K)+(DZ+RV):REM MOVE PARTICLE IN Z DIMENSION
     REM     CHECK TO SEE IF PARTICLE HAS BEEN MOVED ABOVE WATER TABLE
     IF Z(K)>(BOT(IZ,JZ,NL)+THICK(IZ,JZ,NL)) THEN Z(K)=BOT(IZ,JZ,NL)+THICK(IZ,JZ,NL)-.001
     REM     CHECK TO SEE IF PARTICLE HAS BEEN MOVED BELOW BOTTOM
     IF Z(K)<BOT(IZ,JZ,1) THEN Z(K)=BOT(IZ,JZ,1)+.001
3432 III=INT(XP):REM THESE 4 STATEMENTS MOVE THE PARTICLE OUT OF A ZERO VELOCITY FIELD IF THEY
HAPPAN TO GET THERE
3434 JJJ=INT(YP)
     IF LL>NL THEN GOTO 3440:REM SKIP THIS CHECK IF PARTICLE IN CONFINING LAYER
3436 IF VI(III,JJJ,LL)=0 AND VJ(III,JJJ,LL)=0 AND VI(III,JJJ+1,LL)=0 AND VJ(III+1,JJJ,LL)=0 THEN
GOSUB 10400:REM MOVE PARTICLE OUT OF ZERO VELOCITY GRID
3440 NEWX = (XP-.5)*DELX+LLX:REM TRANSLATE PARTICLE POSITION BACK TO REAL COORDINATES
3450 NEWY = (YP-.5)*DELY+LLY
3460 X(K)=NEWX
3470 Y(K)=NEWY
3492 IF XX$="Y" OR XX$="y" THEN IPEN=2 ELSE IPEN=1:REM CHECK FOR COLOR MONITOR AND SET COLOR
      IF MAPTYPE$="A" THEN SCREENX=X(K):SCREENY=Y(K):OLDSCRX=OLDX:OLDSCRY=OLDY
      IF MAPTYPE$="B" THEN SCREENX=X(K):SCREENY=Z(K):OLDSCRX=OLDX:OLDSCRY=OLDZ
      IF MAPTYPE$="C" THEN SCREENX=Y(K):SCREENY=Z(K):OLDSCRX=OLDY:OLDSCRY=OLDZ
3500 LINE (OLDSCRX,OLDSCRY)-(SCREENX,SCREENY),IPEN:REM WRITE PARTICLE TRACE TO SCREEN
     IF XX$="Y" OR XX$="y" THEN PSET (SCREENX,SCREENY),3:REM PUT THE HEAD ON THE SPERM
    REM
    REM THIS CODE REMOVES THE PARTICLE IF NEAR A SINK
    REM
    IF LL>NL THEN GOTO 3580:REM IF PARTICLE IS IN CONFINING LAYER SKIP THE SINK CAPTURE
3510 FOR L%=1 TO NS
3520      II=INT((X1(L%)-LLX)/DELX+1.0):REM FIND ROW AND COLUMN OF SINK
3525      JJ=INT((Y1(L%)-LLY)/DELY+1.0)
          KK=Z1(L%):REM FIND LAYER OF SINK
3540      IF X(K)=X1(L%) AND Y(K)=Y1(L%) AND Z(K)<TOP(II,JJ,KK) AND Z(K)>=BOT(II,JJ,KK) THEN GOTO
3600:REM REMOVE PARTICLE
3530      R3=SQR((X(K)-X1(L%))*(X(K)-X1(L%))+(Y(K)-Y1(L%))*(Y(K)-Y1(L%)))
3555 REM  CHECK IF PARTICLE REACHES SINK DURING TIME STEP
          CAPTUREMAX=DMAX:REM PARTICLE MUST BE WITHIN MAX MOVE OF WELL TO BE CAPTURED
          BOTCAPTURE=BOT(II,JJ,KK):REM SET BOTTOM OF LAYER AS CAPTURE ZONE - ASSUMES FULL
PENETRATION
3560      IF KK<NL THEN            :REM  SET TOP OF LAYER AS CAPTURE ZONE
3561         TOPCAPTURE=TOP(II,JJ,KK)
          ELSE
3562         TOPCAPTURE=BOT(II,JJ,KK)+THICK(II,JJ,KK)
3563         IF TOP(II,JJ,NL)<>0 THEN          :REM IF RIVER IN TOP LAYER THEN MODIFY CAPTURE ZONE
3564            R4=QSUM(L%)*F/DELX/DELY/7.48/E
                BOTCAPTURE=TOP(II,JJ,NL)-R4
3565            CAPTUREMAX=SQR(DELX*DELY)
```

```
               IF BOTCAPTURE<BOT(II,JJ,NL) THEN BOTCAPTURE=BOT(II,JJ,NL)
            END IF
         END IF
3566     IF QSUM(L%)<=0 THEN GOTO 3570:REM SKIP CAPTURE IF INJECTION WELL
3567     IF R3<SQR(QSUM(L%)*DELTA/(3.141592*THICK(II,JJ,KK)*E*7.48)) AND R3<CAPTUREMAX  AND
Z(K)>=BOTCAPTURE AND Z(K)<=TOPCAPTURE THEN GOTO 3600:REM  REMOVE PARTICLE IN SINK
3570 NEXT L%
3580 IF D>0 THEN 2890:REM LOOP BACK TO FINISH MOVE FOR REMAINING TIME INCREMENT
3590 GOTO 2790:REM LOOP BACK FOR NEXT PARTICLE
3600 rem LPRINT "PARTICLE EXITED AT SINK NUMBER ";L
3610 CONC(L%)=CONC(L%)+(PM*119872!)/(QSUM(L%)*DELTA):REM CALC CONCENTRATION  ENTERING SINK
3620 X(K)=X(NP):REM    REMOVE PARTICLE BY
3630 Y(K)=Y(NP):REM    PUTTING LAST PARTICLE IN POSITION
     Z(K)=Z(NP):REM    OF REMOVED PARTICLE
3640 NP=NP-1
3650 GOTO 2830:REM LOOP BACK TO NEXT PARTICLE
3660 REM END OF PARTICLE MOVE AND BEGIN PRINTOUT OF SINK CONCENTRATIONS.
3670 LPRINT "NP=";NP
3690 FOR I= 1 TO NS
3700 IF QSUM(I)>0 AND CONC(I)>0 THEN LPRINT "CONCENTRATION IN PUMPED WELL NUMBER ";I;", IN PPM, IS=
";CONC(I)
3710 IF QSUM(I)>0 AND CONC(I)>0 THEN LPRINT "MASS EXITING IN PUMPED WELL NUMBER ";I;", IN LBS, IS=
";CONC(I)*QSUM(I)*DELTA/119872
3715 NEXT I
3720 LPRINT:LPRINT:LPRINT:LPRINT
3790 LOCATE 1,1
3800 W$=INKEY$:REM END OF MOVE SUBROUTINE, KEEP SCREEN UNTIL KEY IS PRESSED
3810 IF W$="S" OR W$="s" THEN GOSUB 7110:REM SAVE SCREEN
3820 IF W$="" THEN 3800
3830 FOR I=1 TO NP:REM SET ALL PARTICLES TO SLUG TYPE AT END OF MOVE
3840 SWITCH%(I)=0
3850 NEXT I
3860 SCREEN 2
3870 PRINT:PRINT:SCREEN 0:IF XX$="Y" OR XX$="y" THEN COLOR 15,1,1 :CLS ELSE CLS
3880 REM +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
3890 REM +  BUG OUT TO MAIN MENU AS THIS IS THE END OF SUBROUTINE MOVE.  +
3900 REM +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
3910 RETURN
3940 REM ++++++++++++++++++++++++++++++++
3950 REM +      MAP PARTICLES OR        +
3960 REM +      CONCENTRATION           +
3970 REM ++++++++++++++++++++++++++++++++
3980 F$="A":REM SET MAP FLAG
3990 LN = 10:REM FIXED NUMBER OF ROWS IN SCREEN MAP, SCREEN MAPPED 14 BY 10 TO MATCH TYPICAL SCREEN
SIZE
3992 WD = 14:REM FIXED NUMBER OF COLUMNS IN SCREEN MAP
3994 IF ASPECT<=0 THEN PRINT "  YOU DID NOT ENTER AN ASPECT RATIO FOR THE SCREEN ":GOSUB
13100:RETURN
3996 IF (SCALEDEF<=0 AND TEMPSCR=0) OR (SCALEX<=0 AND TEMPSCR=1) THEN PRINT "  YOU DID NOT ENTER A
DEFAULT SCREEN WIDTH ":GOSUB 13100:RETURN
4010 GOSUB 7520:REM GRAPHICS ROUTINE TO DEFINE SCREEN COORDINATES
4020 CDX=(I2!-I1!)/WD:REM J2! = J1! + ((WD - 1) * CDY)  DEFINE GRID SPACING
4030 CDY=(J2!-J1!)/LN:REM I2! = I1! + ((LN - 1) * CDX)
4040 SCREEN 0:WIDTH 80:IF XX$="Y" OR XX$="y" THEN COLOR 15,1,1:CLS ELSE CLS
```

```
4050 PRINT:PRINT:PRINT
4070 PRINT "WHICH TYPE OF MAP DO YOU WANT?:"
4080 PRINT SPC( 5);"ENTER P FOR PARTICLE MAP"
4090 PRINT SPC( 5);"ENTER C FOR CONCENTRATION MAP"
4100 PRINT
4110 INPUT "WHICH ONE......";C$
4120 IF C$<> "C" AND C$<>"c" THEN 4170
4130 INPUT "ENTER SCALE FACTOR FOR CONCENTRATION MAP (T-PPT, M-PPM, B-PPB)";CC$
4140 IF CC$ = "T" OR CC$="t" THEN PPP = .001
4150 IF CC$ = "M" OR CC$="m" THEN PPP = 1
4160 IF CC$ = "B" OR CC$="b" THEN PPP = 1000
4170 IF MAPTYPE$="A" THEN
         PRINT "  ENTER LAYER OF MODEL TO MAP, FOR ALL PARTICLES IN ALL LAYERS VISIBLE,"
         INPUT "  ENTER ZERO ";LAYER
     IF LAYER >NL+(NL-1) THEN GOTO 4170:REM CHECK SELECTION
     END IF
     IF MAPTYPE$="B" THEN
        PRINT "  DO YOU WANT ALL PARTICLES VISIBLE OR ONLY THOSE IN THE"
        INPUT "   SELECTED ROW (A/S) ";Z1$
        IF Z1$="A" OR Z1$="a" THEN LAYER=0 ELSE LAYER=JROWXS
     END IF
     IF MAPTYPE$="C" THEN
        PRINT "  DO YOU WANT ALL PARTICLES VISIBLE OR ONLY THOSE IN THE"
        INPUT "   SELECTED COLUMN (A/S) ";Z1$
        IF Z1$="A" OR Z1$="a" THEN LAYER=0 ELSE LAYER=ICOLXS
     END IF
4172 GOSUB 4220:REM PRINT THE MAP
4180 INPUT "DO A SCREEN PRINT NOW OR PRESS <RETURN> TO GO BACK TO THE MENU.";A$
4210 RETURN
4220 REM   +++++++++++++++++
4230 REM   + PRINT THE MAP +
4240 REM   +++++++++++++++++
4250 FOR I = 1 TO WD:REM INITIALIZE THE ARRAY
4260 FOR J = 1 TO LN
4270 NMAP(I,J)=0!
4280 NEXT J
4290 NEXT I
4300 FOR MM% = 1 TO NP:REM  TEST EACH PARTICLE TO SEE WHERE IT FALLS IN THE MAPPED AREA
      IF SWITCH%(MM%)=1 THEN GOTO 4370:REM SKIP PARTICLE IF NOT CONTINUOUS
      IF MAPTYPE$="A" THEN
        SCREENX=X(MM%)
        SCREENY=Y(MM%)
        IF LAYER =0 THEN 4310:REM SKIP LAYER TESTING IF ALL PARTICLES ARE TO BE INCLUDED
        IZ=INT((X(MM%)-LLX)/DELX+1.0):REM FIN COLUMN OF PARTICLE
        JZ=INT((Y(MM%)-LLY)/DELY+1.0):REM FIND ROW OF PARTICLE
        IF Z(MM%)>=BOT(IZ,JZ,NL) THEN LL=NL:GOTO 4302:REM  FIND LAYER OF PARTICLE
        FOR KLK=1 TO NL-1:REM FIND LAYER OF PARTICLE
           IF Z(MM%)>= BOT(IZ,JZ,KLK) AND Z(MM%)<TOP(IZ,JZ,KLK) THEN LL=KLK:GOTO 4302
           IF Z(MM%)>= TOP(IZ,JZ,KLK) AND Z(MM%)<BOT(IZ,JZ,KLK+1) THEN LL=KLK+NL:GOTO 4302
        NEXT KLK
4302    IF LL<>LAYER THEN 4370:REM SKIP IF IN WRONG LAYER
4304 END IF
     IF MAPTYPE$="B" THEN
        IF LAYER<>0 THEN
```

```
          JZ=INT((Y(MM%)-LLY)/DELY+1.0)
          IF JZ<>LAYER THEN 4370:REM SKIP IF IN WRONG ROW
        END IF
        SCREENX=X(MM%)
        SCREENY=Z(MM%)
     END IF
     IF MAPTYPE$="C" THEN
        IF LAYER<>0 THEN
          IZ=INT((X(MM%)-LLX)/DELX+1.0)
          IF IZ<>LAYER THEN 4370:REM SKIP IF IN WRONG ROW
        END IF
        SCREENX=Y(MM%)
        SCREENY=Z(MM%)
     END IF
4310 I = INT(1.5 + ((SCREENX-I1!)/CDX))
4320 J = INT(1.5 + ((J2!-SCREENY)/CDY))
4330 IF I<1 OR I>WD THEN GOTO 4370
4340 IF J<1 OR J>LN THEN GOTO 4370
4360 NMAP(I,J) = NMAP(I,J)+1
4370 NEXT MM%
4380 FOR IIS=1 TO NS:REM   TEST EACH SINK TO SEE WHERE IT IS AND ASSIGN NEGATIVE NUMBERS TO SINK
LOCATIONS
     IF MAPTYPE$="A" THEN
         SCREENX=X1(IIS)
         SCREENY=Y1(IIS)
         IF LAYER<>Z1(IIS) AND LAYER<>0 THEN GOTO 4460:REM SKIP THIS SINK IF DOING LAYER MAP
     END IF
     IF MAPTYPE$="B" THEN
         SCREENX=X1(IIS)
         IZ=INT((X1(IIS)-LLX)/DELX+1.0):REM FIN COLUMN OF WELL
         JZ=INT((Y1(IIS)-LLY)/DELY+1.0):REM FIND ROW OF WELL
         SCREENY=BOT(IZ,JZ,INT(Z1(IIS)))
         IF LAYER<>0 AND LAYER<>JZ THEN GOTO 4460:REM SKIP THIS SINK IF NOT IN ROW
     END IF
     IF MAPTYPE$="C" THEN
         SCREENX=Y1(IIS)
         IZ=INT((X1(IIS)-LLX)/DELX+1.0):REM FIN COLUMN OF WELL
         JZ=INT((Y1(IIS)-LLY)/DELY+1.0):REM FIND ROW OF WELL
         SCREENY=BOT(IZ,JZ,INT(Z1(IIS)))
         IF LAYER<>0 AND LAYER<>IZ THEN GOTO 4460:REM SKIP THIS SINK IF NOT IN COLUMN
     END IF
4390 I=INT((SCREENX-I1!)/CDX+1.5)
4400 J=INT((J2!-SCREENY)/CDY+1.5)
4410 IF I<1 OR I>WD THEN 4460
4420 IF J<1 OR J>LN THEN 4460
4430 IF QSUM(IIS)>0 THEN NMAP(I,J)=-1:REM SET PUMPING WELL SITE
4450 IF QSUM(IIS)<0 THEN NMAP(I,J)=-2:REM SET INJECTION WELL SITE
4460 NEXT IIS
4470 IF C$<> "C" AND C$<>"c" THEN GOTO 4600
4480 FOR I = 1 TO WD:REM COMPUTE THICKNESS AND CALC CONCENTRATION
4490 FOR J = 1 TO LN
4500 XX=(I-1)*CDX+I1!:REM FIND INDICES OF THICKNESS GRID FOR MIDDLE OF THIS MAP GRID
4510 YY=J2!-(J-1)*CDY
4540 II=INT((XX-LLX)/DELX+1.0)
```

```
4550 JJ=INT((YY-LLY)/DELY+1.0)
     IF MAPTYPE$="A" THEN
         IF II<1 OR II>NC THEN GOTO 4580:REM  IF OUTSIDE GRID TO TO END OF LOOP
         IF JJ<1 OR JJ>NR THEN GOTO 4580
         TEMPTHCK=0
         IF LAYER=0 THEN
           FOR K=1 TO NL-1
            TEMPTHCK=TEMPTHCK+THICK(II,JJ,K)*RETARD(K)
            TEMPTHCK=TEMPTHCK+(BOT(II,JJ,K+1)-TOP(II,JJ,K))*RETARD(K+NL)
           NEXT K
           TEMPTHCK=TEMPTHCK+THICK(II,JJ,NL)*RETARD(NL)
         ELSE
           IF LAYER<=NL THEN TEMPTHCK=THICK(II,JJ,LAYER) ELSE TEMPTHCK=BOT(II,JJ,LAYER-NL+1)-
TOP(II,JJ,LAYER-NL)
         END IF
     END IF
     IF MAPTYPE$="B" THEN
         IF II<1 OR II>NC THEN GOTO 4580:REM  IF OUTSIDE GRID TO TO END OF LOOP
         FOR K=1 TO NL-1
             IF YY>=BOT(II,JROWXS,K) AND YY<TOP(II,JROWXS,K) THEN RD=RETARD(K)
             IF YY>=TOP(II,JROWXS,K) AND YY<BOT(II,JROWXS,K+1) THEN RD=RETARD(K+NL)
         NEXT K
         IF YY>BOT(II,JROWXS,NL) THEN RD=RETARD(NL)
         IF LAYER=0 THEN TEMPTHCK=NR*DELY*RD ELSE TEMPTHCK=DELY
     END IF
     IF MAPTYPE$="C" THEN
         II=INT((XX-LLY)/DELY+1.0)
         IF II<1 OR II>NR THEN GOTO 4580
         FOR K=1 TO NL-1
             IF YY>=BOT(ICOLXS,II,K) AND YY<TOP(ICOLXS,II,K) THEN RD=RETARD(K)
             IF YY>=TOP(ICOLXS,II,K) AND YY<BOT(ICOLXS,II,K+1) THEN RD=RETARD(K+NL)
         NEXT K
         IF YY>BOT(ICOLXS,II,NL) THEN RD=RETARD(NL)
         IF LAYER=0 THEN TEMPTHCK=NC*DELX*RD ELSE TEMPTHCK=DELX
     END IF
4560 IF NMAP(I,J)<0 THEN 4580
4570 NMAP(I,J)=(16030*NMAP(I,J)*PM)/(E*CDX*CDY*TEMPTHCK)*PPP:REM CONCENTRATION CALCULATION
4580 NEXT J
4590 NEXT I
4600 CLS:PRINT "ACCUMULATED TIME = ";T2;" DAYS";"  PARTICLES= ";NP;
     LPRINT "  MAP AT TIME =";T2;" DAYS OF ";
     IF MAPTYPE$="A" THEN PRINT "   TOP VIEW   LAYER=";:LPRINT "TOP VIEW   LAYER=";
     IF MAPTYPE$="B" THEN PRINT "   ROW X-SECTION ROW=";:LPRINT "   ROW X-SECTION ROW=";
     IF MAPTYPE$="C" THEN PRINT "   COL X-SECTION COL=";:LPRINT "   COL X-SECTION COL=";
     IF LAYER=0 THEN PRINT "ALL":LPRINT "ALL" ELSE PRINT LAYER:LPRINT LAYER
4610 IF C$ <> "C" AND C$<>"c" THEN PRINT "PARTICLE MAP   (P SIGNIFIES PUMP LOCATION.  I SIGNIFIES
LOCATION OF INJECTION)"
4620 IF C$ = "P" OR C$="p" THEN 4670
4630 PRINT " CONCENTRATION MAP IN PP";RIGHT$(CC$,1);"  (P SIGNIFIES PUMPAGE, I SIGNIFIES INJECTION)"
     LPRINT " CONCENTRATION MAP IN PP";RIGHT$(CC$,1);"  (P SIGNIFIES PUMPAGE, I SIGNIFIES
INJECTION)"
4670 PRINT " "
4720 YYY = J2!
4730 FOR J = 1 TO LN
```

```
4740 M$ = STRING$(8-LEN(STR$(INT(YYY)))," ")+STR$(INT(YYY))
4750 PRINT  M$;"┤";
4760 FOR I = 1 TO WD
4770 IF NMAP(I,J) = -1 THEN PRINT "   P";:GOTO 4840
4780 IF NMAP(I,J) = -2 THEN PRINT "   I";:GOTO 4840
4790 IF ABS(NMAP(I,J))<.5 THEN PRINT "   •";:GOTO 4840
4800 M$ = STR$(INT(NMAP(I,J)+.5))
4810 M$ = RIGHT$(M$,LEN(M$)-1)
4820 IF LEN(M$) < 4 THEN M$ = SPACE$(4-LEN(M$)) + M$
4830 PRINT LEFT$(M$,4);
4840 NEXT I
4850 PRINT
4860 YYY = YYY - CDY
4870 NEXT J
4880 PRINT "         ";
4890 FOR I = 1 TO WD
4900 PRINT "───┬";
4910 NEXT I
4920 PRINT:PRINT
4930 FOR J = 1 TO 7
4940 PRINT "         ";
4950 XXX = I1!
4960 FOR I = 1 TO WD
4970 M$=STR$(INT(XXX)) + "      "
4980 M$ = MID$(M$,2,J)
4990 M$ = RIGHT$ (M$,1)
5000 PRINT "   ";M$;
5010 XXX = XXX + CDX
5020 NEXT I
5030 PRINT
5040 NEXT J
5050 PRINT
5060 REM LOCATE 9,55:PRINT "X AND Y COORDINATES"
5070 REM LOCATE 10,55:PRINT " SHOWN ARE IN FEET"
5080 REM LOCATE 11,55:PRINT "    FROM ORIGIN"
5090 REM LOCATE 24,1
5100 RETURN
     REM **************************************************
5110 REM Subroutine to read plume data from external file
     REM **************************************************
5120 PRINT
5130 PRINT " Enter filename, including extension (example PEORIA.DAT) [";FILE$;"] ";:INPUT TEMP$
     IF LEN(TEMP$)>0 THEN FILE$=TEMP$
5140 OPEN "I",#1,FILE$
     LPRINT
     LPRINT "  READING IN PLUME DATA FROM EXTERNAL FILE ";FILE$
     LPRINT
5150 LINE INPUT #1,A$
5160 LINE INPUT #1,R$:REM INPUT NAME OF FLOW DATA FILE
5165 LINE INPUT #1,SPFSAVE$:REM INPUT NAME OF SPF FILE
5170 INPUT
#1,T2,E,DL,DT,DV,PM,LAMDBA,I1!,J1!,CDX,CDY,NP,I2!,J2!,O,SCALEX,NL,TEMPSCR,MAPTYPE$,JROWXS,ICOLXS
     FOR I=1 TO NL+(NL-1):REM READ IN RETARDATION COEFFICIENTS
        INPUT#1,RETARD(I)
     NEXT I
5180 IF NP=0 THEN 5230
5190 FOR I=1 TO NP
5200 INPUT#1,X(I),Y(I),Z(I),SWITCH%(I)
5220 NEXT I
5230 CLOSE 1
5240 GOSUB 9260:REM OPEN AND READ IN DATA FROM FLOW DATA FILE
5245 IF LEN(SPFSAVE$)>0 THEN GOSUB 12290:REM OPEN AND READ SPECIAL FEATURE FILE
5250 G$="B"
5260 F$="A"
5270 RETURN
     REM ******************************************************
5280 REM Subroutine to write data to external file
     REM ******************************************************
5290 INPUT " Enter the name of the external file to be opened, including ext. ";FILE$
5320 OPEN "O",1,FILE$
     LPRINT
     LPRINT "  WRITING PLUME DATA TO EXTERNAL FILE ";FILE$
     LPRINT
5330 PRINT #1,"name:";FILE$
5340 PRINT #1,R$
5345 PRINT #1,SPFSAVE$:REM WRITE NAME OF SPECIAL FEATURES FILE
5350 PRINT
#1,T2;E;DL;DT;DV;PM;LAMDBA;I1!;J1!;CDX;CDY;NP;I2!;J2!;O;SCALEX;NL;TEMPSCR;CHR$(34);MAPTYPE$;CHR$(34)
;JROWXS;ICOLXS
     FOR I=1 TO NL+(NL-1):REM WRITE RETARDATION COEFFICIENTS
        PRINT#1,RETARD(I);
     NEXT I
     PRINT#1," "
5360 IF NP=0 THEN 5410
5370 FOR I=1 TO NP:REM WRITE PARTICLE LOCATIONS
5380 PRINT#1,X(I);Y(I);Z(I);SWITCH%(I)
5400 NEXT I
5410 CLOSE #1
5420 RETURN
     REM ******************************************************
     REM  ERROR ROUTINE - THIS ROUTINE IS BRANCHED TO ON ERROR, IT
     REM      RETURNS ERROR # AND TAKES YOU BACK TO THE MENU
     REM ******************************************************
5430 LPRINT "  ERR =";ERR;"  ERL # =";ERL
     IF ERR=6 THEN RESUME:REM THIS STATEMENT TRAPS A MYSTERIOUS ERROR CONDITION AND THE PROGRAM
CONTINUES
5435 PRINT "HEY, YOU'VE DONE SOMETHING WRONG!  YOU THINK ABOUT WHAT HAPPENED.  TRYING TO GO TO THE
MAIN MENU NOW.":PRINT
5440 PRINT "  ERR =";ERR;"  ERL # =";ERL
5460 FOR I = 1 TO 6000:NEXT I
5470 GOTO 760
     REM ******************************************************
5480 REM SUBROUTINE FOR SETTING COEFFICIENTS
     REM ******************************************************
5490 PRINT:G$="B":REM SET COEFFICIENT FLAG
5500 PRINT "////////////////BASIC TRANSPORT COEFFICIENTS\\\\\\\\\\\\\\"
5510 PRINT "     ENTER POROSITY [";E;"] ";:INPUT TEMP$:IF LEN(TEMP$)>0 THEN E=VAL(TEMP$)
5520 LPRINT "           POROSITY = ";E
```

```
     IF E<=0 OR E>1 THEN PRINT "  BAD POROSITY, PLEASE REENTER ":GOTO 5510
5560 PRINT "     ENTER PARTICLE MASS (LBS/PARTICLE) [";PM;"] ";:INPUT TEMP$:IF LEN(TEMP$)>0 THEN
PM=VAL(TEMP$)
5570 LPRINT "           PARTICLE MASS (LBS/PARTICLE) = ";PM
     IF PM<=0 THEN PRINT "  BAD PARTICLE MASS, PLEASE REENTER ":GOTO 5560
5580 PRINT "     ENTER LONGITUDINAL DISPERSIVITY (FT) [";DL;"] ";:INPUT TEMP$:IF LEN(TEMP$)>0 THEN
DL=VAL(TEMP$)
5590 LPRINT "           LONGITUDINAL DISPERSIVITY (FT)=";DL
5592 PRINT "     ENTER TRANSVERSE DISPERSIVITY (FT) [";DT;"] ";:INPUT TEMP$:IF LEN(TEMP$)>0 THEN
DT=VAL(TEMP$)
5594 LPRINT "           TRANSVERSE DISPERSIVITY (FT)=";DT
     PRINT "     ENTER VERTICAL DISPERSIVITY (FT) [";DV;"] ";:INPUT TEMP$:IF LEN(TEMP$)>0 THEN
DV=VAL(TEMP$)
     LPRINT "           VERTICAL DISPERSIVITY (FT)=";DV
     PRINT "     ENTER HALF-LIFE FOR FIRST ORDER DECAY (YEARS)(ZERO FOR NONE) [";LAMDBA;"] ";:INPUT
TEMP$:IF LEN(TEMP$)>0 THEN LAMDBA=VAL(TEMP$)
     IF LAMDBA<=0 THEN LAMDBA=1E10
     LPRINT "           HALF-LIFE FOR FIRST ORDER DECAY (YEARS)=";LAMDBA
     FOR I=1 TO NL+(NL-1)
      IF RETARD(I) <=0 THEN RETARD(I)=1:REM SET DEFAULT VALUES
5630  PRINT "      ENTER RETARDATION COEFFICIENT FOR LAYER ";I;" [";RETARD(I);"] ";
      INPUT TEMP$:IF LEN(TEMP$)>0 THEN RETARD(I)=VAL(TEMP$)
      IF RETARD(I)<1 THEN PRINT "  BAD RETARDATION COEFFICIENT, PLEASE REENTER ":GOTO 5630
      IF I<=NL THEN LPRINT "      RETARDATION COEFFICIENT FOR LAYER ";I;" IS ";RETARD(I)
      IF I>NL THEN LPRINT "      RETARDATION COEFFICIENT FOR CONFINING LAYER ";I-NL;" IS ";RETARD(I)
     NEXT I
5640 PRINT ""
5660 RETURN
     REM**********************************************
     REM  CREATE PLOT FILE FOR GOLDEN SOFTWARE SURFER
     REM **********************************************
6080 CLS
6090 IF F$<> "A" THEN PRINT "HEY, YOU HAVEN'T GONE THROUGH MAP--DO IT NOW!!!":GOSUB 13100:RETURN
6100 IF G$ <> "B" THEN PRINT "PLEASE CHOOSE COEFFICIENTS MENU ITEM NOW!!!@#":GOSUB 13100:RETURN
     IF PPP=0 THEN PRINT "MUST DISPLAY CONCENTRATION MAP TO SELECT CONCENTRATION RANGE":GOSUB
13100:RETURN
6220 PRINT:PRINT:INPUT "ENTER LOWER LEFT CORNER COORDINATES OF PLOT MAP (X,Y)";LOWERLX,LOWERLY
6230 LPRINT "LOWER LEFT-HAND CORNER OF PLOT POSITION = ";LOWERLX,LOWERLY
6240 INPUT "ENTER DESIRED GRID SIZE OF MAP, IN FEET ";SIZE
6250 LPRINT "GRID SIZE OF PLOT MAP, IN FEET = ";SIZE
6260 INPUT "ENTER NUMBER OF ROWS & COLUMNS OF PLOT MAP (ROWS, COLUMNS) ";ROWS,COLUMNS
6270 LPRINT "NUMBER OF ROWS AND COLUMNS OF PLOT MAP = ";ROWS,COLUMNS:LPRINT
6272 INPUT "ENTER LAYER OF MODEL TO GENERATE PLOT FILE FOR, FOR ALL LAYERS ENTER ZERO "; LAYER
     IF LAYER >NL+(NL-1) THEN GOTO 6272:REM CHECK SELECTION
6280 PRINT:PRINT
6290 PRINT "ENTER A 3-LETTER CODE FOR THE PLOT FILE"
6300 INPUT "File will be stored on default disk (Example abc01.dat)";code$
     IF LAYER<=0 THEN KKK=1:NLAYER=NL ELSE KKK=LAYER:NLAYER=LAYER:REM SET LOOP INDICES FOR CORRECT
PASSES THROUGH LOOP
     FOR K=KKK TO NLAYER:REM START OF LAYER LOOP
6310 FILE$=code$+RIGHT$(str$(K),LEN(STR$(K))-1)+".DAT"
6320 LPRINT "Plot file name is =";FILE$
6330 OPEN "O",#1,FILE$
6340 FOR I = 1 TO COLUMNS:REM  INITIALIZE NMAP
```

```
6350 FOR J = 1 TO ROWS
6360 NMAP(I,J)=0
6370 NEXT J
6380 NEXT I
6390 FOR MM% = 1 TO NP:REM     PARTICLE LOOP
        IF SWITCH%(MM%)=1 THEN GOTO 6400:REM   SKIP IF A CONTINUOUS SOURCE PARTICLE
        I = INT(1.0 + ((X(MM%)-LOWERLX)/SIZE)):REM  FIND COLUMN AND ROW OF PARTICLE
        J = INT(1.0 + ((Y(MM%)-LOWERLY)/SIZE))
        IF I<1 OR I>COLUMNS THEN GOTO 6400
        IF J<1 OR J>ROWS THEN GOTO 6400
        IF Z(MM%)>=BOT(I,J,NL) THEN LL=NL:GOTO 6395:REM  FIND LAYER OF PARTICLE
        FOR KLK=1 TO NL-1:REM FIND LAYER OF PARTICLE
           IF Z(MM%)>= BOT(I,J,KLK) AND Z(MM%)<TOP(I,J,KLK) THEN LL=KLK:GOTO 6395
           IF Z(MM%)>= TOP(I,J,KLK) AND Z(MM%)<BOT(I,J,KLK+1) THEN LL=KLK+NL:GOTO 6395
        NEXT KLK
6395    IF LL<>K THEN 6400:REM SKIP IF IN WRONG LAYER
        NMAP(I,J) = NMAP(I,J)+1:REM INCREMENT GRID PARTICLE COUNTER
6400 NEXT MM%
     FOR I=1 TO COLUMNS
        XX=(I-.5)*SIZE+LOWERLX:REM FIND INDICES OF THICKNESS GRID FOR MIDDLE OF THIS MAP GRID
        FOR J=1 TO ROWS
          YY=(J-.5)*SIZE +LOWERLY
          II=INT((XX-LLX)/DELX+1):REM FIND GRID NODE FOR PLOT GRID NODE
          JJ=INT((YY-LLY)/DELY+1)
          IF II<1 OR II>NC THEN GOTO 6500:REM  IF OUTSIDE GRID TO TO END OF LOOP
          IF JJ<1 OR JJ>NR THEN GOTO 6500
          IF K>NL THEN THICK1=BOT(II,JJ,K-NL+1)-TOP(II,JJ,K-NL) ELSE THICK1=THICK(II,JJ,K)
          IF THICK1<=0 THEN PRINT II,JJ,THICK(II,JJ,K):GOTO 6500
          IF RETARD(K)<=0 THEN PRINT II,JJ,RETARD(K),"RETARDATION =0 ":END
          NMAP(I,J)=(16019*NMAP(I,J)*PM)/(RETARD(K)*E*SIZE*SIZE*THICK1)*PPP
          PRINT #1, XX,YY,NMAP(I,J)
6500    NEXT J
6510 NEXT I
6520 CLOSE #1
     NEXT K
6530 RETURN
7110   REM ++++++++++++++++++++++++++++++++++++
7120   REM +  THIS IS THE SUBROUTINE TO SAVE  +
7130   REM +  THE PLUME POSITION FOR A SLIDE  +
7140 REM ++++++++++++++++++++++++++++++++++++
7200 DEF SEG = &HB800
7210 L$ = STR$(O)
7220 L$ =RIGHT$(L$,LEN(L$)-1)
7230 FILE$ = "SLIDE" + L$ + ".DAT":REM MAKE SLIDE NAME
7240 BSAVE FILE$,0,&H4000
7245 REM CALL BSAVEA(FILE$,&HB800,0,&H4000,ERRCODE)
7250 O = O+1:REM INCREMENT SLIDE COUNTER
7260 RETURN
7270 REM ++++++++++++++++++++++++++++++++++++++
7280 REM +  THIS IS THE SUBROUTINE TO DISPLAY  +
7290 REM +  THE SAVED SCREENS OF THE SLIDESHOW +
7300 REM ++++++++++++++++++++++++++++++++++++++
7310 PRINT:DEF SEG = &HB800
7320 IF O = 0 THEN PRINT " NO SCREENS HAVE BEEN SAVED YET---CHOOSE ANOTHER COMMAND #@#!!":GOTO 760
```

```
7330 PRINT:PRINT
7340 PRINT "                          Which mode do you want to operate in?"
7350 PRINT "                          FORWARD in time?  Or BACKWARD in time?"
7360 PRINT ""
7370 INPUT "          Enter the mode that you want.  F - FORWARD or B - BACKWARD";HHH$
7380 IF HHH$ = "f" OR HHH$ = "F" OR HHH$= "b" OR HHH$ = "B" THEN 7390 ELSE 7370
7390 REM
7400 SCREEN 1:IF XX$="y" OR XX$="Y" THEN COLOR 1,1:CLS ELSE CLS
7410 IF HHH$ = "B" OR HHH$ = "b" THEN 7440
7412 FOR I = 0 TO O-1
7414   I$ = STR$(I)
7416   I$ =RIGHT$(I$,LEN(I$)-1)
7418   FILE$ = "SLIDE" + I$ + ".DAT"
7420   BLOAD FILE$,0
7425   K$=INKEY$
7426   IF K$="" THEN GOTO 7425
7427 NEXT I
7430 GOTO 7500
7440 FOR I = O-1 TO 0 STEP -1
7450   I$ = STR$(I)
7460   I$ =RIGHT$(I$,LEN(I$)-1)
7470   FILE$ = "SLIDE" + I$ + ".DAT"
7480   BLOAD FILE$,0
7490   K$=INKEY$
       IF K$="" THEN GOTO 7490
     NEXT I
7500 SCREEN 0: WIDTH 80:IF XX$="Y" OR XX$="y" THEN COLOR 15,1,1:CLS ELSE CLS
7510 RETURN
7515 REM **********************************************************************
7520 REM THIS SUBROUTINE SELECTS THE AREA FOR PLOTTING, THE USER MAY ZOOM IN TO
7530 REM  ANY AREA AND DISPLAY ANY SPECIAL FEATURE FILE
7540 REM    by Don Koch, May 1986, July 1986, DECEMBER 1988
7542 REM ************************************************************************
     IF MAPTYPE$="B" THEN GOSUB 14115:IF Z1$="G" OR Z1$="g" THEN RETURN:REM PROFILE ROW ON SCREEN
     IF MAPTYPE$="C" THEN GOSUB 14615:IF Z1$="G" OR Z1$="g" THEN RETURN:REM PROFILE COLUMN ON SCREEN
7545 IF TEMPSCR=0 OR (I1!=0 AND SCALEX<=0) THEN GOSUB 8510:REM SET WINDOW TO DEFAULT COORDINATES
7570 YASPECT=ASPECT:GOSUB 7770:REM call window definition subroutine
7590 GOSUB 8110:REM DEFINE ZOOM BOX AREA
7600 IF TEMPSCR=1 THEN GOSUB 8700:REM RESTORE LAST SCREEN
7605 Z2$="":Z1$="":REM SET STRING OF COMMANDS TO NULL
     MAPTYPE$="A":REM INITIALIZE MAP TYPE TO PLAN VIEW
7610 Z1$=INKEY$
7620 GOSUB 10100:REM CALL SUBROUTINE WITH SPECIAL FEATURES
7630 IF Z1$="W" OR Z1$="w" THEN GOSUB 8580:REM DRAW SOURCES/SINKS AS CIRCLES
7660 IF Z1$="Z" OR Z1$="z" THEN Z3$=Z3$+Z2$:GOSUB 8750:REM CALL ZOOM SUBROUTINE
7700 IF Z1$="G" OR Z1$="g" THEN GOSUB 8640:RETURN:REM SAVE SCREEN AND QUIT SUBROUTINE
7710 IF Z1$="c" OR Z1$="C" THEN CLS:Z3$="":Z2$="":REM clear screen
7720 IF Z1$="m" OR Z1$="M" THEN GOSUB 8640:GOSUB 8270:GOSUB 8700:REM call menu, restore screen, and
reset windows
7730 IF Z1$="N" OR Z1$="n" THEN GOSUB 8510:GOSUB 7770:Z2$="":REM return to settings at beginning of
subroutine
7735 IF Z1$="D" OR Z1$="d" THEN GOSUB 12600:REM DRAW MODEL GRID ON SCREEN
7737 IF Z1$="X" OR Z1$="x" THEN MAPTYPE$="B":GOSUB 15000:GOSUB 14000:IF Z1$="G" OR Z1$="g" THEN
RETURN ELSE GOTO 7545:REM PROFILE ROW
```

```
7738 IF Z1$="Y" OR Z1$="y" THEN MAPTYPE$="C":GOSUB 15200:GOSUB 14500:IF Z1$="G" OR Z1$="g" THEN
RETURN ELSE GOTO 7545:REM PROVILE COLUMN
7740 Z2$=Z2$+Z1$
7750 Z1$=""
7760 GOTO 7610
7765 REM *************************************************
7770 REM  Subroutine to define plotting area of screen
7780 REM     by Don Koch, May 1985
7790 REM *************************************************
7820 CLS
7830 PRINT
7850 I2!=I1!+SCALEX:REM COMPUTE UPPER LIMITS OF VIEWING AREA
7860 J2!=J1!+SCALEX/YASPECT
7880 IF I2!<=I1! THEN PRINT "  Bad x coordinates":GOSUB 8060:GOTO 7820
7890 IF J2!<=J1! THEN PRINT "  Bad y coordinates":GOSUB 8060:GOTO 7820
7900 G1=0:G2=0:G3=319:G4=199
      REM G1=0:G2=0:G3=639:G4=349:REM FOR SCREEN MODE 9 WITH EGA CARD
7910 KEY OFF
7920 CLS
7930 SCREEN 1:IF XX$="Y" OR XX$="y" THEN COLOR 1,1
7940 VIEW (G1,G2)-(G3,G4),0
7950 WINDOW (I1!,J1!)-(I2!,J2!):REM SET SCREEN IN WORLD COORDINATES
7960 RETURN
7970 REM ***************************************************************
7975 REM SUBROUTINE TO READ SPECIAL FEATURE FILE AND PLOT IT ON THE SCREEN
7980 REM
7985 REM   INPUT FILE MUST BE IN THE GOLDEN SOFTWARE NEW PLOTCALL FORMAT
7990 REM   SPFFILES$(I)-NAME OF SPECIAL FEATURE FILE
7995 REM *****************************************************************
7997 IPEN=3:REM SET DEFAULT PEN COLOR
7999 IF XX$="Y" OR XX$="y" THEN MAXPEN=3 ELSE MAXPEN=1:REM SET MAXIMUM PEN COLOR
8000 OPEN "I",1,SPFFILES$(I)
8005 ACTION$="":X$=""
8010 SNUM$="":NTEMP=0
8015 X$=INPUT$(1,#1):REM READ FILE CHARACTERS ONE AT A TIME
8017 IF EOF(1) THEN CLOSE 1:RETURN
8020 IF X$=" " THEN 8015
8025 IF INSTR("MSPAR",X$)>0 THEN ACTION$=ACTION$+X$:REM IF ACTION VERB ADD TO ACTION STRING
8030 IF INSTR("-0123456789.",X$)>0 THEN 8035 ELSE 8015:REM IF A COORDINATE ADD TO COORDINATE STRING
AND STOP CHECKING FOR ACTION VERBS
8035 SNUM$=X$
8040 X$=INPUT$(1,#1):REM READ FILE CHARACTERS ONE AT A TIME
8045 IF X$=CHR$(13) THEN 8050 ELSE SNUM$=SNUM$+X$:GOTO 8040:REM IF END OF LINE THEN EXEUTE COMMAND
8050 IF ACTION$="SP" THEN IPEN=VAL(SNUM$):GOTO 8005:REM CHANGE PEN
8055 NTEMP=INSTR(SNUM$,","):REM DECODE THE CARTESIAN COORDINATES
8060 IF NTEMP<=0 THEN NTEMP=INSTR(SNUM$," ")
8065 IF NTEMP<=0 THEN 8005
8070 X=VAL(LEFT$(SNUM$,NTEMP-1))
8075 Y=VAL(RIGHT$(SNUM$,LEN(SNUM$)-NTEMP))
8076 IF IPEN>MAXPEN THEN IPEN=MAXPEN:REM CHECK FOR VALID PEN COLOR
8077 IF ACTION$="MA" THEN PSET (X,Y),IPEN:REM EXECUTE THE ACTION
8080 IF ACTION$="MR" THEN PSET STEP (X,Y),IPEN
8082 IF ACTION$="PA" THEN LINE -(X,Y),IPEN
8085 IF ACTION$="PR" THEN LINE -STEP (X,Y),IPEN
```

```
8090 IF EOF(1) THEN CLOSE 1:RETURN ELSE GOTO 8005:REM RETURN FOR NEXT LINE
8105 REM *****************************************
8110 REM  SET DEFAULT ZOOM BOX COORDINATES
8120 REM *****************************************
8130 X1V=I1!
8140 Y1V=J1!
8150 SVIEWX=(I2!-I1!)/100
8160 X2V=X1V+SVIEWX
8170 Y2V=Y1V+SVIEWX/ASPECT
8180 RETURN
8185 REM ******************************************
8190 REM  SUBROUTINE RESETS WINDOW AREA TO ZOOM BOX
8200 REM ******************************************
8210 I1!=X1V
8220 J1!=Y1V
8230 J2!=Y2V
8240 I2!=X2V
8250 SCALEX=X2V-X1V
8260 RETURN
8270 REM  *********************************************************************
8280 REM   MENU ROUTINE  FOR SPECIAL FEATURE FILES
8290 REM  *********************************************************************
8300 CLS:WIDTH 80
8320 PRINT TAB(20)"Menu"
8330 PRINT
8350 PRINT TAB(2)"w - plot sinks as circles";TAB(40) "c - clear screen"
8390 PRINT TAB(2)"m - this menu";TAB(40) "n - return to default settings"
8410 PRINT TAB(2)"z - zoom routine";TAB(40) "d - draw model grid"
8420 PRINT TAB(2)"x - profile row ";TAB(40) "y - profile column"
8450 PRINT TAB(2)"g - leave special feature subroutine"
8452 FOR I=1 TO NSPF
8454    PRINT TAB(2)SPFKEY$(I,1);" - ";SPFFILE$(I)
8456 NEXT I
8460 PRINT TAB(2)"                    "
8470 PRINT "  Hit return to return to the plotting screen"
8480 INPUT Z4$
8490 CLS:GOSUB 7770:REM DEFINE SCREEN WINDOW AGAIN
8500 RETURN
8510 REM ****************************************
8515 REM   SET WINDOW TO DEFAULT COORDINATES
8520 REM ****************************************
8525 I1!=I1DEF!
8530 J1!=J1DEF!
8535 SCALEX=SCALEDEF
8540 RETURN
8570 REM ****************************************************
8580 REM   Plot sources/sinks as circles
8590 REM *****************************************************
8600 FOR I=1 TO NS
8610 CIRCLE (X1(I),Y1(I)),(I2!-I1!)/180,3,,,(ASPECT*(J2!-J1!)/(I2!-I1!))
8620 NEXT I
8630 RETURN
8638 REM *****************************************************
8640 REM  TEMP SAVE OF SCREEN IMAGE
```

```
8650 REM ***************************************************
8660 DEF SEG=&HB800:REM  SET SEG TO VIDEO MEM
8670 BSAVE "TEMP.SCR",0,&H4000
8675 REM CALL BSAVEA("TEMP.SCR",&HB800,0,&H4000,ERRCODE):REM USE THIS CALL WITH BLBS.OBJ AND QB 1.02
8680 TEMPSCR=1:REM SET FLAG FOR SAVED VARIABLE
8685 Z3$=Z3$+Z2$:REM ADD CURRENT ADDITIONS TO SPECIAL FEATURES STRING
8690 RETURN
8698 REM *******************************************************
8700 REM   RESTORE SCREEN IMAGE FROM TEMP.SCR
8710 REM *******************************************************
8720 DEF SEG=&HB800
8730 BLOAD "TEMP.SCR",0
8735 REM CALL BLOADA("TEMP.SCR",&HB800,0,BYTES,ERRCODE):REM USE THIS CALL WITH BLBS.OBJ AND QB 1.02
8740 RETURN
8745 REM ***************************************************
8750 REM   ZOOM SUBROUTINE  THIS SUBROUTINE IS CALLED WHEN ZOOMING IS DESIRED
8760 REM ******************************************************
8770 Z1$=INKEY$
8780 LINE (X1V,Y1V)-(X2V,Y2V),1,B:REM DRAW ZOOM BOX
8784 REM  NEXT STATEMENT PERFORMS THE ZOOM ON A CARRIAGE RETURN
8786 REM   FIRST WINDOW IS SET TO ZOOM BOX, THEN ACCUMULATED COMMANDS ARE PLOTTED
8790 IF Z1$=CHR$(13) THEN GOSUB 8190:GOSUB 7770:GOSUB 10300:RETURN:REM perform the zoom
8800 IF Z1$=CHR$(0)+CHR$(72) THEN LINE (X1V,Y1V)-(X2V,Y2V),0,B:Y1V=Y1V+(I2!-I1!)/50:GOSUB 8890:REM
MOVE LL CORNER OF ZOOM BOX
8810 IF Z1$=CHR$(0)+CHR$(80) THEN LINE (X1V,Y1V)-(X2V,Y2V),0,B:Y1V=Y1V-(I2!-I1!)/50:GOSUB 8890
8820 IF Z1$=CHR$(0)+CHR$(75) THEN LINE (X1V,Y1V)-(X2V,Y2V),0,B:X1V=X1V-(I2!-I1!)/50:GOSUB 8890
8830 IF Z1$=CHR$(0)+CHR$(77) THEN LINE (X1V,Y1V)-(X2V,Y2V),0,B:X1V=X1V+(I2!-I1!)/50:GOSUB 8890
8840 IF Z1$="e" OR Z1$="E" THEN LINE (X1V,Y1V)-(X2V,Y2V),0,B:SVIEWX=SVIEWX+(I2!-I1!)/50:GOSUB
8890:REM EXPAND ZOOM BOX
8850 IF Z1$="S" OR Z1$="s" THEN LINE (X1V,Y1V)-(X2V,Y2V),0,B:SVIEWX=SVIEWX-(I2!-I1!)/50:GOSUB
8890:REM SHRINK ZOOM BOX
8860 IF Z1$="M" OR Z1$="m" THEN GOSUB 8640:GOSUB 8950:GOSUB 8700:REM SAVE SCREEN, VIEW MENU, AND
RESTORE SCREEN
8862 IF Z1$="R" OR Z1$="r" THEN GOSUB 10300:REM REFRESH SCREEN
8870 GOTO 8770
8880 RETURN:REM end of zoom routine
8890 REM  DRAW ZOOM BOX SUBROUTINE
8900 IF SVIEWX<(I2!-I1!)/1000 THEN SVIEWX=(I2!-I1!)/1000
8910 X2V=X1V+SVIEWX:REM RESIZE THE ZOOM BOX
8920 Y2V=Y1V+SVIEWX/ASPECT
8930 LINE (X1V,Y1V)-(X2V,Y2V),1,B:REM DRAW ZOOM BOX
8940 RETURN
8950 REM **********************************************************************
8960 REM   ZOOM ROUTINE MENU
8970 REM **********************************************************************
8980 CLS
8990 PRINT
9000 PRINT TAB(10)"Menu"
9010 PRINT
9020 PRINT TAB(2)"s - shrink the zoom box"
9030 PRINT TAB(2)"e - expand the zoom box"
9040 PRINT TAB(2)"use arrow keys to move the lower left"
9050 PRINT TAB(2)"  corner of the zoom box"
9060 PRINT TAB(2)"m - this menu"
```

```
9062 PRINT TAB(2)"r - refresh screen"
9070 PRINT TAB(2)"return to perform zoom"
9080 PRINT TAB(2)"                   "
9090 PRINT "  Hit return to return to the plotting screen"
9100 INPUT Z4$
9110 CLS
9120 RETURN
9128 REM *****************************************************
9130 REM  INPUT VELOCITY AND THICKNESS ARRAY DATA
9132 REM *****************************************************
9140 CLS:PRINT:PRINT SPC(18);"VELOCITY FILE INPUT DATA MENU":PRINT
9150 FILES "*.RND"
9160 PRINT
9170 PRINT "Enter the name of the external VELOCITY file from above--DO NOT"
9180 PRINT "use any extensions or drive designation nor more than "
9190 PRINT "eight (8) characters.  You MUST choose from the above list--"
9200 PRINT "-------------------are you ready??"
9210 PRINT
9220 INPUT "Enter the name of the external file for input (Example HEAD1) ",FILE$
9230 IF LEN(FILE$)>8 THEN PRINT "FILE NAME MUST BE LESS THAN 9 CHARACTERS":GOTO 9220
9240 CLS
9250 R$=FILE$+".RND"
     rem     lprint "  string space free = ";fre("");fre(0);fre(-1)
     9260 PRINT " NOW PROCESSING FILE ";R$
     LPRINT "  INPUT VELOCITY FILE ";R$
9270    OPEN "I",#1,R$
9280 REM   INPUT #1,CTIME,TIMESTEP
9290 REM    INPUT #1, NSTEPS,DE,PLASMER,KS
9292 NPER=NPER+1:REM INCREMENT PERIOD COUNTER
9300    INPUT #1,NC,NR,NL,DELX,DELY,LLX,LLY,LLZ:REM INPUT GRID CONSTANTS FOR FLOW MODEL
9310    FOR I=1 TO NC:REM INPUT THICKNESSES (FT) AND VELOCITIES (FT/DAY)
9320        FOR J=1 TO NR
9322         FOR K=1 TO NL
9330            INPUT #1,II,JJ,KK,VTHICK,VVI,VVJ,VVK,TBOT,TTOP
9340            THICK(II,JJ,KK)=VTHICK:VI(II,JJ,KK)=VVI:VJ(II,JJ,KK)=VVJ:VK(II,JJ,KK)=VVK
9341            BOT(II,JJ,KK)=TBOT:TOP(II,JJ,KK)=TTOP
9342         NEXT K
9350        NEXT J
9360    NEXT I
9370 REM   FOR I=1 TO NC+1
9380 REM       INPUT #1,DELX(I)
9390 REM   NEXT I
9400 REM   FOR J=1 TO NR+1
9410 REM       INPUT #1,DELY(J)
9420 REM   NEXT J
9430 REM   INPUT #1,NS:REM INPUT NUMBER OF SINKS
9440 REM LPRINT "There are ";NS;" sinks":LPRINT
9450 REM LPRINT "Sink # ","I coordinate","J coordinate"," Withdrawal rate"
9458     I=0
         IF EOF(1) THEN NS=I:GOTO 9500:REM IF NO SINKS THEN END ROUTINE
9460        I=I+1:REM INPUT SINKS AS COORDINATES, AND DISCHARGE (GPD)
9470            INPUT #1,X1(I),Y1(I),Z1(I),QSUM(I)
9480 REM        LPRINT I,X1(I),Y1(I),Z1(I),QSUM(I)
9490        IF(EOF(1)) THEN NS=I ELSE GOTO 9460
```

```
9500    CLOSE 1
9510 REM LPRINT:LPRINT:LPRINT
9520 REM PRINT "STARTING TIME MUST BE LESS THAN ";CTIME;" AND GREATER THAN OR EQUAL TO ";CTIME-
TIMESTEP
9530 REM INPUT "   ENTER STARTING TIME OF SIMULATION ",T2
9540 REM IF T2>=CTIME OR T2<CTIME-TIMESTEP THEN PRINT "STARTING TIME MUST BE LESS THAN ";CTIME;" AND
GREATER THAN ";CTIME-TIMESTEP:GOTO 21030
9550 REM PRINT "          STARTING TIME OF SIMULATION (DAYS) = ";T2
9560 REM LPRINT "          STARTING TIME OF SIMULATIN (DAYS) = ";T2
9570 REM LPRINT:LPRINT
9580 REM INPUT "DO YOU WANT TO REDO THIS SCREEN DATA ? (Y-Yes, N-No) ",ANS$
9590 REM IF ANS$<>"N" AND ANS$<>"n" AND ANS$<>"Y" AND ANS$<>"y" THEN PRINT "(Y-Yes, N-No)":GOTO
21080
9600 REM IF ANS$="Y" OR ANS$="y" THEN GOTO 21030
9610 RETURN
10100 REM **********************************************************
10110 REM   SUBROUTINE THAT CONTAINS KEYS FOR SPECIAL FEATURES FILE
10120 REM **********************************************************
10130 FOR I=1 TO NSPF
10140   IF Z1$=SPFKEY$(I,1) OR Z1$=SPFKEY$(I,2) THEN GOSUB 7970
10150 NEXT I
10160 IF Z1$="D" OR Z1$="d" THEN GOSUB 12610:REM RESTORE GRID
10170 IF Z1$="W" OR Z1$="w" THEN GOSUB 8580:REM RESTORE WELLS
10250 RETURN
10300 REM ************************************************************
10310 REM   SUBROUTINE THAT WRITES SPECIAL FEATURES TO SCREEN AFTER ZOOM
10320 REM ************************************************************
10330 IZ2=LEN(Z3$)
10340 FOR K=1 TO IZ2
10350   Z1$=MID$(Z3$,K,1)
10360   GOSUB 10130:REM MATCH KEY TO SPECIAL FILE
10370 NEXT K
10375 Z1$=""
10380 RETURN
10400 REM ***************************************************************
10410 REM   SUBROUTINE TO MOVE PARTICLE OUT OF ZERO VELOCITY GRID
10420 REM ***************************************************************
10440 XP=III+1.01
10470 YP=JJJ+1.01
10480 IF III+2>=NC THEN XP=III-.01:GOTO 10500
10490 IF VJ(III+2,JJJ,K)=0 THEN XP=III-.01
10500 IF JJJ+2>=NR THEN YP=JJJ-.01:GOTO 10520
10510 IF VI(III,JJJ+2,K)=0 THEN YP=JJJ-.01
10520 RETURN
11000 REM **********************************************************
11010 REM   SETUP ROUTINE FOR SPECIAL FEATURE FILES
11020 REM      BY DON KOCH, JULY 1986
11030 REM
11040 REM   SPECIAL FEATURE FILES ARE ASCII SEQUENTIAL FILES OF CARTESIAN
11050 REM    COORDINATES THAT DESCRIBE SPECIAL GEOGRAPHIC FEATURES THAT
11060 REM    THE USER WISHES TO HAVE SUPERIMPOSED ON THE SCREEN
11070 REM
11080 REM   THIS SUBROUTINE CONTROLS THE MENU THAT THE USER USES TO DEFINE
11090 REM    SPECIAL FEATURES FILES AND ASSIGN KEYS TO RETRIEVE THE FILES
```

```
11100 REM   SCREEN ASPECT RATIO AND THE DEFAULT VIEW WINDOW MAY ALSO BE SET
11110 REM ************************************************************
11120 REM
11130 CLS
11140 PRINT
11150 PRINT TAB(15)"SPECIAL FEATURE SETUP MENU"
11160 PRINT
11170 PRINT TAB(5)"1   SET DEFAULT VIEW WINDOWS"
11180 PRINT TAB(5)"2   SET ASPECT RATIOS"
11190 PRINT TAB(5)"3   SETUP SPECIAL FEATURE FILES"
11192 PRINT TAB(5)"4   SAVE SPECIAL FEATURE FILE SETUP TO DISK"
11194 PRINT TAB(5)"5   LOAD SPECIAL FEATURE FILE SETUP TO DISK"
11196 PRINT TAB(5)"6   SETUP TIME PERIODS FOR TRANSIENT SIMULATION"
11200 PRINT TAB(5)"7   RETURN TO MAIN MENU"
11210 PRINT
11220 PRINT
11230 INPUT "  ENTER CHOICE ";NCHOICE
11240 IF NCHOICE=7 THEN RETURN
11250 ON NCHOICE GOSUB 11300,11540,11700,12000,12200,12800
11260 GOTO 11130
11300 REM ***********************************************************
11310 REM   SET DEFAULT WINDOW SIZE
11320 REM
11330 REM   I1DEF!,J1DEF! - LOWER LEFT CORNER OF DEFAULT VEIWING WINDOW
11340 REM   SCALEDEF - HORIZONTAL WIDTH OF DEFAULT VIEWING WINDOW
11350 REM ************************************************************
11360 CLS
11362 TEMPSCR=0:REM DO NOT RESTORE LAST SCREEN SAVED AFTER RESETTING DEFAULTS
11370 PRINT
11380 IF I1DEF!=0 AND SCALEDEF=0 THEN I1DEF!=LLX:I2DEF!=LLX+NC*DELX:J1DEF!=LLY:SCALEDEF=I2DEF!-
I1DEF!
11390 PRINT "  ENTER LOWER LEFT CORNER OF DEFAULT VIEW WINDOW ";"[";I1DEF!;",";J1DEF!;"] ";
11400 LINE INPUT TEMP$
11410 IF LEN(TEMP$)>0 THEN NTEMP=INSTR(TEMP$,","):IF NTEMP<=0 THEN GOSUB 11490:GOTO 11390:ELSE
I1DEF!=VAL(LEFT$(TEMP$,NTEMP-1)):J1DEF!=VAL(RIGHT$(TEMP$,LEN(TEMP$)-NTEMP))
11420 PRINT "  ENTER HORIZONTAL WIDTH OF DEFAULT VIEW WINDOW ";"[";SCALEDEF;"] ";
11430 INPUT TEMP$
11440 IF LEN(TEMP$)>0 THEN SCALEDEF=VAL(TEMP$)
11444 IF SCALEDEF<=0 THEN PRINT "  HORIZONTAL WIDTH OF DEFAULT VIEWING AREA MUST BE GREATER THAN
ZERO ":GOTO 11390
11450 RETURN
11460 REM ************************************************************
11470 REM  BAD ENTRY OF COORDINATE PAIR ROUTINE
11480 REM ************************************************************
11490 PRINT
11500 PRINT "  USE COMMA TO SEPARATE COORDINATE PAIR"
11510 PRINT
11520 RETURN
11530 REM ************************************************************
11540 REM   SET ASPECT RATIO OF SCREEN
11550 REM ************************************************************
11560 CLS
11570 PRINT
11580 IF ASPECT<=0 THEN ASPECT=1.4
```

```
     IF ZASPECT<=0 AND SCALEDEF>0 THEN ZASPECT=SCALEDEF/(THICK(1,1,NL)+BOT(1,1,NL)+10.-LLZ)
11590 PRINT "  ENTER ASPECT RATIO (HORIZONTAL/VERTICAL) OF SCREEEN ";"[";ASPECT;"] ";
11600 INPUT TEMP$
11610 IF LEN(TEMP$)>0 THEN ASPECT=VAL(TEMP$)
11612 PRINT "  ENTER VERTICAL EXAGGERATION OF CROSS-SECTION PLOT ";"[";ZASPECT;"]";
11614 INPUT TEMP$
11616 IF LEN(TEMP$)>0 THEN ZASPECT=VAL(TEMP$)
11620 RETURN
11700 REM ******************************************************************
11710 REM  SUBROUTINE TO ENTER SPECIAL FEATURE FILE NAMES AND KEYS
11720 REM
11730 REM    EACH SPECIAL FEATURE IS DESCRIBED BY A SERIES OF CARTESIAN COORDINATES IN A FILE
11740 REM      AND ASSIGNED TO A KEY, SO WHEN SCREEN GRAPHICS ARE ACTIVE, THE
11750 REM      SPECIAL FEATURE MAY BE PLOTTED ON THE SCREEN MERLELY BY PRESSING
11760 REM      THE KEY
11770 REM   SPFFILE$() - ARRAY OF SPECIAL FEATURE FILE NAMES
11780 REM   SPFKEY$() - ARRAY OF KEYS FOR THESE SPECIAL FILE NAMES
11790 REM   NSPF - NUMBER OF KEYS ASSIGNED
11800 REM
11810 CLS
11820 PRINT
11830 PRINT "  ASSIGN SPECIAL FEATURE FILE NAMES TO KEYS, ENTER EACH FILE NAME"
11840 PRINT "  AND KEY ASSIGNMENT IN ORDER, ENTER A CARRIAGE RETURN FOR A BLANK"
11850 PRINT "  FILE NAME TO EXIT THIS ROUTINE"
11860 PRINT
11870 I=0
11880 I=I+1
11882 IF I>20 THEN PRINT "  MAXIMUM NUMBER OF SPECIAL FEATURES ALREADY ASSIGNED":FOR K=1 TO
1000:NEXT K:RETURN
11890 PRINT "  ENTER SPECIAL FEATURE FILE NAME ";"[";SPFFILE$(I);"] ";
11900 LINE INPUT TEMP$
11910 IF LEN(TEMP$)>0 THEN SPFFILE$(I)=TEMP$
11920 IF LEN(TEMP$)=0 AND LEN(SPFFILE$(I))=0 THEN RETURN
11930 PRINT "  ENTER ONE KEY FOR THIS SPECIAL FEATURE ";"[";SPFKEY$(I,1);"] ";
11940 INPUT TEMP$
11950 IF LEN(TEMP$)>1 THEN PRINT "  ONLY ONE KEY ":GOTO 11930
11960 IF LEN(TEMP$)>0 THEN SPFKEY$(I,1)=TEMP$:SPFKEY$(I,2)=TEMP$
11962 IF ASC(SPFKEY$(I,1))>64 AND ASC(SPFKEY$(I,1))<91 THEN SPFKEY$(I,2)=CHR$(ASC(SPFKEY$(I,1))+32)
11964 IF ASC(SPFKEY$(I,1))>96 AND ASC(SPFKEY$(I,1))<123 THEN SPFKEY$(I,2)=CHR$(ASC(SPFKEY$(I,1))-32)
11966 GOSUB 12400:REM CHECK SPFKEY FOR DUPLICATION
11970 NSPF=I
11980 GOTO 11880
12000 REM ****************************************************************
12010 REM    SAVE FILE OF SPECIAL FEATURE DATA
12020 REM ****************************************************************
12030 REM
12040 CLS
12050 PRINT
12060 PRINT "  ENTER FILE SPECIFICATION TO SAVE SPECIAL FEATURE DATA ON [";SPFSAVE$;"] ";
12070 LINE INPUT TEMP$
12080 IF LEN(TEMP$)>0 THEN SPFSAVE$=TEMP$
12090 OPEN "O",1,SPFSAVE$
12100 PRINT#1,I1DEF!;J1DEF!;SCALEDEF
12110 PRINT#1,ASPECT;ZASPECT
```

```
12115 PRINT#1,NSPF
12120 FOR I=1 TO NSPF
12130     WRITE#1,SPFKEY$(I,1);SPFKEY$(I,2);SPFFILE$(I)
12140 NEXT I
12150 CLOSE 1
12160 RETURN
12200 REM *****************************************************************
12210 REM   LOAD SPECIAL FEATURE DATA FROM FILE
12220 REM *****************************************************************
12230 REM
12240 CLS
12250 PRINT
12260 PRINT "  ENTER FILE SPECIFICATION TO LOAD SPECIAL FEATURE DATA FROM [";SPFSAVE$;"] ";
12270 LINE INPUT TEMP$
12280 IF LEN(TEMP$)>0 THEN SPFSAVE$=TEMP$
12290 OPEN "I",1,SPFSAVE$
12300 INPUT#1,I1DEF!,J1DEF!,SCALEDEF
12310 INPUT#1,ASPECT,ZASPECT
12320 INPUT#1,NSPF
12330 FOR I=1 TO NSPF
12340     INPUT#1,SPFKEY$(I,1),SPFKEY$(I,2),SPFFILE$(I)
12350 NEXT I
12360 CLOSE 1
12370 RETURN
12400 REM ************************************************************
12410 REM  SUBROUTINE TO CHECK SPECIAL FEATURE KEYS FOR DUPLICATION
12420 REM ************************************************************
12430 REM  SPFKEY$(I,1) (I,2) - ASSIGNED SPECIAL FEATURE KEY TO CHECK
12440 IF SPFKEY$(I,1)="W" OR SPFKEY$(I,1)="w" THEN 12540
12450 IF SPFKEY$(I,1)="C" OR SPFKEY$(I,1)="c" THEN 12540
12460 IF SPFKEY$(I,1)="M" OR SPFKEY$(I,1)="m" THEN 12540
12470 IF SPFKEY$(I,1)="N" OR SPFKEY$(I,1)="n" THEN 12540
12480 IF SPFKEY$(I,1)="Z" OR SPFKEY$(I,1)="z" THEN 12540
12490 IF SPFKEY$(I,1)="G" OR SPFKEY$(I,1)="g" THEN 12540
12495 IF SPFKEY$(I,1)="D" OR SPFKEY$(I,1)="d" THEN 12540
12497 IF SPFKEY$(I,1)="X" OR SPFKEY$(I,1)="x" THEN 12540
12498 IF SPFKEY$(I,1)="Y" OR SPFKEY$(I,1)="y" THEN 12540
12500 FOR K=1 TO NSPF
12505 IF I=K THEN 12520
12510   IF SPFKEY$(I,1)=SPFKEY$(K,1) OR SPFKEY$(I,2)=SPFKEY$(K,1) THEN 12540
12520 NEXT K
12525 IF ASC(SPFKEY$(I,1)) < 33 OR ASC(SPFKEY$(I,1)) > 126 THEN PRINT "  BAD KEY ASSIGNMENT ":I=I-
1:FOR K=1 TO 1000:NEXT K:RETURN
12530 RETURN
12540 PRINT:REM ERROR MESSAGE FOR DUPLICATE KEY
12550 PRINT "  THAT KEY IS ALREADY ASSIGNED TO ANOTHER SPECIAL FEATURE, CHOOSE ANOTHER":FOR K=1 TO
1000:NEXT K
12560 I=I-1
12570 RETURN
12600 REM ***************************************************************
12610 REM  SUBROUTINE TO DRAW MODEL GRID ON SCREEN
12620 REM ***************************************************************
12630 YBOT=LLY
12640 YTOP=LLY+(NR)*DELY
```

```
12650 FOR I=0 TO NC
12660    X=LLX+I*DELX
12670    LINE (X,YBOT)-(X,YTOP),1
12680 NEXT I
12690 XBOT=LLX
12700 XTOP=LLX+(NC)*DELX
12710 FOR J=0 TO NR
12720   Y=LLY+J*DELY
12730   LINE (XBOT,Y)-(XTOP,Y),1
12740 NEXT J
12750 RETURN
12800 REM ************************************************************
12810 REM  ROUTINE TO INPUT VELOCITY FILE END TIMES
12820 REM
12830 REM   NENDTIME - NUMBER OF VELOCITY FILES TO BE USED DURING SIMULATION
12840 REM   ENDTIME() - ARRAY OF ENDING TIMES FOR EACH VELOCITY FILE
12850 REM ************************************************************
12860 CLS
12862 ENDTIME(0)=0
12870 PRINT
12880 PRINT "  IF MULTIPLE VELOCITY FILES ARE TO BE USED (TRANSIENT SIMULATION)"
12890 PRINT "  THE USER MUST ENTER THE NUMBER OF VELOCITY FILES THAT WILL BE USED"
12900 PRINT "  AND THE ENDING TIME FOR EACH VELOCITY FILE"
12910 PRINT
12920 PRINT "  ENTER THE NUMBER OF VELOCITY FILES (TIME PERIODS) ";"[";NENDTIME;"] ";
12930 INPUT TEMP$
12940 IF LEN(TEMP$)>0 THEN NENDTIME=VAL(TEMP$)
12950 IF NENDTIME<=0 THEN PRINT "  STEADY STATE ANALYSIS":FOR K=1 TO 6000:NEXT K:RETURN
12960 IF NENDTIME>20 THEN PRINT "  MAXIMUM NUMBER OF TIME PERIODS IS 20 ":GOTO 12920
12970 FOR I=1 TO NENDTIME
12980   PRINT "  ENTER THE ENDING TIME (DAYS) FOR VELOCITY FILE #";I;" [";ENDTIME(I);"] ";
12990   INPUT TEMP$
13000   IF LEN(TEMP$)>0 THEN ENDTIME(I)=VAL(TEMP$)
13002   IF ENDTIME(I)<ENDTIME(I-1) THEN PRINT "  ENDING TIME MUST BE GREATER THAN PREVIOUS
TIME":GOTO 12980
13010 NEXT I
13020 RETURN
13100 REM **********************************************
13110 REM   PAUSE ON SCREEN ROUTINE
13120 REM **********************************************
13130 PRINT
13140 INPUT "  PRESS ENTER TO CONTINUE ";Z1$
13150 RETURN
14000 REM ********************************************************
14010 REM  PLOT ROW CROSS SECTION ON SCREEN
14020 REM ********************************************************
14030 REM
14040 REM   ZASPECT - HORIZONTAL/VERTICAL SCREEN RATIO - THE INVERSE
14050 REM                OF THE VERTICAL EXAGGERATION RATIO
14060 REM   I1!,J1!, - COORDINATES OF LOWER LEFT CORNER OF SCREEN
14070 REM   JROWXS - SELCTED ROW TO PLOT ON SCREEN
14080 REM   LLX,LLZ - ENTERED COORDINATES OF LOWER LEFT CORNER OF SCREEN
14110 REM USE SAME I1! AND I2! COORDINATES AS BEFORE=LLX
      IF I2!<=I1! THEN I2!=I1!+SCALEDEF
      IF SCALEDEF<=0 THEN SCALEDEF=NC*DELX
      J1!=LLZ:  REM  COORDINATES OF LOWER LEFT CORNER OF SCREEN
      J2!=LLZ+SCALEDEF/ZASPECT
14115 GOSUB 7880:REM CALL ROUTINE TO SET UP SCREEN WINDOW,NOTICE NOT AT BEGINNING OF SUBROUTINE
14120 J=JROWXS
14130 FOR K=1 TO NL-1
        ISTART=0
14140   ISTART=ISTART+1
        IF THICK(ISTART,J,K)<0 THEN GOTO 14140
        PSET (LLX+DELX/2,BOT(ISTART,J,K)):REM SET BEGINNING POINT OF BOTTOM
14150   FOR I=ISTART TO NC :REM    PLOT BOTTOM OF LAYER
14160     DIST=LLX+DELX*I
          IF THICK(I,J,K) < 0 THEN GOTO 14180
14170     LINE -(DIST-DELX/2,BOT(I,J,K))
14180   NEXT I
14190   PSET (LLX+DELX/2,TOP(ISTART,J,K)):REM SET BEGINNING POINT OF TOP
14200   FOR I=ISTART TO NC :REM    PLOT TOP OF LAYER
14210     DIST=LLX+DELX*I
          IF THICK(I,J,K) <0 THEN GOTO 14230
14220     LINE -(DIST-DELX/2,TOP(I,J,K))
14230   NEXT I
14240 NEXT K
        ISTART=0
14250   ISTART=ISTART+1
        IF THICK(ISTART,J,K)<0 THEN GOTO 14250
        PSET (LLX+DELX/2,BOT(ISTART,J,NL)):REM SET BEGINNING POINT OF BOTTOM OF TOP LAYER
14260   FOR I=ISTART TO NC :REM    PLOT BOTTOM OF TOP LAYER
14270     DIST=LLX+DELX*I
          IF THICK(I,J,K) < 0 THEN GOTO 14290
14280     LINE -(DIST-DELX/2,BOT(I,J,NL))
14290   NEXT I
14300 PSET(LLX+DELX/2,BOT(ISTART,J,NL)+THICK(ISTART,J,NL)):REM SET BEGINNING OF WATER TABLE TOP
14310 FOR I=ISTART TO NC :REM    PLOT BOTTOM OF TOP LAYER
14320   DIST=LLX+DELX*I
        IF THICK(I,J,K) < 0 THEN GOTO 14340
14330   LINE -(DIST-DELX/2,BOT(I,J,NL)+THICK(I,J,NL))
14340 NEXT I
14350 FOR II=1 TO NS :REM PLOT WELLS AND RIVERS ON SCREEN
14360   IF INT((Y1(II)-LLY)/DELY+1.0) <> J THEN 14440:REM ONLY PLOT WELLS IN THIS RO
14370   I=INT((X1(II)-LLX)/DELX+1.0):REM FIND COLUMN OF WELL
14380 REM  FOR KLK=1 TO NL-1:REM FIND LAYER OF WELL
14390 REM    IF Z1(II)>=BOT(I,J,KLK) AND Z1(II) <TOP(I,J,KLK) THEN ZK=BOT(I,J,KLK)
14400 REM   NEXT KLK
      ZK=BOT(I,J,INT(Z1(II)))
14410 REM  IF Z1(II)>=BOT(I,J,NL) AND Z1(II)<(BOT(I,J,NL)+THICK(I,J,NL)) THEN ZK=BOT(I,J,NL)
14420   IF TOP(I,J,NL) = 0 OR INT(Z1(II)) <> NL THEN LINE (X1(II),BOT(I,J,NL)+THICK(I,J,NL)+5.)-
(X1(II),ZK):LINE (X1(II),ZK)-(X1(II)+DELX/50,ZK):LINE (X1(II)+DELX/50,ZK)-
(X1(II)+DELX/50,BOT(I,J,NL)+THICK(I,J,NL)+5.)
14430   IF TOP(I,J,NL) >= BOT(I,J,NL) THEN RAD=(BOT(I,J,NL)+THICK(I,J,NL)-TOP(I,J,NL)):LINE (X1(II)-
DELX/4,BOT(I,J,NL)+THICK(I,J,NL))-(X1(II),BOT(I,J,NL)+THICK(I,J,NL)-RAD):LINE -
(X1(II)+DELX/4,BOT(I,J,NL)+THICK(I,J,NL))
14440 NEXT II
      Z1$=""
      TEMPSCR=0:REM RESET SCREEN FLAG FOR PLAN VIEW WITH DEFAULT COORDINATES (MAPTYPE$="A")14450
RETURN
14442 Z1$=INKEY$:REM DO NOT CHANGE SCREEN UNTIL USER HITS KEY
      IF Z1$="g" or z1$="G" or z1$=chr$(13) THEN RETURN ELSE GOTO 14442
      RETURN
14500 REM ***************************************************
14510 REM  PLOT COLUMN CROSS SECTION ON SCREEN
14520 REM ***************************************************
14530 REM
14540 REM   ZASPECT - HORIZONTAL/VERTICAL SCREEN RATIO - THE INVERSE
14550 REM              OF THE VERTICAL EXAGGERATION RATIO
14560 REM   I1!,J1!, - COORDINATES OF LOWER LEFT CORNER OF SCREEN
14570 REM   ICOLXS - SELCTED ROW TO PLOT ON SCREEN
14580 REM   LLY,LLZ - ENTERED COORDINATES OF LOWER LEFT CORNER OF SCREEN
      IF J2!<=J1! THEN J2!=J1!+NR*DELY
14610 I1!=J1!:I2!=J2!:REM TRANSPOSE SCREEN FOR COLUMNS
      IF SCALEDEF<=0 THEN SCALEDEF=NC*DELX
      J1!=LLZ:  REM  COORDINATES OF LOWER LEFT CORNER OF SCREEN
      J2!=LLZ+SCALEDEF/ZASPECT
14615 GOSUB 7880:REM CALL ROUTINE TO SET UP SCREEN WINDOW
14620 I=ICOLXS
14630 FOR K=1 TO NL-1
        JSTART=0
14640   JSTART=JSTART+1
        IF THICK(I,JSTART,K)<0 THEN GOTO 14640
        PSET (LLY+DELY/2,BOT(I,JSTART,K)):REM SET BEGINNING POINT OF BOTTOM
14650   FOR J=JSTART TO NR :REM    PLOT BOTTOM OF LAYER
14660     DIST=LLY+DELY*J
          IF THICK(I,J,K) < 0 THEN GOTO 14680
14670     LINE -(DIST-DELY/2,BOT(I,J,K))
14680   NEXT J
14690   PSET (LLY+DELY/2,TOP(I,JSTART,K)):REM SET BEGINNING POINT OF TOP
14700   FOR J=JSTART TO NR :REM    PLOT TOP OF LAYER
14710     DIST=LLY+DELY*J
          IF THICK(I,J,K) < 0 THEN GOTO 14730
14720     LINE -(DIST-DELY/2,TOP(I,J,K))
14730   NEXT J
14740 NEXT K
14750   PSET (LLY+DELY/2,BOT(I,1,NL)):REM SET BEGINNING POINT OF BOTTOM OF TOP LAYER
14760   FOR J=2 TO NR :REM    PLOT BOTTOM OF TOP LAYER
14770     DIST=LLY+DELY*J
14780     LINE -(DIST-DELY/2,BOT(I,J,NL))
14790   NEXT J
        JSTART=0
14800   JSTART=ISTART+1
        IF THICK(I,JSTART,K)<0 THEN GOTO 14640
        PSET(LLY+DELY/2,BOT(I,JSTART,NL)+THICK(I,JSTART,NL)):REM SET BEGINNING OF WATER TABLE TOP
14810 FOR J=JSTART TO NR :REM    PLOT BOTTOM OF TOP LAYER
14820   DIST=LLY+DELY*J
        IF THICK(I,J,K) < 0 THEN GOTO 14840
14830   LINE -(DIST-DELY/2,BOT(I,J,NL)+THICK(I,J,NL))
14840 NEXT J
14850 FOR II=1 TO NS :REM PLOT WELLS AND RIVERS ON SCREEN
14860   IF INT((X1(II)-LLX)/DELX+1.0) <> I THEN 14940:REM ONLY PLOT WELLS IN THIS RO
```

```
14870   J=INT((Y1(II)-LLY)/DELY+1.0):REM FIND ROW OF WELL
14880 REM  FOR KLK=1 TO NL-1:REM FIND LAYER OF WELL
14890 REM    IF Z1(II)>=BOT(I,J,KLK) AND Z1(II) <TOP(I,J,KLK) THEN ZK=BOT(I,J,KLK)
14900 REM  NEXT KLK
      ZK=BOT(I,J,INT(Z1(II))):REM FIND LAYER OF WELL
14910 REM  IF Z1(II)>=BOT(I,J,NL) AND Z1(II)<(BOT(I,J,NL)+THICK(I,J,NL)) THEN ZK=BOT(I,J,NL)
14920   IF TOP(I,J,NL) = 0 OR INT(Z1(II)) <> NL THEN LINE (Y1(II),BOT(I,J,NL)+THICK(I,J,NL)+5.)-
(Y1(II),ZK):LINE (Y1(II),ZK)-(Y1(II)+DELY/50,ZK):LINE (Y1(II)+DELY/50,ZK)-
(Y1(II)+DELY/50,BOT(I,J,NL)+THICK(I,J,NL)+5.)
14930   IF TOP(I,J,NL) >= BOT(I,J,NL) THEN RAD=(BOT(I,J,NL)+THICK(I,J,NL)-TOP(I,J,NL)):LINE (Y1(II)-
DELY/4,BOT(I,J,NL)+THICK(I,J,NL))-(Y1(II),BOT(I,J,NL)+THICK(I,J,NL)-RAD):LINE -
(Y1(II)+DELY/4,BOT(I,J,NL)+THICK(I,J,NL))
14940 NEXT II
      Z1$=""
      TEMPSCR=0:REM RESET SCREEN FLAG FOR PLAN VIEW WITH DEFAULT COORDINATES (MAPTYPE$="A")
14942 Z1$=INKEY$:REM DO NOT CHANGE SCREEN UNTIL USER HITS "G"
      IF Z1$="g" or z1$="G" or z1$=chr$(13) THEN RETURN ELSE GOTO 14942
14950 RETURN
15000 rem **************************************************************
15010 rem   profile row setup and row selection
15020 rem **************************************************************
15030 rem   jrowxs - row to profile
15100 if jrowxs<=0 then jrowxs=int(((j2!-j1!)/2-lly)/dely+1.0)
      IF JROWXS<1 THEN JROWXS=1:REM CHECK POSITION OF ROW, KEEP ON GRID
      IF JROWXS>NR THEN JROWXS=NR
15110 z1$=inkey$
15120 y1x=(jrowxs*dely)+lly-dely/2:rem  compute position of middle of selected row
15130 locate 1,32:print "row=";jrowxs
      IF JROWXS<1 THEN JROWXS=1:REM CHECK POSITION OF ROW, KEEP ON GRID
      IF JROWXS>NR THEN JROWXS=NR
15140 line (i1!,y1x-dely/4)-(i2!,y1x+dely/4),2,b
15150 if z1$=chr$(13) then return :rem exit routine on carriage return with row selected
15160 if z1$=chr$(0)+chr$(72) then line (i1!,y1x-dely/4)-(i2!,y1x+dely/4),0,b:jrowxs=jrowxs+1
15165 if z1$=chr$(0)+chr$(80) then line (i1!,y1x-dely/4)-(i2!,y1x+dely/4),0,b:jrowxs=jrowxs-1
15170 goto 15110:rem return to beginning of subroutine
15200 rem **************************************************************
15210 rem   profile column setup and row selection
15220 rem **************************************************************
15230 rem   icolxs - column to profile
15300 if icolxs<=0 then icolxs=int(((i2!-i1!)/2-llx)/delx+1.0)
      if icolxs<1 then icolxs=1:rem check position of col and keep on grid
      if icolxs>nc then icolxs=nc
15310 z1$=inkey$
15320 x1x=(icolxs*delx)+llx-delx/2:rem  compute position of middle of selected row
15330 locate 1,32:print "col=";icolxs
      if icolxs<1 then icolxs=1:rem check position of col and keep on grid
      if icolxs>nc then icolxs=nc
15340 line (x1x-delx/4,j1!)-(x1x+delx/4,j2!),2,b
15350 if z1$=chr$(13) then return :rem exit routine on carriage return with row·selected
15360 if z1$=chr$(0)+chr$(77) then line (x1x-delx/4,j1!)-(x1x+delx/4,j2!),0,b:icolxs=icolxs+1
15365 if z1$=chr$(0)+chr$(75) then line (x1x-delx/4,j1!)-(x1x+delx/4,j2!),0,b:icolxs=icolxs-1
15370 goto 15310:rem return to beginning of subroutine
```

APPENDIX E

Typical Model Output

```
  RANDOM WALK 3-D MASS TRANSPORT MODEL WITH GEOGRAPHIC FEATURE OVERLAYS
                                BY
                           DONALD KOCH
                   ENGINEERING TECHNOLOGIES ASSOCIATES
                    3458 ELLICOTT CENTER DRIVE
                    ELLICOTT CITY, MD,  21043

                    PHONE: 301-461-9920



   READING IN PLUME DATA FROM EXTERNAL FILE PICCLV.DAT

   INPUT VELOCITY FILE CLV101.RND
       PRESENT SIMULATION TIME =  0   DAYS
        INCREMENTAL SIMULATION TIME =  60.8333  DAYS
 DMAX =  10  FT   ZMAX =  1  FT

 NP= 7827
 CONCENTRATION IN PUMPED WELL NUMBER  2 , IN PPM, IS=  1.174009
 MASS EXITING IN PUMPED WELL NUMBER  2 , IN LBS, IS=  68.63529
 CONCENTRATION IN PUMPED WELL NUMBER  3 , IN PPM, IS=  8.854937E-02
 MASS EXITING IN PUMPED WELL NUMBER  3 , IN LBS, IS=  2.329561
 CONCENTRATION IN PUMPED WELL NUMBER  10 , IN PPM, IS=  .0249045
 MASS EXITING IN PUMPED WELL NUMBER  10 , IN LBS, IS=  .12942
 CONCENTRATION IN PUMPED WELL NUMBER  11 , IN PPM, IS=  9.02222E-03
 MASS EXITING IN PUMPED WELL NUMBER  11 , IN LBS, IS=  .04314
 CONCENTRATION IN PUMPED WELL NUMBER  12 , IN PPM, IS=  2.068306E-02
 MASS EXITING IN PUMPED WELL NUMBER  12 , IN LBS, IS=  .08628
 CONCENTRATION IN PUMPED WELL NUMBER  13 , IN PPM, IS=  .0953318
 MASS EXITING IN PUMPED WELL NUMBER  13 , IN LBS, IS=  .4314
 CONCENTRATION IN PUMPED WELL NUMBER  14 , IN PPM, IS=  .27379
 MASS EXITING IN PUMPED WELL NUMBER  14 , IN LBS, IS=  1.33734
 CONCENTRATION IN PUMPED WELL NUMBER  15 , IN PPM, IS=  .1436282
 MASS EXITING IN PUMPED WELL NUMBER  15 , IN LBS, IS=  .60396
 CONCENTRATION IN PUMPED WELL NUMBER  16 , IN PPM, IS=  .2243432
 MASS EXITING IN PUMPED WELL NUMBER  16 , IN LBS, IS=  1.03536
 CONCENTRATION IN PUMPED WELL NUMBER  17 , IN PPM, IS=  6.589717E-02
 MASS EXITING IN PUMPED WELL NUMBER  17 , IN LBS, IS=  .38826
 CONCENTRATION IN PUMPED WELL NUMBER  18 , IN PPM, IS=  4.821198E-02
 MASS EXITING IN PUMPED WELL NUMBER  18 , IN LBS, IS=  .2157
 CONCENTRATION IN PUMPED WELL NUMBER  19 , IN PPM, IS=  2.017261E-02
 MASS EXITING IN PUMPED WELL NUMBER  19 , IN LBS, IS=  .08628



   INPUT VELOCITY FILE CLV102.RND
       PRESENT SIMULATION TIME =  60.8333   DAYS
        INCREMENTAL SIMULATION TIME =  60.8333  DAYS
 DMAX =  10  FT   ZMAX =  1  FT

 NP= 6182
 CONCENTRATION IN PUMPED WELL NUMBER  2 , IN PPM, IS=  1.151134
 MASS EXITING IN PUMPED WELL NUMBER  2 , IN LBS, IS=  67.29797
 CONCENTRATION IN PUMPED WELL NUMBER  3 , IN PPM, IS=  7.543092E-02
 MASS EXITING IN PUMPED WELL NUMBER  3 , IN LBS, IS=  1.98444
 CONCENTRATION IN PUMPED WELL NUMBER  8 , IN PPM, IS=  .0092309
 MASS EXITING IN PUMPED WELL NUMBER  8 , IN LBS, IS=  .04314
 CONCENTRATION IN PUMPED WELL NUMBER  11 , IN PPM, IS=  9.759743E-03
 MASS EXITING IN PUMPED WELL NUMBER  11 , IN LBS, IS=  .04314
```

```
CONCENTRATION IN PUMPED WELL NUMBER  14 , IN PPM, IS=  .1240565
MASS EXITING IN PUMPED WELL NUMBER  14 , IN LBS, IS=  .56082
CONCENTRATION IN PUMPED WELL NUMBER  15 , IN PPM, IS=  6.710225E-02
MASS EXITING IN PUMPED WELL NUMBER  15 , IN LBS, IS=  .25884
CONCENTRATION IN PUMPED WELL NUMBER  16 , IN PPM, IS=  7.041197E-02
MASS EXITING IN PUMPED WELL NUMBER  16 , IN LBS, IS=  .30198
CONCENTRATION IN PUMPED WELL NUMBER  17 , IN PPM, IS=  7.720923E-03
MASS EXITING IN PUMPED WELL NUMBER  17 , IN LBS, IS=  .04314
CONCENTRATION IN PUMPED WELL NUMBER  18 , IN PPM, IS=  3.309396E-02
MASS EXITING IN PUMPED WELL NUMBER  18 , IN LBS, IS=  .12942



  INPUT VELOCITY FILE CLV103.RND
      PRESENT SIMULATION TIME =  121.6666   DAYS
       INCREMENTAL SIMULATION TIME =  60.8333  DAYS
DMAX =  10  FT   ZMAX =  1  FT

NP= 4948
CONCENTRATION IN PUMPED WELL NUMBER  1 , IN PPM, IS=  3.279605E-03
MASS EXITING IN PUMPED WELL NUMBER  1 , IN LBS, IS=  4.314001E-02
CONCENTRATION IN PUMPED WELL NUMBER  2 , IN PPM, IS=  .8699918
MASS EXITING IN PUMPED WELL NUMBER  2 , IN LBS, IS=  50.86175
CONCENTRATION IN PUMPED WELL NUMBER  3 , IN PPM, IS=  3.771545E-02
MASS EXITING IN PUMPED WELL NUMBER  3 , IN LBS, IS=  .9922199
CONCENTRATION IN PUMPED WELL NUMBER  11 , IN PPM, IS=  2.003237E-02
MASS EXITING IN PUMPED WELL NUMBER  11 , IN LBS, IS=  .08628
CONCENTRATION IN PUMPED WELL NUMBER  13 , IN PPM, IS=  2.131044E-02
MASS EXITING IN PUMPED WELL NUMBER  13 , IN LBS, IS=  .08628
CONCENTRATION IN PUMPED WELL NUMBER  14 , IN PPM, IS=  6.847543E-02
MASS EXITING IN PUMPED WELL NUMBER  14 , IN LBS, IS=  .30198
CONCENTRATION IN PUMPED WELL NUMBER  15 , IN PPM, IS=  9.201175E-02
MASS EXITING IN PUMPED WELL NUMBER  15 , IN LBS, IS=  .34512
CONCENTRATION IN PUMPED WELL NUMBER  16 , IN PPM, IS=  5.151961E-02
MASS EXITING IN PUMPED WELL NUMBER  16 , IN LBS, IS=  .2157
CONCENTRATION IN PUMPED WELL NUMBER  17 , IN PPM, IS=  2.361315E-02
MASS EXITING IN PUMPED WELL NUMBER  17 , IN LBS, IS=  .12942
CONCENTRATION IN PUMPED WELL NUMBER  18 , IN PPM, IS=  3.448108E-02
MASS EXITING IN PUMPED WELL NUMBER  18 , IN LBS, IS=  .12942
CONCENTRATION IN PUMPED WELL NUMBER  19 , IN PPM, IS=  1.145189E-02
MASS EXITING IN PUMPED WELL NUMBER  19 , IN LBS, IS=  .04314



  INPUT VELOCITY FILE CLV104.RND
      PRESENT SIMULATION TIME =  182.4999   DAYS
       INCREMENTAL SIMULATION TIME =  60.8333  DAYS
DMAX =  10  FT   ZMAX =  1  FT

NP= 4009
CONCENTRATION IN PUMPED WELL NUMBER  2 , IN PPM, IS=  .6559992
MASS EXITING IN PUMPED WELL NUMBER  2 , IN LBS, IS=  38.35124
CONCENTRATION IN PUMPED WELL NUMBER  3 , IN PPM, IS=  4.591446E-02
MASS EXITING IN PUMPED WELL NUMBER  3 , IN LBS, IS=  1.20792
CONCENTRATION IN PUMPED WELL NUMBER  11 , IN PPM, IS=  1.011029E-02
MASS EXITING IN PUMPED WELL NUMBER  11 , IN LBS, IS=  .04314
CONCENTRATION IN PUMPED WELL NUMBER  12 , IN PPM, IS=  1.178857E-02
MASS EXITING IN PUMPED WELL NUMBER  12 , IN LBS, IS=  4.314001E-02
CONCENTRATION IN PUMPED WELL NUMBER  13 , IN PPM, IS=  1.076179E-02
MASS EXITING IN PUMPED WELL NUMBER  13 , IN LBS, IS=  .04314
```

```
CONCENTRATION IN PUMPED WELL NUMBER  16 , IN PPM, IS=  5.197955E-02
MASS EXITING IN PUMPED WELL NUMBER  16 , IN LBS, IS=  .2157
CONCENTRATION IN PUMPED WELL NUMBER  17 , IN PPM, IS=  7.929791E-03
MASS EXITING IN PUMPED WELL NUMBER  17 , IN LBS, IS=  .04314
CONCENTRATION IN PUMPED WELL NUMBER  18 , IN PPM, IS=  2.331844E-02
MASS EXITING IN PUMPED WELL NUMBER  18 , IN LBS, IS=  .08628
CONCENTRATION IN PUMPED WELL NUMBER  19 , IN PPM, IS=  1.157192E-02
MASS EXITING IN PUMPED WELL NUMBER  19 , IN LBS, IS=  .04314



  INPUT VELOCITY FILE CLV105.RND
      PRESENT SIMULATION TIME =  243.3332   DAYS
       INCREMENTAL SIMULATION TIME =  60.8333  DAYS
DMAX =  10  FT   ZMAX =  1  FT

NP= 3308
CONCENTRATION IN PUMPED WELL NUMBER  2 , IN PPM, IS=  .4884946
MASS EXITING IN PUMPED WELL NUMBER  2 , IN LBS, IS=  28.55853
CONCENTRATION IN PUMPED WELL NUMBER  3 , IN PPM, IS=  2.951644E-02
MASS EXITING IN PUMPED WELL NUMBER  3 , IN LBS, IS=  .7765199
CONCENTRATION IN PUMPED WELL NUMBER  11 , IN PPM, IS=  1.015256E-02
MASS EXITING IN PUMPED WELL NUMBER  11 , IN LBS, IS=  .04314
CONCENTRATION IN PUMPED WELL NUMBER  12 , IN PPM, IS=  1.184606E-02
MASS EXITING IN PUMPED WELL NUMBER  12 , IN LBS, IS=  .04314
CONCENTRATION IN PUMPED WELL NUMBER  13 , IN PPM, IS=  1.080282E-02
MASS EXITING IN PUMPED WELL NUMBER  13 , IN LBS, IS=  .04314
CONCENTRATION IN PUMPED WELL NUMBER  14 , IN PPM, IS=  4.953232E-02
MASS EXITING IN PUMPED WELL NUMBER  14 , IN LBS, IS=  .2157
CONCENTRATION IN PUMPED WELL NUMBER  15 , IN PPM, IS=  9.333776E-02
MASS EXITING IN PUMPED WELL NUMBER  15 , IN LBS, IS=  .34512
CONCENTRATION IN PUMPED WELL NUMBER  16 , IN PPM, IS=  2.085047E-02
MASS EXITING IN PUMPED WELL NUMBER  16 , IN LBS, IS=  .08628
CONCENTRATION IN PUMPED WELL NUMBER  17 , IN PPM, IS=  1.588923E-02
MASS EXITING IN PUMPED WELL NUMBER  17 , IN LBS, IS=  .08628
CONCENTRATION IN PUMPED WELL NUMBER  18 , IN PPM, IS=  1.172515E-02
MASS EXITING IN PUMPED WELL NUMBER  18 , IN LBS, IS=  .04314



  INPUT VELOCITY FILE CLV106.RND
      PRESENT SIMULATION TIME =  304.1665   DAYS
       INCREMENTAL SIMULATION TIME =  60.8333  DAYS
DMAX =  10  FT   ZMAX =  1  FT

NP= 2792
CONCENTRATION IN PUMPED WELL NUMBER  2 , IN PPM, IS=  .3637886
MASS EXITING IN PUMPED WELL NUMBER  2 , IN LBS, IS=  21.26793
CONCENTRATION IN PUMPED WELL NUMBER  3 , IN PPM, IS=  2.951644E-02
MASS EXITING IN PUMPED WELL NUMBER  3 , IN LBS, IS=  .7765199
CONCENTRATION IN PUMPED WELL NUMBER  12 , IN PPM, IS=  1.186094E-02
MASS EXITING IN PUMPED WELL NUMBER  12 , IN LBS, IS=  4.314001E-02
CONCENTRATION IN PUMPED WELL NUMBER  14 , IN PPM, IS=  2.977143E-02
MASS EXITING IN PUMPED WELL NUMBER  14 , IN LBS, IS=  .12942
CONCENTRATION IN PUMPED WELL NUMBER  18 , IN PPM, IS=  1.174784E-02
MASS EXITING IN PUMPED WELL NUMBER  18 , IN LBS, IS=  .04314
```

```
 CONCENTRATION MAP IN PPB  (P SIGNIFIES PUMPAGE, I SIGNIFIES INJECTION)
LOWER LEFT-HAND CORNER OF PLOT POSITION =  2090          840
GRID SIZE OF PLOT MAP, IN FEET =  60
NUMBER OF ROWS AND COLUMNS OF PLOT MAP =  32             21

Plot file name is =CL13.DAT
LOWER LEFT-HAND CORNER OF PLOT POSITION =  2090          840
GRID SIZE OF PLOT MAP, IN FEET =  60
NUMBER OF ROWS AND COLUMNS OF PLOT MAP =  32             21

Plot file name is =CL12.DAT
  INPUT VELOCITY FILE CLV201.RND
      PRESENT SIMULATION TIME =  364.9998   DAYS
       INCREMENTAL SIMULATION TIME =  60.8333  DAYS
DMAX =  10  FT   ZMAX =  1  FT

NP= 2490
CONCENTRATION IN PUMPED WELL NUMBER  2 , IN PPM, IS=  .28139
MASS EXITING IN PUMPED WELL NUMBER  2 , IN LBS, IS=  12.33804
CONCENTRATION IN PUMPED WELL NUMBER  3 , IN PPM, IS=  1.311842E-02
MASS EXITING IN PUMPED WELL NUMBER  3 , IN LBS, IS=  .34512
CONCENTRATION IN PUMPED WELL NUMBER  13 , IN PPM, IS=  1.033524E-02
MASS EXITING IN PUMPED WELL NUMBER  13 , IN LBS, IS=  4.314001E-02
CONCENTRATION IN PUMPED WELL NUMBER  14 , IN PPM, IS=  .0095225
MASS EXITING IN PUMPED WELL NUMBER  14 , IN LBS, IS=  4.314001E-02
CONCENTRATION IN PUMPED WELL NUMBER  15 , IN PPM, IS=  3.355112E-02
MASS EXITING IN PUMPED WELL NUMBER  15 , IN LBS, IS=  .12942
CONCENTRATION IN PUMPED WELL NUMBER  16 , IN PPM, IS=  2.018458E-02
MASS EXITING IN PUMPED WELL NUMBER  16 , IN LBS, IS=  .08628
CONCENTRATION IN PUMPED WELL NUMBER  18 , IN PPM, IS=  1.105715E-02
MASS EXITING IN PUMPED WELL NUMBER  18 , IN LBS, IS=  .04314



  INPUT VELOCITY FILE CLV202.RND
      PRESENT SIMULATION TIME =  425.8331   DAYS
       INCREMENTAL SIMULATION TIME =  60.8333  DAYS
DMAX =  10  FT   ZMAX =  1  FT

NP= 2249
CONCENTRATION IN PUMPED WELL NUMBER  2 , IN PPM, IS=  .2262924
MASS EXITING IN PUMPED WELL NUMBER  2 , IN LBS, IS=  9.922184
CONCENTRATION IN PUMPED WELL NUMBER  3 , IN PPM, IS=  1.311842E-02
MASS EXITING IN PUMPED WELL NUMBER  3 , IN LBS, IS=  .34512
CONCENTRATION IN PUMPED WELL NUMBER  14 , IN PPM, IS=  1.879446E-02
MASS EXITING IN PUMPED WELL NUMBER  14 , IN LBS, IS=  8.628001E-02
CONCENTRATION IN PUMPED WELL NUMBER  15 , IN PPM, IS=  1.102417E-02
MASS EXITING IN PUMPED WELL NUMBER  15 , IN LBS, IS=  .04314



  INPUT VELOCITY FILE CLV203.RND
      PRESENT SIMULATION TIME =  486.6664   DAYS
       INCREMENTAL SIMULATION TIME =  60.8333  DAYS
DMAX =  10  FT   ZMAX =  1  FT

NP= 2025
CONCENTRATION IN PUMPED WELL NUMBER  2 , IN PPM, IS=  .2085825
MASS EXITING IN PUMPED WELL NUMBER  2 , IN LBS, IS=  9.145666
CONCENTRATION IN PUMPED WELL NUMBER  3 , IN PPM, IS=  9.838814E-03
```

```
MASS EXITING IN PUMPED WELL NUMBER  15 , IN LBS, IS=  .08628
CONCENTRATION IN PUMPED WELL NUMBER  17 , IN PPM, IS=  7.692974E-03
MASS EXITING IN PUMPED WELL NUMBER  17 , IN LBS, IS=  .04314



  INPUT VELOCITY FILE CLV204.RND
      PRESENT SIMULATION TIME =  547.4998   DAYS
       INCREMENTAL SIMULATION TIME =  60.8333  DAYS
DMAX =  10  FT   ZMAX =  1  FT

NP= 1856
CONCENTRATION IN PUMPED WELL NUMBER  2 , IN PPM, IS=  .1495498
MASS EXITING IN PUMPED WELL NUMBER  2 , IN LBS, IS=  6.557272
CONCENTRATION IN PUMPED WELL NUMBER  3 , IN PPM, IS=  2.295723E-02
MASS EXITING IN PUMPED WELL NUMBER  3 , IN LBS, IS=  .6039599
CONCENTRATION IN PUMPED WELL NUMBER  13 , IN PPM, IS=  2.025673E-02
MASS EXITING IN PUMPED WELL NUMBER  13 , IN LBS, IS=  .08628
CONCENTRATION IN PUMPED WELL NUMBER  18 , IN PPM, IS=  1.074411E-02
MASS EXITING IN PUMPED WELL NUMBER  18 , IN LBS, IS=  .04314



  INPUT VELOCITY FILE CLV205.RND
      PRESENT SIMULATION TIME =  608.3331   DAYS
       INCREMENTAL SIMULATION TIME =  60.8333  DAYS
DMAX =  10  FT   ZMAX =  1  FT

NP= 1710
CONCENTRATION IN PUMPED WELL NUMBER  1 , IN PPM, IS=  3.279605E-03
MASS EXITING IN PUMPED WELL NUMBER  1 , IN LBS, IS=  4.314001E-02
CONCENTRATION IN PUMPED WELL NUMBER  2 , IN PPM, IS=  .1269206
MASS EXITING IN PUMPED WELL NUMBER  2 , IN LBS, IS=  5.565055
CONCENTRATION IN PUMPED WELL NUMBER  3 , IN PPM, IS=  2.295723E-02
MASS EXITING IN PUMPED WELL NUMBER  3 , IN LBS, IS=  .6039599
CONCENTRATION IN PUMPED WELL NUMBER  13 , IN PPM, IS=  1.012836E-02
MASS EXITING IN PUMPED WELL NUMBER  13 , IN LBS, IS=  .04314
CONCENTRATION IN PUMPED WELL NUMBER  18 , IN PPM, IS=  1.073868E-02
MASS EXITING IN PUMPED WELL NUMBER  18 , IN LBS, IS=  .04314



  INPUT VELOCITY FILE CLV206.RND
      PRESENT SIMULATION TIME =  669.1664   DAYS
       INCREMENTAL SIMULATION TIME =  60.8333  DAYS
DMAX =  10  FT   ZMAX =  1  FT

NP= 1581
CONCENTRATION IN PUMPED WELL NUMBER  2 , IN PPM, IS=  .1111785
MASS EXITING IN PUMPED WELL NUMBER  2 , IN LBS, IS=  4.874816
CONCENTRATION IN PUMPED WELL NUMBER  3 , IN PPM, IS=  1.639802E-02
MASS EXITING IN PUMPED WELL NUMBER  3 , IN LBS, IS=  .4314
CONCENTRATION IN PUMPED WELL NUMBER  12 , IN PPM, IS=  1.103132E-02
MASS EXITING IN PUMPED WELL NUMBER  12 , IN LBS, IS=  4.314001E-02
CONCENTRATION IN PUMPED WELL NUMBER  13 , IN PPM, IS=  1.012233E-02
MASS EXITING IN PUMPED WELL NUMBER  13 , IN LBS, IS=  .04314
CONCENTRATION IN PUMPED WELL NUMBER  14 , IN PPM, IS=  9.346603E-03
MASS EXITING IN PUMPED WELL NUMBER  14 , IN LBS, IS=  .04314
CONCENTRATION IN PUMPED WELL NUMBER  15 , IN PPM, IS=  2.190629E-02
```

```
LOWER LEFT-HAND CORNER OF PLOT POSITION =  2090          840
GRID SIZE OF PLOT MAP, IN FEET =  60
NUMBER OF ROWS AND COLUMNS OF PLOT MAP =  32             21

Plot file name is =CL23.DAT
  INPUT VELOCITY FILE CLV301.RND
      PRESENT SIMULATION TIME =  729.9997   DAYS
       INCREMENTAL SIMULATION TIME =  60.8333  DAYS
DMAX =  10  FT   ZMAX =  1  FT

NP= 1476
CONCENTRATION IN PUMPED WELL NUMBER  2 , IN PPM, IS=  9.510852E-02
MASS EXITING IN PUMPED WELL NUMBER  2 , IN LBS, IS=  3.75318
CONCENTRATION IN PUMPED WELL NUMBER  3 , IN PPM, IS=  1.147862E-02
MASS EXITING IN PUMPED WELL NUMBER  3 , IN LBS, IS=  .30198
CONCENTRATION IN PUMPED WELL NUMBER  11 , IN PPM, IS=  9.444213E-03
MASS EXITING IN PUMPED WELL NUMBER  11 , IN LBS, IS=  .04314
CONCENTRATION IN PUMPED WELL NUMBER  13 , IN PPM, IS=  1.996181E-02
MASS EXITING IN PUMPED WELL NUMBER  13 , IN LBS, IS=  .08628
CONCENTRATION IN PUMPED WELL NUMBER  14 , IN PPM, IS=  2.770774E-02
MASS EXITING IN PUMPED WELL NUMBER  14 , IN LBS, IS=  .12942
CONCENTRATION IN PUMPED WELL NUMBER  15 , IN PPM, IS=  1.081243E-02
MASS EXITING IN PUMPED WELL NUMBER  15 , IN LBS, IS=  .04314
CONCENTRATION IN PUMPED WELL NUMBER  16 , IN PPM, IS=  9.810428E-03
MASS EXITING IN PUMPED WELL NUMBER  16 , IN LBS, IS=  .04314
CONCENTRATION IN PUMPED WELL NUMBER  18 , IN PPM, IS=  .031613
MASS EXITING IN PUMPED WELL NUMBER  18 , IN LBS, IS=  .12942



  INPUT VELOCITY FILE CLV302.RND
      PRESENT SIMULATION TIME =  790.833   DAYS
       INCREMENTAL SIMULATION TIME =  60.8333  DAYS
DMAX =  10  FT   ZMAX =  1  FT

NP= 1379
CONCENTRATION IN PUMPED WELL NUMBER  2 , IN PPM, IS=  9.073572E-02
MASS EXITING IN PUMPED WELL NUMBER  2 , IN LBS, IS=  3.58062
CONCENTRATION IN PUMPED WELL NUMBER  3 , IN PPM, IS=  9.838814E-03
MASS EXITING IN PUMPED WELL NUMBER  3 , IN LBS, IS=  .25884
CONCENTRATION IN PUMPED WELL NUMBER  15 , IN PPM, IS=  .0215263
MASS EXITING IN PUMPED WELL NUMBER  15 , IN LBS, IS=  .08628
CONCENTRATION IN PUMPED WELL NUMBER  16 , IN PPM, IS=  4.889414E-02
MASS EXITING IN PUMPED WELL NUMBER  16 , IN LBS, IS=  .2157
CONCENTRATION IN PUMPED WELL NUMBER  17 , IN PPM, IS=  7.603521E-03
MASS EXITING IN PUMPED WELL NUMBER  17 , IN LBS, IS=  .04314



  INPUT VELOCITY FILE CLV303.RND
      PRESENT SIMULATION TIME =  851.6663   DAYS
       INCREMENTAL SIMULATION TIME =  60.8333  DAYS
DMAX =  10  FT   ZMAX =  1  FT

NP= 1300
CONCENTRATION IN PUMPED WELL NUMBER  2 , IN PPM, IS=  .0732445
MASS EXITING IN PUMPED WELL NUMBER  2 , IN LBS, IS=  2.89038
```

```
CONCENTRATION IN PUMPED WELL NUMBER  16 , IN PPM, IS=  2.929268E-02
MASS EXITING IN PUMPED WELL NUMBER  16 , IN LBS, IS=  .12942
CONCENTRATION IN PUMPED WELL NUMBER  18 , IN PPM, IS=  1.045472E-02
MASS EXITING IN PUMPED WELL NUMBER  18 , IN LBS, IS=  .04314



  INPUT VELOCITY FILE CLV304.RND
      PRESENT SIMULATION TIME =  912.4996   DAYS
       INCREMENTAL SIMULATION TIME =  60.8333  DAYS
DMAX =  10  FT   ZMAX =  1  FT

NP= 1223
CONCENTRATION IN PUMPED WELL NUMBER  2 , IN PPM, IS=  .0776173
MASS EXITING IN PUMPED WELL NUMBER  2 , IN LBS, IS=  3.06294
CONCENTRATION IN PUMPED WELL NUMBER  3 , IN PPM, IS=  3.279605E-03
MASS EXITING IN PUMPED WELL NUMBER  3 , IN LBS, IS=  8.628001E-02
CONCENTRATION IN PUMPED WELL NUMBER  13 , IN PPM, IS=  9.92381E-03
MASS EXITING IN PUMPED WELL NUMBER  13 , IN LBS, IS=  .04314
CONCENTRATION IN PUMPED WELL NUMBER  14 , IN PPM, IS=  9.181052E-03
MASS EXITING IN PUMPED WELL NUMBER  14 , IN LBS, IS=  .04314
CONCENTRATION IN PUMPED WELL NUMBER  18 , IN PPM, IS=  2.090945E-02
MASS EXITING IN PUMPED WELL NUMBER  18 , IN LBS, IS=  .08628



  INPUT VELOCITY FILE CLV305.RND
      PRESENT SIMULATION TIME =  973.3329   DAYS
       INCREMENTAL SIMULATION TIME =  60.8333  DAYS
DMAX =  10  FT   ZMAX =  1  FT

NP= 1159
CONCENTRATION IN PUMPED WELL NUMBER  2 , IN PPM, IS=  6.012608E-02
MASS EXITING IN PUMPED WELL NUMBER  2 , IN LBS, IS=  2.3727
CONCENTRATION IN PUMPED WELL NUMBER  3 , IN PPM, IS=  9.838814E-03
MASS EXITING IN PUMPED WELL NUMBER  3 , IN LBS, IS=  .25884
CONCENTRATION IN PUMPED WELL NUMBER  14 , IN PPM, IS=  9.176097E-03
MASS EXITING IN PUMPED WELL NUMBER  14 , IN LBS, IS=  4.314001E-02
CONCENTRATION IN PUMPED WELL NUMBER  15 , IN PPM, IS=  1.074411E-02
MASS EXITING IN PUMPED WELL NUMBER  15 , IN LBS, IS=  .04314
CONCENTRATION IN PUMPED WELL NUMBER  18 , IN PPM, IS=  .0104483
MASS EXITING IN PUMPED WELL NUMBER  18 , IN LBS, IS=  .04314



  INPUT VELOCITY FILE CLV306.RND
      PRESENT SIMULATION TIME =  1034.166   DAYS
       INCREMENTAL SIMULATION TIME =  60.8333  DAYS
DMAX =  10  FT   ZMAX =  1  FT

NP= 1091
CONCENTRATION IN PUMPED WELL NUMBER  2 , IN PPM, IS=  6.887169E-02
MASS EXITING IN PUMPED WELL NUMBER  2 , IN LBS, IS=  2.71782
CONCENTRATION IN PUMPED WELL NUMBER  3 , IN PPM, IS=  3.279605E-03
MASS EXITING IN PUMPED WELL NUMBER  3 , IN LBS, IS=  8.628001E-02
CONCENTRATION IN PUMPED WELL NUMBER  14 , IN PPM, IS=  9.181052E-03
MASS EXITING IN PUMPED WELL NUMBER  14 , IN LBS, IS=  .04314
CONCENTRATION IN PUMPED WELL NUMBER  15 , IN PPM, IS=  1.074411E-02
MASS EXITING IN PUMPED WELL NUMBER  15 , IN LBS, IS=  .04314
```

```
LOWER LEFT-HAND CORNER OF PLOT POSITION =  2090          840
GRID SIZE OF PLOT MAP, IN FEET =  60
NUMBER OF ROWS AND COLUMNS OF PLOT MAP =  32             21

Plot file name is =CL33.DAT
LOWER LEFT-HAND CORNER OF PLOT POSITION =  2090          840
GRID SIZE OF PLOT MAP, IN FEET =  60
NUMBER OF ROWS AND COLUMNS OF PLOT MAP =  32             21

Plot file name is =CL32.DAT
LOWER LEFT-HAND CORNER OF PLOT POSITION =  2090          840
GRID SIZE OF PLOT MAP, IN FEET =  60
NUMBER OF ROWS AND COLUMNS OF PLOT MAP =  32             21

Plot file name is =CL35.DAT
      PRESENT SIMULATION TIME =  1095   DAYS
       INCREMENTAL SIMULATION TIME =  365  DAYS
DMAX =  10  FT   ZMAX =  1  FT

NP= 832
CONCENTRATION IN PUMPED WELL NUMBER  1 , IN PPM, IS=  1.093201E-03
MASS EXITING IN PUMPED WELL NUMBER  1 , IN LBS, IS=  8.628001E-02
CONCENTRATION IN PUMPED WELL NUMBER  2 , IN PPM, IS=  4.281705E-02
MASS EXITING IN PUMPED WELL NUMBER  2 , IN LBS, IS=  10.1379
CONCENTRATION IN PUMPED WELL NUMBER  3 , IN PPM, IS=  3.006303E-03
MASS EXITING IN PUMPED WELL NUMBER  3 , IN LBS, IS=  .4745401
CONCENTRATION IN PUMPED WELL NUMBER  11 , IN PPM, IS=  1.563612E-03
MASS EXITING IN PUMPED WELL NUMBER  11 , IN LBS, IS=  .04314
CONCENTRATION IN PUMPED WELL NUMBER  13 , IN PPM, IS=  1.653968E-03
MASS EXITING IN PUMPED WELL NUMBER  13 , IN LBS, IS=  4.314001E-02
CONCENTRATION IN PUMPED WELL NUMBER  14 , IN PPM, IS=  3.060349E-03
MASS EXITING IN PUMPED WELL NUMBER  14 , IN LBS, IS=  .08628
CONCENTRATION IN PUMPED WELL NUMBER  15 , IN PPM, IS=  8.953416E-03
MASS EXITING IN PUMPED WELL NUMBER  15 , IN LBS, IS=  .2157
CONCENTRATION IN PUMPED WELL NUMBER  16 , IN PPM, IS=  1.62737E-03
MASS EXITING IN PUMPED WELL NUMBER  16 , IN LBS, IS=  .04314
CONCENTRATION IN PUMPED WELL NUMBER  17 , IN PPM, IS=  1.26612E-03
MASS EXITING IN PUMPED WELL NUMBER  17 , IN LBS, IS=  .04314



      PRESENT SIMULATION TIME =  1460   DAYS
       INCREMENTAL SIMULATION TIME =  365  DAYS
DMAX =  10  FT   ZMAX =  1  FT

NP= 669
CONCENTRATION IN PUMPED WELL NUMBER  1 , IN PPM, IS=  5.466006E-04
MASS EXITING IN PUMPED WELL NUMBER  1 , IN LBS, IS=  4.314001E-02
CONCENTRATION IN PUMPED WELL NUMBER  2 , IN PPM, IS=  2.860543E-02
MASS EXITING IN PUMPED WELL NUMBER  2 , IN LBS, IS=  6.772982
CONCENTRATION IN PUMPED WELL NUMBER  3 , IN PPM, IS=  8.199008E-04
MASS EXITING IN PUMPED WELL NUMBER  3 , IN LBS, IS=  .12942
CONCENTRATION IN PUMPED WELL NUMBER  17 , IN PPM, IS=  1.26612E-03
MASS EXITING IN PUMPED WELL NUMBER  17 , IN LBS, IS=  .04314
CONCENTRATION IN PUMPED WELL NUMBER  18 , IN PPM, IS=  1.742453E-03
MASS EXITING IN PUMPED WELL NUMBER  18 , IN LBS, IS=  4.314001E-02
```

```
NP= 518
CONCENTRATION IN PUMPED WELL NUMBER  2 , IN PPM, IS=  2.714783E-02
MASS EXITING IN PUMPED WELL NUMBER  2 , IN LBS, IS=  6.427862
CONCENTRATION IN PUMPED WELL NUMBER  3 , IN PPM, IS=  5.466006E-04
MASS EXITING IN PUMPED WELL NUMBER  3 , IN LBS, IS=  8.628001E-02



LOWER LEFT-HAND CORNER OF PLOT POSITION =  2090          840
GRID SIZE OF PLOT MAP, IN FEET =  60
NUMBER OF ROWS AND COLUMNS OF PLOT MAP =  32             21

Plot file name is =CL63.DAT
LOWER LEFT-HAND CORNER OF PLOT POSITION =  2090          840
GRID SIZE OF PLOT MAP, IN FEET =  60
NUMBER OF ROWS AND COLUMNS OF PLOT MAP =  32             21

Plot file name is =CL65.DAT

  WRITING PLUME DATA TO EXTERNAL FILE CLVOUT6.DAT

      PRESENT SIMULATION TIME =  2190   DAYS
       INCREMENTAL SIMULATION TIME =  365  DAYS
DMAX =  10  FT   ZMAX =  1  FT

NP= 391
CONCENTRATION IN PUMPED WELL NUMBER  2 , IN PPM, IS=  2.295723E-02
MASS EXITING IN PUMPED WELL NUMBER  2 , IN LBS, IS=  5.435641
CONCENTRATION IN PUMPED WELL NUMBER  3 , IN PPM, IS=  2.733003E-04
MASS EXITING IN PUMPED WELL NUMBER  3 , IN LBS, IS=  4.314001E-02



      PRESENT SIMULATION TIME =  2555   DAYS
       INCREMENTAL SIMULATION TIME =  365  DAYS
DMAX =  10  FT   ZMAX =  1  FT

NP= 311
CONCENTRATION IN PUMPED WELL NUMBER  2 , IN PPM, IS=  1.421162E-02
MASS EXITING IN PUMPED WELL NUMBER  2 , IN LBS, IS=  3.364921
CONCENTRATION IN PUMPED WELL NUMBER  3 , IN PPM, IS=  2.733003E-04
MASS EXITING IN PUMPED WELL NUMBER  3 , IN LBS, IS=  4.314001E-02
CONCENTRATION IN PUMPED WELL NUMBER  10 , IN PPM, IS=  1.453415E-03
MASS EXITING IN PUMPED WELL NUMBER  10 , IN LBS, IS=  .04314



      PRESENT SIMULATION TIME =  2920   DAYS
       INCREMENTAL SIMULATION TIME =  365  DAYS
DMAX =  10  FT   ZMAX =  1  FT

NP= 272
CONCENTRATION IN PUMPED WELL NUMBER  2 , IN PPM, IS=  6.923608E-03
MASS EXITING IN PUMPED WELL NUMBER  2 , IN LBS, IS=  1.63932
CONCENTRATION IN PUMPED WELL NUMBER  4 , IN PPM, IS=  1.461511E-03
MASS EXITING IN PUMPED WELL NUMBER  4 , IN LBS, IS=  .04314
```

```
NUMBER OF ROWS AND COLUMNS OF PLOT MAP =  32            21

Plot file name is =CL93.DAT
      PRESENT SIMULATION TIME =  3285   DAYS
       INCREMENTAL SIMULATION TIME =  365  DAYS
DMAX =  10  FT   ZMAX =  1  FT

NP= 259
CONCENTRATION IN PUMPED WELL NUMBER  2 , IN PPM, IS=  2.004202E-03
MASS EXITING IN PUMPED WELL NUMBER  2 , IN LBS, IS=  .4745401
CONCENTRATION IN PUMPED WELL NUMBER  4 , IN PPM, IS=  2.923022E-03
MASS EXITING IN PUMPED WELL NUMBER  4 , IN LBS, IS=  .08628



      PRESENT SIMULATION TIME =  3650   DAYS
       INCREMENTAL SIMULATION TIME =  365  DAYS
DMAX =  10  FT   ZMAX =  1  FT

NP= 253
CONCENTRATION IN PUMPED WELL NUMBER  2 , IN PPM, IS=  5.466006E-04
MASS EXITING IN PUMPED WELL NUMBER  2 , IN LBS, IS=  .12942
CONCENTRATION IN PUMPED WELL NUMBER  4 , IN PPM, IS=  1.461511E-03
MASS EXITING IN PUMPED WELL NUMBER  4 , IN LBS, IS=  .04314
CONCENTRATION IN PUMPED WELL NUMBER  7 , IN PPM, IS=  1.647237E-03
MASS EXITING IN PUMPED WELL NUMBER  7 , IN LBS, IS=  .04314
CONCENTRATION IN PUMPED WELL NUMBER  10 , IN PPM, IS=  1.453415E-03
MASS EXITING IN PUMPED WELL NUMBER  10 , IN LBS, IS=  .04314



      PRESENT SIMULATION TIME =  4015   DAYS
       INCREMENTAL SIMULATION TIME =  365  DAYS
DMAX =  10  FT   ZMAX =  1  FT

NP= 247
CONCENTRATION IN PUMPED WELL NUMBER  2 , IN PPM, IS=  9.110009E-04
MASS EXITING IN PUMPED WELL NUMBER  2 , IN LBS, IS=  .2157
CONCENTRATION IN PUMPED WELL NUMBER  5 , IN PPM, IS=  1.58442E-03
MASS EXITING IN PUMPED WELL NUMBER  5 , IN LBS, IS=  .04314



LOWER LEFT-HAND CORNER OF PLOT POSITION =  2090          840
GRID SIZE OF PLOT MAP, IN FEET =  60
NUMBER OF ROWS AND COLUMNS OF PLOT MAP =  32            21

Plot file name is =C123.DAT
      PRESENT SIMULATION TIME =  4380   DAYS
       INCREMENTAL SIMULATION TIME =  1825  DAYS
DMAX =  10  FT   ZMAX =  1  FT

NP= 234
CONCENTRATION IN PUMPED WELL NUMBER  2 , IN PPM, IS=  3.644004E-04
MASS EXITING IN PUMPED WELL NUMBER  2 , IN LBS, IS=  .4314
CONCENTRATION IN PUMPED WELL NUMBER  4 , IN PPM, IS=  8.769065E-04
MASS EXITING IN PUMPED WELL NUMBER  4 , IN LBS, IS=  .12942
```

```
DMAX =   10   FT    ZMAX =   1   FT

NP= 230
CONCENTRATION IN PUMPED WELL NUMBER  2 , IN PPM, IS=  1.093201E-04
MASS EXITING IN PUMPED WELL NUMBER  2 , IN LBS, IS=  .12942
CONCENTRATION IN PUMPED WELL NUMBER  4 , IN PPM, IS=  2.923022E-04
MASS EXITING IN PUMPED WELL NUMBER  4 , IN LBS, IS=  4.314001E-02



      PRESENT SIMULATION TIME =  8030   DAYS
       INCREMENTAL SIMULATION TIME =  1825  DAYS
DMAX =   10   FT    ZMAX =   1   FT

NP= 229
CONCENTRATION IN PUMPED WELL NUMBER  4 , IN PPM, IS=  2.923022E-04
MASS EXITING IN PUMPED WELL NUMBER  4 , IN LBS, IS=  4.314001E-02
```