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

\[ h = 0.3665 \left(\frac{Q}{T}\right) \log \left(\frac{\text{del}}{4.81r_w}\right) \]

where

- \( h \) = head loss
- \( Q \) = pumping rate
- \( T \) = effective transmissivity
- \( \text{del} \) = 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)

A-1

Engineering Technologies Associates, Inc.
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

WELL1AL

1. Data: MXWELL IWELCB
   Format: I10 I10

FOR EACH STRESS PERIOD

WELL1RP

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 I16 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 calculate 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     SPECIFICATIONS:
C  **********************************************************************************************
INTEGER*4 LENX, ISUM
COMMON X(65000)
DIMENSION HEADNG(32), VBVL(4, 20), IUNIT(24)
C
C CHARACTER*4 VBVM(4, 20)
C
C DOUBLE PRECISION DUMMY
EQUIVALENCE (DUMMY, X(1))

----------------------SET SIZE OF X ARRAY. REMEMBER TO REDIMENSION X.
LENX=65000
C
C ASSIGN BASIC INPUT UNIT AND PRINTER UNIT.
INBAS=1
IOUT=6
C
C DEFINE PROBLEM ROWS, COLUMNS, LAYERS, STRESS PERIODS, PACKAGES
CALL BASIDF(ISUM, HEADNG, NPER, ITMUNI, TOTIM, NCOL, NROW, NLAY,
1     NODES, INBAS, IOUT, IUNIT)
C
C ALLOCATE SPACE IN "X" ARRAY.
CALL BASIAL(ISUM, LENX, LCHNEW, LCHOLD, LCIROU, LCCR, LCCC, LCCV,
1     LCHCOF, LCRHS, LCDELR, LCDELCL, LCSTRT, LCBUFF, LCOFL,
2     INBAS, ISTRT, NCOL, NROW, NLAY, IOUT)
IF(IUNIT(1).GT.0) CALL BCFIAL(ISUM, LENX, LCBOT, LCTOP, LCSR, LCTRYPY,
1     IUNIT(1), ISS,
2     NCOL, NROW, NLAY, IOUT, IBCFCB)
IF(IUNIT(2).GT.0) CALL WELIAL(ISUM, LENX, MXWELL, NWELLS, IUNIT(2), IOUT, IWELCB)
IF(IUNIT(3).GT.0) CALL DRNIAL(ISUM, LENX, MXDRAI, NDRAIN, IDFRCN, IUNIT(3), IOUT, IDRCNB)
IF(IUNIT(8).GT.0) CALL RCHIAL(ISUM, LENX, LCRCH, LCRECH, NRCHOP, IUNIT(8), IOUT, IRCHCB)


A-6 Engineering Technologies Associates, Inc.
IF(IUNIT(5).GT.O) CALL EVT1AL(ISUM,LENX,LCIEVT,LCEVTR,LCEXDP,
   1 LCSURF,NCOL,NRCV,NEVTOP,IUNIT(5),IOUT,IEVTCC)
IF(IUNIT(4).GT.O) CALL RIV1AL(ISUM,LENX,LCRIVR,MXRIVR,NRIVER,
   1 UINIT(4),IOUT,IRIVCS)
IF(IUNIT(7).GT.O) CALL GHBlAL(ISUM,LENX,LCBNDs,NBOUND,MXBND,
   1 UINIT(7),IOUT,IGHSCB)
IF(IUNIT(9).GT.O) CALL SIPtAL(ISUM,LENX,LCFL,LCGL,LCV,
   1 LCHDCG,LCLRCH,LCW,MXITER,NPARM,NCOL,NROW,NLAY,
   2 UUNIT(9),IOUT)
IF(IUNIT(11).GT.O) CALL SCRlAL(ISUM,LENX,LCA,LCRES,LCHDCG,LCLRCH,
   1 LCIEGP,MXITER,NCOL,NRGW,NLAY,NSLICE,MBW,UNIT(11),IOUT)

C5------IF THE "X" ARRAY IS NOT BIG ENOUGH THEN STOP.
IF(ISUM-1.GT.LENX) STOP

C6------READ AND PREPARE INFORMATION FOR ENTIRE SIMULATION.
   CALL BASlRP(X(LCIBOU),X(LCHNEW),X(LCSTRT),X(LCHOLD),
   1 ISTRT,INBAS,HEADNG,NCOL,NROW,NLAY,NODES,VBVL,X(LCIOFL),
   2 IUNIT(12),IHEDFM,IDDNFM,INDDUN,INOUT)
IF(IUNIT(1).GT.O) CALL BCF1RP(X(LCIBOU),X(LCHNEW),X(LCS1),
   1 X(LCHY),X(LCCR),X(LCC),X(LCCV),X(LCDELR),
   2 X(LCDELC),X(LCBOI),X(LCTOP),X(LCSC2),X(LCTRNY),
   3 UNIT(1),ISS,INCOL,NROW,NLAY,NODES,IOUT)
IF(IUNIT(9).GT.O) CALL SIP1RP(NPARM,MXITER,ACCL,HCLOSE,X(LCW),
   1 UNIT(9),IPCALC,IPRSIP,UOUT)
IF(IUNIT(11).GT.O) CALL SCR1RP(MXITER,ACCL,HCLOSE,UNIT(11),
   1 IPRSOR,OUT)

C7------SIMULATE EACH STRESS PERIOD.
   DO 300 KPER=1,NPER

C7A------READ STRESS PERIOD TIMING INFORMATION.
   CALL BASlST(NSTP,DELT,TSMULT,PERTIM,KPER,INBAS,IOUT)

C7B------READ AND PREPARE INFORMATION FOR STRESS PERIOD.
   IF(IUNIT(2).GT.O) CALL WELTRP(X(LCwellx),NWELLS,MAXWELL,UNIT(2),
   1 IOUT)
   IF(IUNIT(3).GT.O) CALL DRNTRP(X(LCDRA),NDRAIN,MAXDRN,UNIT(3),
   1 IOUT)
   IF(IUNIT(8).GT.O) CALL RCHTRP(NRCHIP,X(LCIRCH),X(LCRECH),
   1 X(LCDELR),X(LCDEL),X(LCDELR),X(LCDRA),NROW,NCOL,NLAY,UNIT(8),IOUT)
   IF(IUNIT(5).GT.O) CALL EVTRP(NEVTOP,X(LCIEVT),X(LCEVIR),
   1 X(LCEXDP),X(LCSURF),X(LCDELR),X(LCDELR),X(LCDELR),X(LCDRA),NROW,
   1 NLAY,UNIT(5),IOUT)
   IF(IUNIT(4).GT.O) CALL RIV1RP(X(LCIVR),NRIVER,MXRIVR,UNIT(4),
   1 IOUT)
   IF(IUNIT(7).GT.O) CALL GH1RP(X(LCBNDs),NBOUND,MXBND,UNIT(7),
   1 IOUT)

C7C------SIMULATE EACH TIME STEP.
   DO 200 KSTP=1,NSTP

C7C1------CALCULATE TIME STEP LENGTH. SET HOLD=HNEW.
   CALL BACTIO(D4ELT,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 BASTFM(X(LCHCOF),X(LCRHS),NCOL,NROW,NLAY,NODES)
  IF(IUNIT(1).GT.0) CALL BCF1FM(X(LCHCOF),X(LCRHS),X(LCHOLD),
      1 X(LCC1),X(LCHNEW),X(LCIBOU),X(LCCR),X(LCC),X(LCCV),
      2 X(LCHY),X(LCTRYP),X(LCBOT),X(LCTOP),X(LCC2),
      3 X(LCDLR),X(LCDEL),DEL,T,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 DNF1FM(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(NEVENTOP,X(LCEVT),X(LCEVTR),
      1 X(LCLEXD),X(LCSURF),X(LCRHS),X(LCHCOF),X(LCIBOU),
      1 X(LCHNEW),NCOL,NROW,NLAY)
  IF(IUNIT(4).GT.0) CALL RIV1FM(NRIVER,MXRIVR,X(LCHNEW),
      1 X(LCHCOF),X(LCRHS),X(LCIBOU),NCOL,NROW,NLAY)
  IF(IUNIT(7).GT.0) CALL GHBFM(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(LCCV),
      1 X(LCCV),X(LCHCOF),X(LCRHS),X(LCEL),X(LCLFL),X(LCGL),X(LCV),
      2 X(LC),X(LCHDCG),X(LCLRCH),NPARM,KITER,KCLOSE,ACCL,ICNVG,
      3 KSTP,KPER,IPRSIP,MXITER,NSTP,NCOL,NROW,NLAY,NODES,
      4 IOUT)
  IF(IUNIT(11).GT.0) CALL SOR1AP(X(LCHNEW),X(LCIBOU),X(LCCR),
      1 X(LCC),X(LCCV),X(LCHCOF),X(LCRHS),X(LCEL),X(LCRES),X(LCEGDP),
      2 X(LCHDCG),X(LCLRCH),KITER,KCLOSE,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,INDF1L,IUNIT(12),IOUT)
C
C7C4----CALCULATE BUDGET TERMS. SAVE CELL-BY-CELL FLOW TERMS.
  MSNBC=1
  IF(IUNIT(1).GT.0) CALL BCF1BD(VBNM,VBVL,MSUM,X(LCHNEW),
      1 X(LCIBOU),X(LCHOLD),X(LCC1),X(LCCR),X(LCC),X(LCCV),
      2 X(LCTOP),X(LCC2),DEL,T,ISS,KITER,KSTP,KPER,
      3 IBFC0,ICBCFL,ICBUFF,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 IDANCB, ICBCFL, X(LCBUFF), IOUT)
   IF(IUNIT(5).GT.0) CALL RCHIBD(NRCHOP, X(LCIRCH), X(LCRECH),
1 X(LCIBOU), NROW, NCOL, NLAY, DELT, VBVL, VBNM, MSUM, KSTP, KPER,
2 ICHCB, ICBCFL, X(LCBUFF), IOUT)
   IF(IUNIT(6).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, XRIVR, 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 GH1BD(NBOUND, XBDND, VBNM, VBVL, MSUM,
1 X(LCBNDS), DELT, X(LCHNEW), NCOL, NROW, NLAY, X(LCIBOU), KSTP, KPER,
2 IGBCB, ICBCFL, X(LCBUFF), IOUT)
   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 IHDFD, IBUDFL, IHEDFM, IHEDUN, IDDNFM, IDDNUN, IOUT)

PRINT WELL DRAWDOWNS

CALL BAS1OT( X(LCHNEW), X(LCSTRT), TOTIM, X(LCBUFF), X(LCIOFL),
1 NCOL, NROW, NLAY,
2 IHDFD, IOUT,
3 X(LCWELL), X(LCIBOU), X(LCR), X(LCDEL), X(LCDELC),
4 X(LCTRYP), X(LCIBOU), X(LCHY), X(LCBOT), X(LCTOP),
5 NWELS, MWELL)

G7C6---IF ITERATION FAILED TO CONVERGE THEN STOP.
   IF(INCNV.GE.0.0) STOP
200 CONTINUE
300 CONTINUE

C8-----END PROGRAM
STOP

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 CONVERGENCE LOSS AND WELL EFFICIENCY
C LOSS
C
C SPECIFICATIONS:
C
DOUBLE PRECISION HNEW

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)

C66

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 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)

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.

\[ h = 0.3665\frac{Q}{T}\log_{10}\left(\frac{\text{del}}{4.81r_w}\right) \]

where
- \( h \) = head loss
- \( Q \) = pumping rate
- \( T \) = effective transmissivity
- \( \text{del} \) = 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, 
THY,SCT,TOP,NWELLS,MXWELL,NCOL,NROW,NLAY,IOUT,TOTIM)

C
C
C----VERSION 2.0 NOV 1986 SBAS2W
C
******************************************************************************
C THIS ROUTINE COMPUTES THE HEAD IN A PUMPING WELL INCLUDING
C CONVERGENCE 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 CONVERGENCE LOSS IS CALCULATED USING THE SEMIEMPIRICAL RESULTS
C OF PRICKETT AND LONQUIST, "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
C CLOSS = 0.3665 * (Q/TRANS) *LOG10 (DEL/(4.81 * RWELL))
C
C WELL EFFICIENCY LOSS IS CALCULATED AS
C
C WLOSS = ALOSS * Q ** AEXP
C
C WELL NAMES MAY ALSO BE USED, VARIABLE WELLNAM 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 WELIRP
C
C----------SPECIFICATIONS
C
CHARACTER*8 WELLNAM
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/WELLNAM(100)
COMMON/FLWCOM/LAYCON(80)
C
--------------------------------------------------------------
C
C PRINT A HEADER
C
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)

A-13

Engineering Technologies Associates, Inc.
```fortran
RWELL=WELL(7,II)
IWELL=WELL(8,II)

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)
      KB=KB+1
   IF(LAYCON(KCOUNT) .EQ. 2 .OR. LAYCON(KCOUNT) .EQ. 3)
      KT=KT+1
55 CONTINUE

WRITE(*,*)II,I,J,K,KT, KB

C CALCULATE THE NODE TRANSMISSIVITY ?????
TRANS=0.
ICOUNT=0
TEMP1=0.
IF(J .GT. 1) TEMP1=CC(J-1,1,K)*(DELC(I)+DELC(1-1))/2./DELR(J)
IF(TEMP1 .GT. 0) ICOUNT=ICOUNT+1
TEMP2=0.
IF(J .GT. 1) TEMP2=CR(J-1,1,K)*(DELR(J)+DELR(J-1))/2./DELC(I)
IF(TEMP2 .GT. 0) ICOUNT=ICOUNT+1
IF(CC(J,1,K) .GT. 0) ICOUNT=ICOUNT+1
IF(CR(J,1,K) .GT. 0) ICOUNT=ICOUNT+1
IF(ICOUNT .GT. 0) TRANS=(CC(J,1,K)*(DELC(I+1)+DELC(I))/2./
     DELR(J)+CR(J,1,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
   SBAS2H(HNEW, IBOUND,CC,HY,TRPY,BOT,TOP,K, I,J,KB,KT,NCLNROW,
     NLAY,TRANS)

C CALCULATE CONVERGENCE 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*0/TRANS*
1   ALOG10(SORT(DELR(1))/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,1,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,1,KB) .OR. TRANS .LE. 0)WELL(4,II)=0.
```

A-14 Engineering Technologies Associates, Inc.
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 COMPUTE TRANSMISSIVITY FROM SATURATED THICKNESS AND HYDRAULIC
C CONDUCTIVITY
C
C THIS ROUTINE WRITTEN BY DONALD KOCH, ENGINEERING TECHNOLOGIES
C ASSOCIATES, NOVEMBER 1986
C
C SPECIFICATIONS:
C
DOUBLE PRECISION HNEW

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)

COMMON /FLWCOM/LAYCON(80)

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

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)

C4------CHECK TO SEE IF SATURATED THICKNESS IS GREATER THAN ZERO.

IF(THCK.LE.0.) GO TO 100

C5------IF SATURATED THICKNESS>0 THEN T=K*THICKNESS.

100 CC(J,I,K)=THCK*HY(J,I,KB)
   GO TO 200

C6------WHEN SATURATED THICKNESS < 0, SET
C7------TRANSMISSIVITY=0

200 CONTINUE

C COMPUTE GEOMETRIC MEAN TRANSMISSIVITY IN CELL
TRANS=SGRT(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.

\[ v_i = c_R \frac{k_i^r(k_{i+1,r} + k_{i-1,r})}{D_{R,i}} \times \frac{\text{CONV}}{7.48} \]

B-1

Engineering Technologies Associates, Inc.
\[ VJ = CC_{j,1-1,k} (H_{j,1,k} - H_{j,1-1,k}) \times 2 \times CONV / 7.48 / (DELC_{i} + DELC_{j}) \]

\[ VZ = CV_{j,1,k} (H_{j,1,k} - HOLD_{j,1,k}) \times CONV / 7.48 \]

for \( k = 2 \) to \( NL \)

\[ VZ = (H_{j,1,k} - HOLD_{j,1,k}) / DELC \times CONV / 7.48 - RECH_{j,1} - RIV / DELR_{j} / DELC_{i} - ET_{j,1} \]

for \( k = 1 \)

where

\( V_{I} \) = velocity in row direction from middle of node \( j, i, k \) to middle of node \( j+1, i, k \)

\( V_{J} \) = velocity in column direction from middle of node \( j, i, k \) to middle of node \( j, i-1, k \) (ft/day)

\( V_{Z} \) = vertical velocity from node \( j, i, k \) to node \( j, i, k-1 \) (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 \( j, i+1, k \) (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 \( j, i, k \)

\( DELR_{j} \) = 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 (ft²/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 GALLON-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 coordinates (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:probw1hd.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)

```plaintext
<table>
<thead>
<tr>
<th></th>
<th>20</th>
<th>20</th>
<th>3</th>
<th>100.0</th>
<th>100.0</th>
<th>.0</th>
<th>.0</th>
<th>490.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>20</td>
<td>3</td>
<td>40.0</td>
<td>-.984E-01</td>
<td>.000</td>
<td>.413E-01</td>
<td>660.</td>
<td>695.</td>
</tr>
<tr>
<td>2</td>
<td>20</td>
<td>3</td>
<td>42.0</td>
<td>-.867E-01</td>
<td>.000</td>
<td>-.288E-02</td>
<td>660.</td>
<td>.000</td>
</tr>
<tr>
<td>3</td>
<td>20</td>
<td>3</td>
<td>43.7</td>
<td>-.771E-01</td>
<td>.000</td>
<td>-.339E-02</td>
<td>660.</td>
<td>.000</td>
</tr>
<tr>
<td>4</td>
<td>20</td>
<td>3</td>
<td>45.3</td>
<td>-.686E-01</td>
<td>.000</td>
<td>-.386E-02</td>
<td>660.</td>
<td>.000</td>
</tr>
<tr>
<td>5</td>
<td>20</td>
<td>3</td>
<td>46.7</td>
<td>-.611E-01</td>
<td>.000</td>
<td>-.427E-02</td>
<td>660.</td>
<td>.000</td>
</tr>
<tr>
<td>6</td>
<td>20</td>
<td>3</td>
<td>47.9</td>
<td>-.546E-01</td>
<td>.000</td>
<td>-.464E-02</td>
<td>660.</td>
<td>.000</td>
</tr>
<tr>
<td>7</td>
<td>20</td>
<td>3</td>
<td>49.0</td>
<td>-.487E-01</td>
<td>.000</td>
<td>-.498E-02</td>
<td>660.</td>
<td>.000</td>
</tr>
<tr>
<td>8</td>
<td>20</td>
<td>3</td>
<td>50.0</td>
<td>-.435E-01</td>
<td>.000</td>
<td>-.527E-02</td>
<td>660.</td>
<td>.000</td>
</tr>
<tr>
<td>9</td>
<td>20</td>
<td>3</td>
<td>50.8</td>
<td>-.388E-01</td>
<td>.000</td>
<td>-.552E-02</td>
<td>660.</td>
<td>.000</td>
</tr>
<tr>
<td>10</td>
<td>20</td>
<td>3</td>
<td>51.6</td>
<td>-.346E-01</td>
<td>.000</td>
<td>-.573E-02</td>
<td>660.</td>
<td>.000</td>
</tr>
<tr>
<td>11</td>
<td>20</td>
<td>3</td>
<td>52.3</td>
<td>-.307E-01</td>
<td>.000</td>
<td>-.591E-02</td>
<td>660.</td>
<td>.000</td>
</tr>
<tr>
<td>12</td>
<td>20</td>
<td>3</td>
<td>52.9</td>
<td>-.271E-01</td>
<td>.000</td>
<td>-.606E-02</td>
<td>660.</td>
<td>.000</td>
</tr>
<tr>
<td>13</td>
<td>20</td>
<td>3</td>
<td>53.4</td>
<td>-.236E-01</td>
<td>.000</td>
<td>-.617E-02</td>
<td>660.</td>
<td>.000</td>
</tr>
<tr>
<td>14</td>
<td>20</td>
<td>3</td>
<td>53.9</td>
<td>-.202E-01</td>
<td>.000</td>
<td>-.626E-02</td>
<td>660.</td>
<td>.000</td>
</tr>
<tr>
<td>15</td>
<td>20</td>
<td>3</td>
<td>54.3</td>
<td>-.169E-01</td>
<td>.000</td>
<td>-.633E-02</td>
<td>660.</td>
<td>.000</td>
</tr>
<tr>
<td>16</td>
<td>20</td>
<td>3</td>
<td>54.7</td>
<td>-.135E-01</td>
<td>.000</td>
<td>-.637E-02</td>
<td>660.</td>
<td>.000</td>
</tr>
<tr>
<td>17</td>
<td>20</td>
<td>3</td>
<td>54.9</td>
<td>-.102E-01</td>
<td>.000</td>
<td>-.640E-02</td>
<td>660.</td>
<td>.000</td>
</tr>
<tr>
<td>18</td>
<td>20</td>
<td>3</td>
<td>55.1</td>
<td>-.680E-02</td>
<td>.000</td>
<td>-.642E-02</td>
<td>660.</td>
<td>.000</td>
</tr>
<tr>
<td>19</td>
<td>20</td>
<td>3</td>
<td>55.3</td>
<td>-.340E-02</td>
<td>.000</td>
<td>-.643E-02</td>
<td>660.</td>
<td>.000</td>
</tr>
<tr>
<td>20</td>
<td>20</td>
<td>3</td>
<td>55.3</td>
<td>.000</td>
<td>.000</td>
<td>-.644E-02</td>
<td>660.</td>
<td>.000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>1</th>
<th>100.</th>
<th>-.174E-02</th>
<th>.183E-04</th>
<th>-.115E-03</th>
<th>500.</th>
<th>660.</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>1</td>
<td>1</td>
<td>100.</td>
<td>-.160E-02</td>
<td>.183E-04</td>
<td>-.148E-03</td>
<td>500.</td>
<td>660.</td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td>1</td>
<td>100.</td>
<td>-.142E-02</td>
<td>.366E-04</td>
<td>-.176E-03</td>
<td>500.</td>
<td>660.</td>
</tr>
<tr>
<td>14</td>
<td>1</td>
<td>1</td>
<td>100.</td>
<td>-.123E-02</td>
<td>.183E-04</td>
<td>-.201E-03</td>
<td>500.</td>
<td>660.</td>
</tr>
<tr>
<td>15</td>
<td>1</td>
<td>1</td>
<td>100.</td>
<td>-.101E-02</td>
<td>.183E-04</td>
<td>-.221E-03</td>
<td>500.</td>
<td>660.</td>
</tr>
<tr>
<td>16</td>
<td>1</td>
<td>1</td>
<td>100.</td>
<td>-.769E-03</td>
<td>.183E-04</td>
<td>-.238E-03</td>
<td>500.</td>
<td>660.</td>
</tr>
<tr>
<td>17</td>
<td>1</td>
<td>1</td>
<td>100.</td>
<td>-.522E-03</td>
<td>.183E-04</td>
<td>-.250E-03</td>
<td>500.</td>
<td>660.</td>
</tr>
<tr>
<td>18</td>
<td>1</td>
<td>1</td>
<td>100.</td>
<td>-.266E-03</td>
<td>.183E-04</td>
<td>-.259E-03</td>
<td>500.</td>
<td>660.</td>
</tr>
<tr>
<td>19</td>
<td>1</td>
<td>1</td>
<td>100.</td>
<td>.000</td>
<td>.183E-04</td>
<td>-.262E-03</td>
<td>500.</td>
<td>660.</td>
</tr>
<tr>
<td>20</td>
<td>1</td>
<td>1</td>
<td>100.</td>
<td>.000</td>
<td>.183E-04</td>
<td>-.262E-03</td>
<td>500.</td>
<td>660.</td>
</tr>
</tbody>
</table>

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.  
```

Engineering Technologies Associates, Inc.
<table>
<thead>
<tr>
<th>Value 1</th>
<th>Value 2</th>
<th>Value 3</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>50.0</td>
<td>850.0</td>
<td>3.0</td>
<td>3173.</td>
</tr>
<tr>
<td>50.0</td>
<td>750.0</td>
<td>3.0</td>
<td>3178.</td>
</tr>
<tr>
<td>50.0</td>
<td>650.0</td>
<td>3.0</td>
<td>3187.</td>
</tr>
<tr>
<td>50.0</td>
<td>550.0</td>
<td>3.0</td>
<td>3196.</td>
</tr>
<tr>
<td>50.0</td>
<td>450.0</td>
<td>3.0</td>
<td>3209.</td>
</tr>
<tr>
<td>50.0</td>
<td>350.0</td>
<td>3.0</td>
<td>3219.</td>
</tr>
<tr>
<td>50.0</td>
<td>250.0</td>
<td>3.0</td>
<td>3232.</td>
</tr>
<tr>
<td>50.0</td>
<td>150.0</td>
<td>3.0</td>
<td>3237.</td>
</tr>
<tr>
<td>50.0</td>
<td>50.0</td>
<td>3.0</td>
<td>3241.</td>
</tr>
</tbody>
</table>
PREMOD3D

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
C-----VERSION 1116 28DEC1983 MAIN1
C
C*******************************************************************************
C
C SPECIFICATIONS:
C
C---------------------------------------------- --
CHARACTER*4 VBNM(4,20)
COMMON X(90000)
DIMENSION HEADNG(32),VBVL(4,20),IUNIT(24)

CHARACTER*4 VBNM(4,20)
CHARACTER*1 ANSWER
CHARACTER*3 CODE

DOUBLE PRECISION DUMMY
EQUIVALENCE (DUMMY,X(1))

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

C

- ------

SET SIZE OF X ARRAY. REMEMBER TO REDIMENSION X.
LENX=90000

C

ASSIGN BASIC INPUT UNIT AND PRINTER UNIT.
INBAS=1
IOUT=6

C

DEFINE PROBLEM ROWS, COLUMNS, LAYERS, STRESS PERIODS, PACKAGES
CALL BASDF(ISUM,HEADNG,NPER,ITMUNI,TOTIM,NCOL,NROW,NLAY,
1 NODES,INBAS,IOUT,IUNIT)

C

ALLOCATE SPACE IN "X" ARRAY.
CALL BASIALCISUM,LENX,LCHNEW,LCHOLD,LCIBOT,LCCCR,LCCC,LCCV,
1 LCRCOF,LCRHS,LCDELC,LCSTRT,LCBUFF,LCIOFL,
2 INBAS,ISTRT,NCOL,NROW,NLAY,IOUT)

IF(IUNIT(1).GT.0) CALL BCFTAL(ISUM,LENX,LCSTC1,LCHY,
1 LCBOT,LCTOP,LCSC2,LCTRPY,IUNIT(1),ISS,
2 NCOL,NROW,NLAY,IOUT,IBCFCB)

IF(IUNIT(2).GT.0) CALL WELTAL(ISUM,LENX,LCWELL,MXWELL,NWELLS,
1 IUNIT(2),IOUT,IEWELCB)

IF(IUNIT(3).GT.0) CALL DRNTAL(ISUM,LENX,LCRAI,MXRAIN,MXRAN,
1 IUNIT(3),IOUT,IDRNCB)

IF(IUNIT(8).GT.0) CALL RCHTAL(ISUM,LENX,LCIRCH,LCRECH,NRCHOP,
1 IUNIT(8),IOUT,ICRCHCB)

B-9  Engineering Technologies Associates, Inc.
NCOL, NROW, IUNIT(5), IOUT, ICUT, IRCHCB)
IF(IUNIT(5).GT.0) CALL EVTIAL(ISUM, LENX, LCEVT, LCEVRT, LCEXDP, 1
LCSURF, NCOL, NROW, NEVTOP, IUNIT(5), IOUT, LEVTCB)
IF(IUNIT(4).GT.0) CALL RIVIAL(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 5-----IF THE "X" ARRAY IS NOT BIG ENOUGH THEN STOP.
C IF(ISUM-1.GT.LENX) STOP
C
C 6-----READ AND PREPARE INFORMATION FOR ENTIRE SIMULATION.
C CALL BASRPF(X(LCIBOU), X(LCHNEW), X(LCSTRT), X(LCHOLD), 1
ISTRT, INBAS, HEADING, NCOL, NROW, NODES, VBVL, X(LCIOFL), 2
IUNIT(12), IHEDFM, IDDFM, IHEDUN, IDDUN, IOUT)
IF(IUNIT(1).GT.0) CALL BCF1RP(X(LCIBOU), X(LCHNEW), X(LCBS2), 1
X(LCTRPY), X(LCSTRT), X(LCHOLD), X(LCBOT), X(LCTOP), X(LCDELR), 2
X(LCDLRE), X(LCSCI), X(LCHY), X(LCCR), X(LCCC), X(LCCV), X(LCDLRE), 3
IUNIT(1), ISS, NCOL, NROW, NODES, ILOUT)

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
C GAL-DAY UNITS'
READ(*,*)CONVRT
WRITE(IOUT,*)' CONVERSION FACTOR = ', CONVRT

C ENTER SECTION OF MODEL TO BE USED IN TRANSPORT SIMULATION
C
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 ', LUX
WRITE(IOUT,*)' ROWS IN TRANSPORT MODEL FROM ', LLY, TO ', LUY

C 7-----SIMULATE EACH STRESS PERIOD.
DO 300 KPER=1,NPER

C  
C7A----READ STRESS PERIOD TIMING INFORMATION.
   CALL BASIST(NSTF,DELT,TSMULT,PERTIM,KPER,INBSA,IOUT)
C
C  
C7B----READ AND PREPARE INFORMATION FOR STRESS PERIOD.
   IF(IUNIT(2).GT.0) CALL WE1IP(X(LCNEW),NWELL,X(LCNEW),1,IUNIT(2),
      1 IOUT)
   IF(IUNIT(3).GT.0) CALL DRN1RP(X(LCNEW),MDFR,NDRAIN,IUNIT(3),
      1 IOUT)
   IF(IUNIT(8).GT.0) CALL RCH1RP(NRCHOP,X(LCIRCH),X(LCCRECH),
      1 X(LCDEL),X(LCDEL),NROW,NCOL,NLAY,IUNIT(8),IOUT)
   IF(IUNIT(5).GT.0) CALL EVT1RP(NEVTOP,X(LCEVT),X(LCEVT),
      1 X(LCNEW),X(LCNEW),X(LCFLUX),X(LCFLUX),X(LCDEL),X(LCDEL),NROW,
      2 NCOL,NLAY,IUNIT(5),IOUT)
   IF(IUNIT(4).GT.0) CALL RIV1RP(X(LCRIVR),NHRIVER,MRIVER,IUNIT(4),
      1 IOUT)
   IF(IUNIT(7).GT.0) CALL GHST1RP(X(LCMBS),NBOUND,MXBND,IUNIT(7),
      1 IOUT)
C
C QUERY USER IF THIS STRESS PERIOD IS TO BE USED. IF NOT JUMP TO END OF STRESS PERIOD LOOP
C
C WRITE(*,1002)KPER
   1002 FORMAT(1X,' DO YOU WISH TO SIMULATE SOLUTE TRANSPORT IN STRESS PERIOD '//
      1 PERIOD ' ,13,' (Y/N) '\)
   READ(*,1003)ANSWER
   1003 FORMAT(A1)
   IF(ANSWER .EQ. 'Y' .OR. ANSWER .EQ. 'y') GO TO 300
   WRITE(IOUT,'*') ' TIME STEPS IN STRESS PERIOD ',KPER,' CONVERTED '
C
C TIME STEP LOOP
   DO 200 KSTEP=1,NSTEP
C
C  
C READ IN HEADS FROM EACH TIME STEP.
C
C   CALL BASPRE(X(LCHNEW),X(LCBSF),X(LCFL),KSTEP,KPER,NCOL,
      1 NROW,NLAY,IOUT,IPREDM,IPREDN,IPFLAG,PERTIM,TOTIM)
C
C OPEN OUTPUT FILE ON UNIT IOUT WITH NAME CODE+KSTEP+''.RND'
C
C CALL OUTDAT(IOUT,CODE,KPER,KSTEP,IOUT)
C
C WRITE HEADER TO FILE
C
C   CALL HEADER(NCOL,NROW,NLAY,X(LCDEL),X(LCDEL),IOUT,LLX,LUX,
      1 LLY,LUX,X(LCRHS),X(LCHDF),X(LCBOT))
C
C CALCULATE APPARENT VELOCITY OF WATER TABLE FROM RECHARGE
C AND PUT INTO ZVEL AND X(LCRHS)
C
C   IF(IUNIT(5).GT.0) CALL PRERCH(NRCHOP,X(LCIRCH),X(LCCRECH),
      1 ,X(LCRHS),X(LCBOU),NROW,NCOL,NLAY)
C
C
C SWITCH HNEW TO HOLD
CALL HSWITCH(X(LCHNEW),X(LCHOLD),X(LCRHS),NCOL,NROW,NLAY)
END OF TIME PERIOD LOOP

200 CONTINUE
END OF STRESS PERIOD LOOP

C
C8
------
END PROGRAM
STOP
C
ERROR MESSAGE FOR BAD COLUMNS OR ROW SELECTION
401 WRITE(*,'(A)') 'OUTSIDE MODEL - TRY AGAIN'
GO TO 5
C
END
C
SUBROUTINE HSWITCH(HNEW,HOLD,ZVEL,NCOL,NROW,NLAY)
C
THIS SUBROUTINE REPLACES HOLD WITH HNEW AT THE END OF THE TIME LOOP
C
WRITTEN BY DONALD KOCH, ENGINEERING TECHNOLOGIES ASSOCIATES, INC
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
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
WRITE HEADER RECORD INTO IVOUT AND INITIALIZE ZVEL AND RBOT
CALCULATE LOWER LEFT CORNER
WRITTEN BY DONALD KOCH, ENGINEERING TECHNOLOGIES ASSOCIATES, INC
DECEMBER 1988
C
DIMENSION DELR(NCOL),DELC(NROW),ZVEL(NCOL,NROW),RBOT(NCOL,NROW),
CALCULATE APPARENT VELOCITY OF WATER TABLE FROM
EVPOTRANSPIRATION AND PUT INTO ZVEL AND X(LCRHS)

IF(UNIT(5).GT.0)CALL PREEVT(NEVENT,X(LCEVT),X(LCEVTR),
1X(LCEXDP),X(LCSURF),X(LCRHS),X(LCIBOU),X(LCHNEW),
2NCOL,NROW,NLAY)

CALCULATE APPARENT VELOCITY OF WATER TABLE FROM
RIVER LEAKAGE AND PUT INTO ZVELAND X(LCRHS)

IF(UNIT(4).GT.0)CALL PRERIV(NRIVER,MXIRVR,X(LCRIVR),
1X(LCHNEW),X(LCIBOU),X(LCHCF),X(LCDEL),X(LCDELC),X(LCRHS),
2NCOL,NROW,NLAY,IVOUT,CONVRT)

CALCULATE VELOCITIES AND WRITE TO FILE

CALL VELOCITY(X(LCHNEW),X(LCHOLD),X(LCIBOU),X(LCCR),X(LCC),
1X(LCCV),X(LCHY),X(LCTRPY),X(LCBOT),X(LCTOP),X(LCDEL),
2X(LCDEL),DELT,ISS,KSTP,KPER,NCOL,NROW,NLAY,IVOUT,IVOUT,
3X(LCRHS),X(LCHCF),LLX,LUX,LLY,LUY,CONVRT)

CALCULATE SINK LOCATIONS AND WRITE TO FILE

SINKS AT CONSTANT HEAD NODES WHERE WATER LEAVES MODEL

CALL CHDPRE;

CREATE SINKS AT WELLS

IF(UNIT(2).GT.0)CALL WELPRE(NWELLS,MXWELL,X(LCWELL),
1X(LCIBOU),X(LCDEL),X(LCDEL),NCOL,NROW,NLAY,IVOUT,CONVRT,
2LLX,LUX,LLY,LUY)

CREATE SINKS AT DRAINS

IF(UNIT(3).GT.0)CALL DRNPRE(NDRAIN,MXDRN,X(LCDRAI),
1X(LCHNEW),X(LCIBOU),X(LCDEL),X(LCDELC),NCOL,NROW,NLAY,
2IVOUT,CONVRT,LLX,LUX,LLY,LUY)

CREATE SINKS AT RIVERS

IF(UNIT(4).GT.0)CALL RIVPRE(NRIVER,MXIRVR,X(LCRIVR),
1X(LCHNEW),X(LCIBOU),X(LCDEL),X(LCDELC),NCOL,NROW,
2NLAY,IVOUT,CONVRT,LLX,LUX,LLY,LUY)

CREATE SINKS AT GENERALIZED HEAD BOUNDARIES

IF(UNIT(7).GT.0)CALL GHBPRE(NBOUND,MXBND,X(LCBNDS),
1X(LCHNEW),X(LCDEL),X(LCDELC),NCOL,NROW,NLAY,IVOUT,CONVRT,
2LLX,LUX,LLY,LUY)

CLOSE OUTPUT UNIT

CLOSE(UNIT=IVOUT)
SUBROUTINE OUTDAT(IVOUT, CODE, KPER, KSTEP, IOUT)

C THIS SUBROUTINE CREATES NAMES FOR AND OPENS THE VELOCITY AND
C SINK OUTPUT FILES

C WRITTEN BY DONALD KOCH, ENGINEERING TECHNOLOGIES ASSOCIATES, INC
C DECEMBER 1988

IVOUT - OUTPUT UNIT FOR VELOCITY FILE
CODE* - 3 LETTER CODE TO BE USED IN FILES
VOTFIL - NAME OF VELOCITY OUTPUT FILE
KSTEP - TIME STEP NUMBER
KPER - PER:OD NUMBER
KNUM - OUTPUT FILE NUMBER

CHARACTER*3 CODE
CHARACTER*10 VOTFIL
DATA VOTFIL/'XXXDOO.RND'/

OUTDAT(IVOUT, CODE, KPER, KSTEP, IOUT)

C CHECK FOR UNEQUAL GRID SPACING
IF(DELC(LLY) .NE. DELC(LUY))GO TO 100
IF(DELRL(LLX) .NE. DELR(LLX))GO TO 100
C
C ASSIGN COORDINATES OF LOWER LEFT CORNER OF GRID
C
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 XYZPCS(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),
         XLL,YLL,ZLL
1001 FORMAT(3I3,5F0.1)
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
C ERROR FOR UNEQUAL GRID SPACING
100 WRITE(*,*) ' UNEQUAL GRID SPACING - CANNOT BE USED WITH RAND3D'
   STOP 3333
END
KNUM=KPER*100+KSTEP
WRITE(VOTFIL(1:6),1001)CODE,KNUM
C WRITE(*,*)VOTFIL,KSTEP
1001 FORMAT(A3,I13.3)
OPEN(IVOUT,FILE=VOTFIL,STATUS='NEW')
WRITE(*,*)' 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
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
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
SUBROUTINE BCF1AL(ISUM,LENX,LCSC1,LCHY,LCBOT,
1   LCTOP,LCSC2,LCTRPY,IN,ISS,NCOL,NROW,NLAY,IOUT,IBCFCB)

C
C ---- version 0931 08DEC83 BCF1AL
C
C ALLOCATE ARRAY STORAGE FOR BLOCK-CENTERED FLOW PACKAGE
C
C SPECIFICATIONS:

C

INTEGER*4 LENX
COMMON /FLWCOM/LAYCON(80)
C
IDENTIFY PACKAGE
WRITE(IOUT,1)IN
1 FORMAT(IHO,'BCF1 -- BLOCK-CENTERED FLOW PACKAGE, VERSION 1',
1' ,12/08/83', ' INPUT READ FROM UNIT',13)
C
READ AND PRINT ISS (STEADY-STATE FLAG) AND IBCFCB (FLAG FOR
PRINTING OR UNIT# FOR RECORDING CELL-BY-CELL FLOW TERMS)
READ(IN,2) ISS,IBCFCB
2 FORMAT(2110)
   IF(ISS.EQ.0) WRITE(IOUT,3)
   3 FORMAT(IX,'TRANSIENT SIMULATION')
   IF(ISS.NE.0) WRITE(IOUT,4)
   4 FORMAT(IX,'STEADY-STATE SIMULATION')
   IF(IBCFCB.GT.0) WRITE(IOUT,9) IBCFCB
   9 FORMAT(IX,'CELL-BY-CELL FLOWS WILL BE RECORDED ON UNIT',13)
   IF(IBCFCB.LT.0) WRITE(IOUT,88)
   88 FORMAT('CONSANT HEAD CELL-BY-CELL FLOWS WILL BE PRINTED')
C
READ TYPE CODE FOR EACH LAYER AND COUNT TOPS AND BOTTOMS
IF(NLAY.LE.80) GO TO 50
WRITE(IOUT,11)
11 FORMAT(IHO,'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(4012)
  C BOTTOM IS READ FOR TYPES 1,3  TOP IS READ FOR TYPES 2,3
  WRITE(IOUT,52)
  52 FORMAT(IX,5X,'LAYER AQUIFER TYPE',/1X,5X,19('-'))
C
C3B ---- INITIALIZE TOP AND BOTTOM COUNTERS.
    NBOT=0
    NTOP=0
C3C----PRINT LAYER TYPE AND COUNT TOPS AND BOTTOMS NEEDED.
   DO 100 I=1,NLAY

C3C1----PRINT LAYER NUMBER AND LAYER TYPE CODE.
   L=LAYCON(I)
   WRITE(IOUT,7) I,L
    7 FORMAT(IX,I9,I10)

C3C2----ONLY THE TOP LAYER CAN BE UNCONFINED(LAYCON=1).
   IF(L.NE.1 .OR. L.EQ.1) GO TO 70
   WRITE(IOUT,8)
    8 FORMAT(IX,"AQUIFER TYPE 1 IS ONLY ALLOWED IN TOP LAYER")
   STOP

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

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

C4------COMPUTE DIMENSIONS FOR ARRAYS.
   NRC=NROW*NCOL
   ISIZ=NRC*NLAY

C5------ALLOCATE SPACE FOR ARRAYS. IF RUN IS TRANSIENT(ISS=0)
   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

C6------PRINT THE AMOUNT OF SPACE USED BY THE BCF PACKAGE.
   ISP=ISUM-ISOLD
   WRITE(IOUT,101) ISP
    101 FORMAT(IX,16," ELEMENTS IN X ARRAY ARE USED BY BCF")
   ISUM=ISUM+1
   WRITE(IOUT,102) ISUM,LENX
    102 FORMAT(IX," ELEMENTS OF X ARRAY USED OUT OF",17)
   IF(ISUM.GT.LENX) WRITE(IOUT,103)
    103 FORMAT(IX," ***X ARRAY MUST BE DIMENSIONED LARGER***")

C7------RETURN
RETURN
END
SUBROUTINE BCFlRP(IBOUND,HNEW,SC1,HY,CR,CC,CV,DELR,DELC,
1 BOT,TOP,SC2,TRPY,IN,ISS,NCOL,NROW,NLAY,NCDES,ICUT)
C
C----VERSION 1003 03MAY1983 BCFlRP
C
C
C
READ AND INITIALIZE DATA FOR BLOCK-CENTERED FLOW PACKAGE
C
C
SPECIFICATIONS:
C
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)
C
DIMENSION ANAME(6,1O)
C
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) /'TOP', '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) /'BO', 'TTOM'/
DATA ANAME(1,7),ANAME(2,7),ANAME(3,7),ANAME(4,7),ANAME(5,7),
1 ANAME(6,7) /'BO', 'TTOM'/
DATA ANAME(1,8),ANAME(2,8),ANAME(3,8),ANAME(4,8),ANAME(5,8),
1 ANAME(6,8) /'COLU', 'MN', 'ROW AN', 'ISDT', '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
C1------CALCULATE NUMBER OF NODES IN A LAYER AND READ TRPY,DELR,DELC
C
NJO=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),NRW,IN,IOUT)

C2-----READ ALL PARAMETERS FOR EACH LAYER
KT=0
KB=0
DO 200 K=1,NLAY

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.1) KT=KT+1
LOC=1+(K-1)*NIJ
LOCB=(KB-1)*NIJ
LOCT=1+(KT-1)*NIJ

C2B-----READ PRIMARY STORAGE COEFFICIENT INTO ARRAY SC1 IF TRANSIENT
IF(ISS.EQ.0) CALL UZDREL(SC1(LOC),ANAME(1,1),NRW,NCOL,K,IN,IOUT)

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 UZDREL(CC(LOC),ANAME(1,2),NRW,NCOL,K,IN,IOUT)
GO TO 110

C2D-----READ HYDRAULIC CONDUCTIVITY(HY) AND BOTTOM ELEVATION(BOT)
C2D-----IF LAYER TYPE IS 1 OR 3
100 CALL UZDREL(HY(LOCB),ANAME(1,3),NRW,NCOL,K,IN,IOUT)
CALL UZDREL(BOT(LOCB),ANAME(1,5),NRW,NCOL,K,IN,IOUT)

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 UZDREL(CV(LOC),ANAME(1,4),NRW,NCOL,K,IN,IOUT)

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 UZDREL(SC2(LOCT),ANAME(1,7),NRW,NCOL,K,IN,IOUT)

C2G-----READ TOP ELEVATION(TOP) IF LAYER TYPE IS 2 OR 3
CALL UZDREL(TOP(LOCT),ANAME(1,6),NRW,NCOL,K,IN,IOUT)
200 CONTINUE

C3-----PREPARE AND CHECK BCF DATA
CALL SBICF(HNEW,IBOUND,SC1,SC2,CR,CC,CV,HY,TRPY,DELR,DELC,ISS,
1 NCOL,NROW,NLAY,IOUT)

C4-----RETURN
RETURN
END

SUBROUTINE VELOCITY(HNEW,HOLD,IBOUND,CR,CC,CV,HY,TRPY,
1 BOT, TOP, DELR, DELC, ISS, KSTP, KPER,
2 NCOL, NRW, NLAY, IOUT, IOUT, ZVEL, RBOT,
3 LLX, LUX, LLY, LUH, CONV)

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

B-19
C CALCULATE VELOCITIES AND WRITE TO FILE ON UNIT IFOUT
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 /FLWCM/LAYCON(80)
C
KB=0
KT=0
C
C 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 T40 I=LLY,LOY
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
   V=0.0
   VZ=0.0
   THICK=-99.
ELSE
   COMPUTE VELOCITY IN RCW DIRECTION
B-20
IF(J .EQ. NCOL .OR. IBOUND(J+1,1,K) .EQ. 0) THEN
  VI=0.0
ELSE
  VI=CR(J,1,K)*(HNEW(J,1,K)-HNEW(J+1,1,K))*2.0/(DELR(J)+DELR(J))*CONV/7.48
ENDIF

COMPUTE VELOCITY IN COLUMN DIRECTION
IF(I .EQ. 1 .OR. IBOUND(J,1-1,K) .EQ. 0) THEN
  VJ=0.0
ELSE
  VJ=CM(J,1-1,K)*(HNEW(J,1,K)-HNEW(J-1,1,K))*2.0/(DELC(I)+DELC(1))*CONV/7.48
ENDIF

COMPUTE VERTICAL VELOCITY
IF(K .EQ. 1) THEN
  VZ=VZ*(HNEW(J,1,K)-HNEW(J,1,K-1))/CONV
  IF(IS .EQ. 0) VZ=VZ*(HNEW(J,1,K)-HOLD(J,1,K))/DELT*CONV/7.48
  THICK=HNEW(J,1,K)-BOT(J,1,K)
  IF(RBOT(J,1) .GT. 0) THEN
    TP=RBOT(J,1)
  ELSE
    TP=0.0
  ENDIF
  ELSE
    VZ=CV(J,1,K-1)*(HNEW(J,1,K)-HNEW(J,1,K-1))*CONV/7.48
  TP=TOP(J,1,K)
C ZERO VERTICAL VELOCITY IF LAYER UNDER WATER TABLE
CONDITIONS
IF(HNEW(J,1,K) .LT. TP) VZ=0.0
C CODE FOR THICKNESS IN PRICKET RAND3D
IF(K .EQ. 2) THICK = HNEW(J,1,K-1) - (TOP(J,1,K) +
BOT(J,1,K))/2
IF(K .EQ. 3) THICK = (TOP(J,1,K-1) + BOT(J,1,K))/2
C
THICK = TOP(J,1,K) - BOT(J,1,K)
ENDIF
ENDIF

WRITE VELOCITY
NEWROW=LUY+LLY-1
WRITE(IVOUT,1001)J-LLX+1,NEWROW-I+LLY,NLAY-K+1,THICK,VI,VJ,
VZ,BOT(J,1,K),TP
1001 FORMAT(313,6(1X,G10.3))
CONTINUE
200 CONTINUE
C WRITE GRID SIZE DATA TO FILE
WRITE(IVOUT,1005)(DEL(R,J),J=1,NCOL)
1005 FORMAT(10F8.2)
WRITE(IVOUT,1001)(DELC(1),I=1,NROW)
RETURN
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,TRY,K,NCOL,NROW,NLAY)

C
C COMPUTE BRANCH HYDRAULIC CONDUCTIVITY USING HARMONIC MEAN OF BLOCK
C HYDRAULIC CONDUCTIVITIES BLOCK HY IS IN CC UPON ENTRY

C SPECIFICATIONS:

C

DIMENSION CR(NCOL,NROW,NLAY), CC(NCOL,NROW,NLAY)
C2, TRY(NLAY)

C

YX=TRY(K)*2.

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)
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, IBCFCT, BUDG, IOUT)

C
C VERSION 1004 03MAY1983 SBCF1B
COMPUTE FLOW ACROSS EACH CELL WALL

SPECIFICATIONS:

DOUBLE PRECISION HNEW, HD

DIMENSION HNEW(NCOL,NROW,NLAY), IBOUND(NCOL,NROW,NLAY),
CR(NCOL,NROW,NLAY), CC(NCOL,NROW,NLAY),
CV(NCOL,NROW,NLAY), TOP(NCOL,NROW,NLAY),
BUFF(NCOL,NROW,NLAY)

COMMON /FLWCOM/LAYCON(80)

DIMENSION TEXT(12)

CHARACTER*4 TEXT(12)

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', 'MT F', 'ACE', 'FLOW', 'LOW', 'ER F', 'ACE '/

NCM1=NCOL-1
IF(NCM1.LT.1) GO TO 405

C1-----CLEAR THE BUFFER
DO 310 K=1,NLAY
DO 310 I=1,NROW
DO 310 J=1,NCM1
BUFF(J,I,K)=0.
310 CONTINUE

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

C3-----RECORD CONTENTS OF BUFFER
CALL UBUDSV(KSTP,KPER,TEXT(1),IBCFCB,BUFF,NCOL,NROW,NLAY,IOUT)

C4-----CLEAR THE BUFFER
NRM1=NROW-1
IF(NRM1.LT.1) GO TO 505
DO 410 K=1,NLAY
DO 410 I=1,NRCW
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((IBOUNO(J, I, K).LE.0) .AND. (IBOUNO(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, IKT+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, IBCFL, 2 BUFF, IVCOT, CONVRT)
C
C
******************************************************************************

 Engineering Technologies Associates, Inc.
COMPUTE FLOW FROM CONSTANT HEAD NODES

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

SPECIFICATIONS:

------------------------------------------------------------------

DOUBLE PRECISION HNEW,HD

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)

COMMON /FLWCOM/LAYCON(80)

CF66

DIMENSION TEXT(4),VBNM(4,20)

CF66

CF77

CHARACTER*4 TEXT(4),VBNM(4,20)

CF77

DATA TEXT(1),TEXT(2),TEXT(3),TEXT(4) /'C','ONST','ANT ','HEAD'/

------------------------------------------------------------------

Cl......CLEAR BUDGET ACCUMULATORS

CHIN=0.

CHOUT=0.

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

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.

1 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 7-17 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-1,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+1,I,K)-HNEW(J-1,I,K)
X2=HDIFF*CR(J+1,I,K)
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,1-1,K).LE.0) GO TO 90
HDIFF=HNEW(J,1-1,K)-HNEW(J,1+1,K)
X3=HDIFF*BC(J,1-1,K)
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,1+1,K).LE.0) GO TO 120
HDIFF=HNEW(J,1+1,K)-HNEW(J,1-1,K)
X4=HDIFF*BC(J,1+1,K)
10 CHOUT=CHOUT-X4
IF (X4) 100,120,110
B-26 

Engineering Technologies Associates, Inc.
100 CHOUT=CHOUT-X4
GO TO 120
110 CHIN=CHIN+X4
C
C6----CALCULATE FLOW THROUGH THE UPPER FACE
120 IF(K.EQ.1) GO TO 150
   IF (IBOUND(J,1,K-1).LE.0) GO TO 150
   HD=HNEW(J,1,K)
   IF(LC.NE.3 .AND. LC.NE.2) GO TO 122
   TMP=HD
   IF(TMP.LT.TOP(J,1,KT)) HD=TOP(J,1,KT)
122 HDIFF=HD-HNEW(J,1,K-1)
   X5=HDIFF*CV(J,1,K-1)
   IF(X5) 130,150,140
130 CHOUT=CHOUT-X5
   GO TO 150
140 CHIN=CHIN+X5
C
C6----CALCULATE FLOW THROUGH THE LOWER FACE.
150 IF(K.EQ.NLAY) GO TO 180
   IF(IBOUND(J,1,K+1).LE.0) GO TO 180
   HD=HNEW(J,1,K+1)
   IF(LAYCON(K+1).NE.3 .AND. LAYCON(K+1).NE.2) GO TO 152
   TMP=HD
   IF(TMP.LT.TOP(J,1,KT+1)) HD=TOP(J,1,KT+1)
152 HDIFF=HNEW(J,1,K)-HD
   X6=HDIFF*CV(J,1,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<0.AND.ICBCFL.NE.0) WRITE(IOUT,900) (TEXT(N),N=1,4),
      1 ,KPER,KSTP,K,I,J,RATE
900 FORMAT(1HO,4A4,' PERIOO',I3,' STEP',I3,' LAYER',13,'
         1 ',ROW',I4,' COL',I4,' RATE ',G15.7)
C
C CALL XYZPOS(J,1,K,X,Y,Z,DELR,DELC,NROW,NCOL,NLAYER)
   IF(RATE .LT. 0)WRITE(IVOUT,1001)X,Y,Z,RATE*CONVRT
1001 FORMAT(313,F12.4)
200 CONTINUE
C17----RETURN
RETURN
END
SUBROUTINE HYCALC(CC,HY,TRPY,IBOUND,1,RO,TOP,K,KT,NCOL,NLAY)
C
C CALCULATE POSITION OF SINK
   CALL XYZPOS(J,1,K,X,Y,Z,DELR,DELC,NROW,NCOL,NLAYER)
C
C COMPUTE HYDRAULIC CONDUCTIVITY FROM TRANSMISSIVITY AND THICKNESS

B-27

Engineering Technologies Associates, Inc.
**SPECIFICATIONS:**

DOUBLE PRECISION HNEW

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)

COMMON /FLWCCM/LAYCON(80)

------------------------------------------------------------------

CALCULATE TRANSMISSIVITY AT EACH ACTIVE CELL. TRANSMISSIVITY
WILL BE STORED TEMPORARILY IN THE CC ARRAY.

DO 200 I=1,NROW
DO 200 J=I,NCOL

CALCULATE THICKNESS.

THCK=TOP(J,I,KT)-BOT(J,I,KB)

CHECK TO SEE IF THICKNESS IS GREATER THAN ZERO.

IF(THCK.LE.0.) GO TO 200

IF THICKNESS>O THEN K=T/THICKNESS.

CC(J,I,K)=CC(J,I,K)/THCK
GO TO 200

WHEN THICKNESS < 0, PRINT A MESSAGE AND SET
TRANSMISSIVITY, AND VERTICAL CONDUCTANCE =0

WRITE(IOUT,150) J,I,K
150 FORMAT(1H10,10('**'),9NCOL,314,' (COL,ROW,LAYER) IS DRY')

HNEW(J,I,K)=1.E30

CC(J,I,K)=0.

IBOUND(J,I,K)=0

IF(K.LT.NLAY) CV(J,I,K)=0.

IF(K.GT.1) CV(J,I,K-1)=0.

CONTINUE

RETURN

RETURN

END

SUBROUTINE SBCFIN(HNEW, IBOUND, SC1, SC2, CR, CC, CV, HY, TRPY, DELR, DELC,
1 ISS, NCOL, NROW, NLAY, IOUT)

-------VERSION 1007 03MAY1983 SBCFIN

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

INITIALIZE AND CHECK BCF DATA

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

SPECIFICATIONS:
DOUBLE PRECISION HNEW, HCNV

DIMENSION HNEW(NCOL, NROW, NLAY), IBOUND(NCOL, NROW, NLAY)
1 , SC(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)

COMMON /FLWCOM/LAYCON(80)

C

C

1

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.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 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) 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)',314, I

1 Eliminated because all conductances to node are 0')

54 CONTINUE

C

C2B

WHEN LAYER TYPE 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

59 CONTINUE
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 HNEW(J,I,K)=HCNV
CC(J,I,K)=0.
WRITE(IOUT,52) K,I,J
59 CONTINUE
60 CONTINUE
C
C3------CALCULATE HORIZONTAL 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 SBCFIC(CR,CC,TRPY,DELR,DELC,K,NCOL,NROW,NLAY)
65 CONTINUE
C
C4------MULTIPLY VERTICAL LEAKANCE BY AREA TO MAKE CONDUCTANCE
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 & 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(1)
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(1)
75 CONTINUE
C
80 CONTINUE
C
C7------RETURN
100 RETURN
END
$DEBUG
$LARGEOAD
$NOCALLS

SUBROUTINE BASPRE(HNEW,BUFF,IOFLG,KSTP,KPER,NCOL,NROW,
1  NLAY,IOUT,IHEDDM,IHEDDU,IPFLG,PERTIM,TOTIM)
C
C ******************************************************************************
C READ PREVIOUSLY SAVED HEADS FROM BINARY HEAD SAVE FILE
C ******************************************************************************
C MODIFIED FOR PREMOD3D BY DON KOCH, ENGINEERING TECHNOLOGIES
ASSOCIATES FROM USGS MODFLOW - SBASIH SUBROUTINE
C
C K1-STEP # IN INPUT FILE
C K2-PERIOD # IN INPUT FILE
C IHEEDUN- UNIT # TO READ INPUT FROM
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=IHEEDUN
   CALL ULAREAD(BUFF,TEXT(1),K1,K2,PERTIM,TOTIM,NCOL,NROW,ILAYER,
1  IHEEDU)
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
C
C5-------FOR EACH LAYER: PRINT HEAD IF REQUESTED.
   DO 39 K=1,NLAY
   C
B-31
Engineering Technologies Associates, Inc.
C6------TEST IOFLG TO SEE IF HEAD SHOULD BE PRINTED.
   IF(IOFLG(K,I).EQ.0) GO TO 39
      IPFLG=1

C7------COPY HEADS FOR THIS LAYER INTO BUFFER.
   DO 32 J=1,NCOL
   DO 32 I=1,NROW
      BUFF(J,I,1)=HNEW(J,I,K)
      CONTINUE

C8------CALL UTILITY MODULE TO PRINT CONTENTS OF BUFFER.
   IF(IHEDFM.LT.0) CALL ULAPRS(BUFF,TEXT(1),KSTP,KPER,NCOL,NROW,K,
                                -IHEDFM,IOUT)
   IF(IHEDFM.GE.0) CALL ULAPRW(BUFF,TEXT(1),KSTP,KPER,NCOL,NROW,K,
                               IHEDFM,IOUT)
   CONTINUE

C10------RETURN
      RETURN

C11------PRINT ERROR MESSAGE IF NOT READING CORRECT DATA
   WRITE(*,*)' NOT READING CORRECT DATA IN FILE'
      STOP 0001
   END
SUBROUTINE ULAREAD(BUF,TEXT,KSTP,KPER,PERTIM,TOTIM,NCOL,
                   NROW,ILAY,ICHN)
   READ AN UNFORMATTED RECORD CONTAINING IDENTIFYING
   INFORMATION.
   READ(ICHN) KSTP,KPER,PERTIM,TOTIM,TEXT,NCOL,NROW,K

   READ AN UNFORMATTED RECORD CONTAINING ARRAY VALUES
   THE ARRAY IS DIMENSIONED (NCOL,NROW)
   READ(ICHN) ((BUF(IC,IR),IC=I,NCOL),IR=1,NROW)

   RETURN
RETURN
   END
SUBROUTINE UBUDSV(KSTP,KPER,TEXT,IBDCHN,BUFF,NCOL,NROW,NLAY,IOUT)

   -------------------------------
   READ 1 LAYER ARRAY FROM BINARY FILE
   -------------------------------

   -------------------------------
   SPECIFICATIONS:
   -------------------------------

   DIMENSION BUF(NCOL,NROW)

   CHARACTER*4 TEXT(4)

   READ AN UNFORMATTED RECORD CONTAINING IDENTIFYING
   INFORMATION.
   READ(ICHN) KSTP,KPER,PERTIM,TOTIM,TEXT,NCOL,NROW,ILAY

   READ AN UNFORMATTED RECORD CONTAINING ARRAY VALUES
   THE ARRAY IS DIMENSIONED (NCOL,NROW)
   ((BUF(IC,IR),IC=I,NCOL),IR=1,NROW)

   RETURN
RETURN
   END
SUBROUTINE UBUDSV(KSTP,KPER,TEXT,IBDCHN,BUFF,NCOL,NROW,NLAY,IOUT)
RECORD CELL-BY-CELL FLOW TERMS FOR ONE COMPONENT OF FLOW.

SPECIFICATIONS:

DIMENSION BUFF(NCOL,NROW,NLAY)

DIMENSION TEXT(4)

CHARACTER*4 TEXT(4)

WRITE(IOUT,1) TEXT,IBDCHN,KSTP,KPER

1 FORMAT(1X,'""','4A4',' BUDGET VALUES WILL BE SAVED ON UNIT',13,'
  AT END OF TIME STEP',13,' ',STRESS PERIOD',13)

WRITE(IBDCHN) KSTP,KPER,TEXT,NCOL,NROW,NLAY

WRITE(IBDCHN) BUFF

RETURN

RETURN

END

SUBROUTINE UCOLNO(NLBL1,NLBL2,NSPACE,NCPL,NDIG,IOUT)

SPECIFICATIONS:

DIMENSION BF(130),DG(10)

CHARACTER*4 BF(130),DG(10),DOT,SPACE

VERSION 1446 20APR1983 UCOLNO

OUTPUT COLUMN NUMBERS ABOVE A MATRIX PRINTOUT

NLBL1 IS THE START COLUMN LABEL (NUMBER)

NLBL2 IS THE STOP COLUMN LABEL (NUMBER)

NSPACE IS NUMBER OF BLANK SPACES TO LEAVE AT START OF LINE

NCPL IS NUMBER OF COLUMN NUMBERS PER LINE

NDIG IS NUMBER OF CHARACTERS IN EACH COLUMN FIELD

IOUT IS OUTPUT CHANNEL

SPECIFICATIONS:

DIMENSION BF(130),DG(10)

CHARACTER*4 BF(130),DG(10),DOT,SPACE
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 DC,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
  12=J/10
  11=J-12*10+1
  BF(NBF)=DG(11)
  IF(12.EQ.0) GO TO 30
  13=12/10
  12=12-13*10+1
  BF(NBF-1)=DG(12)
  IF(13.EQ.0) GO TO 30
  BF(NBF-2)=DG(13+1)
30 CONTINUE
C
  WRITE(IOUT,31) (BF(I),I=1,NBF)
31 FORMAT(1X,130A1)
C
C  40 CONTINUE
C
C7------PRINT A LINE OF DOTS (FOR ESTHETIC PURPOSES ONLY).
      NTOT=NTOT+5
IF(NTOT.GT.130) NTOT=130
WRITE(ICUT,51) (DCT,I=1,NTOT)
51 FORMAT(1X,130A1)
C
C8-----RETURN
RETURN
END

SUBROUTINE ULAPRS(BUF,TEXT,KSTP,KPER,NCOL,NROW,ILAY,IPRN,IOUT)
C
C8
C--...VERSION 1448 20APR1983 ULAPRS
C
C PRINT A 1 LAYER ARRAY IN STRIPS
C
C SPECIFICATIONS:
C
DIMENSION BUF(NCOL,NROW)
C
C DIMENSION TEXT(4)
C
CHARACTER*4 TEXT(4)
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(M1,H1,10X,4A4,' IN LAYER',I3,' AT END OF TIME STEP',I3,
      1   ' IN STRESS PERIOD',I3/1X,7(' -'))
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(M1,H1,13,2X,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(M1,H1,13,2X,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(M1,H1,13,1X,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(M1,H1,13,1X,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(M1,H1,13,1X,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(M1,H1,13,1X,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(M1,H1,13,1X,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(M1,H1,13,1X,20(1X,F5.1))
    GO TO 1000
C
C--------FORMAT 20F5.2
    90 WRITE(IOUT,91) I,(BUF(J,I),J=J1,J2)
SUBROUTINE ULAPRW(BUF, TEXT, KSTP, KPER, NCOL, NROW, ILAY, IPRN, IOUT)

C
C
C
C SPECIFICATIONS:
DIMENSION BUF(NCOL, NROW)

CF66
C DIMENSION TEXT(4)
CF66
CF77
CHARACTER*4 TEXT(4)
CF77
C
C1-----PRINT A HEADER
IF(ILAY.LE.0) GO TO 5
WRITE(IOUT, 1) TEXT, ILAY, KSTP, KPER
1 FORMAT(1H1, 1OX, 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 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, 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)

CLOOP 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(1HO, 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(1HO, 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(1HO, 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(1HO, 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(1HO, 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(1HO, 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(1HO, 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(1HO, 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(1HO, I3, 1X, 20(1X, F5.2)/(5X, 20(1X, F5.2)))
   GO TO 1000

Engineering Technologies Associates, Inc.
C             FORMAT 20F5.3
   100 WRITE(IOUT,101) I,(BUF(J,I),J=1,NCOL)
   101 FORMAT(1HO,13,1X,20(1X,F5.3)/(5X,20(1X,F5.3)))
   GO TO 1000
C             FORMAT 20F5.4
   110 WRITE(IOUT,111) I,(BUF(J,I),J=1,NCOL)
   111 FORMAT(1HO,13,1X,20(IX,F5.4)/(5X,20(IX,F5.4)))
   GO TO 1000
C             FORMAT 1OG11.4
   120 WRITE(IOUT,121) I,(BUF(J,I),J=1,NCOL)
   121 FORMAT(1HO,13,2X,1PGI1.4,9(1X,G11.4)/(SX,10(1X,G11.4)))
1000 CONTINUE
C      C5------RETURN
      RETURN
      END
SUBROUTINE ULASAV(BUF,TEXT,KSTP,KPER,PERTIM,TOTIM,NCOL,
                  NROW,ILAY,ICHN)
C             VERSION 1445 20APR1983 ULASAV
C             *******************************************
C             SAVE 1 LAYER ARRAY ON DISK
C             *******************************************
C             SPECIFICATIONS:
C             *******************************************
DIMENSION BUF(NCOL,NROW)
C DIMENSION TEXT(4)
CHARACTER*4 TEXT(4)
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 UIDREL(A,ANAME,JJ,IN,IOUT)
C             VERSION 1436 20MAY1983 UIDREL

Engineering Technologies Associates, Inc.
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 SPECIFICATIONS:
C
DIMENSION A(JJ)
CF66
C DIMENSION ANAME(6),FMTIN(5)
CF66
CF77
CHARACTER ANAME(6)*4,FMTIN*20
CF77
C
C1------READ ARRAY CONTROL RECORD.
READ (IN,1) LOCAT,CNSTNT,FMTIN,IPRN
CF66
C 1 FORMAT(I10,F10.0,5A4,11O)
CF66
CF77
1 FORMAT(I10,F10.0,A20,11O)
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(1HO,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(1HO,///30X,6A4,' WILL BE READ ON UNIT',13,
C 1 ' USING FORMAT: ',5A4///30X,79(''-')/)
CF66
CF77
5 FORMAT(1HO,///30X,6A4,' WILL BE READ ON UNIT',13,
C 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
C6------ IF PRINT CODE (IPRN) => O THEN PRINT ARRAY VALUES.
120 IF(IPRN.LT.O) RETURN
     WRITE(IOUT,1001) (A(J),J=1,JJ)
1001 FORMAT((1X,1PG12.5,9(I1X,G12.5)))
     RETURN

C7------ CONTINUE

CONTINUE

END

SUBROUTINE U2DINT(IA,ANAME,II,JJ,K, IN,IOUT)

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 O)
C IN IS INPUT UNIT
C IOUT IS OUTPUT UNIT

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

C SPECIFICATIONS:

DIMENSION IA(JJ,II)

C DIMENSION ANAME(6),FMTIN(5)

CHARACTER ANAME(6)*4,FMTIN*20

C CHARACTER ANAME(6)*4,FMTIN*20

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

C1------- READ ARRAY CONTROL RECORD.
     READ (IN,1) LOCAT,ICONST,FMTIN,IPRN

C FORMAT(I10,110,5A4,110)

CF77
     1 FORMAT(I10,110,5A4,110)
     WRITE(IOUT,*) IN = ',IN

C2------- USE LOCAT TO SEE WHERE ARRAY VALUES COME FROM.
     IF(LOCAT) 200,50,90

C3------- IF LOCAT=O 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.O) WRITE(IOUT,2) ANAME,ICONST,K
2 FORMAT(1H0,52X,6A4,' = ',115, ' FOR LAYER',13)
   IF(K.LE.0) WRITE(IOUT,3) ANAME,ICONST
3 FORMAT(1H0,52X,6A4,' = ',115)
   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
C66
C 4 FORMAT(1H0,///30X,6A4,' FOR LAYER',13,' WILL BE READ ON UNIT',
C     1 13,' USING FORMAT: ','5A4/30X,96('--'))
C66
C77
C 4 FORMAT(IHO,///30X,6A4,' FOR LAYER',13,' WILL BE READ ON UNIT',
C     1 13,' USING FORMAT: ','A20/30X,96('--'))
C77

IF(K.LE.0) WRITE(IOUT,5) ANAME,LOCAT,FMTIN
C66
C 5 FORMAT(IHO,///30X,6A4,' WILL BE READ ON UNIT',
C     1 13,' USING FORMAT: ','5A4/30X,83('--'))
C66
C77
C 5 FORMAT(IHO,///30X,6A4,' WILL BE READ ON UNIT',
C     1 13,' USING FORMAT: ','A20/30X,83('--'))
C77

DO 100 I=1,11
   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',13,
C     1 ' WILL BE READ UNFORMATTED ON UNIT',13/30X,73('--'))
   IF(K.LE.0) WRITE(IOUT,202) ANAME,LOCAT
202 FORMAT(1H0,///30X,6A4,
C     1 ' WILL BE READ UNFORMATTED ON UNIT',13/30X,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=1,11
      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, 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.
   DC 110 I=1,II
C
C10------SELECT THE FORMAT
   GO TO(101, 102, 103, 104, 105, 106), IPRN
C
C----------FORMAT 1011
101 WRITE(IOUT,1001) I,(IA(J,I),J=1, JJ)
   1001 FORMAT(1HO,13,2X,111,9(1X,II1)/(5X,10(1X,II1)))
   GO TO 110
C
C----------FORMAT 6011
102 WRITE(IOUT,1002) I,(IA(J,I),J=1, JJ)
   1002 FORMAT(1HO,13,1X,60(1X,II1)/(5X,60(1X,II1)))
   GO TO 110
C
C----------FORMAT 4012
103 WRITE(IOUT,1003) I,(IA(J,I),J=1, JJ)
   1003 FORMAT(1HO,13,1X,40(1X,II2)/(5X,40(1X,II2)))
   GO TO 110
C
C----------FORMAT 3013
104 WRITE(IOUT,1004) I,(IA(J,I),J=1, JJ)
   1004 FORMAT(1HO,13,1X,30(1X,II3)/(5X,30(1X,II3)))
   GO TO 110
C
C----------FORMAT 2514
105 WRITE(IOUT,1005) I,(IA(J,I),J=1, JJ)
   1005 FORMAT(1HO,13,1X,25(1X,II4)/(5X,25(1X,II4)))
   GO TO 110
C
C----------FORMAT 2015
106 WRITE(IOUT,1006) I,(IA(J,I),J=1, JJ)
   1006 FORMAT(1HO,13,1X,20(1X,II5)/(5X,20(1X,II5)))
   110 CONTINUE
   RETURN
C
C11------RETURN
   END
SUBROUTINE U2DREL(A,ANAME,II,JJ,K, IN,IOUT)
C
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

Engineering Technologies Associates, Inc.
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
******************************************************************************
C
DIMENSION A(JJ,II)
C
C DIMENSION ANAME(6),FMTIN(5)
C
CHARACTER ANAME(6)*4,FMTIN*20
C
C
C1------READ ARRAY CONTROL RECORD.
C
READ (IN,1) LOCAT,CNSTNT,FMTIN,IPRN
C
1 FORMAT(I10,F10.0,5A4,110)
C
1 FORMAT(I10,F10.0,A20,110)
C
C2------USE LOCAT TO SEE WHERE ARRAY VALUES COME FROM.
C
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(1HO,52X,6A4,' =',G15.7,' FOR LAYER',I3)
      IF(K.LE.0) WRITE(IOUT,3) ANAME,CNSTNT
      3 FORMAT(1HO,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
C
        4 FORMAT(1HO,///30X,6A4,' FOR LAYER',I3,' WILL BE READ ON UNIT',13,' USING FORMAT: ',5A4/30X,96('-'))
C
        1 13,' USING FORMAT: ',5A4/30X,96('-')
C
C77
        4 FORMAT(1HO,///30X,6A4,' FOR LAYER',I3,' WILL BE READ ON UNIT',13,' USING FORMAT: ',A20/30X,96('-'))
C
        IF(K.LE.0) WRITE(IOUT,5) ANAME,LOCAT,FMTIN
C
        5 FORMAT(1HO,///30X,6A4,' WILL BE READ ON UNIT',13,' USING FORMAT: ',5A4/30X,83('-'))
C
        IF(K.LE.0) WRITE(IOUT,5) ANAME,LOCAT,FMTIN
C
        5 FORMAT(1HO,///30X,6A4,' WILL BE READ ON UNIT',13,' USING FORMAT: ',5A4/30X,83('-'))
C
        IF(K.LE.0) WRITE(IOUT,5) ANAME,LOCAT,FMTIN

B-44

Engineering Technologies Associates, Inc.
5 FORMAT(1HO,///30X,6A4,' WILL BE READ ON UNIT',
1           13,' USING FORMAT: ',A2/30X,B3(''-'))

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(1HO,///30X,6A4,', LAYER',13,
1           ' WILL BE READ UNFORMATTED ON UNIT',13/30X,73('-'))
   IF(K.LE.0) WRITE(COUT,202) ANAME,LOCAT
202 FORMAT(IHO,///30X,
1           ' WILL BE READ UNFORMATTED ON UNIT',13/30X,60('-'))

C5A------READ AN UNFORMATTED DUMMY RECORD FIRST.
   READ(LOCAT)
   READ(LOCAT) A

C6------IF CNSTNT NOT ZERO THEN MULTIPLY ARRAY VALUES BY CNSTNT.
300 IF(CNSTNT.EQ.0.) GO TO 320
   DO 310 I=1,11
      DO 310 J=1,JJ
         A(J,I)=A(J,I)*CNSTNT
   310 CONTINUE

C7------IF PRINT CODE (IPRN) =>0 THEN PRINT ARRAY VALUES.
320 IF(IPRN.LT.0) RETURN
   CALL ULPWRW(A,ANAME,0,0,JJ,II,0,IPRN,IOUT)
   RETURN

C8------RETURN
END

SUBROUTINE BASIAD(DELT,TSMULT,TOTIM,PERTIM,HNEW,HOLD,KSTP,
1                  NCOL,NROW,NLAY)

C-----VERSION 1412 22FEB1982 BASIAD
C
C ********************************************************************************
C ADVANCE TO NEXT TIME STEP
C ********************************************************************************
C
C SPECIFICATIONS:
C
C DOUBLE PRECISION HNEW
C
C DIMENSION HNEW(NCOL,NROW,NLAY), HOLD(NCOL,NROW,NLAY)
C
C1------IF NOT FIRST TIME STEP THEN CALCULATE TIME STEP LENGTH.
   IF(KSTP.NE.1) DELT=TSMULT*DELT
C2------ACCUMULATE ELAPSED TIME IN SIMULATION(TOTIM) AND IN THIS
C2------STRESS PERIOD(PERTIM).
   TOTIM=TOTIM+DELT
   PERTIM=PERTIM+DELT
C
C3------COPY HHNEW TO HOLD.
   DO 10 K=1,NLAY
       DO 10 I=1,NROW
           DO 10 J=1,NCOL
           10 HOLD(J,I,K)=HHNEW(J,I,K)
C
C4------RETURN
   RETURN
END
SUBROUTINE BAS1AL(ISUM,LENX,LCHNEW,LCHOLD,LCIBOU,LCCR,LCCC,LCCV,
   LCHCOF,LCRHS,LCDEL,LCDLC,LCASTR,LCBUFF,LCIOLFL,BAS1AS,
   ISTRT,NCOL,NROW,NLAY,IOUT)
C
C1------PRINT A MESSAGE IDENTIFYING THE PACKAGE.
   WRITE(IOUT,1)INBAS
   1 FORMAT(1HO,'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(2110)
       IF(IAPART.EQ.O) WRITE(IOUT,3)
       3 FORMAT(IX,'ARRAYS RHS AND BUFF WILL SHARE MEMORY.')
       IF(ISTRT.NE.O) WRITE(IOUT,4)
       4 FORMAT(IX,'START HEAD WILL BE SAVED')
       IF(ISTRT.EQ.D) 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

Engineering Technologies Associates, Inc.
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
LCDEL=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',17)
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 SPECIFICATIONS:
C
DIMENSION HEADNG(32),IUNIT(24)
PRINT THE NAME OF THE PROGRAM.
WRITE(IO1UT,1)
1 FORMAT(1X,'ETA PREPROCESSOR - PREPARES MODFLOW FILES AND',
      1       ' OUTPUT FOR INPUT TO RAND3D')
C
READ AND PRINT A HEADING.
READ(INBAS,2) HEADNG
2 FORMAT(20A4)
WRITE(IO1UT,3) HEADNG
3 FORMAT(1X,32A4)
C
READ NUMBER OF LAYERS, ROWS, COLUMNS, STRESS PERIODS AND
UNITS OF TIME CODE.
READ(INBAS,4) NLAY, NROW, NCOL, NPER, ITMUNI
4 FORMAT(8I10)
C
PRINT # OF LAYERS, ROWS, COLUMNS AND STRESS PERIODS.
WRITE(IO1UT,5) NLAY, NROW, NCOL
5 FORMAT(1X,14,' LAYERS',110,' ROWS',110,' COLUMNS')
WRITE(IO1UT,6) NPER
6 FORMAT(1X,13,' STRESS PERIOD(S) IN SIMULATION')
C
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(IO1UT,9)
9 FORMAT(1X,'MODEL TIME UNITS ARE UNDEFINED')
GO TO 100
10 WRITE(IO1UT,11)
11 FORMAT(1X,'MODEL TIME UNIT IS SECONDS')
GO TO 100
20 WRITE(IO1UT,21)
21 FORMAT(1X,'MODEL TIME UNIT IS MINUTES')
GO TO 100
30 WRITE(IO1UT,31)
31 FORMAT(1X,'MODEL TIME UNIT IS HOURS')
GO TO 100
40 WRITE(IO1UT,41)
41 FORMAT(1X,'MODEL TIME UNIT IS DAYS')
GO TO 100
50 WRITE(IO1UT,51)
51 FORMAT(1X,'MODEL TIME UNIT IS YEARS')
C
READ & PRINT INPUT UNIT NUMBERS (IUNIT) FOR MAJOR OPTIONS.
100 READ(INBAS,101) IUNIT
101 FORMAT(24I3)
WRITE(IO1UT,102) (I,I=1,24),IUNIT
102 FORMAT(1H0,'/I/O UNITS:',24I3,
      1        '/IUNIT:',NUNIT)
C
INITIALIZE TOAL ELAPSED TIME COUNTER STORAGE ARRAY COUNTER
C
AND CALCULATE NUMBER OF CELLS.
SUBROUTINE BASIFM(HCOF,RHS,NCOL,NROW,NLAY,NODES)

C
C
C

C SET HCOF=RHS=0.
C
C
C SPECIFICATIONS:

DIMENSION HCOF(NODES),RHS(NODES)

C
C FOR EACH CELL INITIALIZE HCOF AND RHS ACCUMULATORS.
DO 100 I=1,NODES
HCOF(I)=0.
RHS(I)=0.
100 CONTINUE
C
C
C
C
RETURN
RETURN
END

SUBROUTINE BASOCC(NSTP,KSTP,KPER,ISTRT,ICNVG,IOFLG,NLAY,
IBUDFL,ICBCFL,IHDDFL,INOC,IOUT)

C
C
C

C OUTPUT CONTROLLER FOR HEAD, DRAWDOWN, AND BUDGET

C
C SPECIFICATIONS:

DIMENSION IOFLG(NLAY,4)

C
C TEST UNIT NUMBER (INOC (INOC=IUNIT(12))) TO SEE IF
C
IF(INOC.NE.0)GO TO 500
C
C
C IF OUTPUT CONTROL IS INACTIVE THEN SET DEFAULTS AND RETURN.

C
C
C IF(INOC.EQ.0 .OR. KSTP.EQ.NSTP)IHDDFL=I
IBUDFL=0
IF(INOC.EQ.0 .OR. KSTP.EQ.NSTP)IBUDFL=1
ICBCFL=0
GO TO 1000
C3------READ AND PRINT OUTPUT FLAGS AND CODE FOR DEFINING IOFLG.
500 READ(INOC,1) INCODE, IHDDFL, IBUDFL, ICBCFL
   1 FORMAT(4:10)
   WRITE(IOUT,3) IHDDFL, IBUDFL, ICBCFL
   3 FORMAT(1H0,'HEAD/DRAWDOWN PRINTOUT FLAG =',12,
            5X,'TOTAL BUDGET PRINTOUT FLAG =',12,
            5X,'CELL-BY-CELL FLOW TERM FLAG =',12)
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(1H0,'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(K,1),M=1,4)
   201 FORMAT(410)
      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:'/
            1X,'HEAD DRAWDOWN HEAD DRAWDOWN'/
            1X,'PRINTOUT PRINTOUT SAVE SAVE'/
            1X,34('-')/1X,I5,110,I8,I8)
   GO TO 600
C
C7------READ IOFLG IN ENTIRETY
   300 READ(INOC,301) ((IOFLG(K,1),I=1,4),K=1,NLAY)
   301 FORMAT(410)
   WRITE(IOUT,302)
   302 FORMAT(1H0,'OUTPUT FLAGS FOR EACH LAYER:'/
            1X,'HEAD DRAWDOWN HEAD DRAWDOWN'/
            1X,'LAYER PRINTOUT PRINTOUT SAVE SAVE'/
            1X,41('-'))
   WRITE(IOUT,303) (K,(IOFLG(K,1),I=1,4),K=1,NLAY)
   303 FORMAT(1X,I4,18,110,18,IS)
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,
                   VBV, KSTP,KPER,DELT,PERTIM,TOTIM,ITMUNI,NCOL,NROW,NLAY,ICNVG,
                   IHDDFL, IBUDFL, IHEDFM, IHEDUN, IBDUN, IBDNUM, IOUT)

Engineering Technologies Associates, Inc.
SUBROUTINE BASIRP(IBOUND,HNEW,STRT,HOLD,ISTRT,INBAS,HEADNG,NCOL,NROW,NLAY,NODES,VBVL,IOFLG,INOC,IHEDFM,IPFLG)

C-----OUTPUT TIME, VOLUMETRIC BUDGET, HEAD, AND DRAWDOWN
C
C SPECIFICATIONS:
C
DOUBLE PRECISION HNEW
C
DIMENSION HNEW(NCOL,NROW,NLAY),STRT(NCOL,NROW,NLAY),
1 VBLVL(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
C CLEAR PRINTOUT FLAG (IPFLG)
IPFLG=0
C
IF ITERATIVE PROCEDURE FAILED TO CONVERGE PRINT MESSAGE
IF(INCNV.EQ.0) WRITE(IOUT,1) KSTP,KPER
1 FORMAT(IHO,10X,'****FAILED TO CONVERGE IN TIME STEP',I3,
1 ' OF STRESS PERIOD',I3,'****')
C
IF HEAD AND DRAWDOWN FLAG (IHDDFL) IS SET WRITE HEAD AND
C DRAWDOWN IN ACCORDANCE WITH FLAGS IN IOFLG.
IF(IHDDFL.EQ.0) GO TO 100
C
CALL SBASIH(HNEW,BUFF,IOFLG,KSTP,KPER,NCOL,NROW,INBAS,HEADNG,NCOL,NROW,NLAY,NODES,VBVL,IOFLG,INOC,IHEDFM,IPFLG,DELT,PERTIM,TOTIM)
CALL SBAS1D(HNEW,BUFF,IOFLG,KSTP,KPER,NCOL,NROW,NLAY,INBAS,HEADNG,NCOL,NROW,NLAY,NODES,VBVL,IOFLG,INOC,IHEDFM,IPFLG,DELT,PERTIM,TOTIM)
C
PRINT TOTAL BUDGET IF REQUESTED
100 IF(IBUDFL.EQ.0) GO TO 120
CALL SBASIV(MSUM,VBNM,VBVL,KSTP,KPER,IOUT)
IPFLG=1
C
END PRINTOUT WITH TIME SUMMARY AND FORM FEED IF ANY PRINTOUT
C WILL BE PRODUCED.
120 IF(IPFLG.EQ.0) RETURN
CALL SBASIT(KSTP,KPER,DELT,PERTIM,TOTIM,ITMUN,IOUT)
WRITE(IOUT,101)
101 FORMAT(1H1)
C
RETURN
RETURN
END
C READ AND INITIALIZE BASIC MODEL ARRAYS

C SPECIFICATIONS:

DOUBLE PRECISION HNEW,HNOFLO

DIMENSION HNEW(NODES),IBOUND(NCDES),STRT(NODES),HOLD(NODES),
1 VBVL(4,20),IOFLG(NLAY,4),HEADNG(32)

DIMENSION ANAME(6,2)

CHARACTER*4 ANAME(6,2)

DATA ANAME(I,I),ANAME(2,1),ANAME(3,1),ANAME(4,1),ANAME(5,1),
1 ANAME(6,1) /'i',' ','BO','UNDA','RY A','RARY'/

DATA ANAME(1,2),ANAME(2,2),ANAME(3,2),ANAME(4,2),ANAME(5,2),
1 ANAME(6,2) /'i',' ','','INIT','IAL ','HEAD'/

C PRINT SIMULATION TITLE, CALCULATE # OF CELLS IN A LAYER.
WRITE(IOUT,1) HEADNG
1 FORMAT(1H1,32A4)
NCR=NCOL*NROW

C 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 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 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 COPY INITIAL HEADS FROM HOLD TO HNEW.
DO 400 I=1,NODES
HNEW(I)=HOLD(I)
1 IF(IBOUND(I).EQ.0) HNEW(I)=HNOFLO
400 CONTINUE
C4-------INITIALIZE PERTIM (ELAPSED TIME WITHIN STRESS PERIOD).
PERTIM=0.

C5-------RETURN
RETURN
END

SUBROUTINE SBAS1D(HNEW,BUFF,IOFLG,KSTP,KPER,NCOL,NROW,1
NLAY,IOUT,IDDNM,STRT,ISTRT,IBOUND,IPFLG,2
PERTIM,TOTIM)

C-----VERSION 1147 29MAR1984 SBAS1D

C ***CALCULATE PRINT AND RECORD DRAWDOWNS***

C ***SPECIFICATIONS***

DOUBLE PRECISION HNEW

DIMENSION HNEW(NCOL,NROW,NLAY),IOFLG(NLAY,4),
BUFF(NCOL,NROW,NLAY),STRT(NCOL,NROW,NLAY), 1
IBOUND(NCOL,NROW,NLAY)

CHARACTER*4 TEXT(4)

DATA TEXT(1),TEXT(2),TEXT(3),TEXT(4)
/I',',',DRAW',1
'DOWN'/

C1-------FOR EACH LAYER CALCULATE DRAWDOWN IF PRINT OR RECORD
C1-------IS REQUESTED
DO 59 K=1,NLAY

C2-------DRAWDOWN NEEDED FOR THIS LAYER?
IF(IOFLG(K,2).EQ.O .AND. IOFLG(K,4).EQ.O) GO TO 59

C3-------DRAWDOWN IS NEEDED. WERE STARTING HEADS SAVED?
IF(ISTRT.NE.O) GO TO 53

C4-------STARTING HEADS WERE NOT SAVED. PRINT MESSAGE AND STOP.
WRITE(IOUT,52)
52 FORMAT(IHO,'CANNOT CALCULATE DRAWDOWN BECAUSE START',1
' HEADS WERE NOT SAVED')
STOP

C5-------CALCULATE DRAWDOWN FOR THE LAYER.
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.O) BUFF(J,I,K)=STRT(J,I,K)-HSING

58

59
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(NLAYISTRT,IOFLG,INOC,IOUT,IHEDFM,
      1      IDDNFM,IHEDUN,IDDNMUN)
C
C9------RETURN
1000 RETURN
END
SUBROUTINE BASIST(NSTP,DELT,TSMULT,PERTIM,KPER,INBAS,IOUT)
C
C
C-----VERSION 1614 08SEP1982 BASIST
C
C SETUP TIME PARAMETERS FOR NEW TIME PERIOD
C
C SPECIFICATIONS:
C
C---------------------
READ LENGTH OF STRESS PERIOD, NUMBER OF TIME STEPS AND.
C---------------------
READ (INBAS,1) PERLEN,NSTP,TSMULT
1 FORMAT(F10.0,0,110,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 (1OU7,2) KPER,PERLEN,NSTP,TSMULT,DELT
2 FORMAT(1H1,51X,'STRESS PERIOD NO.',14,' ',LENGTH=',G15.7/52X
      1,46(':'')/52X,'NUMBER OF TIME STEPS =',16
      2//53X,'MULTIPLIER FOR DELT =',F10.3
      3//50X,'INITIAL TIME STEP SIZE =',G15.7)
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.O) GO TO 69
IF(IDDNFM.LT.O) CALL ULAPRS(BUFF(1,1,K),TEXT(1),KSTP,KPER,
1 NCOL,NROW,K,-IDDNFM,IOUT)
IF(IDDNFM.GE.O) 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.O) GO TO 80
DO 79 K=1,NLAY
IF(IOFLG(K,4).LE.O) GO TO 79
IF(IFIRST.EQ.1) WRITE(IOUT,74) IDDNUN,KSTPKPER
74 FORMAT(IHO,'DRAWDOWN WILL BE SAVED ON UNIT',13,
1 ' AT END OF TIME STEP',13,' STRESS PERIOD',13)
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 SPECIFICATIONS
C
C DOUBLE PRECISION HNEW
C
C DIMENSION HNEW(NCOL,NROW,NLAY),IOFLG(NLAY,4),
1 BUFF(NCOL,NROW,NLAY)
CF66
C DIMENSION TEXT(4)
CF66
C77
CHARACTER*4 TEXT(4)
C77
DATA TEXT(1),TEXT(2),TEXT(3),TEXT(4) /'','','','
1 'HEAD'/
FOR EACH LAYER: PRINT HEAD IF REQUESTED.
DO 39 K=1,NLAY

TEST IOFLG TO SEE IF HEAD SHOULD BE PRINTED.
IF(IOFLG(K,1).EQ.0) GO TO 39
IPFLG=1

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

CALL UTILITY MODULE TO PRINT CONTENTS OF BUFFER.
IF(IHEDFM.LT.0) CALL ULAPRS(BUFF,TEXT(1),KSTP,KPER,NCOL,NROW,K,
1 1IHEDFM,IOUT)
IF(IHEDFM.GE.0) CALL ULAPRW(BUFF,TEXT(1),KSTP,KPER,NCOL,NROW,K,
1 IHEDFM,IOUT)
39 CONTINUE

IF UNIT FOR RECORDING HEADS <= 0: THEN RETURN.
IF(IHEDUN.LE.0)GO TO 50
IFIRST=1

FOR EACH LAYER: RECORD HEAD IF REQUESTED.
DO 49 K=1,NLAY

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(IHO,'HEAD WILL BE SAVED ON UNIT',13,' AT END OF TIME STEP',
1 13,', STRESS PERIOD',13)
IFIRST=0

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

RECORD CONTENTS OF BUFFER ON UNIT=IHEDUN
CALL ULASAV(BUFF,TEXT(1),KSTP,KPER,PERTIM,TOTIM,NCOL,NROW,K,
1 1IHEDUN)
49 CONTINUE

RETURN
50 RETURN
END

SUBROUTINE SBAS1I(NLAY,ISTRT,IOFLG,INOC,IOUT,IHEDFM, 
1 IDDNM, IHEDUN, IDDUN)

VERSION 1138 03NOV1982 SBAS1I
C SET UP OUTPUT CONTROL

SPECIFICATIONS

DIMENSION IOFLG(NLAY,4)

C1-----TEST UNIT NUMBER FROM IUNIT (INOC) TO SEE IF OUTPUT
C1-----CONTROL IS ACTIVE.
IF(INOC.LE.0) GO TO 600

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 (4110)
  WRITE (IOUT,3)IHEDFM,IDDNFM
  3 FORMAT (IHO,'HEAD PRINT FORMAT IS FORMAT NUMBER',14,
    1 ' DRAWDOWN PRINT FORMAT IS FORMAT NUMBER',14)
  WRITE (IOUT,4)IHEDUN,IDDNUN
  4 FORMAT (IHO,'HEADS WILL BE SAVED ON UNIT',13,
    1 ' DRAWDOWNS WILL BE SAVED ON UNIT',13)
  WRITE(IOUT,561)
  561 FORMAT(IH0,'OUTPUT CONTROL IS SPECIFIED EVERY TIME STEP')
  GO TO 1000

C3-----OUTPUT CONTROL IS INACTIVE. PRINT A MESSAGE LISTING DEFAULTS.
600 WRITE(IOUT,641)
  641 FORMAT(1HO,'DEFAULT OUTPUT CONTROL -- THE FOLLOWING OUTPUT',
    1 ' COMES AT THE END OF EACH STRESS PERIOD:')
  WRITE(IOUT,642)
  642 FORMAT(IX,'TOTAL VOLUMETRIC BUDGET')
  WRITE(IOUT,643)
  643 FORMAT(1X,10X,'HEAD')
  IF(ISTRT.NE.0)WRITE(IOUT,644)
  644 FORMAT(1X,10X,'DRAWDOWN')

C4-----SET THE FORMAT CODES EQUAL TO THE DEFAULT FORMAT.
  IHEDFM=0
  IDDNFM=0

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
C6------RETURN
1000 RETURN
END

SUBROUTINE SEAS1T(KSTP,KPER,DELT,PERTIM,TOTIM,ITMUNI,IOUT)

C

C

C------VERSION 0837 09APR1982 SEAS1T

C *************************************************************
C PRINT SIMULATION TIME
C *************************************************************
C
C SPECIFICATIONS:
C
WRITE(IOUT,199) KSTP,KPER
199 FORMAT(IHO,///1OX,'TIME SUMMARY AT END OF TIME STEP',13,
1 ' IN STRESS PERIOD',13)

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.

Engineering Technologies Associates, Inc.
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', ' 1OX, ' 1DAYS', ' 75(', ' 1X', ' 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 SBASIV(MSUM, VBNM, VBVL, KSTP, KPER, IOUT)

C
C
C6
C7

C
C
C
C
C
C------VERSION 1153 03NOV1982 SBASIV
C
C    ************************************************************************
C
C    PRINT VOLUMETRIC BUDGET
C
C    ************************************************************************

DIMENSION VBVL(4,20)

CHARACTER*4 VBNM(4,20)

C
C
C
C------DETERMINE NUMBER OF INDIVIDUAL BUDGET ENTRIES.
   MSUM1 = MSUM - 1
   IF (MSUM1 LE 0) RETURN
C
C
C2------CLEAR RATE AND VOLUME ACCUMULATORS.
   TOTRIN = 0.
   TOTROT = 0.
   TOTVIN = 0.
   TOTVOT = 0.
C
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
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(1HO,///30X,'VOLUMETRIC BUDGET FOR ENTIRE MODEL AT END OF'
     1,' TIME STEP',I3,' IN STRESS PERIOD',I3//30X,77('(''))
265 FORMAT(1HO,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,44X,'=',G14.5,44X,'=',G14.5)
286 FORMAT(1HO,26X,'TOTAL IN '='G14.5,'TOTAL IN '='
     1,G14.5)
287 FORMAT(1HO,24X,'OUT:','67X,'OUT:','25X,4('('')',67X,4('('')'
298 FORMAT(1HO,25X,'TOTAL OUT '='G14.5,'TOTAL OUT '='
     1,G14.5)
299 FORMAT(1HO,26X,'IN - OUT =',G14.5,47X,'IN - OUT =',G14.5)
300 FORMAT(1HO,15X,'PERCENT DISCREPANCY =',F20.2
1,30X,'PERCENT DISCREPANCY =',F20.2,///)
C
END
SUBROUTINE WELTL(ISUM,LENX,LCELL,MNWELL,NWELL,NWELLS,IN,IOUT,
          IVELCB)

C------VERSION 0533 OBEC1983 WELTL
C***************************************************************
C ALLOCATE ARRAY STORAGE FOR WELL PACKAGE
C***************************************************************
C
C SPECIFICATIONS:
C
INTEGER*4 LENX
C
C1------IDENTIFY PACKAGE AND INITIALIZE NWELLS
WRITE(IOUT,1)IN
1 FORMAT(1HO,'WELL1 -- WELL PACKAGE, VERSION 1, 12/08/83,'
     12 INPUT READ FROM',13)
NWELLS=0
C
C2------READ MAX NUMBER OF WELLS AND
C2------UNIT OR FLAG FOR CELL-BY-CELL FLOW TERMS.
READ(IN,2) MXWELL,IVELCB
2 FORMAT(21I0)
WRITE(IOUT,3) MXWELL
3 FORMAT(1H,'MAXIMUM OF',15,' WELLS')
IF(IVELCB.GT.0) WRITE(IOUT,9) IVELCB
9 FORMAT(1X,'CELL-BY-CELL FLOWS WILL BE RECORDED ON UNIT',13)
IF(IVELCB.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,16,' ELEMENTS IN X ARRAY ARE USED FOR WELLS')
ISUM=ISUM-1
WRITE(IOUT,5) ISUM,LENX
5 FORMAT(1X,16,' ELEMENTS OF X ARRAY USED OUT OF',17)
C
C6------IF THERE ISN'T ENOUGH SPACE IN THE X ARRAY THEN PRINT
C6------A WARNING MESSAGE.
IF(ISUM.GT.LENX) WRITE(IOUT,6)
6 FORMAT(1X,' ***X ARRAY MUST BE DIMENSIONED LARGER***')
C7------RETURN
RETURN
END
SUBROUTINE WELRP(WELL,NWELLS,MXWELL,IN,IOUT)
C
C-----VERSION 1544 22DEC1982 WELRP
C
C READ NEW WELL LOCATIONS AND STRESS RATES
C
C SPECIFICATIONS:
C
DIMENSION WELL(4,MXWELL)
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
CIA-----IF ITMP LESS THAN ZERO REUSE DATA. PRINT MESSAGE AND RETURN.
WRITE(IOUT,6)
6 FORMAT(1HO,'REUSING WELLS FROM LAST STRESS PERIOD')
RETURN
C
CIB-----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(IHO,'NWELLS(',14,') IS GREATER THAN MXWELL(',14,')')
STOP
C
C3-------PRINT NUMBER OF WELLS IN CURRENT STRESS PERIOD.
100 WRITE (IOUT,2) NWELLS
2 FORMAT(1HO,10X,14,' 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(1HO,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(3110,F10.0)
WRITE (IOUT,5) K,I,J,0,II
5 FORMAT(48X,13,I8,17,G16.5,I8)
WELL(1,II)=K
WELL(2,II)=I
WELL(3,II)=J
WELL(4,II)=Q
250 CONTINUE

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 MODIFIED FROM
C ----- USGS 3D MODFLOW MODEL, VERSION 1001 26AUG1982 WELIFM
C
C ADD WELLS TO LIST OF SINKS
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,105)X,Y,Z,-Q*CONVRT
105 FORMAT(3F12.1,G12.4)
100 CONTINUE
C
C3------RETURN
RETURN
END
SUBROUTINE EVTIAL(ISUM, LENX, LCIEVT, LCEVTR, LCEXDP, LCSURF, NCOL, NROW, NEVTOP, IN, IOUT, IEVTCB)

C
C------VERSION 0943 0EDEC1983 EVTIAL
C
C ALLOCATE ARRAY STORAGE FOR EVAPOTRANSPIRATION
C
C SPECIFICATIONS:
C
INTEGER*4 LENX

C
C1------IDENTIFY PACKAGE.
WRITE(IOUT,1)IN
1 FORMAT(1HO,'EVTIAL-- EVAPOTRANSPIRATION PACKAGE, VERSION ',
1 '12/08/83',' INPUT READ FROM UNIT',13)

C
C2------READ NEVTOP AND IEVTCB.
READ(IN,3)NEVTOP,IEVTCB
3 FORMAT(2110)

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(IX,'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(IX,'OPTION 1 -- EVAPOTRANSPIRATION FROM TOP LAYER')
IF(NEVTOP.EQ.2) WRITE(IOUT,202)
202 FORMAT(IX,'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(IX,'CELL-BY-CELL FLOW TERMS WILL BE SAVED ON UNIT',13)

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

B-65

Engineering Technologies Associates, Inc.
LCIEVT = ISUM
ISUM = ISUM - NCOL * NROW

C
C----CALCULATE & PRINT AMOUNT OF SPACE USED BY ET PACKAGE.
300 IRK = ISUM - IRK
WRITE(IOUT, 4) IRK
4 FORMAT(1X, 16, 'ELEMENTS OF X ARRAY USED FOR EVAPOTRANSPIRATION')
ISUM = ISUM - 1
WRITE(IOUT, 5) ISUM1, LENX
5 FORMAT(1X, 'ELEMENTS OF X ARRAY USED OUT OF', 17)
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 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', 'INDEX'/
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, IENEVT, IEXDP, IIEVT
6 FORMAT(4110)
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(1HO,'REUSING SURF FROM LAST STRESS PERIOD')
   GO TO 35
C
C3------IF INSURF=>O THEN CALL MODULE U2DREL TO READ SURFACE.
   32 CALL U2DREL(SURF,ANAME(1,2),NROW,NCOL,O,IN,IOUT)
C
C4------TEST INEVTR TO SEE WHERE MAX ET RATE COMES FROM.
   35 IF(INEVTR.GE.O)GO TO 37
C
C4A------IF INEVTR<0 THEN REUSE MAX ET RATE.
   WRITE(IOUT,4)
   4 FORMAT(1HO,'REUSING EVTR FROM LAST STRESS PERIOD')
   GO TO 45
C
C5------IF INEVTR=>O CALL MODULE U2DREL TO READ MAX ET RATE.
   37 CALL U2DREL(EVTR,ANAME(1,3),NROW,NCOL,O,IN,IOUT)
C
C6------MULTIPLY MAX ET RATE BY CELL AREA TO GET VOLUMETRIC RATE
   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.O)GO TO 47
C
C7A------IF INEXDP<0 REUSE EXTINCTION DEPTH FROM LAST STRESS PERIOD
   WRITE(IOUT,5)
   5 FORMAT(1HO,'REUSING EXDP FROM LAST STRESS PERIOD')
   GO TO 48
C
C8------IF INEXDP=>O CALL MODULE U2DREL TO READ EXTINCTION DEPTH
   47 CALL U2DREL(EXDP,ANAME(1,4),NROW,NCOL,O,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.O)GO TO 49
       WRITE(IOUT,2)
       2 FORMAT(1HO,'REUSING IEVT FROM LAST STRESS PERIOD')
       GO TO 50
C
C11------IF INIEVT=>O THEN CALL MODULE U2DINT TO READ INDICATOR ARRAY.
   49 CALL U2DINT(IEVT,ANAME(I,I),NROW,NCOL,O,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 ADD EVAPOTRANSPIRATION TO RHS AND HCOF

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

C1------PROCESS EACH HORIZONTAL CELL LOCATION

DO 10 IR=1,NROW
  DO 10 IC=1,NCOL

  C2------SET THE LAYER INDEX EQUAL TO 1
  IL=1
  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(BOUND(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)

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

  D=S-H
  X=EXDP(IC,IR)
  IF(D.GE.X)GO TO 10

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
SUBROUTINE GHB1AL(ISUM, LENX, LCBNDS, NBOUND, MXBND, IN, IOUT,  
1  IGHBCB)

C-----VERSION 0940 08DEC1983 GHB1AL
C ****************************************************
C ALLOCATE ARRAY STORAGE FOR HEAD-DEPENDENT BOUNDARIES
C *****************************************************
C SPECIFICATIONS:
C -------------------------------------------------------
INTEGER*4 LENX
C -------------------------------------------------------

C1-------IDENTIFY PACKAGE AND INITIALIZE # OF GENERAL HEAD BOUNDS
WRITE(IOUT,1)IN
1 FORMAT(IH0,'GHBI -- GHB PACKAGE, VERSION 1, 12/08/83',  
2 ' INPUT READ FROM UNIT',13)
NBOUND=0

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(2110)
WRITE(IOUT,3) MXBND
3 FORMAT(IX,'MAXIMUM OF',15,' HEAD-DEPENDENT BOUNDARY NODES')
IF(IGHBCB.GT.O) WRITE(IOUT,9) IGHBCB
9 FORMAT(IX,'CELL-BY-CELL FLOW WILL BE RECORDED ON UNIT',13)
IF(IGHBCB.LT.O) WRITE(IOUT,8)
8 FORMAT(IX,'CELL-BY-CELL FLOW WILL BE PRINTED WHEN ICBCFL NOT 0')

C3-------SET LCBNDS EQUAL TO ADDRESS OF FIRST UNUSED SPACE IN X.
LCBNDS=ISUM

C4-------CALCULATE AMOUNT OF SPACE USED BY THE GENERAL HEAD LIST.
ISP=5*MXBND
ISUM=ISUM+ISP

C5-------PRINT AMOUNT OF SPACE USED BY THE GHB PACKAGE
WRITE(IOUT,4) ISP
4 FORMAT(1X,'ELEMENTS IN X ARRAY ARE USED FOR HEAD',  
1 'DEPENDENT BOUNDARIES')
IF(ISUM1.GT.LENX) WRITE(IOUT,6)
6 FORMAT(1X,'***X ARRAY MUST BE DIMENSIONED LARGER***')

C6-------RETURN
RETURN
END
SUBROUTINE GHB1RP(BNDS, NBOUND, MXBND, IN, IOUT)

Engineering Technologies Associates, Inc.
C
C-----VERSION 1651 02FEB1983 GHB1RP
C
C*******************************************************************
C
C SPECIFICATIONS:
C
DIMENSION BNDS(5,MXBND)
C
C
C1------READ ITMP(# OF GENERAL HEAD BOUNDS OR FLAG TO REUSE DATA.)
C   READ(IN,8) ITMP
C
C2------TEST ITMP
C   IF(ITMP.GE.0) GO TO 50
C
C2A------IF ITMP<0 THEN REUSE DATA FROM LAST STRESS PERIOD
C   WRITE(IOUT,7)
C   7 FORMAT(1HO,'REUSING HEAD-DEPENDENT BOUNDS FROM LAST STRESS',
C     ' PERIOD')
C   GO TO 260
C
C3------IF ITMP=>0 THEN IT IS THE # OF GENERAL HEAD BOUNDS.
C   50 NBOUND=ITMP
C
C4------IF MAX NUMBER OF BOUNDS IS EXCEEDED THEN STOP
C   IF(NBOUND.LE.MXBND) GO TO 100
C   WRITE(IOUT,99) NBOUND,MXBND
C   99 FORMAT(1HO,'NBOUND(',I4,') IS GREATER THAN MXBND(',14,')')
C
C4A------ABNORMAL STOP
C   STOP
C
C5------PRINT # OF GENERAL HEAD BOUNDS THIS STRESS PERIOD
C   100 WRITE(IOUT,1) NBOUND
C     1 FORMAT(1HO,//1X,15,' HEAD-DEPENDENT BOUNDARY NODES')
C
C6------IF THERE ARE NO GENERAL HEAD BOUNDS THEN RETURN.
C   IF(NBOUND.EQ.0) GO TO 260
C
C7------READ & PRINT DATA FOR EACH GENERAL HEAD BOUNDARY.
C   WRITE(IOUT,3)
C     3 FORMAT(1HO,15X,'LAYER',5X,'ROW',5X
C     'ELEVATION CONDUCTANCE BOUND NO.'/1X,15X,60(''-'))
C     DO 250 II=1,NBOUND
C     READ (IN,4) K,I,J,BNDS(4,II),BNDS(5,II)
C     4 FORMAT(3110,2F10.0)
C     WRITE (IOUT,5) K,I,J,BNDS(4,II),BNDS(5,II),II
C     5 FORMAT(1X,15X,14,19,18,613.4,G14.4,G14.4,18)
C     BNDS(1,II)=<
C     BNDS(2,II)=I
C     BNDS(3,II)=J
C
C
CREATE SINKS AT GENERAL HEAD BOUNDARIES MODIFIED FROM USGS 3D

**Version 1605 02FEB1983 GHB1FM**

ADD GHB BOUNDARIES TO SINK LIST

**Specifications:**

- DOUBLE PRECISION HNEW
- DIMENSION BNDS(5,MXBND),HNEW(NCOL,NROW,NLAY),
  
- DELR(NCOL),DELC(NROW),IBOUND(NCOL,NROW,NLAY)

**1.** IF NBOUND<=0 THEN THERE ARE NO GENERAL HEAD BOUNDS. RETURN.

**2.** PROCESS EACH ENTRY IN THE GENERAL HEAD BOUND LIST (BNDS)

DO 100 L=1,NBOUND

**3.** GET COLUMN, ROW AND LAYER OF CELL CONTAINING BOUNDARY

IL=BNDS(1,L)
IR=BNDS(2,L)
IC=BNDS(3,L)

**4.** 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

**5.** SINCE THE CELL IS INTERNAL GET THE BOUNDARY DATA.

HB=BNDS(4,L)
C=BNDS(5,L)

**6.** 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. 0) WRITE(6,1005)X,Y,Z,-RATE*CONVRT

1005 FORMAT(3F12.1,G12.4)

100 CONTINUE

**7.** RETURN

RETURN

END
SUBROUTINE RIV1AL(ISUM, LENX, LCRIVR, MRIVR, NRIVER, IN, IOUT, 1 IRIVCB)
C
C------VERSION 0935 08DEC1983 RIV1AL
C
C ALLOCATE ARRAY STORAGE FOR RIVERS
C
C SPECIFICATIONS:
C
INTEGER*4 LENX
C
C IDENTIFY PACKAGE AND INITIALIZE NRIVER.
WRITE(IOUT,1)IN
1 FORMAT(1HO,'RIV1 AL RIVER PACKAGE, VERSION 1, 12/08/83',  12/08/83',
   2' INPUT READ FROM UNIT',I3)
NRIVER=0
C
C 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(IH ,'MAXIMUM OF',I5,' RIVER NODES')
IF(IRIVCB.GT.O) WRITE(IOUT,9) IRIVCB
9 FORMAT(IX,'CELL-BY-CELL FLOWS WILL BE RECORDED ON UNIT',I3)
IF(IRIVCB.LT.O) WRITE(IOUT,8)
8 FORMAT(IX,'CELL-BY-CELL FLOWS WILL BE PRINTED')
C
C SET LCRIVR EQUAL TO ADDRESS OF FIRST UNUSED SPACE IN X.
LCRIVR=ISUM
C
C CALCULATE AMOUNT OF SPACE USED BY RIVER LIST.
ISP=6*MXRIVR
ISUM=ISUM+ISP
C
C PRINT AMOUNT OF SPACE USED BY RIVER PACKAGE.
WRITE (IOUT,4)ISP
4 FORMAT(1X,16,' ELEMENTS IN X ARRAY ARE USED FOR RIVERS')
ISUM1=ISUM-1
WRITE(IOUT,5)ISUM1,LENX
5 FORMAT(1X,16,' ELEMENTS OF X ARRAY USED OUT OF',17)
IF(ISUM1.GT.LENX) WRITE(IOUT,6)
6 FORMAT(1X,'***X ARRAY MUST BE DIMENSIONED LARGER***')
C
RETURN
RETURN
END
SUBROUTINE RIV1RP(RIVR, NRIVER, MRIVR, IN, IOUT)
C-----VERSION 1319 25AUG1982 RIV1RP
C
******************************************************************************
C READ RIVER HEAD, CONDUCTANCE AND BOTTOM ELEVATION
C******************************************************************************
C
C SPECIFICATIONS:
C
DIMENSION RIVR(6,MXRIVR)

C
C1-----READ ITMP(NUMBER OF RIVER REACHES OR FLAG TO REUSE DATA)
READ(IN,B)ITMP
B 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(IHO,'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(IHO,'NRIVER(',I4,') IS GREATER THAN MXRIVR(',14,')')
C
C4A-----ABNORMAL STOP.
   STOP
C
C5-----PRINT NUMBER OF RIVER REACHES IN THIS STRESS PERIOD.
100 WRITE(IOUT,1)NRIVER
1 FORMAT(IHO,//IX,15,' 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(IHO,15X,'LAYER',5X,'ROW',5X,'COL ',1,' STAGE CONDUCTANCE BOTTOM ELEVATION RIVER REACH'
2/1X,15X,80('-'))
   DO 250 II1=1,NRIVER
   READ(IN,4)I,J,RIVR(4,11),RIVR(5,II),RIVR(6,II)
4 FORMAT(3110,3F10.0)
   WRITE(IOUT,5)I,J,RIVR(4,II),RIVR(5,II),RIVR(6,II)
5 FORMAT(1X,15X,'14,18,613.4,614.4,619.4,110)
   RIVR(1,II)=K
   RIVR(2,II)=I
   RIVR(3,II)=J

B-73  Engineering Technologies Associates, Inc.
C ROUTINE TO ADD RIVER NODES TO SINK LIST MODIFIED FROM USGS 3D
C----VERSION 0915 27AUG1982 RIV1FM
C
C SPECIFICATIONS:
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: IF NRIVER<=O THERE ARE NO RIVERS. RETURN.
  IF(NRIVER.LE.0)RETURN
C
C PROCESS EACH CELL IN THE RIVER LIST.
DO 100 L=1,NRIVER
  C
C GET COLUMN, ROW, AND LAYER OF CELL CONTAINING REACH
  IL=RIVR(1,L)
  IR=RIVR(2,L)
  IC=RIVR(3,L)
C
C IF THE CELL IS EXTERNAL SKIP IT.
  IF(BOUND(IL,IR,IC).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
C 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
C COMPARE AQUIFER HEAD TO BOTTOM OF STREAM BED.
  IF(HHNEW.LE.RBOT)GO TO 96
C
C SINCE HEAD>BOTTOM CREATE SINK.
  RATE=-CRIV*(HHNEW-HRIV)
  GO TO 98
SUBROUTINE PRERIV(NRIVER, MXRIVR, RIVR, HNEW, IBOUND, RBOT,  
  DELR, DELC, ZVEL, NCOL, NROW, NLAY, IVOUT, CONVRT)

C
C ROUTINE TO COMPUTE VELOCITY OF WATER TABLE IN RIVER NODES
C MODIFIED FROM USGS 3D
C
C COMPUTE WATER TABLE VELOCITY UNDER RIVERS AND FILL RBOT ARRAY
C
C SPECIFICATIONS:
C
DOUBLE PRECISION HNEW
DIMENSION RIVR(6,MXRIVR),HNEW(NCOL,NROW,NLAY),  
  DELR(NCOL),DELC(NROW),ZVEL(NCOL,NROW),  
  IBOUND(NCOL,NROW,NLAY),RBOT(NCOL,NROW)
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(BOUND(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-HRV)
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)/DELIC(IR)
100 CONTINUE

C
C9-----RETURN
RETURN
END
SUBROUTINE RCH1AL(ISUM, LENX, LCIIRCH, LCRECH, NRCHOP, NCOL, NROW, IN, IOUT, IRCHCB)

C
C
C ALLOCATE ARRAY STORAGE FOR RECHARGE
C
C
C SPECIFICATIONS:
C
C
C
C
C IDENTIFY PACKAGE.
WRITE(IOUT,1)IN
1 FORMAT(1HO,'RCH1 -- RECHARGE PACKAGE, VERSION 1, 12/08/83',
2' INPUT READ FROM UNIT',f3)
C
C READ NRCHOP AND IRCHCB.
READ(IN,2)NRCHOP, IRCHCB
2 FORMAT(2110)
C
C CHECK TO SEE THAT OPTION IS LEGAL.
IF(NRCHOP.GE.1.AND.NRCHOP.LE.3)GO TO 200
C
C IF ILLEGAL PRINT A MESSAGE AND ABORT SIMULATION
WRITE(IOUT,8)
8 FORMAT(IX,'ILLEGAL OPTION CODE. SIMULATION ABORTING')
STOP
C
C 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
C 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',f3)
C
C ALLOCATE SPACE FOR THE RECHARGE ARRAY(RECH).
LCRECH=ISUM
ISUM=ISUM+NCOL*NROW
C
C IF OPTION 2 THEN ALLOCATE SPACE FOR INDICATOR ARRAY(IRCH)
IF(NRCHOP.NE.2)GO TO 300
LCIRCH=ISUM
ISUM=ISUM+NCOL*NROW 

C8-----CALCULATE AND PRINT AMOUNT OF SPACE USED BY RECHARGE.
300 IRK=ISUM-IRK
   WRITE(IOUT,4)IRK
4 FORMAT(1X,16,' ELEMENTS OF X ARRAY USED FOR RECHARGE')
   ISUM=ISUM-1
   WRITE(IOUT,5)ISUM,LENX
5 FORMAT(1X,16,' ELEMENTS OF X ARRAY USED OUT OF ',17)
   IF(ISUM.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
C9-----VERSION 1513 22DEC1982 RCH1RP
C
C READ RECHARGE RATES
C
C SPECIFICATIONS:
C
C DIMENSION IRCH(NROW),RECH(NCOL),DELR(NROW),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
C1-----READ FLAGS SHOWING WHETHER DATA IS TO BE REUSED.
   READ(IN,4)INRECH,INIRCH
4 FORMAT(2110)

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,NRCW
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(1HO,'REUSING IRCH FROM LAST STRESS PERIOD')
   GO TO 60
C
C7-------IF INIRCH=>O CALL U2DINT TO READ LAYER IND ARRAY(IRCH)
   CALL U2DINT(IRCH,ANAME(1,1),NRCW,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 RCHIFM
C
C ADD RECHARGE TO RHS
C
C SPECIFICATIONS:
DIMENSION IRCH(NCOL,NROW),RECH(NCOL,NROW),
1 ZVEL(NCOL,NROW),IBOUND(NCOL,NROW,NLAY)
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)
   WRITE(*,1001)IC,IR,ZVEL(IC,IR)
   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=RCH(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
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 2S
C
C3C------SUBTRACT RECHARGE FROM RIGHT-HAND-SIDE.
      ZVEL(IC,IR)=-RECH(IC,IR)
   28 CONTINUE
   30 CONTINUE
   100 CONTINUE
C
C4------RETURN
      RETURN
      END
SUBROUTINE DRAIN(LENX,LCDRAI,NDRAIN,MXDRN,IN,IOUT)

C-----VERSION 0956 08DEC1983 DRAIN

C ALLOCATE ARRAY STORAGE FOR DRAIN PACKAGE

C SPECIFICATIONS:

INTEGER*4 LENX

C IDENTIFY PACKAGE AND INITIALIZE NDRAIN.

WRITE(IOUT,1)IN

1 FORMAT(1H0,' DRAIN -- DRAIN PACKAGE, VERSION 1, 12/08/83',
2' INPUT READ FROM UNIT',13)

NDRAIN=0

C 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(IH,'MAXIMUM OF',15,' DRAINS')

IF(IDRNCB.GT.0)WRITE(IOUT,9)IDRNCB

9 FORMAT(IH,'CELL-BY-CELL FLOWS WILL BE RECORDED ON UNIT',13)

IF(IDRNCB.LT.0)WRITE(IOUT,8)

8 FORMAT(IH,'CELL-BY-CELL FLOWS WILL BE PRINTED WHEN ICBCFL NOT 0')

C SET LCDRAI EQUAL TO ADDRESS OF FIRST UNUSED SPACE IN X.

LCDRAI=ISUM

C CALCULATE AMOUNT OF SPACE USED BY THE DRAIN PACKAGE.

ISP=5*MXDRN

ISUM=ISUM+ISP

C PRINT AMOUNT OF SPACE USED BY DRAIN PACKAGE.

WRITE(IOUT,4)ISP

4 FORMAT(1X,16,' ELEMENTS IN X ARRAY ARE USED FOR DRAINS')

ISUM=ISUM-1

WRITE(IOUT,5)ISUM,LENX

5 FORMAT(1X,16,' ELEMENTS OF X ARRAY USED OUT OF',17)

IF(ISUM1.GT.LENX)WRITE(IOUT,6)

6 FORMAT(1X,'***X ARRAY MUST BE DIMENSIONED LARGER***')

C RETURN

RETURN

END

SUBROUTINE DRN1RP(DRAI,NDRAIN,MXDRN,IN,IOUT)
READ DRAIN LOCATIONS, ELEVATIONS, AND CONDUCTANCES

SPECIFICATIONS:

DIMENSION DRAI(5,MXDRN)

C1-----READ ITMP(NUMBER OF DRAIN CELLS OR FLAG TO REUSE DATA)
READ(IN,8) ITMP
8 FORMAT(I10)

C2------TEST ITMP
IF(ITMP.LE.0) GO TO 50

C2A-----IF ITMP<0 THEN REUSE DATA FROM LAST STRESS PERIOD.
WRITE(IOUT,7)
7 FORMAT(1HO,'REUSING DRAINS FROM LAST STRESS PERIOD')
RETURN

C3-------IF ITMP=>0 THEN IT IS THE NUMBER OF DRAINS.
50 NDRAIN=ITMP
IF(NDRAIN.LE.MXDRN) GO TO 100

C4------IF NDRAIN>MXDRN THEN STOP
WRITE(IOUT,99) NDRAIN,MXDRN
99 FORMAT(1HO,'NDRAIN(',I4,') IS GREATER THAN MXDRN(',I4,')')
STOP

C5------PRINT NUMBER OF DRAINS IN THIS STRESS PERIOD.
100 WRITE(IOUT,1) NDRAIN
1 FORMAT(1HO,'//1X,15,' DRAINS')

C6-------IF THERE ARE NO DRAINS THEN RETURN.
IF(NDRAIN.EQ.0) GO TO 260

C7-------READ AND PRINT DATA FOR EACH DRAIN.
WRITE(IOUT,3)
3 FORMAT(1HO,15X,'LAYER',5X,'ROW',5X,'COL',5X,'ELEVATION',15X,60(''-'))
NOW I=1,NDRAIN
DO 250 II=1,NDRAIN
READ (IN,4) K,I,J,DRAI(4,II),DRAI(5,II)
4 FORMAT(310,2F10.0)
WRITE (IOUT,5) K,I,J,DRAI(4,II),DRAI(5,II)
5 FORMAT(1X,15X,4X,18,G13.4,G14.4,18)
250 CONTINUE

C8------RETURN
260 RETURN
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
C -----VERSION 1638 25APR1983 DRN1FM
C
C ADD DRAINS TO SINKS
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
C1-------IF NDRAIN<=O THERE ARE NO DRAINS. RETURN
IF(NDRAIN.LE.O) 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(BOUND(IC,IR,IL).LE.O) 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 HOOF.
C=DRAI(5,L)
RATE=-C*(HHNEW-EL)
CALL XYZPOS(IC,IR,IL,X,Y,Z,DELR,DELC,NROW,NCOL,NLAY)
IF(RATE .LT. O)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:

- 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 into 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 PRND3D model solves the solute transport equation in three dimensions. One form of the governing equation for solute transport in one dimension is

\[
\frac{\partial}{\partial x} \left( \frac{D}{R_d} \frac{\partial C}{\partial x} \right) - \frac{v}{R_d} \frac{\partial C}{\partial x} + C_s Q = \frac{\partial C}{\partial t}
\]

(1)

dispersion - advection + sources/sinks = concentration

where

\[
\begin{align*}
V &= \text{interstitial (seepage) velocity} \\
D &= \text{coefficient of hydrodynamic dispersion} \\
D &= dxV, \text{ where } dx = \text{longitudinal dispersivity} \\
X &= \text{space dimension} \\
R_d &= \text{retardation factor} \\
C_s Q &= \text{source or sink function having a concentration } C_s \\
C &= \text{concentration} \\
t &= \text{time}
\end{align*}
\]

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

The two other commonly used algorithms in groundwater 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.
Basic Concept of Random Walk
(from Prickett, Naymik, and Lonnquist, 1981)

Figure 1

Normal Distribution Curves for Dispersion

Mean Flow

TRANSVERSE DISPERSION

Particle

Porous Medium

Longitudinal Dispersion

Mean Flow

Convective Component

Random Component
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 \( C = 0 \) equal to zero, equation 1 describes the concentration of the slug as it moves downstream. Bear (1972) describes the solution as

\[
C(x,t) = \left[ \frac{1}{(4\pi d_L vt)^{0.5}} \right] \exp\left[ -\frac{(x-vt)^2}{4d_L vt} \right] \tag{2}
\]

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

\[
n(x) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left[ -\frac{(x-u)^2}{2\sigma^2} \right] \tag{3}
\]

where

- \( \sigma \) = standard deviation of the distribution
- \( u \) = mean of the distribution

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

\[
\sigma = \sqrt{2d_L vt} \tag{4}
\]

\[
u = vt \tag{5}
\]

\[
n(x) = C(x,t) \tag{6}
\]

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)
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.

Example new particle position

\[ \overline{\sigma_L} \] = \sqrt{2d_L \Delta x} \]

Old position, \( xx, yy \)

\[ \text{CONVECT} \]

\[ \text{DISPERSE} \]

\[ \Delta x = V_x \Delta \text{ELP} \]

\[ V_x = \Delta \text{ELP} \]

Where:

\[ \Delta \text{L} \Delta x = 2 \Delta \text{L} \Delta x \times \text{ANORM} (0) \]

New position = Old position + Convection + Dispersion

\[ xx = xx + \Delta x + \Delta \text{L} \Delta x \]

\[ yy = yy \]

(B) TRANSVERSE DISPERSION

Example new particle position \((xx, yy)\)

\[ \overline{\sigma_T} \] = \sqrt{2d_T \Delta x} \]

Old position, \( xx, yy \)

\[ \text{DISPERSE} \]

\[ \text{CONVECT} \]

\[ \Delta x = V_x \Delta \text{ELP} \]

Where:

\[ \Delta \text{T} \Delta x = 2 \Delta \text{T} \Delta x \times \text{ANORM} (0) \]

New position = Old position + Convection + Dispersion

\[ xx = xx + \Delta x + 0 \]

\[ yy = yy + 0 - \Delta \text{T} \Delta x \]
where
\[ \sqrt{2dL} = \sigma = \sqrt{2dL} vt \]
\( v = \) velocity in \( x \) direction
\( DX = vt = \) advective move distance
\( t = \) time step
\( \text{ANORM}(0) = \) 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
\[ (\sqrt{2dL}DX)\text{ANORM}(0) \]

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
\[ C(x,t) - n(x) - N/dx = N_0/(\sqrt{4\pi dL}DX) \exp\left[-(x-\mu)^2/4dLt\right] \]
(8)

where
\( N_0 = \) 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 \( \text{ANORM}(0) \). The function \( \text{ANORM}(0) \) 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, Noykin, and Lonnquist, 1981)

(A) LONGITUDINAL DISPERSION

\[ d_L = 0 \quad d_T > 0 \]

\[ dx = V_x \cdot DLP \]

\[ dy = V_y \cdot DLP \]

\[ DD = \sqrt{dx^2 + dy^2} \]

\[ RT \cdot DD = \sqrt{2d_L \cdot DD} \quad ANORM (0) \]

New position = Old position + Convection + Dispersion

\[ xx = xx + DX + RL \cdot DX \]

\[ yy = yy + DY + RL \cdot DY \]

TRANSVERSE DISPERSION

\[ d_L = 0 \quad d_T > 0 \]

\[ dx = V_x \cdot DLP \]

\[ dy = V_y \cdot DLP \]

\[ DD = \sqrt{dx^2 + dy^2} \]

\[ RT \cdot DD = \sqrt{2d_T \cdot DD} \quad ANORM (0) \]

New position = Old position + Convection + Dispersion

\[ xx = xx + DX - RT \cdot DX \]

\[ yy = yy + DY - RT \cdot DY \]
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

$$\frac{v}{v_c} = 1 + \frac{K_d p}{n} = R_d$$ (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:

$$\frac{dc}{dt} = K_c$$
Figure 5

General scheme for convection and dispersion

(from Frickett, Neymik, and Lonnquist, 1981)

Longitudinal and transverse dispersion

\[
\text{New position} = \text{Old position} + \text{Convection} + \text{Longitudinal Dispersion} + \text{Transverse Dispersion}
\]

\[
xx = xx + DX + RL*DX + RT*DY
\]

\[
yy = yy + DY + RL*DY - RT*DX
\]

\[
RL*DD = \sqrt{2\sigma_{DD}^2 \text{ANORM}(0)}
\]

\[
RT*DD = \sqrt{2\sigma_{DD}^2 \text{ANORM}(0)}
\]
Figure 6

Mixing and dilution effects in water quality problems.

(from Prickett, Naymik, and Lonnquist, 1981)

(A) PROBLEM

\[ Q = 200,000 \text{ gpd} \]
\[ C = ? \text{ at concentration} \]
\[ V = ? \]

Relative velocity

(B) SOLUTION

\[ Q = 200,000 \text{ gpd} \]
\[ C = 300 \text{ mg/l} \]
\[ V = 1 \text{ units} \]
\[ Q = 100,000 \text{ gpd} \]
\[ C = 200 \text{ mg/l} \]
\[ V = 5 \text{ units} \]

\[ Q = 300,000 \text{ gpd} \]
\[ C = 200 \text{ mg/l} \]
\[ V = 1.5 \]

(C) With 1 particle = 10 mg/l

\[ N = 30/\text{day} \]
\[ N = 10/\text{day} \]

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

(D) \[ N/V = 30 \]
\[ N/V = 20 \]

Space density of particles in computer model on any particular day
where
\[ c = \text{concentration} \]
\[ t = \text{time} \]
\[ K = \text{decay coefficient} \]

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

\[ \text{half-life} = \frac{\ln(2)}{K} \]

where
\[ \ln = \text{natural log function} \]

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.

\[ M = M_0 \exp(-Kt) \]

where
\[ M = \text{particle mass at end of time step} \]
\[ M_0 = \text{particle mass at beginning of time step} \]
\[ t = \text{time step} \]

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:

\[ \text{fraction} = [1 - 0.5Kt/\ln(2)] \]

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

SUCCESSIVE PARTICLE POSITIONS – VELOCITIES ARE REINTERPOLATED AT THESE POINTS.
\[
\begin{align*}
XP &= (X_K - LLX) / DELX + 0.5 \\
YP &= (Y_K - LLY) / DELY + 0.5 \\
I &= \text{INT} (XP) \\
J &= \text{INT} (YP) \\
AX &= XP - \text{INT} (XP) \\
AY &= YP - \text{INT} (YP) \\
VX &= ((1 - AY) * VI_{I,J,LL} + AY * VI_{I,J+1,LL}) / E \\
VY &= ((1 - AX) * VJ_{I,J,LL} + AX * VJ_{I+1,J,LL}) / E
\end{align*}
\]

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

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.

\[
\begin{align*}
I &= \text{INT} (XP + 0.5) \\
J &= \text{INT} (YP + 0.5) \\
AZ &= (Z_K - BOT_{I,J,LL}) / (TOP_{I,J,LL} - BOT_{I,J,LL}) \\
VZ &= (AZ * VK_{I,J,LL} + (1 - AZ) * VK_{I,J,LL-1}) / E
\end{align*}
\]
Figure 9

Horizontal Velocity Interpolation

\[ V_{i-1,j+1,LL}, V_{i-1,j,LL}, V_{i,j,LL} \]

\[ \text{PARTICLE POSITION} \]
where

\[ Z_K = \text{z coordinate of particle (ft)} \]
\[ \text{BOT}_{I,J,LL} = \text{elevation of bottom of aquifer LL in grid } I,J^T \text{ (ft)} \]
\[ \text{TOP}_{I,J,LL} = \text{elevation of top of aquifer LL in grid } I,J^T \text{ (ft)} \]
\[ AZ = \text{fractional distance of particle above bottom of aquifer} \]
\[ V_Z = \text{vertical velocity of particle (ft/day)} \]
\[ V_{K,I,J,LL} = \text{vertical velocity vector from layer LL to LL-1 in grid } I,J \text{ (ft/day)} \]
\[ E = \text{effective porosity} \]

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

\[ V_Z = \frac{V_{K,I,J,LL}}{E} \]

when

\[ Z_K > \text{TOP}_{I,J,LL} \text{ and } Z_K < \text{BOT}_{I,J,LL+1} \]

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).

\[ DX = V_X \times F \]
\[ DY = V_Y \times F \]
\[ DZ = V_Z \times F \]

where

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

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.
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 relies, 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.

\[ \text{PHI} = \tan^{-1}(DY/DX) \]
\[ DD = \sqrt{DX^2 + DY^2} \]
\[ DDX = \sqrt{DX^2 + DY^2 + DZ^2} \]

where
- \( \text{PHI} \) = 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.

\[
\begin{align*}
RL &= \sqrt{\frac{2*DL*DD}{DD*RN}} \\
RT &= \sqrt{\frac{2*DT*DD}{DD*RN}} \\
RV &= \sqrt{\frac{2*DV*DDX}{DD*RN}}
\end{align*}
\]

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.

\[
\begin{align*}
X_K &= X_K + (DX + RL*DX + RT*DY) \\
Y_K &= Y_K + (DY + RL*DY - RT*DX) \\
Z_K &= Z_K + DZ + RV
\end{align*}
\]

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
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.

\[
\begin{align*}
V_X &= V_X / \text{RETARD}_{LL} \\
V_Y &= V_Y / \text{RETARD}_{LL} \\
V_Z &= V_Z / \text{RETARD}_{LL}
\end{align*}
\]

where

- \(V_X\) = velocity in x direction (ft/day)
- \(V_Y\) = velocity in y direction (ft/day)
- \(V_Z\) = velocity in z direction (ft/day)
- \(\text{RETARD}_{LL}\) = 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.

\[
\begin{align*}
\text{ZZZ} &= \text{ZZZ} + (1.5 \times (\text{DELTA} / \text{LAMDBA} / 365)) \\
\text{IF} \quad \text{ZZZ} &\geq 1 \quad \text{THEN} \\
\text{remove a particle} \\
X_K &= X_{NP} \\
Y_K &= Y_{NP} \\
Z_K &= Z_{NP} \\
\text{NP} &= \text{NP} - 1 \\
\text{ZZZ} &= \text{ZZZ} - 1.0
\end{align*}
\]

where

- \(\text{ZZZ}\) = fraction of particle removed
- \(\text{LAMDBA}\) = half-life (years)
- \(\text{DELTA}\) = time step (days)
- \(X_K, Y_K, Z_K\) = particle position (ft)
- \(\text{NP}\) = 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 \(D_{\text{MAX}}, Z_{\text{MAX}}, \text{ and } \Delta T \) above), the horizontal distance from each sink to the particle is calculated. Which layer the particle is in is also computed.

\[
R_3 = \sqrt{(X_K - X_L) \cdot (X_K - X_L) + (Y_K - Y_L) \cdot (Y_K - Y_L)}
\]

where

- \(R_3\) = distance from sink L to particle K (ft)
- \(X_K, Y_K\) = particle position (ft)
- \(X_L, Y_L\) = 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 (\(D_{\text{MAX}}\)) and the analytical distance that a particle could move toward the well during the time step.

\[
R_4 = \min[D_{\text{MAX}}, \sqrt{Q_{\text{SUM}, L} \cdot \Delta \tau / 3.14 / \text{THICK}_{II, JJ, KK} \cdot E^{7.48}}]
\]

where

- \(R_4\) = capture radius (ft)
- \(D_{\text{MAX}}\) = maximum horizontal move before recomputing the time step (ft)
- \(Q_{\text{SUM}, L}\) = flow rate for sink L (gpd)
- \(\Delta \tau\) = time step (days)
- \(\text{THICK}_{II, JJ, KK}\) = 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 (\( R_3 < R_4 \)) 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 \( \Delta T \), 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 \( D_{MAX} \), 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.

\[
\text{CONC}_L = \text{CONC}_L + \frac{(PM \times 119872)}{(QSUM \times \Delta T)}
\]

\[
X_k = X_{NP}
\]

\[
Y_k = Y_{NP}
\]

\[
Z_k = Z_{NP}
\]

\[
NP = NP - 1
\]

where

- \( \text{CONC}_L \) = 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.

\[
R_4 = \frac{QSUM \times F}{DELX/DELY/7.48/E}
\]

\[
\text{BOTCAPTURE} = \text{TOPII}, JJ, NL - R_4
\]
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 ZK>=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.

\[
I = \text{INT}(1.5 + ((\text{SCREENX}-I!) / \text{CDX}))
\]
\[
J = \text{INT}(1.5 + ((J2!-\text{SCREENY}) / \text{CDY}))
\]
\[
\text{NMAP}_{I,J} = \text{NMAP}_{I,J+1}
\]

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 NMAP\(_{I,J}\) 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:

\[
\text{NMAP}_{I,J} = 16030 \times \text{NMAP}_{I,J} \times \text{PM} / (E \times \text{CDX} \times \text{CDY} \times \text{TEMPTHCK}) \times \text{PPP}
\]

where
- \( \text{PM} \) = particle mass (lbs)
- \( E \) = effective porosity
- \( \text{TEMPTHCK} \) = relevant thickness times retardation coefficient
- \( \text{PPP} \) = 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.

\[
\begin{align*}
DX &= X_6 - X_5 \\
DY &= Y_6 - Y_5 \\
DZ &= Z_6 - Z_5 \\
\text{FOR } I &= 1 \text{ TO } M \\
X &= X_5 + DX \times \text{RND} \\
Y &= Y_5 + DY \times \text{RND} \\
Z &= Z_5 + DZ \times \text{RND} \\
NP &= NP + 1 \\
x_{NP} &= X \\
y_{NP} &= Y \\
z_{NP} &= Z \\
\text{IF } A \text{"S"} &= \text{"C" OR } A \text{"S"} = \text{"C" THEN } \text{SWITCH}\%_NP = 1 \text{ ELSE } \text{SWITCH}\%_NP = 0 \\
\text{NEXT } M
\end{align*}
\]
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
XNP,YNP,ZNP = 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.

\[
\text{VIEW (G1,G2)-(G3,G4),0}\\
\text{WINDOW (Il!,Jl!)-(I2!,J2!)}
\]

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
- \(Il!,Jl!\) = 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.

- \(\text{MA } x,y\) - move absolute, move to position \(x,y\) with the pen up
- \(\text{PA } x,y\) - plot absolute, plot a line from the present position to \(x,y\)
- \(\text{MR } x,y\) - move relative, move \(x\) units to the right and \(y\) units up with the pen up
- \(\text{PR } x,y\) - plot relative, plot a line from the present position to a position \(x\) units to the right and \(y\) units up
- \(\text{SP ipen}\) - select pen, use \(ipen\) as current pen color

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

\[
\text{MA 0,0}\\
\text{PA 1000,0}\\
\text{PA 1000,1500}\\
\text{PA 0,1500}\\
\text{PA 0,0}
\]
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 redefine 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
- THICKI,J,K - saturated thickness (ft)
- VII,J,K - Darcy velocity from I to I+1 (ft/day)
- VJI,J,K - Darcy velocity from J to J+1 (ft/day)
- VKI,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
- BOTI,J,K - bottom elevation of layer (ft)
- TOPI,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:

$X_{1L}$ - x coordinate of sink (ft)
$Y_{1L}$ - y coordinate of sink (ft)
$Z_{1L}$ - layer of sink
$Q_{1L}$ - discharge of sink (gpd), positive is flow out of model
### Example Velocity file (abc110.rnd)

<p>| | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>20</td>
<td>3</td>
<td>100.0</td>
<td>100.0</td>
<td>.0</td>
<td>.0</td>
<td>490.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>20</td>
<td>3</td>
<td>40.0</td>
<td>.984E-01</td>
<td>.000</td>
<td>.413E-01</td>
<td>660.0</td>
<td>695.0</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>20</td>
<td>3</td>
<td>42.0</td>
<td>.869E-01</td>
<td>.000</td>
<td>.288E-02</td>
<td>660.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>20</td>
<td>3</td>
<td>43.7</td>
<td>.771E-01</td>
<td>.000</td>
<td>.359E-02</td>
<td>660.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>20</td>
<td>3</td>
<td>45.3</td>
<td>.686E-01</td>
<td>.000</td>
<td>.386E-02</td>
<td>660.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>20</td>
<td>3</td>
<td>46.7</td>
<td>.611E-01</td>
<td>.000</td>
<td>.427E-02</td>
<td>660.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>20</td>
<td>3</td>
<td>47.9</td>
<td>.546E-01</td>
<td>.000</td>
<td>.464E-02</td>
<td>660.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>20</td>
<td>3</td>
<td>49.0</td>
<td>.487E-01</td>
<td>.000</td>
<td>.498E-02</td>
<td>660.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>20</td>
<td>3</td>
<td>50.0</td>
<td>.435E-01</td>
<td>.000</td>
<td>.527E-02</td>
<td>660.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>20</td>
<td>3</td>
<td>50.8</td>
<td>.386E-01</td>
<td>.000</td>
<td>.552E-02</td>
<td>660.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>20</td>
<td>3</td>
<td>51.6</td>
<td>.346E-01</td>
<td>.000</td>
<td>.573E-02</td>
<td>660.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>20</td>
<td>3</td>
<td>52.3</td>
<td>.307E-01</td>
<td>.000</td>
<td>.591E-02</td>
<td>660.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>20</td>
<td>3</td>
<td>52.9</td>
<td>.271E-01</td>
<td>.000</td>
<td>.606E-02</td>
<td>660.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>20</td>
<td>3</td>
<td>53.4</td>
<td>.236E-01</td>
<td>.000</td>
<td>.617E-02</td>
<td>660.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>20</td>
<td>3</td>
<td>53.9</td>
<td>.202E-01</td>
<td>.000</td>
<td>.626E-02</td>
<td>660.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>20</td>
<td>3</td>
<td>54.3</td>
<td>.169E-01</td>
<td>.000</td>
<td>.633E-02</td>
<td>660.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>20</td>
<td>3</td>
<td>54.7</td>
<td>.135E-01</td>
<td>.000</td>
<td>.637E-02</td>
<td>660.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>20</td>
<td>3</td>
<td>54.9</td>
<td>.102E-01</td>
<td>.000</td>
<td>.640E-02</td>
<td>660.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>20</td>
<td>3</td>
<td>55.1</td>
<td>.680E-02</td>
<td>.000</td>
<td>.642E-02</td>
<td>660.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>20</td>
<td>3</td>
<td>55.3</td>
<td>.340E-02</td>
<td>.000</td>
<td>.643E-02</td>
<td>660.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>20</td>
<td>3</td>
<td>55.3</td>
<td>.000</td>
<td>.000</td>
<td>.644E-02</td>
<td>660.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>100.0</td>
<td>.174E-02</td>
<td>.183E-04</td>
<td>.115E-03</td>
<td>500.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>100.0</td>
<td>.160E-02</td>
<td>.183E-04</td>
<td>.148E-03</td>
<td>500.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>100.0</td>
<td>.142E-02</td>
<td>.366E-04</td>
<td>.176E-03</td>
<td>500.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>100.0</td>
<td>.123E-02</td>
<td>.183E-04</td>
<td>.201E-03</td>
<td>500.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>100.0</td>
<td>.101E-02</td>
<td>.183E-04</td>
<td>.221E-03</td>
<td>500.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>100.0</td>
<td>.769E-03</td>
<td>.183E-04</td>
<td>.238E-03</td>
<td>500.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>100.0</td>
<td>.522E-03</td>
<td>.000</td>
<td>.250E-03</td>
<td>500.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>100.0</td>
<td>.266E-03</td>
<td>.183E-04</td>
<td>.259E-03</td>
<td>500.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>100.0</td>
<td>.000</td>
<td>.183E-04</td>
<td>.262E-03</td>
<td>500.0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<p>| | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>1</td>
<td>1</td>
<td>100.0</td>
<td>.174E-02</td>
<td>.183E-04</td>
<td>.115E-03</td>
<td>500.0</td>
<td>600.0</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td>1</td>
<td>100.0</td>
<td>.160E-02</td>
<td>.183E-04</td>
<td>.148E-03</td>
<td>500.0</td>
<td>600.0</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>1</td>
<td>1</td>
<td>100.0</td>
<td>.142E-02</td>
<td>.366E-04</td>
<td>.176E-03</td>
<td>500.0</td>
<td>600.0</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>1</td>
<td>1</td>
<td>100.0</td>
<td>.123E-02</td>
<td>.183E-04</td>
<td>.201E-03</td>
<td>500.0</td>
<td>600.0</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>1</td>
<td>1</td>
<td>100.0</td>
<td>.101E-02</td>
<td>.183E-04</td>
<td>.221E-03</td>
<td>500.0</td>
<td>600.0</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>1</td>
<td>1</td>
<td>100.0</td>
<td>.769E-03</td>
<td>.183E-04</td>
<td>.238E-03</td>
<td>500.0</td>
<td>600.0</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>1</td>
<td>1</td>
<td>100.0</td>
<td>.522E-03</td>
<td>.000</td>
<td>.250E-03</td>
<td>500.0</td>
<td>600.0</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>1</td>
<td>1</td>
<td>100.0</td>
<td>.266E-03</td>
<td>.183E-04</td>
<td>.259E-03</td>
<td>500.0</td>
<td>600.0</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>1</td>
<td>1</td>
<td>100.0</td>
<td>.000</td>
<td>.183E-04</td>
<td>.262E-03</td>
<td>500.0</td>
<td>600.0</td>
<td></td>
</tr>
</tbody>
</table>

<p>| | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1150.0</td>
<td>950.0</td>
<td>3.0</td>
<td>.1440E+05</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1950.0</td>
<td>3.0</td>
<td>3260.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1850.0</td>
<td>3.0</td>
<td>3260.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1750.0</td>
<td>3.0</td>
<td>3255.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1650.0</td>
<td>3.0</td>
<td>3246.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1550.0</td>
<td>3.0</td>
<td>3232.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1450.0</td>
<td>3.0</td>
<td>3219.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1350.0</td>
<td>3.0</td>
<td>3205.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1250.0</td>
<td>3.0</td>
<td>3191.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1150.0</td>
<td>3.0</td>
<td>3182.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1050.0</td>
<td>3.0</td>
<td>3173.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>950.0</td>
<td>3.0</td>
<td>3173.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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
PICCOL.DAT
MOVE 60.8333
10 1
GS
INPUT
COLO2
MOVE 60.8333
10 1
GS
INPUT
COLO3
MOVE 60.8333
10 1
GS
WR
COLTEST.DAT
MAP
GC
B
```

read in previously saved file PICCOL.DAT

move particles for 60 days with max moves of 10' and 1'

go to move and save screen

input a new velocity file

move again

input a new velocity file

move again

save restart file

map current screen area

go - concentration map

ppb
3

map 3rd layer only
end of map (blank line)

PLT
2090,840
60
32,21
0
CL1
Q

create files for all aquifers
seed for plot file names
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 adequately 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 calculated 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

<table>
<thead>
<tr>
<th>COMMAND</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>SET</td>
<td>INPUT - READ IN FLOW</td>
</tr>
<tr>
<td></td>
<td>COEFFICIENTS</td>
</tr>
<tr>
<td>RE</td>
<td>P - ADD PARTICLES</td>
</tr>
<tr>
<td>WR</td>
<td>(POLLUTANTS)</td>
</tr>
<tr>
<td>PLT</td>
<td>MOVE - MOVE PARTICLES</td>
</tr>
<tr>
<td></td>
<td>FOR TIME STEP</td>
</tr>
<tr>
<td>XS</td>
<td>CLEAR SINKS</td>
</tr>
<tr>
<td>XP</td>
<td>MAP - MAP PARTICLES OR</td>
</tr>
<tr>
<td>XT</td>
<td>CONCENTRATIONS</td>
</tr>
<tr>
<td>SPF</td>
<td>VIEW - REVIEW SLIDES</td>
</tr>
<tr>
<td></td>
<td>Q - QUIT</td>
</tr>
</tbody>
</table>

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.
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"

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

<table>
<thead>
<tr>
<th>FORMAT: COMMAND - DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>SET - MASS TRANSPORT COEFFICIENTS - INPUT - READ IN FLOW DATA FILE</td>
</tr>
<tr>
<td>RE - READ DATA - P - ADD PARTICLES (POLUTANTS)</td>
</tr>
<tr>
<td>WR - WRITE DATA</td>
</tr>
<tr>
<td>PLT - SAVE PLOT DATA - MOVE - MOVE PARTICLES FOR TIME STEP</td>
</tr>
<tr>
<td>XS - CLEAR SINKS</td>
</tr>
<tr>
<td>XP - CLEAR PARTICLES - MAP - MAP PARTICLES OR CONCENTRATIONS</td>
</tr>
<tr>
<td>XT - SET TIME TO 0</td>
</tr>
<tr>
<td>SPF - SETUP SPECIAL FEATURES - VIEW - REVIEW SLIDES</td>
</tr>
<tr>
<td>Q - QUIT</td>
</tr>
</tbody>
</table>

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.
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 must 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.

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.

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

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.

\[ \text{ADD PARTICLES TO SYSTEM? (Y-YES, <RETURN>-NO)? y} \]

\[ \text{CYLINDER, LINE, OR RECTANGULAR (C, L, R) ? l} \]

\[ \text{LINE NUMBER 1} \]

\[ \text{COORDINATES OF LINE BEGINNING (X,Y,Z) ? 2300,1000,600} \]
\[ \text{COORDINATES OF END OF LINE (X,Y,Z) ? 2400,1200,650} \]

Enter the letter C for CONTINUOUS POLLUTION—otherwise a <RETURN> or any other character will assume a SLUG input?

\[ \text{NUMBER OF PARTICLES ? 43} \]

\[ \text{DO YOU WANT TO REDO THIS SCREEN? (Y-YES, <RETURN>-NO)?} \]

\[ \text{ADD PARTICLES TO SYSTEM? (Y-YES, <RETURN>-NO)?} \]

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 PEGRIA.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.
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:

![Diagram 1](image1)

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

![Diagram 2](image2)
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

- plot sinks as circles
- this menu
- zoom routine
- profile row
- leave special feature subroutine
- rivers.dat
- bldgs.dat
- streets.dat
- clear screen
- return to default settings
- draw model grid
- profile column

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 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
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)

19671: * * 3 1 1 * * * * * * * * *
18551: * * 1 1 * * * * * * * * * * *
17421: * 1 2 * * * * * * * * * * * * *
16291: * * * * * * * * * * * * * * * *
15161: P * * * P * * * * * * * * * * *
14031: * * * * * * * * * * * P * * * *
12911: * * * * * * * * * * * * * * * *
11781: * * * * * * * * * * * * * * * *
10651: * * * * P P P P P P P P P * * *
9521: * * * * * * * * * * * * * * * *

1 2 2 2 1 2 2 2 2 2 3 3 3 3 3
2 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**

<table>
<thead>
<tr>
<th>FORMAT: COMMAND</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>SET - MASS TRANSPORT COEFFICIENTS</td>
<td>INPUT - READ IN FLOW DATA FILE</td>
</tr>
<tr>
<td>RE - READ DATA</td>
<td>P - ADD PARTICLES (POLLUTANTS)</td>
</tr>
<tr>
<td>WR - WRITE DATA</td>
<td></td>
</tr>
<tr>
<td>PLT - SAVE PLOT DATA</td>
<td>MOVE - MOVE PARTICLES FOR TIME STEP</td>
</tr>
<tr>
<td>XS - CLEAR SINKS</td>
<td>MAP - MAP PARTICLES OR CONCENTRATIONS</td>
</tr>
<tr>
<td>XP - CLEAR PARTICLES</td>
<td></td>
</tr>
<tr>
<td>XT - SET TIME TO 0</td>
<td></td>
</tr>
<tr>
<td>SPF - SETUP SPECIAL FEATURES</td>
<td>VIEW - REVIEW SLIDES</td>
</tr>
<tr>
<td></td>
<td>G - QUIT</td>
</tr>
</tbody>
</table>

**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

<table>
<thead>
<tr>
<th>COMMAND</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>SET</td>
<td>MASS TRANSPORT</td>
</tr>
<tr>
<td>COEFFICIENTS</td>
<td>INPUT - READ IN FLOW</td>
</tr>
<tr>
<td>RE</td>
<td>READ DATA</td>
</tr>
<tr>
<td>WR</td>
<td>WRITE DATA</td>
</tr>
<tr>
<td>PLT</td>
<td>SAVE PLOT DATA</td>
</tr>
<tr>
<td>XS</td>
<td>CLEAR SINKS</td>
</tr>
<tr>
<td>XP</td>
<td>CLEAR PARTICLES</td>
</tr>
<tr>
<td>XT</td>
<td>SET TIME TO 0</td>
</tr>
<tr>
<td>SPF</td>
<td>SETUP SPECIAL</td>
</tr>
<tr>
<td>FEATURES</td>
<td>VIEW - REVIEW SLIDES</td>
</tr>
<tr>
<td>Q</td>
<td>QUIT</td>
</tr>
</tbody>
</table>

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.

C-73
V. References


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)
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)
DET. - 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)
TEMPSKR - 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

V\text{I}(I,J,K) - VELOCITY IN X DIRECTION FROM I,J,K TO I+1,J,K
(FT/DAY)

V\text{J}(I,J,K) - VELOCITY IN Y DIRECTION FROM I,J,K TO I,J+1,K
(FT/DAY)

V\text{K}(I,J,K) - VELOCITY IN Z DIRECTION FROM I,J,K TO I,J,K+1
(FT/DAY)

V\text{X} - INTERPOLATED X VELOCITY (FT/DAY)

V\text{Y} - INTERPOLATED Y VELOCITY (FT/DAY)

V\text{Z} - INTERPOLATED Z VELOCITY (FT/DAY)

W\text{D} - COLUMNS OF MAP

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

X\text{1}(L) - X COORDINATE OF SINK L (FT)

X\text{P} - 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)

Y\text{1}(L) - Y COORDINATE OF SINK L (FT)

Y\text{ASPECT} - ASPECT RATIO OF SCREEN FOR TOP VIEW

Y\text{P} - Y COORDINATE OF PARTICLE POSITION IN GRID UNITS

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

Z\text{1}(L) - LAYER OF SINK L

Z\text{2}\$ - STRING OF KEYS PRESSED TO PUT SPECIAL FEATURES ON
SCREEN

Z\text{3}\$ - STRING OF KEYS PRESSED TO PUT SPECIAL FEATURES ON
SCREEN

Z\text{ASPECT} - ASPECT RATIO OF SCREEN FOR CROSS-SECTION VIEW

Z\text{MAX} - 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

\[
C = \frac{M}{(8\pi t^3 D_x D_y D_z)^{0.5}} \exp\left(-\frac{(x-vt)^2}{4D_xt} - \frac{y^2}{4D_yt} - \frac{z^2}{4D_zt}\right)
\]

(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 = \(\alpha_x v\)
- \(D_y\) = dispersion in y direction = \(\alpha_y v\)
- \(D_z\) = dispersion in z direction = \(\alpha_z v\)
- \(n\) = 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 ft²/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
110 INPUT " FILE "; A$; " OPENED FOR OUTPUT"
120 INPUT " ENTER # OF COLUMNS "; NC
130 INPUT " ENTER # OF ROWS "; NR
140 INPUT " ENTER # OF LAYERS "; NL
150 INPUT " ENTER GRID SPACING FOR COLUMNS "; DELX
160 INPUT " ENTER GRID SPACING FOR ROWS "; DELY
170 INPUT " ENTER X,Y,Z COORDINATES OF LOWER LEFT
CORNER OF MODEL "; LLX, LLY, LLZ
180 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 12

Analytical Results - layer 2 - xy plane
Figure 13

RAN3D verification - row 6 - xz plane
Figure 14

Analytical Results - row 6 - xz plane
Example 1 - Attachment 1

Velocity File

<p>| | | | | | | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>14</td>
<td>11</td>
<td>3</td>
<td>10</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>1 1</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>1 1</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>1 1</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>1 1</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>1 1</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>2</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>2</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>2 1</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>2 1</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>2 1</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>2 1</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>2 1</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>3</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>3</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>3 1</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>3 1</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>3 1</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>3 1</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>3 1</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>4</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>4</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>4</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>4</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>4</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>4</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>4</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>4</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>5</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>5</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>5</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>5</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>5</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>5</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>5</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>5</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>5</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>5</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>6</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>6</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>6</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>6</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>6</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>6</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>6</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>6</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>6</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>6</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>7</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>7</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>7</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>7</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>7</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>7</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>7</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>7</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>7</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>7</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>7</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>7</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>7</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>8</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>8</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>8</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>8</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>8</td>
<td>1</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

C-87
<table>
<thead>
<tr>
<th></th>
<th>8</th>
<th>120</th>
<th>0.1</th>
<th>0.0</th>
<th>0.0</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>8</td>
<td>120</td>
<td>0.1</td>
<td>0.0</td>
<td>0.0</td>
<td>20</td>
</tr>
<tr>
<td>9</td>
<td>8</td>
<td>120</td>
<td>0.1</td>
<td>0.0</td>
<td>0.0</td>
<td>20</td>
</tr>
<tr>
<td>10</td>
<td>8</td>
<td>120</td>
<td>0.1</td>
<td>0.0</td>
<td>0.0</td>
<td>20</td>
</tr>
<tr>
<td>11</td>
<td>8</td>
<td>120</td>
<td>0.1</td>
<td>0.0</td>
<td>0.0</td>
<td>20</td>
</tr>
<tr>
<td>12</td>
<td>8</td>
<td>120</td>
<td>0.1</td>
<td>0.0</td>
<td>0.0</td>
<td>20</td>
</tr>
<tr>
<td>13</td>
<td>8</td>
<td>120</td>
<td>0.1</td>
<td>0.0</td>
<td>0.0</td>
<td>20</td>
</tr>
<tr>
<td>14</td>
<td>8</td>
<td>120</td>
<td>0.1</td>
<td>0.0</td>
<td>0.0</td>
<td>20</td>
</tr>
<tr>
<td>1</td>
<td>9</td>
<td>120</td>
<td>0.1</td>
<td>0.0</td>
<td>0.0</td>
<td>20</td>
</tr>
<tr>
<td>2</td>
<td>9</td>
<td>120</td>
<td>0.1</td>
<td>0.0</td>
<td>0.0</td>
<td>20</td>
</tr>
<tr>
<td>3</td>
<td>9</td>
<td>120</td>
<td>0.1</td>
<td>0.0</td>
<td>0.0</td>
<td>20</td>
</tr>
<tr>
<td>4</td>
<td>9</td>
<td>120</td>
<td>0.1</td>
<td>0.0</td>
<td>0.0</td>
<td>20</td>
</tr>
<tr>
<td>5</td>
<td>9</td>
<td>120</td>
<td>0.1</td>
<td>0.0</td>
<td>0.0</td>
<td>20</td>
</tr>
<tr>
<td>6</td>
<td>9</td>
<td>120</td>
<td>0.1</td>
<td>0.0</td>
<td>0.0</td>
<td>20</td>
</tr>
<tr>
<td>7</td>
<td>9</td>
<td>120</td>
<td>0.1</td>
<td>0.0</td>
<td>0.0</td>
<td>20</td>
</tr>
<tr>
<td>8</td>
<td>9</td>
<td>120</td>
<td>0.1</td>
<td>0.0</td>
<td>0.0</td>
<td>20</td>
</tr>
<tr>
<td>9</td>
<td>9</td>
<td>120</td>
<td>0.1</td>
<td>0.0</td>
<td>0.0</td>
<td>20</td>
</tr>
<tr>
<td>10</td>
<td>9</td>
<td>120</td>
<td>0.1</td>
<td>0.0</td>
<td>0.0</td>
<td>20</td>
</tr>
<tr>
<td>11</td>
<td>9</td>
<td>120</td>
<td>0.1</td>
<td>0.0</td>
<td>0.0</td>
<td>20</td>
</tr>
<tr>
<td>12</td>
<td>9</td>
<td>120</td>
<td>0.1</td>
<td>0.0</td>
<td>0.0</td>
<td>20</td>
</tr>
<tr>
<td>13</td>
<td>9</td>
<td>120</td>
<td>0.1</td>
<td>0.0</td>
<td>0.0</td>
<td>20</td>
</tr>
<tr>
<td>14</td>
<td>9</td>
<td>120</td>
<td>0.1</td>
<td>0.0</td>
<td>0.0</td>
<td>20</td>
</tr>
<tr>
<td>1</td>
<td>10</td>
<td>120</td>
<td>0.1</td>
<td>0.0</td>
<td>0.0</td>
<td>20</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>120</td>
<td>0.1</td>
<td>0.0</td>
<td>0.0</td>
<td>20</td>
</tr>
<tr>
<td>3</td>
<td>10</td>
<td>120</td>
<td>0.1</td>
<td>0.0</td>
<td>0.0</td>
<td>20</td>
</tr>
<tr>
<td>4</td>
<td>10</td>
<td>120</td>
<td>0.1</td>
<td>0.0</td>
<td>0.0</td>
<td>20</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td>120</td>
<td>0.1</td>
<td>0.0</td>
<td>0.0</td>
<td>20</td>
</tr>
<tr>
<td>6</td>
<td>10</td>
<td>120</td>
<td>0.1</td>
<td>0.0</td>
<td>0.0</td>
<td>20</td>
</tr>
<tr>
<td>7</td>
<td>10</td>
<td>120</td>
<td>0.1</td>
<td>0.0</td>
<td>0.0</td>
<td>20</td>
</tr>
<tr>
<td>8</td>
<td>10</td>
<td>120</td>
<td>0.1</td>
<td>0.0</td>
<td>0.0</td>
<td>20</td>
</tr>
<tr>
<td>9</td>
<td>10</td>
<td>120</td>
<td>0.1</td>
<td>0.0</td>
<td>0.0</td>
<td>20</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>120</td>
<td>0.1</td>
<td>0.0</td>
<td>0.0</td>
<td>20</td>
</tr>
<tr>
<td>11</td>
<td>10</td>
<td>120</td>
<td>0.1</td>
<td>0.0</td>
<td>0.0</td>
<td>20</td>
</tr>
<tr>
<td>12</td>
<td>10</td>
<td>120</td>
<td>0.1</td>
<td>0.0</td>
<td>0.0</td>
<td>20</td>
</tr>
<tr>
<td>13</td>
<td>10</td>
<td>120</td>
<td>0.1</td>
<td>0.0</td>
<td>0.0</td>
<td>20</td>
</tr>
<tr>
<td>14</td>
<td>10</td>
<td>120</td>
<td>0.1</td>
<td>0.0</td>
<td>0.0</td>
<td>20</td>
</tr>
<tr>
<td>1</td>
<td>11</td>
<td>120</td>
<td>0.1</td>
<td>0.0</td>
<td>0.0</td>
<td>20</td>
</tr>
<tr>
<td>2</td>
<td>11</td>
<td>120</td>
<td>0.1</td>
<td>0.0</td>
<td>0.0</td>
<td>20</td>
</tr>
<tr>
<td>3</td>
<td>11</td>
<td>120</td>
<td>0.1</td>
<td>0.0</td>
<td>0.0</td>
<td>20</td>
</tr>
<tr>
<td>4</td>
<td>11</td>
<td>120</td>
<td>0.1</td>
<td>0.0</td>
<td>0.0</td>
<td>20</td>
</tr>
<tr>
<td>5</td>
<td>11</td>
<td>120</td>
<td>0.1</td>
<td>0.0</td>
<td>0.0</td>
<td>20</td>
</tr>
<tr>
<td>6</td>
<td>11</td>
<td>120</td>
<td>0.1</td>
<td>0.0</td>
<td>0.0</td>
<td>20</td>
</tr>
<tr>
<td>7</td>
<td>11</td>
<td>120</td>
<td>0.1</td>
<td>0.0</td>
<td>0.0</td>
<td>20</td>
</tr>
<tr>
<td>8</td>
<td>11</td>
<td>120</td>
<td>0.1</td>
<td>0.0</td>
<td>0.0</td>
<td>20</td>
</tr>
<tr>
<td>9</td>
<td>11</td>
<td>120</td>
<td>0.1</td>
<td>0.0</td>
<td>0.0</td>
<td>20</td>
</tr>
<tr>
<td>10</td>
<td>11</td>
<td>120</td>
<td>0.1</td>
<td>0.0</td>
<td>0.0</td>
<td>20</td>
</tr>
<tr>
<td>11</td>
<td>11</td>
<td>120</td>
<td>0.1</td>
<td>0.0</td>
<td>0.0</td>
<td>20</td>
</tr>
<tr>
<td>12</td>
<td>11</td>
<td>120</td>
<td>0.1</td>
<td>0.0</td>
<td>0.0</td>
<td>20</td>
</tr>
<tr>
<td>13</td>
<td>11</td>
<td>120</td>
<td>0.1</td>
<td>0.0</td>
<td>0.0</td>
<td>20</td>
</tr>
<tr>
<td>14</td>
<td>11</td>
<td>120</td>
<td>0.1</td>
<td>0.0</td>
<td>0.0</td>
<td>20</td>
</tr>
<tr>
<td>1</td>
<td>12</td>
<td>120</td>
<td>0.1</td>
<td>0.0</td>
<td>0.0</td>
<td>20</td>
</tr>
<tr>
<td>2</td>
<td>12</td>
<td>120</td>
<td>0.1</td>
<td>0.0</td>
<td>0.0</td>
<td>20</td>
</tr>
<tr>
<td>3</td>
<td>12</td>
<td>120</td>
<td>0.1</td>
<td>0.0</td>
<td>0.0</td>
<td>20</td>
</tr>
<tr>
<td>4</td>
<td>12</td>
<td>120</td>
<td>0.1</td>
<td>0.0</td>
<td>0.0</td>
<td>20</td>
</tr>
</tbody>
</table>

**C-88**
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

C-89
<p>| | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>5</td>
<td>2</td>
<td>10</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>30</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>2</td>
<td>10</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>30</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>2</td>
<td>10</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>30</td>
</tr>
<tr>
<td>6</td>
<td>5</td>
<td>2</td>
<td>10</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>30</td>
</tr>
<tr>
<td>7</td>
<td>5</td>
<td>2</td>
<td>10</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>30</td>
</tr>
<tr>
<td>8</td>
<td>5</td>
<td>2</td>
<td>10</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>30</td>
</tr>
<tr>
<td>9</td>
<td>5</td>
<td>2</td>
<td>10</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>30</td>
</tr>
<tr>
<td>10</td>
<td>5</td>
<td>2</td>
<td>10</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>30</td>
</tr>
<tr>
<td>11</td>
<td>5</td>
<td>2</td>
<td>10</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>30</td>
</tr>
<tr>
<td>12</td>
<td>5</td>
<td>2</td>
<td>10</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>30</td>
</tr>
<tr>
<td>13</td>
<td>5</td>
<td>2</td>
<td>10</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>30</td>
</tr>
<tr>
<td>14</td>
<td>5</td>
<td>2</td>
<td>10</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>30</td>
</tr>
<tr>
<td>1</td>
<td>6</td>
<td>2</td>
<td>10</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>30</td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td>2</td>
<td>10</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>30</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
<td>2</td>
<td>10</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>30</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>2</td>
<td>10</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>30</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td>2</td>
<td>10</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>30</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>2</td>
<td>10</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>30</td>
</tr>
<tr>
<td>7</td>
<td>6</td>
<td>2</td>
<td>10</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>30</td>
</tr>
<tr>
<td>8</td>
<td>6</td>
<td>2</td>
<td>10</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>30</td>
</tr>
<tr>
<td>9</td>
<td>6</td>
<td>2</td>
<td>10</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>30</td>
</tr>
<tr>
<td>10</td>
<td>6</td>
<td>2</td>
<td>10</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>30</td>
</tr>
<tr>
<td>11</td>
<td>6</td>
<td>2</td>
<td>10</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>30</td>
</tr>
<tr>
<td>12</td>
<td>6</td>
<td>2</td>
<td>10</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>30</td>
</tr>
<tr>
<td>13</td>
<td>6</td>
<td>2</td>
<td>10</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>30</td>
</tr>
<tr>
<td>14</td>
<td>6</td>
<td>2</td>
<td>10</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>30</td>
</tr>
<tr>
<td>1</td>
<td>7</td>
<td>2</td>
<td>10</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>30</td>
</tr>
<tr>
<td>2</td>
<td>7</td>
<td>2</td>
<td>10</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>30</td>
</tr>
<tr>
<td>3</td>
<td>7</td>
<td>2</td>
<td>10</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>30</td>
</tr>
<tr>
<td>4</td>
<td>7</td>
<td>2</td>
<td>10</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>30</td>
</tr>
<tr>
<td>5</td>
<td>7</td>
<td>2</td>
<td>10</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>30</td>
</tr>
<tr>
<td>6</td>
<td>7</td>
<td>2</td>
<td>10</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>30</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>2</td>
<td>10</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>30</td>
</tr>
<tr>
<td>8</td>
<td>7</td>
<td>2</td>
<td>10</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>30</td>
</tr>
<tr>
<td>9</td>
<td>7</td>
<td>2</td>
<td>10</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>30</td>
</tr>
<tr>
<td>10</td>
<td>7</td>
<td>2</td>
<td>10</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>30</td>
</tr>
<tr>
<td>11</td>
<td>7</td>
<td>2</td>
<td>10</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>30</td>
</tr>
<tr>
<td>12</td>
<td>7</td>
<td>2</td>
<td>10</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>30</td>
</tr>
<tr>
<td>13</td>
<td>7</td>
<td>2</td>
<td>10</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>30</td>
</tr>
<tr>
<td>14</td>
<td>7</td>
<td>2</td>
<td>10</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>30</td>
</tr>
<tr>
<td>1</td>
<td>8</td>
<td>2</td>
<td>10</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>30</td>
</tr>
<tr>
<td>2</td>
<td>8</td>
<td>2</td>
<td>10</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>30</td>
</tr>
<tr>
<td>3</td>
<td>8</td>
<td>2</td>
<td>10</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>30</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>2</td>
<td>10</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>30</td>
</tr>
<tr>
<td>5</td>
<td>8</td>
<td>2</td>
<td>10</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>30</td>
</tr>
<tr>
<td>6</td>
<td>8</td>
<td>2</td>
<td>10</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>30</td>
</tr>
<tr>
<td>7</td>
<td>8</td>
<td>2</td>
<td>10</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>30</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>2</td>
<td>10</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>30</td>
</tr>
<tr>
<td>9</td>
<td>8</td>
<td>2</td>
<td>10</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>30</td>
</tr>
<tr>
<td>10</td>
<td>8</td>
<td>2</td>
<td>10</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>30</td>
</tr>
<tr>
<td>11</td>
<td>8</td>
<td>2</td>
<td>10</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>30</td>
</tr>
<tr>
<td>12</td>
<td>8</td>
<td>2</td>
<td>10</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>30</td>
</tr>
<tr>
<td>13</td>
<td>8</td>
<td>2</td>
<td>10</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>30</td>
</tr>
<tr>
<td>14</td>
<td>8</td>
<td>2</td>
<td>10</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>9</td>
<td>2</td>
<td>10</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>9</td>
<td>2</td>
<td>10</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>9</td>
<td>2</td>
<td>10</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>9</td>
<td>2</td>
<td>10</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>9</td>
<td>2</td>
<td>10</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>9</td>
<td>2</td>
<td>10</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>9</td>
<td>2</td>
<td>10</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>9</td>
<td>2</td>
<td>10</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>9</td>
<td>2</td>
<td>10</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>9</td>
<td>2</td>
<td>10</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>9</td>
<td>2</td>
<td>10</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>9</td>
<td>2</td>
<td>10</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>9</td>
<td>2</td>
<td>10</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>9</td>
<td>2</td>
<td>10</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>10</td>
<td>2</td>
<td>10</td>
<td>.1</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>10</td>
<td>2</td>
<td>10</td>
<td>.1</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>10</td>
<td>2</td>
<td>10</td>
<td>.1</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>10</td>
<td>2</td>
<td>10</td>
<td>.1</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>10</td>
<td>2</td>
<td>10</td>
<td>.1</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>10</td>
<td>2</td>
<td>10</td>
<td>.1</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>10</td>
<td>2</td>
<td>10</td>
<td>.1</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>10</td>
<td>2</td>
<td>10</td>
<td>.1</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>10</td>
<td>2</td>
<td>10</td>
<td>.1</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>2</td>
<td>10</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>10</td>
<td>2</td>
<td>10</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>10</td>
<td>2</td>
<td>10</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>10</td>
<td>2</td>
<td>10</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>10</td>
<td>2</td>
<td>10</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>11</td>
<td>2</td>
<td>10</td>
<td>.1</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>11</td>
<td>2</td>
<td>10</td>
<td>.1</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>11</td>
<td>2</td>
<td>10</td>
<td>.1</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>11</td>
<td>2</td>
<td>10</td>
<td>.1</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>11</td>
<td>2</td>
<td>10</td>
<td>.1</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>11</td>
<td>2</td>
<td>10</td>
<td>.1</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>11</td>
<td>2</td>
<td>10</td>
<td>.1</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>11</td>
<td>2</td>
<td>10</td>
<td>.1</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>11</td>
<td>2</td>
<td>10</td>
<td>.1</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>11</td>
<td>2</td>
<td>10</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>11</td>
<td>2</td>
<td>10</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>11</td>
<td>2</td>
<td>10</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>11</td>
<td>2</td>
<td>10</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>11</td>
<td>2</td>
<td>10</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>3</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>3</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>3</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>3</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>3</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>3</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>3</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>3</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>3</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>3</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>3</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>3</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

C-91
<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>1</td>
<td>3</td>
<td>20</td>
<td>.1</td>
<td>0</td>
</tr>
<tr>
<td>14</td>
<td>1</td>
<td>3</td>
<td>20</td>
<td>.1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>20</td>
<td>.1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>3</td>
<td>20</td>
<td>.1</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>3</td>
<td>20</td>
<td>.1</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>3</td>
<td>20</td>
<td>.1</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>3</td>
<td>20</td>
<td>.1</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>3</td>
<td>20</td>
<td>.1</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>3</td>
<td>20</td>
<td>.1</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
<td>3</td>
<td>20</td>
<td>.1</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>2</td>
<td>3</td>
<td>20</td>
<td>.1</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>2</td>
<td>3</td>
<td>20</td>
<td>.1</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>2</td>
<td>3</td>
<td>20</td>
<td>.1</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>2</td>
<td>3</td>
<td>20</td>
<td>.1</td>
<td>0</td>
</tr>
<tr>
<td>13</td>
<td>2</td>
<td>3</td>
<td>20</td>
<td>.1</td>
<td>0</td>
</tr>
<tr>
<td>14</td>
<td>2</td>
<td>3</td>
<td>20</td>
<td>.1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>3</td>
<td>20</td>
<td>.1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>3</td>
<td>20</td>
<td>.1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>3</td>
<td>20</td>
<td>.1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>3</td>
<td>20</td>
<td>.1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>3</td>
<td>20</td>
<td>.1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>3</td>
<td>20</td>
<td>.1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>3</td>
<td>20</td>
<td>.1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>3</td>
<td>20</td>
<td>.1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>3</td>
<td>20</td>
<td>.1</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>3</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>3</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>3</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>13</td>
<td>3</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>14</td>
<td>3</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>4</td>
<td>20</td>
<td>.1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>4</td>
<td>20</td>
<td>.1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>4</td>
<td>20</td>
<td>.1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>4</td>
<td>20</td>
<td>.1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>4</td>
<td>20</td>
<td>.1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>4</td>
<td>20</td>
<td>.1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>4</td>
<td>20</td>
<td>.1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>4</td>
<td>20</td>
<td>.1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>4</td>
<td>20</td>
<td>.1</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>4</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>4</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>4</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>13</td>
<td>4</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>14</td>
<td>4</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>5</td>
<td>20</td>
<td>.1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>5</td>
<td>20</td>
<td>.1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>5</td>
<td>20</td>
<td>.1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>5</td>
<td>20</td>
<td>.1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>5</td>
<td>20</td>
<td>.1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>5</td>
<td>20</td>
<td>.1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>5</td>
<td>20</td>
<td>.1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>5</td>
<td>20</td>
<td>.1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>5</td>
<td>20</td>
<td>.1</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>5</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
C-92
<p>| | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>9</td>
<td>3</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>30</td>
<td>50</td>
</tr>
<tr>
<td>10</td>
<td>9</td>
<td>3</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>30</td>
<td>50</td>
</tr>
<tr>
<td>11</td>
<td>9</td>
<td>3</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>30</td>
<td>50</td>
</tr>
<tr>
<td>12</td>
<td>9</td>
<td>3</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>30</td>
<td>50</td>
</tr>
<tr>
<td>13</td>
<td>9</td>
<td>3</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>30</td>
<td>50</td>
</tr>
<tr>
<td>14</td>
<td>9</td>
<td>3</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>30</td>
<td>50</td>
</tr>
<tr>
<td>1</td>
<td>10</td>
<td>3</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>30</td>
<td>50</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>3</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>30</td>
<td>50</td>
</tr>
<tr>
<td>3</td>
<td>10</td>
<td>3</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>30</td>
<td>50</td>
</tr>
<tr>
<td>4</td>
<td>10</td>
<td>3</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>30</td>
<td>50</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td>3</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>30</td>
<td>50</td>
</tr>
<tr>
<td>6</td>
<td>10</td>
<td>3</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>30</td>
<td>50</td>
</tr>
<tr>
<td>7</td>
<td>10</td>
<td>3</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>30</td>
<td>50</td>
</tr>
<tr>
<td>8</td>
<td>10</td>
<td>3</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>30</td>
<td>50</td>
</tr>
<tr>
<td>9</td>
<td>10</td>
<td>3</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>30</td>
<td>50</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>3</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>30</td>
<td>50</td>
</tr>
<tr>
<td>11</td>
<td>10</td>
<td>3</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>30</td>
<td>50</td>
</tr>
<tr>
<td>12</td>
<td>10</td>
<td>3</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>30</td>
<td>50</td>
</tr>
<tr>
<td>13</td>
<td>10</td>
<td>3</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>30</td>
<td>50</td>
</tr>
<tr>
<td>14</td>
<td>10</td>
<td>3</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>30</td>
<td>50</td>
</tr>
<tr>
<td>1</td>
<td>11</td>
<td>3</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>30</td>
<td>50</td>
</tr>
<tr>
<td>2</td>
<td>11</td>
<td>3</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>30</td>
<td>50</td>
</tr>
<tr>
<td>3</td>
<td>11</td>
<td>3</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>30</td>
<td>50</td>
</tr>
<tr>
<td>4</td>
<td>11</td>
<td>3</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>30</td>
<td>50</td>
</tr>
<tr>
<td>5</td>
<td>11</td>
<td>3</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>30</td>
<td>50</td>
</tr>
<tr>
<td>6</td>
<td>11</td>
<td>3</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>30</td>
<td>50</td>
</tr>
<tr>
<td>7</td>
<td>11</td>
<td>3</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>30</td>
<td>50</td>
</tr>
<tr>
<td>8</td>
<td>11</td>
<td>3</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>30</td>
<td>50</td>
</tr>
<tr>
<td>9</td>
<td>11</td>
<td>3</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>30</td>
<td>50</td>
</tr>
<tr>
<td>10</td>
<td>11</td>
<td>3</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>30</td>
<td>50</td>
</tr>
<tr>
<td>11</td>
<td>11</td>
<td>3</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>30</td>
<td>50</td>
</tr>
<tr>
<td>12</td>
<td>11</td>
<td>3</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>30</td>
<td>50</td>
</tr>
<tr>
<td>13</td>
<td>11</td>
<td>3</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>30</td>
<td>50</td>
</tr>
<tr>
<td>14</td>
<td>11</td>
<td>3</td>
<td>20</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>30</td>
<td>50</td>
</tr>
</tbody>
</table>
Attachment 2 - Example 1
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

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 IN A RECTANGULAR 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

PARTICLE MAP (P SIGNIFIES PUMP LOCATION. I SIGNIFIES LOCATION OF INJECTION)

C-96
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

ACUMULATED TIME = 0 DAYS PARTICLES = 5000 ROW X-SECTION ROW = 6

PARTICLE MAP (P SIGNIFIES PUMP LOCATION, I SIGNIFIES LOCATION OF INJECTION)

70C  I    I    I    I    I    I    I    I    I    I
63C  I    I    I    I    I    I    I    I    I    I
56C  I    I    I    I    I    I    I    I    I    I
49C  I    I    I    I    I    I    I    I    I    I
42C  I    I    I    I    I    I    I    I    I    I
35C  I    I    I    I    I    I    I    I    I    I
28C  I    I    I    I    I    I    I    I    I    I
21C  I    I    I    I    I    I    I    I    I    I
14C  I    I    I    I    I    I    I    I    I    I
7C  I    I    I    I    I    I    I    I    I    I

AA A AA A AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA A

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
I MAX = 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)

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

105C  I    I    I    I    I    I    I    I    I    I
95C  I    I    I    I    I    I    I    I    I    I
85C  I    I    I    I    3    I    I    I    I    I
76C  I    I    I    I    3    3    33    21    30    8    2    2    I
### MAP AT TIME = 10 DAYS OF TOP VIEW LAYER = 2

**CONCENTRATION MAP IN PPm (P SIGNIFIES PUMPAGE, I SIGNIFIES INJECTION)**

<table>
<thead>
<tr>
<th>Concentration</th>
<th>PPm</th>
</tr>
</thead>
<tbody>
<tr>
<td>56C</td>
<td>2</td>
</tr>
<tr>
<td>46C</td>
<td>2</td>
</tr>
<tr>
<td>36C</td>
<td></td>
</tr>
<tr>
<td>26C</td>
<td>2</td>
</tr>
<tr>
<td>16C</td>
<td></td>
</tr>
</tbody>
</table>

### CUMULATED TIME = 10 DAYS PARTICLES = 5000 TOP VIEW LAYER = 2

**CONCENTRATION MAP IN PPm (P SIGNIFIES PUMPAGE, I SIGNIFIES INJECTION)**

<table>
<thead>
<tr>
<th>Concentration</th>
<th>PPm</th>
</tr>
</thead>
<tbody>
<tr>
<td>105C</td>
<td></td>
</tr>
<tr>
<td>95C</td>
<td></td>
</tr>
<tr>
<td>85C</td>
<td></td>
</tr>
<tr>
<td>76C</td>
<td>3</td>
</tr>
<tr>
<td>66C</td>
<td>2</td>
</tr>
<tr>
<td>56C</td>
<td>2</td>
</tr>
<tr>
<td>46C</td>
<td>2</td>
</tr>
<tr>
<td>36C</td>
<td></td>
</tr>
<tr>
<td>26C</td>
<td></td>
</tr>
<tr>
<td>16C</td>
<td></td>
</tr>
</tbody>
</table>

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

**LEFT-HAND CORNER OF PLOT POSITION = 0**

**GRID SIZE OF PLOT MAP, IN FEET = 10**

**NUMBER OF ROWS AND COLUMNS OF PLOT MAP = 11**

**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

C-100
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

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

\[
\begin{array}{cccccccccccccccc}
142 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
128 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
114 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
100 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
85 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
71 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
57 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
42 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
28 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
14 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
\end{array}
\]

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

C-101
### Particle Map

**Accumulated Time = 0 Days Particles = 2500**

*Particle Map* (P signifies pump location, I signifies location of injection)

<table>
<thead>
<tr>
<th>142°C</th>
<th>128°C</th>
<th>114°C</th>
<th>100°C</th>
<th>85°C</th>
<th>71°C</th>
<th>57°C</th>
<th>42°C</th>
<th>28°C</th>
<th>14°C</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>128°C</th>
<th>114°C</th>
<th>100°C</th>
<th>85°C</th>
<th>71°C</th>
<th>57°C</th>
<th>42°C</th>
<th>28°C</th>
<th>14°C</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>142°C</th>
<th>128°C</th>
<th>114°C</th>
<th>100°C</th>
<th>85°C</th>
<th>71°C</th>
<th>57°C</th>
<th>42°C</th>
<th>28°C</th>
<th>14°C</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**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 BOTCAPTURE - BOTTOM ELEVATION AT WHICH PARTICLE MAY BE
REM CAPTURED BY SINK
REM CAPTUREMAX - RIVER MAXIMUM 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 FS - MAP FLAG, IF="A" THEN MAP HAS BEEN RUN
REM G$ - FLAG INDICATING COEFFICIENTS HAVE BEEN ENTERED IF
REM ="B"
REM 11! - X COORDINATE OF LOWER LEFT CORNER OF SCREEN WINDOW
REM (FT)
REM 12! - 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 SCREEN
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 JROWS - 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 MAPTYPES - FLAG INDICATING WHAT VIEW OF THE PROBLEM WAS LAST
REM DISPLAYED ON THE SCREEN, A - TOP VIEW, B - ROW 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 CONCENTRATIONS
REM NP - NUMBER OF PARTICLES
REM NMAP(I,J) - NUMBER OF PARTICLES
REM NMAP(I,J) - NUMBER OF PARTICLES
REM NS - NUMBER OF SINKS
REM NSPF - NUMBER OF SPECIAL FEATURE FILES (WITH KEYS AND FILE 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 R8 - NAME OF VELOCITY FILE (WITH EXTENSION .RND)
REM R9 - 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 (INCHES) (FT)
REM SCREEN - 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 SPPF(I,J,K) - FILE NAMES OF SPECIAL FEATURE FILES
REM SPFKEYS(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
REM SPFSAVE - NAME OF FILE WITH SPECIAL FEATURE DATA
REM SVIEW - X DIMENSION OF ZOOM BOX
REM SWITCH% - ARRAY OF FLAGS INDICATING WHETHER IT IS A SLUG OR CONTINUOUS PARTICLE, 1= CONTINUOUS, 0=SLUG
REM T2 - TIME (DAYS)
REM TEMPSCR - FLAG, IF=1 IT INDICATES A SCREEN IMAGE HAS BEEN 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 LAYER THEN =0 UNLESS A RIVER NODE, THEN = BOTTOM OF RIVER ELEVATION (FT)
REM TOPCAPTURE - TOP ELEVATION AT WHICH PARTICLE MAY 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) - VELOCITY 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 VELOCITY (FT/DAY)
REM WD - COLUMNS OF MAP
REM X(I) - ARRAY OF X COORDINATES FOR PARTICLES (FT)
REM XI(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 Y(I) - ARRAY OF Y COORDINATES FOR PARTICLES (FT)
REM YI(L) - Y COORDINATE OF SINK L (FT)
REM YP - Y COORDINATE OF PARTICLE POSITION IN GRID UNITS
REM Z(I) - ARRAY OF Z COORDINATES FOR PARTICLES (FT)
REM ZI(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 - MAXIMUM VERTICAL MOVE DISTANCE BEFORE RECOMPUTING VELOCITY (DAYS)
REM ZZZ - VARIABLE USED TO IMPLEMENT FIRST ORDER DECAY

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 O:WIDTH 80
330 REM NAME: THREE DIMENSIONAL RANDOM WALK (MICROCOMPUTER VERSION)
350 REM PURPOSE: TO SIMULATE 3-DIMENSIONAL, STEADY OR NONSTEADY
370 REM MASS TRANSPORT PROBLEMS IN HETEROGENEOUS AQUIFERS
380 REM WRITTEN BY: DONALD KOCH
390 REM ENGINEERING TECHNOLOGIES ASSOCIATES, INC.
392 REM 3458 ELLICOTT CENTER DRIVE
394 REM ELLICOTT CITY, MD, 21043
396 REM (301) 461-9920
398 REM BASED ON CODE WRITTEN BY
400 REM THOMAS A. PRICKETT & ASSOCIATES
402 REM URBANA, ILLINOIS 61801
404 REM (217)384-0615
406 DEFINT I,J,K
410 REM ALLOW ARRAYS FOR 10000 PARTICLES AND 99 SINKS AND 45 BY 45 VELOCITY ARRAY
415 REM $DYNAMIC
420 DIM NMAP(45,45),CONC(99),THICK(45,45,3),VI(45,45,3),VJ(45,45,3),VK(45,45,3)
425 DIM X1(99),Y1(99),Z1(99),QSUM(99),TOP(45,45,3),BOT(45,45,3)
430 DIM X(10001),Y(10001),Z(10001),SWITCH%(10001)
435 REM $STATIC
440 REM SPFKEY$(20,2),SPFFILE$(20),ENDTIME(20),RETARD(5)
445 REM SET UP MONITOR FOR COLOR OR WITHOUT COLOR.
450 REM LOCATE 8,30:PRINT "MONITOR SETUP PARAMETERS":PRINT:PRINT:PRINT
452 DIM SPFKEY$(20,2),SPFFILE$(20),ENDTIME(20),RETARD(5)
460 REM SET UP MONITOR FOR COLOR OR WITHOUT COLOR.
462 REM LOCATE 8,30:PRINT "MONITOR SETUP PARAMETERS":PRINT:PRINT:PRINT
500 REM INPUT 
"Do you have a color monitor?? (Y-Yes or <RETURN>-No) ",XX$  
XXS="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:"PRINT "  
580 LPRINT " RANDOM WALK 3-D MASS TRANSPORT MODEL WITH GEOGRAPHIC FEATURE OVERLAYS"  
600 LPRINT "  
610 LPRINT " BY"  
612 LPRINT " DONALD KOCH"  
630 LPRINT " ENGINEERING TECHNOLOGIES ASSOCIATES"  
640 LPRINT " 3458 ELLICOTT CENTER DRIVE"  
650 LPRINT " ELLICOTT CITY, MD, 21043"  
660 LPRINT " PHONE: 301-461-9920";LPRINT ""  
670 LPRINT";LPRINT ""  
680 PRINT"  
690 PRINT "  
700 PRINT "  
705 GS="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 " "  
780 PRINT " "  
790 PRINT " "  
800 PRINT " 
810 PRINT " 
820 PRINT " 
830 PRINT " 
840 PRINT " 
850 PRINT " 
860 PRINT " 
870 PRINT " 
880 PRINT " 
890 PRINT " 
900 PRINT " 
910 PRINT " 
911 PRINT " 
912 PRINT " 
920 PRINT "  
930 PRINT:BEep  
940 INPUT " COMMAND ";AS  
950 REM JUNCTION POINT FOR MENU COMMAND SELECTION.  
960 IF AS="P" OR AS="p" THEN GOSUB 1270  
970 IF AS="XP" OR AS="xp" THEN GOSUB 1180  
980 IF AS="RE" OR AS="re" THEN GOSUB 5110  
990 IF AS="XS" OR AS="xs" THEN GOSUB 1150  
1000 IF AS="WR" OR AS="wr" THEN GOSUB 5210  
1010 IF AS="XT" OR AS="xt" THEN GOSUB 1120  
1020 IF AS="PLT" OR AS="plt" THEN GOSUB 6080  
1030 IF AS="MOVE" OR AS="move" OR AS="Move" THEN GOSUB 2510  

D-4
1040 IF AS="q" OR AS="Q" THEN END
1050 IF AS="MAP" OR AS="map" OR AS="Map" THEN GOSUB 3940
1060 IF AS="SET" OR AS="set" THEN GOSUB 5480
1070 IF AS="EDIT" OR AS="edit" THEN GOSUB 5480
1080 IF AS="VIEW" OR AS="view" THEN GOSUB 7270
1090 IF AS="INPUT" OR AS="input" THEN GOSUB 9140
1092 IF AS="SPF" OR AS="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
1140 RETURN
1150 NS=0:REM ZERO SINKS
1160 PRINT
1170 RETURN
1180 FOR I = 1 TO NP:REM ZERO PARTICLES
1190 SWITCH%(I)=0
1200 NEXT I
1210 PRINT
1220 RETURN
1230 RETURN
1240 REM ++++++++++++++++++++++++++++++++++++++++
1250 REM BEGIN PARTICLE INPUT SUBROUTINE +
1260 REM ++++++++++++++++++++++++++++++++++++++++
1270 PRINT "\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\n
1340 IF AS <> "y" AND AS<>"y" THEN 2420
1350 IF AS="c" OR AS="C" THEN 2400
1360 IF AS="L" OR AS="L" THEN 1750
1370 IF AS="R" OR AS="r" THEN 1410 ELSE 1370
1380 IF AS="R" OR AS="r" THEN 1410 ELSE 1370
1390 IF AS="R" OR AS="r" THEN 1410 ELSE 1370
1400 IF AS="R" OR AS="r" THEN 1410 ELSE 1370
1410 LPRINT:REM INITIATE PARTICLES IN RECTANGULAR PRISM
1420 LPRINT "PARTICLES IN A RECTANGULAR 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 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

D-5
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 25 < BOT(IZ,JZ,1) OR 25 > (BOT(IZ,JZ,NL)+THICK(IZ,JZ,NL)) THEN PRINT "1ST Z COORDINATE OUTSIDE MODEL";GOTO 1430
IF 26 < BOT(IZ,JZ,1) OR 26 > (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>";A$
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 "DO YOU WANT TO REDO THIS SCREEN? (Y-YES, <RETURN>-NO)";OK$
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
1640 FOR I = 1 TO M
1650 X = X5 + DX*RND
1660 Y = Y5 + DY*RND
1670 Z = Z5 +DZ*RND
1680 GOSUB 2100:REM ADD A PARTICLE
1690 NEXT I
1700 PRINT:REM *********************************************
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 "END POINT OF LINE (X,Y,Z) ";X3,Y3,Z3
1830 LPRINT "END POINT OF LINE (X,Y,Z) = ";X3;",";Y3;",";Z3;" FT"
1840 IF X<= (LLX) OR X>=(LLX+DELX*(NC)) THEN PRINT "BEGINNING POINT X COORDINATE IS OUTSIDE THE MODEL GRID";GOTO 1790
1850 IF Y<=(LLY) OR Y>=(LLY+DELY*(NR)) THEN PRINT "BEGINNING POINT Y COORDINATE IS OUTSIDE THE MODEL GRID";GOTO 1790
1860 IF Z<=(LLZ) OR Z>=(LLZ+DELY*(NR)) THEN PRINT "END POINT Y COORDINATE IS OUTSIDE THE MODEL GRID";GOTO 1790
1870 IF X<=(LLX) OR X>=(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 25 < BOT(IZ,JZ,1) OR 25 > (BOT(IZ,JZ,NL)+THICK(IZ,JZ,NL)) THEN PRINT "1ST Z COORDINATE OUTSIDE MODEL";GOTO 1790

D-6
IZ = INT((X3 - LLX) / DELX + 1.0)
JZ = INT((Y3 - LLY) / DELY + 1.0)
IF Z3 < BOT(IZ, JZ, I) 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 M
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
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)
2040 Z = Z2 + (I - 2) * (Z3 - Z2) / (M - 1)
2050 GOSUB 2100: REM ADD A PARTICLE
2060 NEXT I
2070 L = L + 1
2080 GOTO 1330: REM RETURN TO BEGINNING OF PARTICLE SUBROUTINE
REM ****************************************
2090 REM SUBROUTINE TO ADD A PARTICLE
REM ****************************************
2100 NP = NP + 1
2110 X(NP) = X
2120 Y(NP) = Y
2130 Z(NP) = Z
IF A$ = "C" OR A$ = "c" THEN SWITCH%(NP) = 1 ELSE SWITCH%(NP) = 0
2140 RETURN
2150 REM END OF ADD A PARTICLE SECTION
REM ****************************************
2160 LPRINT " SYSTEM PARTICLES = "; NP
2170 L = L + 1
2180 GOTO 1330: REM RETURN TO BEGINNING OF PARTICLE SUBROUTINE
REM ****************************************
2190 REM SUBROUTINE TO ADD A PARTICLES
REM ****************************************
2200 INPUT " SPHERE NUMBER "; C + 1
2210 PRINT " SPHERE NUMBER "; C + 1
2220 INPUT " ENTER SPHERE CENTER COORDINATES (X,Y,Z) "; X4, Y4, Z4
2230 INPUT " ENTER SPHERE CENTER COORDINATES (X,Y,Z) = "; X4; " ; "; Y4; " ; "; Z4; " FT"
2240 INPUT " SPHERE CENTER COORDINATES (X,Y,Z) " = "; X4; " ; "; Y4; " ; "; Z4; " ; "; R; " FT"
2250 IF (X4 - R) < (LLX) OR (X4 + R) > (LLX + DELX * (NC)) THEN PRINT " DO NOT ENTER PARTICLES OUTSIDE FLOW MODEL GRID - BAD X": GOTO 2200
2260 IF (Y4 - R) < (LLY) OR (Y4 + R) > (LLY + DELY * (NR)) THEN PRINT " DO NOT ENTER PARTICLES OUTSIDE FLOW MODEL GRID - BAD Y": GOTO 2200
126 = INT((X4 - LLX) / DELX + 1.0)
127 = 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
2340 FOR II=1 TO M/INT(SQR(M))
2350 PHI=2*3.14159-TH
2360 X = X4+R*SIN(PHI)*COS(TH)
2370 Y = Y4+R*SIN(TH)*SIN(PHI)
2380 Z = Z4+R*COS(PHI)
2390 GOSUB 2100:REM GO TO SUBROUTINE TO INSERT A PARTICLE.
2400 NEXT II
2410 NEXT I
2420 C = C+1
PRINT " SPHERE DONE"
REM *********************************
2430 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
2440 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 - 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 - 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
NUMBER OF PARTICLES = ";M

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

TOTAL SYSTEM PARTICLES = ";NP
TOTAL SYSTEM PARTICLES = ";NP

PRESENT SIMULATION TIME (DAYS) = ";T2
PRESENT SIMULATION TIME = ";T2;" DAYS"
TIME REMAINING FOR THIS VELOCITY FILE IS ";(ENDTIME(NPER)-T2);" DAYS"
ENTER INCREMENTAL SIMULATION TIME (DAYS) [";DELTA;" ];
DELTA MUST BE GREATER THAN ZERO!!!---REDO THIS INPUT NOW PLEASE.
DELTA=ENDTIME(NPER)-T2:GOTO 2580
INCREMENTAL SIMULATION TIME = ";DELTA;" DAYS"
HOW OFTEN DO YOU WANT TO COMPUTE VELOCITY VECTORS
ENTER MAXIMUM HORIZONTAL MOVE (FT) [";DMAX;" ];
ENTER MAXIMUM VERTICAL MOVE (FT) [";ZMAX;" ];
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)=O!
2750 NEXT K
2770 K=0:REM INITIALIZE PARTICLE COUNTER
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)=I THEN D=DELTA*RND ELSE D=DELTA
2840 IF LAMDBA>=1E10 THEN GOTO 2880:REM SKIP FIRST ORDER DECAY TO SAVE TIME IF DECAY IF CONSERVATIVE POLUTION
2850 ZZZ=ZZZ+(1-.5^(D/LAMDBA/365)):REM FIRST ORDER DECAY IMPLEMENTED IN THESE STATEMENTS
2860 IF ZZZ=>1 THEN GOTO 2832 ELSE GOTO 2880
2870 X(K)=X(NP)
2880 Y(K)=Y(NP)
2890 Z(K)=Z(NP)
2900 SWITCH%(K)=SWITCH%(NP)
2910 NP=NP-1
2920 ZZZ=ZZZ-1.0
2930 IF K>NP THEN 3660
2940 REM ++++ ***************************************************
2950 REM WHEN ALL PARTICLES ARE PROCESSED, THIS ++
2960 REM NEXT STATEMENT RETURNS YOU TO THE MENU ++
2970 REM ******************************************************
2980 IF K>NP THEN T2=T2+DELTA:GOTO 3660
2990 XP=(X(K)-LLX)/DELX+.5
3000 YP=(Y(K)-LLY)/DELY+.5
3010 VX=.00001
3020 VY=.00001
3030 VZ=1E-10
3040 I=INT(XP)
3050 J=INT(YP)
3060 IZ=INT(XP+.5)
3070 JZ=INT(YP+.5)
3080 FOR KLK=1 TO NL-1:REM FIND LAYER OF PARTICLE AND INTERPOLATE THE VERTICAL VELOCITY
3090 IF Z(K)>=BOT(IJ,JZ,KLK) AND Z(K)<TOP(IJ,JZ,KLK) THEN
3100 LL=KLK
3110 AZ=(Z(K)-BOT(IJ,JZ,KLK))/(TOP(IJ,JZ,KLK)-BOT(IJ,JZ,KLK))
3120 V3=(AZ*VK(IJ,JZ,KLK)+(1-AZ)*VK(IJ,JZ,KLK-1))/E
3130 GOTO 2948
3140 END IF

D-10
IF $Z(K) > TCP(IZ,JZ,KLK)$ AND $Z(K) < BOT(IZ,JZ,KLK+1)$ THEN
   LL=KLK+NL
   $V_3 = VK(IZ,JZ,KLK)/E$
   GOTO 2948
END IF

NEXT KLK
IF $Z(K) >= BOT(IZ,JZ,NL)$ THEN
   LL=NL
   $AZ = (Z(K) - BOT(IZ,JZ,NL))/THICK(IZ,JZ,NL)$
   $V_3 = (AZ*VK(IZ,JZ,NL) + (1-AZ)*VK(IZ,JZ,NL-1))/E$
END IF

2948 IF LL=NL THEN $V_1 = 0; V_2 = 0$: GOTO 2990:REM IF IN CONFINING LAYER, THEN SKIP HORIZONTAL INTERPOLATION
2950 AX=XP-INT(XP)
2960 AY=YP-INT(YP)
2970 $V_1 = ((1-AY)*VI(I,J,LL)+AY*VI(I,J+1,LL))/E$:REM HORIZONTAL VELOCITY INTERPOLATION
2980 $V_2 = ((1-AX)*VJ(I,J,LL)+AX*VJ(I+1,J,LL))/E$
2990 VX=VX+$V_1$
3000 VY=VY+$V_2$
3040 IF $F = D-ABS(DMAX/VX)) < 0$ THEN 3070
3050 $F = ABS(DMAX/VX)$
3060 GOTO 3080
3070 $F = D$
3080 GOTO 3092
3090 $F = ABS(DMAX/VY)$
3092 IF $F = ABS(ZMAX/VZ)) < 0$ THEN 3100
3100 D=D-$F$:REM CALCULATE REMAINING TIME IN MOVE
3110 DX=VX*$F$
3120 DY=VY*$F$
3130 DZ=VZ*$F$
3135 PHI=ATN(DY/DX)
3140 DD=SQR(DX*DX+DY*DY)
3145 DDX=SQR(DX*DX+DY*DY+DZ*DZ)
3150 RN=0
3210 GOTO 3280
3220 GOTO 3280
REM *******************************************************
REM NORMALLY DISTRIBUTED RANDOM NUMBER GENERATOR
REM *******************************************************
3230 RN=-6
3240 FOR UX=1 TO 12
3250 RN=RN+RND
3260 NEXT UX
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*D*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<=I! 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 MOVES 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
HAPPEN 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)=O AND VJ(III,JJJ,LL)=O AND VI(III,JJJ+1,LL)=O AND VJ(III+1,JJJ,LL)=O 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%=I TO NS
3520 II=INT((XI(L%)-LLX)/DELX+1.0):REM FIND ROW AND COLUMN OF SINK
3525 JJ=INT((YI(L%)-LLY)/DELY+1.0)
KK=ZI(L%):REM FIND LAYER OF SINK
3540 IF X(K)=XI(L%) AND Y(K)=YI(L%) AND Z(K)<TOP(II,JJ,KK) AND Z(K)>=BOT(II,JJ,KK) THEN GOTO
3560 REM REMOVE PARTICLE
3530 R3=SQR((XI(L%)-X1(L%))*(XI(L%)-X1(L%))+(YI(L%)-Y1(L%))*(YI(L%)-Y1(L%)))
3550 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=OSUM(LK)**F/DELY/DLX/7.48/E
3565 CAPTUREMAX=SQR(DELX*DELY)

D-12
IF BOTCAPTURE<BO(I,J,J,NL) THEN BOTCAPTURE=BO(I,J,J,NL)
END IF

3566 IF QSUM(LX)<=0 THEN GOTO 3570:REM SKIP CAPTURE IF INJECTION WELL
3567 IF R3<SQR(QSUM(LX)*DELTA/(3.141592*THICK(I,J,K,K)*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>O 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
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
3679 FOR I= 1 TO NS:REM SET ALL PARTICLES TO SLUG TYPE AT END OF MOVE
3680 SWITCH%(I)=O
3690 NEXT I
3700 FOR I=1 TO NS:REM PRINT OUT OF CONCENTRATION IN PUMPED WELL NUMBER ";; IN PPM, IS=
3701 IF QSUM(I)>O AND CONC(I)>O THEN LPRINT "CONCENTRATION IN PUMPED WELL NUMBER ",; IN LBS, IS=
3702 IF QSUM(I)>O AND CONC(I)>O THEN LPRINT "MASS EXITING IN PUMPED WELL NUMBER ";; IN LBS/119872
3710 NEXT I
3720 LPRINT:LPRINT:LPRINT:LPRINT
3730 LOCATE 1,1
3740 WS=INKEY$:REM END OF MOVE SUBROUTINE, KEEP SCREEN UNTIL KEY IS Pressed
3750 IF WS="y" OR WS="n" THEN GOSUB 7110:REM SAVE SCREEN
3760 IF WS="" THEN 3780
3770 FOR I=1 TO NS:REM SET ALL PARTICLES TO SLUG TYPE AT END OF MOVE
3780 SWITCH%(I)=O
3790 NEXT I
3800 SCREEN 2
3810 PRINT:PRINT:SCREEN 0:IF XX$="y" OR XX$="n" THEN COLOR 15,1,1 :CLS ELSE CLS
3820 REM + BUG OUT TO MAIN MENU AS THIS IS THE END OF SUBROUTINE MOVE.
3830 REM + ++g++sasti++++++++++
3840 FS="a":REM SET MAP FLAG
3850 LN = 10:REM FIXED NUMBER OF ROWS IN SCREEN MAP, SCREEN MAPPED 14 BY 10 TO MATCH TYPICAL SCREEN SIZE
3860 WD = 14:REM FIXED NUMBER OF COLUMNS IN SCREEN MAP
3870 IF ASPECT<=0 THEN PRINT "YOU DID NOT ENTER AN ASPECT RATIO FOR THE SCREEN ":GOSUB 13100:RETURN
3880 IF (SCALEDEF<=0 AND TEMPSCR=0) OR (SCALEX<=0 AND TEMPSCR=1) THEN PRINT "YOU DID NOT ENTER A DEFAULT SCREEN WIDTH ":GOSUB 13100:RETURN
3890 IF SCALEX<=0 AND TEMPSCR=1 THEN PRINT "YOU DID NOT ENTER A SCREEN WIDTH ":GOSUB 13100:RETURN
3900 IF SCALEDEF<=0 AND TEMPSCR=0 THEN PRINT "YOU DID NOT ENTER A SCREEN WIDTH ":GOSUB 13100:RETURN
3910 RETURN
3920 REM + MAP PARTICLES OR
3930 REM + CONCENTRATION
3940 REM ++++++++++++++++++++++++:
3950 IF XX$="y" OR XX$="n" THEN COLOR 15,1,1 :CLS ELSE CLS
3960 GO SUB 7520:REM GRAPHICS ROUTINE TO DEFINE SCREEN COORDINATES
3970 CDX=(12-I1!)/WD:REM DEFINE GRID SPACING
3980 CDY=(J2!-J1!)/LN:REM DEFINE GRID SPACING
3990 FOR I=1 TO NS:REM PRINT OUT OF CONCENTRATION IN PUMPED WELL NUMBER ";; IN LBS/119872
4000 NEXT I
4010 GOSUB 7520:REM GRAPHICS ROUTINE TO DEFINE SCREEN COORDINATES
4020 CDX=(12-J1!)/WD:REM DEFINE GRID SPACING
4030 CDY=(12-J1!)/LN:REM DEFINE GRID SPACING
4040 SCREEN 0:WIDTH 80:IF XX$="y" OR XX$="n" THEN COLOR 15,1,1:CLS ELSE CLS
4050 PRINT:PRINT:PRINT
4070 PRINT "WHICH TYPE OF MAP DO YOU WANT?:"
4080 PRINT SPC(5);"ENTER P FOR PARTICLE MAP"
4090 PRINT SPC(5);"ENTER C FOR CONCENTRATION MAP"
4100 PRINT
4110 INPUT "WHICH ONE.......";CS
4120 IF CS <> "C" AND CS <> "c" THEN 4170
4130 INPUT "ENTER SCALE FACTOR FOR CONCENTRATION MAP (T-PPT, M-PPM, B-PPB)";CCS
4140 IF CCS = "T" OR CCS = "t" THEN PPP = .001
4150 IF CCS = "M" OR CCS = "m" THEN PPP = 1
4160 IF CCS = "B" OR CCS = "b" THEN PPP = 1000
4170 IF MAPTYPES = "A" THEN
  PRINT "ENTER LAYER OF MODEL TO MAP, FOR ALL PARTICLES IN ALL LAYERS VISIBLE,"
  INPUT "ENTER ZERO ";LAYER
END IF
IF MAPTYPES = "B" THEN
  PRINT "DO YOU WANT ALL PARTICLES VISIBLE OR ONLY THOSE IN THE"
  INPUT "SELECTED ROW (A/S) ";Z1S
  IF Z1S = "A" OR Z1S = "a" THEN LAYER = 0 ELSE LAYER = JROWXS
END IF
IF MAPTYPES = "C" THEN
  PRINT "DO YOU WANT ALL PARTICLES VISIBLE OR ONLY THOSE IN THE"
  INPUT "SELECTED COLUMN (A/S) ";Z1S
  IF Z1S = "A" OR Z1S = "a" THEN LAYER = 0 ELSE LAYER = ICOLXS
END IF
4172 GOSUB 4220:REM PRINT THE MAP
4180 INPUT "DO A SCREEN PRINT NOW OR PRESS <RETURN> TO GO BACK TO THE MENU.";A$  
4190 PRINT "WHICH ONE.......";CS
4200 REM ++++++++4210 RETURN
4220 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
END IF
IF SWITCH%(MM%) = 1 THEN GOTO 4370:REM SKIP PARTICLE IF NOT CONTINUOUS
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<>O 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>W0 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$="A" THEN
SCREENX=XI(IIS)
SCREENY=YI(IIS)
IF LAYER<>ZI(IIS) AND LAYER<>O THEN GOTO 4460:REM SKIP THIS SINK IF DOING LAYER MAP
END IF
IF MAPTYPE$="B" THEN
SCREENX=XI(IIS)
IZ=INT((XI(IIS)-LLX)/DELX+1.0):REM FIN COLUMN OF WELL
JZ=INT((YI(IIS)-LLY)/DELY+1.0):REM FIND ROW OF WELL
SCREENY=BOT(IZ,JZ,INT(ZI(IIS)))
IF LAYER<>O AND LAYER<>JZ THEN GOTO 4460:REM SKIP THIS SINK IF NOT IN ROW
END IF
IF MAPTYPE$="C" THEN
SCREENX=YI(IIS)
IZ=INT((XI(IIS)-LLX)/DELX+1.0):REM FIN COLUMN OF WELL
JZ=INT((YI(IIS)-LLY)/DELY+1.0):REM FIND ROW OF WELL
SCREENY=BOT(IZ,JZ,INT(ZI(IIS)))
IF LAYER<>O 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>W0 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 CS<> "C" AND CS<>"c" THEN GOTO 4600
4480 FOR I = 1 TO W0: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 MAPTYPES="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+(TOP(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-1)-
        TOP(II, JJ, LAYER-NL)
    END IF
    END IF
    IF MAPTYPES="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<TOP(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 MAPTYPES="C" THEN
        II=INT((XX-LLX)/DELY+1.0)
        IF II<1 OR II>NR THEN GOTO 4580
        FOR K=1 TO NL-1
            IF YY>=BOT(II, ICOLXS, II, K) AND YY<TOP(II, ICOLXS, II, K) THEN RD=RETARD(K)
            IF YY>=TOP(II, ICOLXS, II, K) AND YY<TOP(II, ICOLXS, II, K+1) THEN RD=RETARD(K+NL)
        NEXT K
        IF YY>BOT(II, ICOLXS, II, NL) THEN RD=RETARD(NL)
        IF LAYER=0 THEN TEMPTHCK=NR*DELY*RD ELSE TEMPTHCK=DELY
    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"; PP; PARTICLES= ";NP;
        LPRINT " MAP AT TIME = ";T2;" DAYS OF ";
        IF MAPTYPES="A" THEN PRINT " TOP VIEW LAYER=";LPRINT "TOP VIEW LAYER=";
        IF MAPTYPES="B" THEN PRINT " ROW XSECTION ROW=";LPRINT " ROW XSECTION ROW=";
        IF MAPTYPES="C" THEN PRINT " COL XSECTION COL=";LPRINT " COL XSECTION COL=";
        IF LAYER=0 THEN PRINT "ALL";LPRINT "ALL" ELSE PRINT LAYER;LPRINT LAYER
4610 IF CS <> "C" AND CS<>"C" THEN PRINT "PARTICLE MAP (P SIGNIFIES PUMP LOCATION. I SIGNIFIES LOCATION OF INJECTION)"
4620 IF CS = "P" OR CS="p" THEN 4670
4630 PRINT " CONCENTRATION MAP IN PP";RIGHTS(CCS,1);"" (P SIGNIFIES PUMPAGE, I SIGNIFIES INJECTION)"
        LPRINT " CONCENTRATION MAP IN PP";RIGHTS(CCS,1);"" (P SIGNIFIES PUMPAGE, I SIGNIFIES INJECTION)"
4670 PRINT 
4720 YYY = J2!
4730 FOR J = 1 TO LN
4740 MS = STR$(8-LEN(STR$(INT(YYY)))) + STR$(INT(YYY))
4750 PRINT MS;"=
4760 FOR I = 1 TO WD
4770 IF NMAP(I,J) = -1 THEN PRINT " P";GOTO 4840
4780 IF NMAP(I,J) = -2 THEN PRINT " I";GOTO 4840
4790 IF ABS(NMAP(I,J))<.5 THEN PRINT " *";GOTO 4840
4800 MS = STR$(INT(NMAP(I,J)+.5))
4810 MS = RIGHT$(MS,LEN(MS)-1)
4820 IF LEN(MS) < 4 THEN MS = SPACES(4-LEN(MS)) + MS
4830 PRINT LEFTS(MS,4);"=
4840 NEXT I
4850 PRINT "=
4860 YYY = YYY - CDY
4870 NEXT J
4880 PRINT "=
4890 FOR I = 1 TO WD
4900 PRINT "---";
4910 NEXT I
4920 PRINT;PRINT
4930 FOR J = 1 TO 7
4940 PRINT "=
4950 XXX = I!
4960 FOR I = 1 TO WD
4970 MS = STR$(INT(XXX)) + 
4980 MS = MIDS(MS,2,J)
4990 MS = RIGHTS(MS,1)
5000 PRINT "=
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
5110 REM Subroutine to read plume data from external file
5120 PRINT
5130 PRINT "Enter filename, including extension (example PEORIA.DAT) ";FILES;o INPUT TEMPS
5140 IF LEN(TEMPS) > 0 THEN FILES = TEMPS
5150 OPEN "I",#1,FILES
5160 LINE INPUT #1,A$
5170 LINE INPUT #1,RS:REM INPUT NAME OF FLOW DATA FILE
5175 LINE INPUT #1,SPFSAVES:REM INPUT NAME OF SPF FILE
5180 INPUT #1,T2,E,DL,DT,DV,PM,LMOB,L,111,111,CDX,CDY,HP,DP,LS,L,32,0,SCALEX,NL,TEMPSR,MAPTYPE,JROW,ICOL
5190 FOR 1 = 1 TO NL+(NL-1):REM READ IN RETARDATION COEFFICIENTS
5200 INPUT#1,RETARD(1)
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$)>0 THEN GOSUB 12290:REM OPEN AND READ SPECIAL FEATURE FILE
5250 GS="B"
5260 FS="A"
5270 RETURN
REM Subroutine to write data to external file
REM *****************************************************
5280 INPUT "Enter the name of the external file to be opened, including ext. ";FILES
5300 OPEN "O",1,FILES
LPRINT
LPRINT "WRITING PLUME DATA TO EXTERNAL FILE ";FILES
LPRINT
5310 PRINT #1,"name:";FILE$
5320 PRINT #1,R$
5325 PRINT #1,SPFSAVE$:REM WRITE NAME OF SPECIAL FEATURES FILE
5340 PRINT #1,'T2;E;DL;DT;DV;PM;LAMDBA;I1;J1;CDX;CDY;NP;I2;J2;O;SCALEX;NL;TEMPSCR;CHR$(34);MAPTYPE$;CHR$(34);JROWXS;ICOLXS
FOR I=1 TO NL+(NL-1):REM WRITE RETARDATION COEFFICIENTS
PRINT#1,RETARD(I); NEX'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 *****************************************************
5430 LPRINT " ERR =";ERR;" ERL # =";ERL
IF ERR=6 THEN RESUME:REM THIS STATEMENT TRAPS A MYSTERIOUS ERROR CONDITION AND THE PROGRAM CONTINUES
5450 PRINT "HEY, YOU'VE DONE SOMETHING WRONG! YOU THINK ABOUT WHAT HAPPENED. TRYING TO GO TO THE MAIN MENU NOW.";PRINT
5460 FOR I = 1 TO 6000:NEXT I
5470 GOTO 760
REM *****************************************************
5480 REM ROUTINE FOR SETTING COEFFICIENTS
REM *****************************************************
5490 PRINT:G$="B":REM SET COEFFICIENT FLAG
5500 PRINT "\\\\\\\"BASIC TRANSPORT COEFFICIENTS\\\\\\\"
5510 PRINT " ENTER POROSITY [";E;"] ";INPUT TEMPS:IF LEN(TEMP$)>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$)>0 THEN DV=VAL(TEMP$)
5640 PRINT ""
5660 RETURN
REM CREATE PLOT FILE FOR GOLDEN SOFTWARE SURFER
6080 CLS
6090 IF FS<>"A" THEN PRINT "HEY, YOU HAVEN'T GONE THROUGH MAP--DO IT NOW!!!" ; GOSUB 13100 ; RETURN
6100 IF GS<>"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 " ENTER LOWER LEFT CORNER COORDINATES OF PLOT MAP (X,Y) " ; LOWERLX, LOWERLY
6230 PRINT " ENTER 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 FILES=CODE$+RIGHT$(STR$(K),LEN(STR$(K))-1)+".DAT"
6320 LPRINT " Plot file name is = " ; FILES
6330 OPEN "O" ; #1 , FILES
6340 FOR I = 1 TO COLUMNS ; REM INITIALIZE NMAP

D-19
FOR J = 1 TO ROWS
    NMAP(I,J)=0
NEXT J

FOR MM% = 1 TO NP:REM PARTICLE LOOP
    IF SWITCHX(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=I 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+I) THEN LL=KLK+NL:GOTO 6395
    NEXT KLK
    IF LL<>K THEN 6400:REM SKIP IF IN WRONG LAYER
    NMAP(I,J)=NMAP(I,J)+1:REM INCREMENT GRID PARTICLE COUNTER
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)
    NEXT J
NEXT I
CLOSE #1
NEXT K
RETURN

REM +++++++++++++++++++++++++++.
REM + THIS IS THE SUBROUTINE TO SAVE +
REM + THE PLUME POSITION FOR A SLIDE +
REM +++++++++++++++++++++++++++++++++++.

DEF SEG = &HB800
LS = STR$(O)
LS = RIGHTS(LS,LEN(LS)-1)
FILES = "SLIDE" + LS + ".DAT":REM MAKE SLIDE NAME
BSAVE FILES,0, &H4000
REM CALL BSAVEA(FILES, &HBSOO, 0, &H4000, ERRCODE)
O = O+1:REM INCREMENT SLIDE COUNTER
RETURN
REM ++++++++++++++++++++++++
REM + THIS IS THE SUBROUTINE TO DISPLAY +
REM + THE SAVED SCREENS OF THE SLIDESHOW +
REM ++++++++++++++++++++++++

PRINT #1, XX, YY, NMAP(I,J)
PRINT "Which mode do you want to operate in?"
PRINT "FORWARD in time? Or BACKWARD in time?"
 INPUT "Enter the mode that you want. F - FORWARD or B - BACKWARD";HHH$
IF HHHS = "f" OR HHHS = "F" OR HHHS = "b" OR HHHS = "B" THEN 7390 ELSE 7370
REM
SCREEN 1:IF XXS="y" OR XXS="Y" THEN COLOR 1,1:CLS ELSE CLS
FOR I = 0 TO 0-1
IS = STRS(I)
IS = RIGHTS(IS,LEN(IS)-1)
FILES = "SLIDE" + IS + ".DAT"
BLOAD FILES,O
KS=INKEY$
IF KS="" THEN GOTO 7425
NEXT I
GOTO 7500
FOR I = 0-1 TO 0 STEP -1
IS = STRS(I)
IS = RIGHTS(IS,LEN(IS)-1)
FILES = "SLIDE" + IS + ".DAT"
BLOAD FILES,O
KS=INKEY$
NEXT I
RETURN
********************************************************************
THIS SUBROUTINE SELECTS THE AREA FOR PLOTTING, THE USER MAY ZOOM IN TO
ANY AREA AND DISPLAY ANY SPECIAL FEATURE FILE
by Don Koch, May 1986, July 1986, DECEMBER 1988
*********************************************************************
IF MAPTYPES="B" THEN GOSUB 14115:IF Z1S="g" OR Z1S="G" THEN RETURN:REM PROFILE ROW ON SCREEN
IF MAPTYPES="C" THEN GOSUB 14615:IF Z1S="g" OR Z1S="G" THEN RETURN:REM PROFILE COLUMN ON SCREEN
IF TEMPSCR=0 OR (11!=0 AND SCALEX<=0) THEN GOSUB 8510:REM SET WINDOW TO DEFAULT COORDINATES
YASPECT=ASPECT:GOSUB 7770:REM call window definition subroutine
GOSUB 8110:REM DEFINE ZOOM BOX AREA
IF TEMPSCR=1 THEN GOSUB 8700:REM RESTORE LAST SCREEN
Z2$="":Z1$="":REM SET STRING OF COMMANDS TO NULL
MAPTYPES="A":REM INITIALIZE MAP TYPE TO PLAN VIEW
IF Z1S=.,N.. OR Z1S=".." THEN GOSUB 8510:GOSUB 7770:Z2$="":REM return to settings at beginning of subroutine
IF Z1S="D" OR Z1S="d" THEN GOSUB 12600:REM DRAW MODEL GRID ON SCREEN
IF Z1S="X" OR Z1S="x" THEN MAPTYPES="B":GOSUB 15000:GOSUB 14000:IF Z1S="G" OR Z1S="g" THEN RETURN ELSE GOTO 7545:REM PROFILE ROW
7738 IF Z1$="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 Z15=""
7760 GOTO 7610
7765
7820 CLS
7830 PRINT
7850 12!=11!+SCALEX:REM COMPUTE UPPER LIMITS OF VIEWING AREA
7860 J2!=JI +SCALEX/YASPECT
7880 IF 12!<=I1! THEN PRINT " Bad x coordinates":GOSUB 8060:GOTO 7820
7890 IF J2!<=JL! THEN PRINT " Bad y coordinates":GOSUB 8060:GOTO 7820
7900 G1=0:G2=0:G3=319:G4=-199
7910 REM G1=0:G2=0:G3=639:G4=349:REM FOR SCREEN MODE 9 WITH EGA CARD
7920 KEY OFF
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 *************************************************************
8000 IREN=3:REM SET DEFAULT PEN COLOR
8005 IF XX$="Y" OR XX$="y" THEN MAXPEN=3 ELSE MAXPEN=1:REM SET MAXIMUM PEN COLOR
8010 OPEN "I",1,SPFFILES(I)
8015 X$=INPUTS(1,#1):REM READ FILE CHARACTERS ONE AT A TIME
8017 IF EOF(1) THEN CLOSE 1:RETURN
8020 IF X$=" " THEN 8015
8025 IF INSTR("MSPAR",X$)>O THEN ACTIONS=ACTIONS+X$:REM IF ACTION VERB ADD TO ACTION STRING
8030 IF INSTR("-O123456789.",X$)>O THEN 8015 ELSE SNUMS=SNUMS+X$:GOTO 8015:REM IF A COORDINATE ADD TO COORDINATE STRING AND STOP CHECKING FOR ACTION VERBS
8035 SNUMS=X$
8040 IF X$="I",#1):REM READ FILE CHARACTERS ONE AT A TIME
8045 IF X$="CRHS(13) THEN 8050 ELSE SNUMS=SNUMS+X$:GOTO 8040:REM IF END OF LINE THEN EXECUTE COMMAND
8050 IF ACTIONS="SP" THEN IPEN=VAL(SN1MS$):GOTO 8005:REM CHANGE PEN
8055 NTMP=INSTR(SN1MS$,"","):REM DECODE THE CARTESIAN COORDINATES
8060 IF NTMP=-1 THEN NTMP=INSTR(SN1MS$,""")
8065 IF NTMP=-1 THEN 8005
8070 X=VAL(LEFTS(SN1MS$,NTMP-1))
8075 Y=VAL(RIGHTS(SN1MS$,LEN(SN1MS$)-NTMP))
8076 IF IPEN>MAXPEN THEN IPEN=MAXPEN:REM CHECK FOR VALID PEN COLOR
8077 IF ACTIONS="MA" THEN PSET (X,Y),IPEN:REM EXECUTE THE ACTION
8080 IF ACTIONS="MR" THEN PSET STEP (X,Y),IPEN
8082 IF ACTIONS="PA" THEN LINE -(X,Y),IPEN
8085 IF ACTIONS="PR" THEN LINE -STEP (X,Y),IPEN

D-22
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=(I2I-I11)/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 I1I=X1V
8220 J1!=Y1V
8230 J2!=Y2V
8240 I2I=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 I11=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)),(I2I-I11)/180,3,,,(ASPECT*(J2I-J11)/(I2I-I11))
8620 NEXT I
8630 RETURN
8638 REM **************************************************
8640 REM TEMP SAVE OF SCREEN IMAGE
8650 REM ####################################################################
8660 DEF SEG=&H800:REM SET SEG TO VIDEO MEM
8670 BSAVE "TEMP.SCR",0,&H4000
8675 REM CALL BSAVEA("TEMP.SCR",&H800,0,&H4000,ERRCODE):REM USE THIS CALL WITH BLBS.OBJ AND QB 1.02
8680 TEMPSRC=1:REM SET FLAG FOR SAVED VARIABLE
8685 Z3S=Z3S+Z2S: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=&H800
8730 BLOAD "TEMP.SCR",0
8735 REM CALL BLOADA("TEMP.SCR",&H800,0,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 ZIS=INKEYS
8780 LINE (XIV,YIV)-(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$=CHR$(O)+CHR$(72) THEN LINE (X1V,YIV)-(X2V,Y2V),O,B:YIV=YIV+(12!-I1I!)/50:GOSUB 8890:REM MOVE LL CORNER OF ZOOM BOX
8810 IF Z1$=CHR$(O)+CHR$(80) THEN LINE (XIV,YIV)-(X2V,Y2V),O,B:YIV=YIV-(12!-I1I!)/50:GOSUB 8890
8820 IF Z1S=CHR$(O)+CHRS(75) THEN LINE (XIV,YIV)-(X2V,Y2V),O,B:XIV=XIV-(12!-I1I!)/50:GOSUB 8890
8830 IF Z1$=CHR$(O)+CHRS(77) THEN LINE (XIV,YIV)-(X2V,Y2V),O,B:XIV=XIV+(12!-I1I!)/50:GOSUB 8890
8840 IF Z1$="e" OR Z1$="E" THEN LINE (XIV,YIV)-(X2V,Y2V),O,B:SVIEWX=SVIEWX+(12!-I1I!)/50:GOSUB 8890:REM EXPAND ZOOM BOX
8850 IF Z1$="S" OR Z1$="s" THEN LINE (XIV,YIV)-(X2V,Y2V),O,B:SVIEWX=SVIEWX-(12!-I1I!)/50:GOSUB 8890:REM SHRINK ZOOM BOX
8860 IF Z1$="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$="r" THEN GOSUB 10300:REM REFRESH SCREEN
8870 GOTO 8770
8880 RETURN:REM end of zoom routine
8890 GOTO D-24
8900 IF SVIEWX<(12I-11I)/1000 THEN SVIEWX=(12I-11I)/1000
8910 X2V=X1V+SVIEWX:REM Resize the zoom box
8920 Y2V=Y1V+SVIEWX/ASPECT
8930 LINE (XIV,YIV)-(X2V,Y2V),1,B:REM DRAW ZOOM BOX
8940 RETURN
8950 REM ####################################################################
8960 REM ZOOM ROUTINE MENU
8970 REM ####################################################################
8980 CLS
8990 PRINT
9000 PRINT TAB(10)"Menu"
9010 PRINT
9020 PRINT TAB(2)"s - shrink the zoom box"
9030 PRINT TAB(2)"e - expand the zoom box"
9040 PRINT TAB(2)"use arrow keys to move the lower left"
9050 PRINT TAB(2)"corner of the zoom box"
9060 PRINT TAB(2)"m - this menu"
9062 PRINT TAB(2)"r - refresh screen"
9070 PRINT TAB(2)"return to perform zoom"
9080 PRINT TAB(2)
9090 PRINT   
9100 INPUT Z4$
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) ",FILES
9230 IF LEN(FILE$)>8 THEN PRINT "FILE NAME MUST BE LESS THAN 9 CHARACTERS";GOTO 9220
9240 CLS
9250 RS$=FILES".RND"    
9260 PRINT "NOW PROCESSING FILE ";RS$
9270 OPEN "I",#1,RS$     
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 #1,11,JJ,KK,VTHICK,WI,WJ,WK,TBOT,TTOP
9341 BOT(II,JJ,KK)=TBOT:TOP(II,JJ,KK)=TTOP
9350 NEXT K
9360 NEXT J
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
9460 IF EOF(1) THEN NS=I:GOTO 9500:REM IF NO SINKS THEN END ROUTINE
9470 IF EOF(1)=1 THEN PRINT INPUT SINKS AS COORDINATES, AND DISCHARGE (GPD)
9480 INPUT #1,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-
9530   REM TIMESTEP
9540   REM IF T2>=CTIME OR T2<CTIME-TIMESTEP THEN PRINT "STARTING TIME MUST BE LESS THAN ";CTIME;" AND
9550   REM GREATER THAN ";CTIME-TIMESTEP:GOTO 21030
9560   REM LPRINT:LPRINT
9570   REM LPRINT:LPRINT
9580   REM IF ANSS<>"Y" AND ANSS<>"N" AND ANSS<>"Y" AND ANSS<>"y" THEN PRINT "(Y-Yes, N-No)";GOTO
9590   REM 21080
9600   REM IF ANSS="Y" OR ANSS="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 ZlS=SPFKEYS(I,1) OR Zl$=SPFKEYS(I,2) THEN GOSUB 7970
10150 NEXT I
10160 IF ZlS="D" OR Z$l="d" THEN GOSUB 12610:REM RESTORE GRID
10170 IF ZlS="W" OR Z$l="w" THEN GOSUB 8580:REM RESTORE WELLS
10250 RETURN
10300  REM *********************************************************
10310  REM SUBROUTINE THAT WRITES SPECIAL FEATURES TO SCREEN AFTER ZOOM
10320  REM *********************************************************
10330  IZ2=LEN(Z3$)
10340  FOR K=1 TO IZ2
10350  Z1S=MIDS(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 SPECIAL FEATURE FILES ARE ASCII SEQUENTIAL FILES OF CARTESIAN
11040  REM COORDINATES THAT DESCRIBE SPECIAL GEOGRAPHIC FEATURES THAT
11050  REM THE USER WISHES TO HAVE SUPERIMPOSED ON THE SCREEN
11060  REM THIS SUBROUTINE CONTROLS THE MENU THAT THE USER USES TO DEFINE
11070  REM SPECIAL FEATURES FILES AND ASSIGN KEYS TO RETRIEVE THE FILES

D-26
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 VIEWING WINDOW
11340 REM SCALEDEF - HORIZONTAL WIDTH OF DEFAULT VIEWING WINDOW
11350 REM **********************************************************
11360 CLS
11362 TEMPS=0:REM DO NOT RESTORE LAST SCREEN SAVED AFTER RESETTING DEFAULTS
11370 PRINT
11380 IF I1DEF!=O AND SCALEDEF=O 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 TEMPS
11410 IF LEN(TEMPS)>O THEN NTEMP=INSTR(TEMPS","):IF NTEMP=0 THEN GOSUB 11490:GOTO 11390:ELSE I1DEF!=VAL(LEFTS(TEMPS,NTEMP-1)):J1DEF!=VAL(RIGHTS(TEMPS,LEN(TEMP$)-NTEMP))
11420 PRINT " ENTER HORIZONTAL WIDTH OF DEFAULT VIEW WINDOW ";"[";SCALEDEF;"] ";
11430 INPUT TEMPS
11440 IF LEN(TEMPS)>O THEN SCALEDEF=VAL(TEMPS)
11444 IF SCALEDEF<=O 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 " USE COMMA TO SEPARATE COORDINATE PAIR"
11500 PRINT
11510 PRINT
11520 RETURN
11530 REM ******************************************************************************************************
11540 REM SET ASPECT RATIO OF SCREEN
11550 REM ******************************************************************************************************
11560 CLS
11570 PRINT
11580 IF ASPECT<=O 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 SCREEN '';ASPECT''
11600 INPUT TEMPS
11610 IF LEN(TEMPS)>0 THEN ASPECT=VAL(TEMPS)
11612 PRINT " ENTER VERTICAL EXAGGERATION OF CROSS-SECTION PLOT '';ZASPECT''
11614 INPUT TEMPS
11616 IF LEN(TEMPS)>0 THEN ZASPECT=VAL(TEMPS)
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 MERELY BY PRESSING
11760 REM THE KEY
11770 REM SPFFILES() - ARRAY OF SPECIAL FEATURE FILE NAMES
11780 REM SPFKEYS() - ARRAY OF KEYS FOR THESE SPECIAL FEATURE 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=1+1
11882 IF I>20 THEN PRINT " MAXIMUM NUMBER OF SPECIAL FEATURES ALREADY ASSIGNED":FOR K=1 TO
11884 NEXT K:RETURN
11890 PRINT " ENTER SPECIAL FEATURE FILE NAME '';SPFFILES(I)'',"
11900 LINE INPUT TEMPS
11910 IF LEN(TEMPS)>0 THEN SPFFILES(I)=TEMPS
11920 IF LEN(TEMPS)=0 AND LEN(SPFFILES(I))=0 THEN RETURN
11930 PRINT " ENTER ONE KEY FOR THIS SPECIAL FEATURE '');SPFKEYS(I,1)'',"
11940 INPUT TEMPS
11950 IF LEN(TEMPS)>1 THEN PRINT " ONLY ONE KEY ':GOTO 11930
11960 IF LEN(TEMPS)>0 THEN SPFKEYS(I,1)=TEMPS
11962 IF ASC(SPFKEYS(I,1))>64 AND ASC(SPFKEYS(I,1))<91 THEN SPFKEYS(I,2)=CHRS(ASC(SPFKEYS(I,1))+32)
11964 IF ASC(SPFKEYS(I,1))>96 AND ASC(SPFKEYS(I,1))<123 THEN SPFKEYS(I,2)=CHRS(ASC(SPFKEYS(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 '';SPFSAVES''"
12070 LINE INPUT TEMPS
12080 IF LEN(TEMPS)>0 THEN SPFSAVES=TEMPS
12090 OPEN "O",1,SPFSAVES
12100 PRINT#1,11DEF!;J1DEF!;SCALEDEF
12110 PRINT#1,ASPECT;ZASPECT
12115 PRINT#1,NSPF
12120 FOR I=1 TO NSPF
12130 PRINT#1,SPFKEY$(1,1),SPFKEY$(1,2),SPFFILE$(I)
12140 NEXT I
12150 CLOSE 1
12160 RETURN
12200 REM *******************************************************
12210 REM LOAD SPECIAL FEATURE DATA FROM FILE
12220 REM *******************************************************
12230 REM
12240 PRINT
12250 PRINT "ENTER FILE SPECIFICATION TO LOAD SPECIAL FEATURE DATA FROM ";SPFSAVES;
12260 IF LEN(TEMP$)>0 THEN SPFSAVES=TEMP$;
12270 OPEN "I",1,SPFSAVES
12280 INPUT#1,11DEF!,JIDEF!,SCALEDEF
12290 INPUT#1,ASPECT,ZASPECT
12300 INPUT#1,NSPF
12310 FOR I=1 TO NSPF
12320 INPUT#1,SPF..Y,ý..,1,,),SPFKEY$(1,2),SPFFILE$(I)
12330 NEXT I
12340 CLOSE 1
12350 RETURN
12400 REM *******************************************************
12410 REM SUBROUTINE TO CHECK SPECIAL FEATURE KEYS FOR DUPLICATION
12420 REM *******************************************************
12430 IF SPFKEY$(1,1)="W" OR SPFKEY$(1,1)="w" THEN 12540
12440 IF SPFKEY$(1,1)="C" OR SPFKEY$(1,1)="c" THEN 12540
12450 IF SPFKEY$(1,1)="M" OR SPFKEY$(1,1)="m" THEN 12540
12460 IF SPFKEY$(1,1)="N" OR SPFKEY$(1,1)="n" THEN 12540
12470 IF SPFKEY$(1,1)="Z" OR SPFKEY$(1,1)="z" THEN 12540
12480 IF SPFKEY$(1,1)="G" OR SPFKEY$(1,1)="g" THEN 12540
12490 IF SPFKEY$(1,1)="D" OR SPFKEY$(1,1)="d" THEN 12540
12495 FOR K=I TO NSPF
12500 IF I=K THEN 12520
12510 IF SPFKEY$(1,1)=SPFKEY$(K,1) OR SPFKEY$(1,2)=SPFKEY$(K,1) THEN 12540
12520 NEXT K
12525 IF ASC(SPFKEY$(1,1)) < 33 OR ASC(SPFKEY$(1,1)) > 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:RETURN
12600 REM *******************************************************
12610 REM SUBROUTINE TO DRAW MODEL GRID ON SCREEN
12620 REM *******************************************************
12630 YBOT=LLY
12640 YTOP=LLY+(NR)*DELY
FOR I=0 TO NC
  X=LLX+I*DELX
  LINE (X,YBOT)-(X,YTOP),1
NEXT I

XTOP=LLX+(NC)*DELX
FOR J=0 TO NR
  Y=LLY+J*DELY
  LINE (XBOT,Y)-(XTOP,Y),1
NEXT J
RETURN

REM *******************************************************
REM ROUTINE TO INPUT VELOCITY FILE END TIMES
REM NENDTIME - NUMBER OF VELOCITY FILES TO BE USED DURING SIMULATION
REM ENDTIME() - ARRAY OF ENDING TIMES FOR EACH VELOCITY FILE
REM *******************************************************
CLS
ENDTIME(0)=0
PRINT
PRINT
"IF MULTIPLE VELOCITY FILES ARE TO BE USED (TRANSIENT SIMULATION)"
PRINT
"THE USER MUST ENTER THE NUMBER OF VELOCITY FILES THAT WILL BE USED"
PRINT
"AND THE ENDING TIME FOR EACH VELOCITY FILE"
PRINT
PRINT
"ENTER THE NUMBER OF VELOCITY FILES (TIME PERIODS) ";"[";NENDTIME;"] ";
INPUT TEMPS
IF LEN(TEMPS)>0 THEN NENDTIME=VAL(TEMPS)
IF NENDTIME<=0 THEN PRINT
"STEADY STATE ANALYSIS":FOR K=I TO 6000:NEXT K:RETURN
IF NENDTIME>20 THEN PRINT
"MAXIMUM NUMBER OF TIME PERIODS IS 20 ":GOTO 12920
FOR I=1 TO NENDTIME
  PRINT
"ENTER THE ENDING TIME (DAYS) FOR VELOCITY FILE ";"[";ENDTIME(I);"] ";
  INPUT TEMPS
  IF LEN(TEMP$)>0 THEN ENDTIME(I)=VAL(TEMP$)
  IF ENDTIME(I)<ENDTIME(I-1) THEN PRINT
"ENDING TIME MUST BE GREATER THAN PREVIOUS TIME":GOTO 12980
NEXT I
RETURN

REM ****************44***********44*******
REM PAUSE ON SCREEN ROUTINE
REM *****************************************
PRINT
INPUT
"PRESS ENTER TO CONTINUE ";Z1S
RETURN

REM ************************
REM PLOT ROW CROSS SECTION ON SCREEN
REM **************************************************
REM ZASPECT - HORIZONTAL/VERTICAL SCREEN RATIO - THE INVERSE
REM I1I,J1!, - COORDINATES OF LOWER LEFT CORNER OF SCREEN
REM LLX,LLZ - ENTERED COORDINATES OF LOWER LEFT CORNER OF SCREEN
REM USE SAME I1! AND I2! COORDINATES AS BEFORE=LLX
IF I2!=11! THEN I2!=I1!+SCALEDEF
IF SCALEDEF=0 THEN SCALEDEF=NC*DEHX
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+DEHX/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+DEHX*I
   IF THICK(I,J,K)<0 THEN GOTO 14180
14170 LINE -(DIST-DEHX/2,BOT(I,J,K))
14180 NEXT I
14190 PSET (LLX+DEHX/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+DEHX*I
   IF THICK(I,J,K)<0 THEN GOTO 14230
14220 LINE -(DIST-DEHX/2,TOP(I,J,K))
14230 NEXT I
14240 NEXT K
14250 ISTART=0
14250 ISTART=ISTART+1
   IF THICK(ISTART,J,K)<0 THEN GOTO 14250
   PSET (LLX+DEHX/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+DEHX*I
   IF THICK(I,J,K)<0 THEN GOTO 14290
14280 LINE -(DIST-DEHX/2,BOT(I,J,NL))
14290 NEXT I
14300 PSET LLX+DEHX/2,BOT(ISTART,J,NL)+THICK(ISTART,J,NL)): REM SET BEGINNING OF WATER TABLE TOP
14310 FOR I=1 TO NC: REM PLOT BOTTOM OF TOP LAYER
14320 DIST=LLX+DEHX*I
   IF THICK(I,J,K)<0 THEN GOTO 14340
14330 LINE -(DIST-DEHX/2,BOT(I,J,NL)+THICK(I,J,NL))
14340 NEXT I
14350 FOR II=1 TO WS: REM PLOT WELLS AND RIVERS ON SCREEN
14360 IF INT(X1(II)-LLY)/DELY+1.O<>J THEN 14440: REM ONLY PLOT WELLS IN THIS ROW
14370 J=INT(X1(II)-LLY)/DELY+1.O: REM FIND COLUMN OF WELLS
14380 REM FOR KLK=1 TO NL-1: REM FIND LAYER OF WELL
14390 IF Z1(II)=BOT(I,J,NL) OR INT(Z1(II))=NL THEN ZK=BOT(I,J,INT(Z1(II)))
14400 REM NEXT KLK
14410 REM IF Z1(II)=BOT(I,J,INT(Z1(II))) AND Z1(II)<TOP(I,J,KLK) THEN ZK=BOT(I,J,KLK)
14420 REM NEXT KLK
14430 REM IF Z1(II)=BOT(I,J,INT(Z1(II))) AND Z1(II)<TOP(I,J,KLK) THEN ZK=BOT(I,J,INT(Z1(II)))
14440 REM NEXT KLK
14450 REM ZK=BOT(I,J,INT(Z1(II)))
14460 REM IF Z1(II)=BOT(I,J,INT(Z1(II))) AND Z1(II)<TOP(I,J,KLK) THEN ZK=BOT(I,J,INT(Z1(II)))
14470 REM NEXT KLK
14480 REM IF Z1(II)=BOT(I,J,INT(Z1(II))) AND Z1(II)<TOP(I,J,KLK) THEN ZK=BOT(I,J,INT(Z1(II)))
14490 REM NEXT KLK
14500 Z1S=""
TEMPSCR=O:REM RESET SCREEN FLAG FOR PLAN VIEW WITH DEFAULT COORDINATES (MAPTYPES="A")

RETURN
14442 Z$=INKEYS:REM DO NOT CHANGE SCREEN UNTIL USER HITS KEY
    IF Z$="g" or Z$="G" or Z$I=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,J1 - COORDINATES OF LOWER LEFT CORNER OF SCREEN
14570 REM COLXS - SELECTED ROW TO PLOT ON SCREEN
14580 REM LLY,LLZ - ENTERED COORDINATES OF LOWER LEFT CORNER OF SCREEN
    IF J21<=J11 THEN J21=J11+NR*DELY
14610 I1=J1:J21=J21:REM TRANSPOSE SCREEN FOR COLUMNS
    IF SCALEDEF<=0 THEN SCALEDEF=NC*DELX
    J11=LLZ: REM COORDINATES OF LOWER LEFT CORNER OF SCREEN
    J21=LLZ+SCALEDEF/ZASPECT
14615 GOSUB 7880:REM CALL ROUTINE TO SET UP SCREEN WINDOW
14620 I=ICOLXS
14630 FOR K=I TO NL-1
14640 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
14800 JSTART=0
14800 JSTART=JSTART+1
    IF THICK(I,JSTART,K)<0 THEN GOTO 14800
    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((X$(II)-LLX)/DELX+1.0) <> 1 THEN 14940:REM ONLY PLOT WELLS IN THIS RO
J=INT((Y1(II)-LLY)/DELY+1.0):REM FIND ROW OF WELL

REM FOR KLK=1 TO NL-1:REM FIND LAYER OF WELL

IF Z1(II)>0 AND Z1(II)<TOP(I,J,KLK) THEN ZK=BOT(I,J,KLK)

NEXT KLK

ZK=BOT(I,J,INT(Z1(II))):REM FIND LAYER OF WELL

IF Z1(II)>=BOT(I,J,NL) AND Z1(II)<(BOT(I,J,NL)+THICK(I,J,NL)) THEN ZK=BOT(I,J,NL)

ZL=INKEY$:REM DO NOT CHANGE SCREEN UNTIL USER HITS "G"

IF ZL='G' OR ZL=CHR$(13) THEN RETURN ELSE GOTO 14942

RETURN

TEMPSCR=O:REM RESET SCREEN FLAG FOR PLAN VIEW WITH DEFAULT COORDINATES (MAPTYPES="A")

IF ZL=CHR$(13) THEN RETURN :REM EXIT ROUTINE ON CARROCH RETURN WITH ROW SELECTED

IF ZL=CHR$(0)+CHR$(72) THEN LINE (I1!,Y1X-DELY/4)-(I2!,Y1X+DELY/4),0,B:ICOLXS=ICOLXS+1

IF ZL=CHR$(0)+CHR$(75) THEN LINE (I1!,Y1X-DELY/4)-(I2!,Y1X+DELY/4),0,B:ICOLXS=ICOLXS-1

GOTO 1510:REM RETURN TO BEGINNING OF SUBROUTINE

INKEY$:REM PROFILE COLUMN SETUP AND ROW SELECTION

IF ICOLXS=NC THEN ICOLXS=NC

IF ICOLXS<1 THEN ICOLXS=1:REM CHECK POSITION OF COL AND KEEP ON GRID

IF ICOLXS>NC THEN ICOLXS=NC

LINE (I1!,Y1X-DELY/4)-(I2!,Y1X+DELY/4),2,B

IF ZL=CHR$(13) THEN RETURN :REM EXIT ROUTINE ON CARROCH RETURN WITH ROW SELECTED

IF ZL=CHR$(0)+CHR$(72) THEN LINE (I1!,Y1X-DELY/4)-(I2!,Y1X+DELY/4),0,B:ICOLXS=ICOLXS+1

IF ZL=CHR$(0)+CHR$(75) THEN LINE (I1!,Y1X-DELY/4)-(I2!,Y1X+DELY/4),0,B:ICOLXS=ICOLXS-1

GOTO 1530:REM RETURN TO BEGINNING OF SUBROUTINE

TEMPSCR=O:REM RESET SCREEN FLAG FOR PLAN VIEW WITH DEFAULT COORDINATES (MAPTYPES="A")

IF ZL=CHR$(13) THEN RETURN :REM EXIT ROUTINE ON CARROCH RETURN WITH ROW SELECTED

IF ZL=CHR$(0)+CHR$(72) THEN LINE (I1!,Y1X-DELY/4)-(I2!,Y1X+DELY/4),0,B:ICOLXS=ICOLXS+1

IF ZL=CHR$(0)+CHR$(75) THEN LINE (I1!,Y1X-DELY/4)-(I2!,Y1X+DELY/4),0,B:ICOLXS=ICOLXS-1

GOTO 1530:REM RETURN TO BEGINNING OF SUBROUTINE

D-33
APPENDIX E

Typical Model Output
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= .103536

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= 0.1240565
MASS EXITING IN PUMPED WELL NUMBER 14, IN LBS, IS= 0.56082
CONCENTRATION IN PUMPED WELL NUMBER 15, IN PPM, IS= 6.710225E-02
MASS EXITING IN PUMPED WELL NUMBER 15, IN LBS, IS= 0.25884
CONCENTRATION IN PUMPED WELL NUMBER 16, IN PPM, IS= 7.041197E-02
MASS EXITING IN PUMPED WELL NUMBER 16, IN LBS, IS= 0.30198
CONCENTRATION IN PUMPED WELL NUMBER 17, IN PPM, IS= 7.720923E-03
MASS EXITING IN PUMPED WELL NUMBER 17, IN LBS, IS= 0.04314
CONCENTRATION IN PUMPED WELL NUMBER 18, IN PPM, IS= 3.309396E-02
MASS EXITING IN PUMPED WELL NUMBER 18, IN LBS, IS= 0.12942

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= 8.699918
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= 99.22199
CONCENTRATION IN PUMPED WELL NUMBER 11, IN PPM, IS= 2.003237E-02
MASS EXITING IN PUMPED WELL NUMBER 11, IN LBS, IS= 0.86628
CONCENTRATION IN PUMPED WELL NUMBER 13, IN PPM, IS= 2.131044E-02
MASS EXITING IN PUMPED WELL NUMBER 13, IN LBS, IS= 0.86628
CONCENTRATION IN PUMPED WELL NUMBER 14, IN PPM, IS= 6.847543E-02
MASS EXITING IN PUMPED WELL NUMBER 14, IN LBS, IS= 30.198
CONCENTRATION IN PUMPED WELL NUMBER 15, IN PPM, IS= 9.201175E-02
MASS EXITING IN PUMPED WELL NUMBER 15, IN LBS, IS= 34.512
CONCENTRATION IN PUMPED WELL NUMBER 16, IN PPM, IS= 5.151961E-02
MASS EXITING IN PUMPED WELL NUMBER 16, IN LBS, IS= 2.157
CONCENTRATION IN PUMPED WELL NUMBER 17, IN PPM, IS= 2.361315E-02
MASS EXITING IN PUMPED WELL NUMBER 17, IN LBS, IS= 1.2942
CONCENTRATION IN PUMPED WELL NUMBER 18, IN PPM, IS= 3.448108E-02
MASS EXITING IN PUMPED WELL NUMBER 18, IN LBS, IS= 1.2942
CONCENTRATION IN PUMPED WELL NUMBER 19, IN PPM, IS= 1.145189E-02
MASS EXITING IN PUMPED WELL NUMBER 19, IN LBS, IS= 0.04314

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= 0.6559992
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= 0.04314
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= 0.04314
INTEGRAL CONCENTRATION IN PUMPED WELL NUMBER 16, IN PPM, IS= 5.1977955E-02
MASS EXITING IN PUMPED WELL NUMBER 16, IN LBS, IS= 0.2157
INTEGRAL CONCENTRATION IN PUMPED WELL NUMBER 17, IN PPM, IS= 7.929791E-03
MASS EXITING IN PUMPED WELL NUMBER 17, IN LBS, IS= 0.04314
INTEGRAL CONCENTRATION IN PUMPED WELL NUMBER 18, IN PPM, IS= 2.331844E-02
MASS EXITING IN PUMPED WELL NUMBER 18, IN LBS, IS= 0.08628
INTEGRAL CONCENTRATION IN PUMPED WELL NUMBER 19, IN PPM, IS= 1.157192E-02
MASS EXITING IN PUMPED WELL NUMBER 19, IN LBS, IS= 0.04314

INPUT VELOCITY FILE CLV105.RND
PRESENT SIMULATION TIME = 243.3332 DAYS
INCREMENTAL SIMULATION TIME = 60.8333 DAYS
D.MAX = 10 FT Z.MAX = 1 FT
NP= 3308
CONCENTRATION IN PUMPED WELL NUMBER 2, IN PPM, IS= 0.4884946
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= 0.7765199
CONCENTRATION IN PUMPED WELL NUMBER 11, IN PPM, IS= 1.015256E-02
MASS EXITING IN PUMPED WELL NUMBER 11, IN LBS, IS= 0.04314
CONCENTRATION IN PUMPED WELL NUMBER 12, IN PPM, IS= 1.184606E-02
MASS EXITING IN PUMPED WELL NUMBER 12, IN LBS, IS= 0.04314
CONCENTRATION IN PUMPED WELL NUMBER 13, IN PPM, IS= 1.080282E-02
MASS EXITING IN PUMPED WELL NUMBER 13, IN LBS, IS= 0.04314
CONCENTRATION IN PUMPED WELL NUMBER 14, IN PPM, IS= 4.953232E-02
MASS EXITING IN PUMPED WELL NUMBER 14, IN LBS, IS= 0.2157
CONCENTRATION IN PUMPED WELL NUMBER 15, IN PPM, IS= 9.333776E-02
MASS EXITING IN PUMPED WELL NUMBER 15, IN LBS, IS= 0.34512
CONCENTRATION IN PUMPED WELL NUMBER 16, IN PPM, IS= 2.085047E-02
MASS EXITING IN PUMPED WELL NUMBER 16, IN LBS, IS= 0.08628
CONCENTRATION IN PUMPED WELL NUMBER 17, IN PPM, IS= 1.588923E-02
MASS EXITING IN PUMPED WELL NUMBER 17, IN LBS, IS= 0.08628
CONCENTRATION IN PUMPED WELL NUMBER 18, IN PPM, IS= 1.172515E-02
MASS EXITING IN PUMPED WELL NUMBER 18, IN LBS, IS= 0.04314

INPUT VELOCITY FILE CLV106.RND
PRESENT SIMULATION TIME = 304.1665 DAYS
INCREMENTAL SIMULATION TIME = 60.8333 DAYS
D.MAX = 10 FT Z.MAX = 1 FT
NP= 2792
CONCENTRATION IN PUMPED WELL NUMBER 2, IN PPM, IS= 0.3637886
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= 0.7765199
CONCENTRATION IN PUMPED WELL NUMBER 12, IN PPM, IS= 1.186094E-02
MASS EXITING IN PUMPED WELL NUMBER 12, IN LBS, IS= 0.08628
CONCENTRATION IN PUMPED WELL NUMBER 14, IN PPM, IS= 4.314001E-02
MASS EXITING IN PUMPED WELL NUMBER 14, IN LBS, IS= 0.12942
CONCENTRATION IN PUMPED WELL NUMBER 18, IN PPM, IS= 1.174784E-02
MASS EXITING IN PUMPED WELL NUMBER 18, IN LBS, IS= 0.04314
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 =CLI3.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 =CLI2.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 = 0.28139
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 = 0.34512
CONCENTRATION IN PUMPED WELL NUMBER 13, IN PPM, IS = 1.033524E-02
MASS EXITING IN PUMPED WELL NUMBER 13, IN LBS, IS = 0.12942
CONCENTRATION IN PUMPED WELL NUMBER 14, IN PPM, IS = 9.838814E-03
MASS EXITING IN PUMPED WELL NUMBER 14, IN LBS, IS = 4.314001E-02
CONCENTRATION IN PUMPED WELL NUMBER 15, IN PPM, IS = 1.105715E-02
MASS EXITING IN PUMPED WELL NUMBER 15, IN LBS, IS = 8.628001E-02
CONCENTRATION IN PUMPED WELL NUMBER 16, IN PPM, IS = 1.05715E-02
MASS EXITING IN PUMPED WELL NUMBER 16, IN LBS, IS = 4.314001E-02

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 = 0.2262924
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 = 0.34512
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 = 0.04314

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 = 0.2085825
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

CONCENTRATION IN PUMPED WELL NUMBER 19, IN PPM, IS= 2.295723E-02
MASS EXITING IN PUMPED WELL NUMBER 19, 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
<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lower left-hand corner of plot position</td>
<td>2090</td>
</tr>
<tr>
<td>Grid size of plot map, in feet</td>
<td>60</td>
</tr>
<tr>
<td>Number of rows and columns of plot map</td>
<td>32</td>
</tr>
<tr>
<td>Plot file name is</td>
<td>CL23.DAT</td>
</tr>
<tr>
<td>Input velocity file</td>
<td>CLV301.RND</td>
</tr>
<tr>
<td>Present simulation time</td>
<td>729.9997 DAYS</td>
</tr>
<tr>
<td>Incremental simulation time</td>
<td>60.8333 DAYS</td>
</tr>
<tr>
<td>DMAX</td>
<td>10 FT</td>
</tr>
<tr>
<td>ZMAX</td>
<td>1 FT</td>
</tr>
<tr>
<td>Concentration in pumped well number 2, in PPM, IS</td>
<td>9.510852E-02</td>
</tr>
<tr>
<td>Mass exiting in pumped well number 2, in LBS, IS</td>
<td>3.75318</td>
</tr>
<tr>
<td>Concentration in pumped well number 3, in PPM, IS</td>
<td>1.147862E-02</td>
</tr>
<tr>
<td>Mass exiting in pumped well number 3, in LBS, IS</td>
<td>.30198</td>
</tr>
<tr>
<td>Concentration in pumped well number 11, in PPM, IS</td>
<td>9.444213E-03</td>
</tr>
<tr>
<td>Mass exiting in pumped well number 11, in LBS, IS</td>
<td>.04314</td>
</tr>
<tr>
<td>Concentration in pumped well number 13, in PPM, IS</td>
<td>1.996181E-02</td>
</tr>
<tr>
<td>Mass exiting in pumped well number 13, in LBS, IS</td>
<td>.08628</td>
</tr>
<tr>
<td>Concentration in pumped well number 14, in PPM, IS</td>
<td>2.770774E-02</td>
</tr>
<tr>
<td>Mass exiting in pumped well number 14, in LBS, IS</td>
<td>.12942</td>
</tr>
<tr>
<td>Concentration in pumped well number 15, in PPM, IS</td>
<td>1.081243E-02</td>
</tr>
<tr>
<td>Mass exiting in pumped well number 15, in LBS, IS</td>
<td>.04314</td>
</tr>
<tr>
<td>Concentration in pumped well number 16, in PPM, IS</td>
<td>9.810428E-03</td>
</tr>
<tr>
<td>Mass exiting in pumped well number 16, in LBS, IS</td>
<td>.04314</td>
</tr>
<tr>
<td>Concentration in pumped well number 18, in PPM, IS</td>
<td>0.31613</td>
</tr>
<tr>
<td>Mass exiting in pumped well number 18, in LBS, IS</td>
<td>.12942</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input velocity file</td>
<td>CLV302.RND</td>
</tr>
<tr>
<td>Present simulation time</td>
<td>790.833 DAYS</td>
</tr>
<tr>
<td>Incremental simulation time</td>
<td>60.8333 DAYS</td>
</tr>
<tr>
<td>DMAX</td>
<td>10 FT</td>
</tr>
<tr>
<td>ZMAX</td>
<td>1 FT</td>
</tr>
<tr>
<td>Concentration in pumped well number 2, in PPM, IS</td>
<td>9.073572E-02</td>
</tr>
<tr>
<td>Mass exiting in pumped well number 2, in LBS, IS</td>
<td>3.58062</td>
</tr>
<tr>
<td>Concentration in pumped well number 3, in PPM, IS</td>
<td>9.838814E-03</td>
</tr>
<tr>
<td>Mass exiting in pumped well number 3, in LBS, IS</td>
<td>.25884</td>
</tr>
<tr>
<td>Concentration in pumped well number 15, in PPM, IS</td>
<td>0.0215263</td>
</tr>
<tr>
<td>Mass exiting in pumped well number 15, in LBS, IS</td>
<td>.08628</td>
</tr>
<tr>
<td>Concentration in pumped well number 16, in PPM, IS</td>
<td>4.889414E-02</td>
</tr>
<tr>
<td>Mass exiting in pumped well number 16, in LBS, IS</td>
<td>.2157</td>
</tr>
<tr>
<td>Concentration in pumped well number 17, in PPM, IS</td>
<td>7.603521E-03</td>
</tr>
<tr>
<td>Mass exiting in pumped well number 17, in LBS, IS</td>
<td>.04314</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input velocity file</td>
<td>CLV303.RND</td>
</tr>
<tr>
<td>Present simulation time</td>
<td>851.6663 DAYS</td>
</tr>
<tr>
<td>Incremental simulation time</td>
<td>60.8333 DAYS</td>
</tr>
<tr>
<td>DMAX</td>
<td>10 FT</td>
</tr>
<tr>
<td>ZMAX</td>
<td>1 FT</td>
</tr>
<tr>
<td>Concentration in pumped well number 2, in PPM, IS</td>
<td>.0732445</td>
</tr>
<tr>
<td>Mass exiting in pumped well number 2, in LBS, IS</td>
<td>2.89038</td>
</tr>
</tbody>
</table>
CONCENTRATION IN PUMPED WELL NUMBER 16, IN PPM, IS= 2.929268E-02
MASS EXITING IN PUMPED WELL NUMBER 16, IN LBS, IS= 0.12942
CONCENTRATION IN PUMPED WELL NUMBER 18, IN PPM, IS= 1.045472E-02
MASS EXITING IN PUMPED WELL NUMBER 18, IN LBS, IS= 0.04314

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= 0.0776173
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= 0.04314

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= 2.71782
CONCENTRATION IN PUMPED WELL NUMBER 3, IN PPM, IS= 9.838814E-03
MASS EXITING IN PUMPED WELL NUMBER 3, IN LBS, IS= 0.04314

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= 0.04314
CONCENTRATION IN PUMPED WELL NUMBER 15, IN PPM, IS= 1.074411E-02
MASS EXITING IN PUMPED WELL NUMBER 15, IN LBS, IS= 0.04314
Plot file name is \text{=CL33.DAT}
Plot file name is \text{=CL32.DAT}
Plot file name is \text{=CL35.DAT}
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= 1.639321
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
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

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