FRONT-TRACKING MODEL FOR CONVECTIVE TRANSPORT IN FLOWING GROUND WATER

by Stephen P. Garabedian and Leonard F. Konikow

U.S. GEOLOGICAL SURVEY
Water-Resources Investigations 83-4034

1983

UNITED STATES DEPARTMENT OF THE INTERIOR

JAMES G. WATT, Secretary

GEOLOGICAL SURVEY

Dallas L. Peck, Director

For additional information write to:

L. F. Konikow, Project Chief
Water Resources Division
U.S. Geological Survey
431 National Center
Reston, Virginia 22092

Copies of this report can be purchased from:

Open-File Services Section
Western Distribution Branch
U.S. Geological Survey
Box 25425, Federal Center
Lakewood, Colorado 80225
(Telephone: [303] 234-5888)

CONTENTS

FRONT-TRACKING MODEL FOR CONVECTIVE TRANSPORT IN FLOWING GROUND WATER

by Stephen P. Garabedian and Leonard F. Konikow

ABSTRACT

This report describes a finite-difference numerical model that simulates the convective transport of water or tracer particles through porous media. It can be applied to one- or two-dimensional problems involving either steady-state or transient flow. The model tracks representative water or tracer particles, initially located along specified lines, as they move in response to the ground-water velocity field. Aquifer properties may be both anisotropic and nonhomogeneous. Included in the report is a listing of the program along with input formats and test problem results. The front-tracking model provides a useful first approximation for determining the movement of solutes in an aquifer, particularly in cases where dispersion and dilution is of minor consideration.

2 Blank

INTRODUCTION

This report presents a two-dimensional, finite-difference, numerical model to simulate the convective transport of a conservative (nonreactive) trace constituent dissolved in flowing ground water. The model computes the change in position of tracer particles in the aquifer from a specified initial location. The specified initial location might represent a chemical front or interface within the aquifer, or perhaps a point source of contamination to the aquifer, such as an injection well or waste disposal pond. The model then indicates how this front migrates through the aquifer with time. Thus, the model provides an estimate of average rates and directions of solute transport. However, because the effects of hydrodynamic dispersion and dilution from recharge or leakage into the aquifer are ignored, the model can not be used as an indicator of actual concentration levels or of first arrival times of contaminants. The model also assumes that the movement of the tracer is unaffected by differences in density or viscosity; hence, it would not be directly applicable to problems in which these factors are significant, such as saltwater encroachment in a coastal aquifer.

The computer program sequentially solves two sets of simultaneous partial differential equations. The first set determines the distribution of hydraulic head in an aquifer for specified aquifer properties, stresses, and initial and boundary conditions. The second set of equations utilizes the head distribution to calculate the velocity field for the aquifer at a given time. Tracer particles are moved to new positions in the aquifer in proportion to the flow velocity at their specific locations and to the length of the time increment. This particle tracking procedure approximates actual particle path lines in the aquifer. Consequently, the location and shape of chemical fronts would be traced most accurately in flow systems dominated by convective transport, where dispersion and dilution of the solute is relatively minor.

The program code for this front-tracking model is adapted from the solute-transport model developed by Konikow and Bredehoeft (1978). The front-tracking model requires less computer storage and computational time for equivalent problems than does the solute-transport model. These advantages make this model a useful tool for rapidly developing an initial understanding of solute transport in a ground-water system under study. The compatibility of the input data between the two models would also allow the front-tracking program to provide a less costly means to develop a preliminary calibration of the solute-transport model. Such a preliminary calibration can include adjustment of those parameters that are relatively independent of the effects of dispersion and dilution.

This report includes only a brief discussion of the theoretical basis and computational procedure of the computer program. A more detailed discussion of input data requirements and of most of the computational algorithm used in this model is presented in the report by Konikow and Bredehoeft (1978), which should be used in conjunction with this report.

MODEL THEORY AND ASSUMPTIONS

The front-tracking model is based on two equations that describe the distribution of hydraulic head and seepage velocity in a ground-water system. An equation describing the transient two-dimensional areal flow of a homogeneous compressible fluid through a nonhomogeneous anisotropic aquifer can be written in Cartesian tensor notation as

$$\frac{\partial}{\partial x_i}\left(T_{ij}\frac{\partial h}{\partial x_j}\right) = S\frac{\partial h}{\partial t} + W \qquad i,j=1,2 \qquad (1)$$

where T_{ij} is the transmissivity tensor, L^2/T;

h is the hydraulic head, L;

S is the storage coefficient, dimensionless;

t is the time, T;

$W = W(x,y,t)$ is the volume flux per unit area (positive sign for outflow and negative for inflow), L/T; and

x_i and x_j are Cartesian coordinates, L.

An equation describing the average seepage velocity of ground water can be derived from Darcy's Law and written in Cartesian tensor notation as

$$V_i = -\frac{K_{ij}}{\varepsilon}\frac{\partial h}{\partial x_j} \qquad (2)$$

where V_i is the seepage velocity in the direction of x_i, L/T;

K_{ij} is the hydraulic conductivity tensor, L/T; and

ε is the effective porosity of the aquifer, (dimensionless).

The solution to equation 1 provides the head distribution in the aquifer for the specified properties, boundary conditions, and stresses. The resultant hydraulic gradients provide the additional information needed to compute seepage velocites using equation 2.

Use of these equations requires that the following assumptions be applicable to the system under study.

1. Darcy's law is valid and hydraulic-head gradients are the only significant driving mechanism for fluid flow.
2. The porosity and hydraulic conductivity of the aquifer are constant with time, and porosity is uniform in space.
3. Gradients of fluid density, viscosity, and temperature do not affect the velocity distribution.
4. Vertical variations in head are negligible, or the simulated heads, transmissivities, and storage coefficients represent vertically averaged values.

The numerical methods used to solve equations (1) and (2) require that the area of interest be subdivided by a grid into a number of smaller subareas. This model is based on the use of a rectangular, uniformly spaced, block-centered, finite-difference grid in which nodes are defined at the centers of the rectangular cells.

Pinder and Bredehoeft (1968) show that if the axes of the Cartesian coordinate system are alined with the principal directions of the transmissivity tensor, the ground-water flow equation may be approximated by the following finite-difference equation:

$$T_{xx[i-\frac{1}{2},j]}\left[\frac{h_{i-1,j,k}-h_{i,j,k}}{(\Delta x)^2}\right] + T_{xx[i+\frac{1}{2},j]}\left[\frac{h_{i+1,j,k}-h_{i,j,k}}{(\Delta x)^2}\right] + T_{yy[i,j-\frac{1}{2}]}\left[\frac{h_{i,j-1,k}-h_{i,j,k}}{(\Delta y)^2}\right] + T_{yy[i,j+\frac{1}{2}]}\left[\frac{h_{i,j+1,k}-h_{i,j,k}}{(\Delta y)^2}\right]$$

$$= S\left[\frac{h_{i,j,k}-h_{i,j,k-1}}{(\Delta t)}\right] + \frac{q_{w(i,j)}}{\Delta x \Delta y} + \frac{K_z}{m}\left[H_{s(i,j)}-h_{i,j,k}\right] \qquad (3)$$

where i,j,k are indices in the x, y, and time dimensions, respectively;

$\Delta x, \Delta y, \Delta t$ are increments in the x, y, and time dimensions, respectively;

q_w is the volumetric rate of withdrawal or recharge at the (i,j) node, L^3/T;

K_z is the vertical hydraulic conductivity of the confining layer, streambed, or lakebed, L/T;

m is the thickness of the confining layer, streambed, or lakebed, L; and

H_s is the hydraulic head in the source bed, stream, or lake, L.

The finite-difference equation is solved numerically for each block of the aquifer grid using an iterative alternating-direction implicit procedure (Konikow and Bredehoeft, 1978; Trescott and others, 1976). After the head distribution is solved for a new time step, the velocity distribution is computed using the following finite-difference approximations of the velocity equation:

$$V_{x(i,j)} = \frac{K_{xx(i,j)}}{\varepsilon} \frac{(h_{i-1,j,k} - h_{i+1,j,k})}{2\Delta x}, \quad (4)$$

for the velocity in the x-direction at node (i,j), and

$$V_{x(i+\frac{1}{2},j)} = \frac{K_{xx(i+\frac{1}{2},j)}}{\varepsilon} \frac{(h_{i,j,k} - h_{i+1,j,k})}{\Delta x} \quad (5)$$

for the velocity in the x-direction at the boundary between node (i,j) and node (i+1,j). Analogous equations are used for the computation of the velocities in the y-direction.

The new coordinates of tracer particles for each step are determined with the following equations:

$$x_{p,k} = x_{p,k-1} + \delta x_p = x_{p,k-1} + \Delta t V_{x[x(p,k),y(p,k)]} \quad (6)$$

$$y_{p,k} = y_{p,k-1} + \delta y_p = y_{p,k-1} + \Delta t V_{y[x_{(p,k)},y_{(p,k)}]} \quad (7)$$

where p is the index number for point identification; and

δx_p and δy_p are the distances tracer particles are moved in the x and y directions, respectively, during a time step.

The x and y components of the particle velocity are determined by bilinear interpolation over the area of half a cell using the x and y velocities computed at adjacent nodes and cell boundaries (Konikow and Bredehoeft, 1978). The distance each tracer particle is moved is equal to the velocity at its location times the duration of the time increment. To minimize the divergence of the subsequent linear position changes from a true curvilinear path line, each time step for the numerical solution to the flow equation is further subdivided for particle tracking into a number of smaller secondary time increments such that the maximum distance traveled by any particle during any secondary time increment does not exceed the width or length of one cell of the finite-difference grid.

The boundary and initial conditions of the aquifer must be specified to obtain a numerical solution to the ground-water flow equation. A variety of boundary conditions may be imposed by using a combination of the two general types (constant-flux and constant-head). A constant-flux boundary indicates that the node has a specified flux representing aquifer underflow, well discharge, or well injection. A no-flow condition, which is one type of constant-flux boundary, may also be specified by setting the nodal transmissivity equal to zero, precluding flow across any boundary of that node. The numerical procedure used in this model requires that the area of interest be surrounded by a no-flow boundary. Therefore, the model will automatically specify the outer rows and columns of the finite-difference grid as no-flow boundaries.

The constant-head condition may be used to represent parts of an aquifer where the head will not change with time, such as recharge boundaries or areas beyond the influence of hydraulic stresses. This is done by setting the leakance term (K_z/m) in the finite-difference equation to a sufficiently high value (1.0 s^{-1}) so that the head at the node will be computed implicity as a value that is essentially equal to the value of H_s, the desired constant-head altitude. The resulting rate of leakage into or out of the node equals the flux required to maintain the head in the aquifer at the specified constant-head altitude.

The initial head at the start of the modeling period should be specified on the basis of field data and (or) previous simulations. An accurate specification of the initial head distribution is particularly important for transient flow problems because errors in initial heads will induce compensating head changes during the simulation.

The no-flow boundary may present a problem related to the movement of particles near the boundary. Because the finite-difference model uses linear segments to approximate a particle path line, a particle may move across a no-flow boundary under certain circumstances. When this occurs, the program relocates the particle by reflection across the boundary. The correction will reposition the particle closer to the true flow line. Another problem occurs at nodes with fluid sources or sinks (withdrawal or injection wells). As these nodes may represent singularities in the velocity field, the finite-difference solution to the velocity equation at such a node may be in significant error. Thus, the velocity within a cell representing a point sink or source is approximated using the velocities computed on the adjacent cell boundaries. The appropriate boundary velocities are selected on the basis of the quadrant of the cell in which the tracer particle of interest is located.

PROGRAM OPERATION

The overall execution of the front-tracking model is controlled through the main program by calling subroutines in proper order. The computer program is written in FORTRAN IV and consists of a main program and seven subroutines. A listing of the source code is included in attachment I.

The aquifer properties, stresses, and initial and boundary conditions are read in by the PARLOD subroutine. A detailed description of the input data and their formats is included in attachment II. The input data formats are structured to resemble closely those used in the transport and dispersion model of Konikow and Bredehoeft (1978); this will allow the user to easily convert from one model to the other.

The model allows different spacing in the x and y directions, and spatially varying values of nodal thickness, transmissivity, leakance coefficients of confining beds, and hydraulic stresses. Anisotropy is defined through the ratio of the y-direction transmissivity to the x-direction transmissivity. This subroutine also reads in the initial particle distribution and the desired output options. The number of particles either at a point source or on any line in the aquifer may be stipulated. The placement and spacing of particles occurs in the GENPT subroutine. If a point source is desired, then the requested number of particles are placed in a circular pattern around that node with a radius equal to one-fourth of the smallest node spacing. If a number of points are stipulated to locate the initial position of the front, then the particles are placed in equal increments along the straight line segments connecting each consecutive point. Flexibility in specifying the hydrologic parameters allows a wide range of aquifer conditions to be simulated by the model.

Subroutine ITERAT includes an alternating-direction implicit procedure to solve the finite-difference approximation of the partial differential equation describing ground-water flow. This yields a computed head distribution for each new time step. Next, the velocities are computed for nodes and node boundaries in the VELO subroutine. The values for the velocities at the nodes and node boundaries are then used in the MOVE subroutine to move the particles. The movement limit (CELDIS) is used to produce sufficiently small time increments so that each particle closely follows a flow line (characteristic curve). The sequence of particle movement in subroutine MOVE is repeated for a sufficient number of time increments to complete the time step for the head calculations. The sequence of head calculations accomplished in subroutine ITERAT and velocity computations accomplished in subroutine VELO is repeated until the simulation of the desired total elapsed time period is completed.

The OUTPUT subroutine prints the hydrologic data, which includes the head distribution and fluid mass balance data. The velocity distribution for nodes and node boundaries can also be printed if desired. The particle locations are printed by the CHMOT subroutine. There are three types of particle location outputs, two of which are optional. A map view of the particle locations is printed each time CHMOT is called, and it prints a symbol indicating the number of particles located within each increment of the map area. The map will be oriented either with the x-axis across the page or down the page, whichever will produce a larger scale map. Node locations are indicated along the margins of the map. Because of printer limitations, there may be a slight distortion between the scales for the x and y directions.

One output option prints the (x,y) position of all particles at selected intervals. As this could be lengthy for a large number of particles, another option allows printing the locations of up to five particles at the end of every time increment. The particles tracked are specified by their identification number; the particle identification numbers range from 1 to NP (the total number of particles generated).

The results of a test case are presented to illustrate the application of the model. The test problem simulates a case in which recharge of water having a different quality causes a ground-water mound to develop in a homogeneous and isotropic aquifer. The input listing for this problem is presented in attachment III. The initial particle distribution is defined to form an arcuate shaped curve near the mound on the upgradient side of the aquifer.

Selected parts of the output listing for this problem are presented in attachment IV. The results show the front curvature increasing as the particles move downgradient. A series of maps showing the sequential change in position of a front serves to demonstrate the spatial variability of the velocity field.

A problem equivalent to this test problem was run with the two-dimensional solute transport and dispersion model documented by Konikow and Bredehoeft (1978) so that the relative efficiencies of these two specific modeling approaches could be compared. Generalizations based on these types of comparisons must always be qualified because the results depend strongly on the size of the grid and on the density of tracer particles used. Nevertheless, the comparisons can provide a reasonable indication of relative computational times and costs. In this case both models were run on a Harris S125 computer[1/]. The front-tracking model required a run time of 7.85 seconds to complete the test

---

[1/] Any use of trade names is for descriptive purposes only and does not constitute endorsement by the U.S. Geological Survey.

problem presented in attachment III. The dispersion model required 20.15 and 35.57 seconds when the initial number of particles per node was 4 and 9, respectively. The relative efficiency of the front-tracking model would be even greater for problems requiring a larger grid than used in this simple test problem.

SUMMARY

The front-tracking model may be used as first approximation for analyzing two-dimensional solute transport problems in ground water. As such, it can be useful in delineating and illustrating the rates and directions of flow of ground water. The model does not predict concentrations or first arrival times for solutes transported in an aquifer, but would instead approximate the arrival time of a mean position within the width of a frontal zone. Therefore, the model may also be useful in the evaluation of tracer tests for determining ground-water velocities. Computational costs required for equivalent problems are significantly less for the front-tracking model than are required for the solute-transport model of Konikow and Bredehoeft (1978); the front-tracking model thus also provides an economical tool for developing a preliminary calibration of the solute-transport model.

SELECTED REFERENCES

Konikow, L. F., and Bredehoeft, J. D., 1978, Computer model of two-dimensional solute transport and dispersion in ground water: U.S. Geological Survey Techniques of Water-Resources Investigations, Book 7, Chapter C2, 90 p.

Pinder, G. F., and Bredehoeft, J. D., 1968, Application of the digital computer for aquifer evaluation: Water Resources Research, v. 4, no. 5, p. 1069-1093.

Trescott, P. C., Pinder, G. F., and Larson, S. P., 1976, Finite-difference model for aquifer simulation in two dimensions with results of numerical experiments: U.S. Geological Survey Techniques of Water-Resources Investigations, Book 7, Chapter C1, 116 p.

Attachment I
FORTRAN IV Program Listing

```
C     MAIN PROGRAM
C     ****************************************************************
C     *                                                              *
C     *        FRONT-TRACKING MODEL FOR GROUND-WATER FLOW            *
C     *                  2-D;  UNIFORM DENSITY                       *
C     *      NUMERICAL SOLUTION --- METHOD OF CHARACTERISTICS        *
C     *    PROGRAMMED BY:  L. F. KONIKOW  AND  S. P. GARABEDIAN      *
C     *                                                              *
C     ****************************************************************
      DOUBLE PRECISION TMRX,VPRM,HI,HR,HC,HK,WT,REC,RECH,TIM,AOPT,TITLE,
     1XDEL,YDEL,S,AREA,SUMT,RHO,PARAM,TEST,TOL,PINT,HMIN,PYR,PYRSUM
      DOUBLE PRECISION TINT,ALPHA1,ANITP,TDEL
C     SPECIAL COMMON PRMI,PRMC,HEDA,HEDB,CHMA
      COMMON IS,IO
      COMMON /LSET/ SET(100),SEY(100)
      COMMON /GOIN/ INDT,I1,IT,KTOUT
      COMMON /PRMI/ NTIM,NPMP,NPNT,NITP,N,NX,NY,NP,NREC,INT,NNX,NNY,NMOV
     1,IMOV,ITMAX,NPNTMV,NPNTVL,NPRINP,NOBS
      COMMON /PRMC/ NODEID(20,20),INOBS(5)
      COMMON /HEDA/ THCK(20,20),PERM(20,20),ANFCTR
      COMMON /HEDB/ TMRX(20,20,2),VPRM(20,20),HI(20,20),HR(20,20),HC(20,
     120),HK(20,20),WT(20,20),REC(20,20),RECH(20,20),TIM(100),AOPT(20),T
     2ITLE(20),XDEL,YDEL,S,AREA,SUMT,RHO,PARAM,TEST,TOL,PINT,HMIN,PYR,PY
     3RSUM
      COMMON /CHMA/ PART(2,600),VX(20,20),VY(20,20),POROS,SUMTCH,TIMV,CE
     1LDIS
      IS=5
      IO=6
C     **************************************************************
C      ---LOAD DATA---
      INT=0
      CALL PARLOD
      CALL GENPT
C     **************************************************************
C     ---START COMPUTATIONS---
C        ---COMPUTE ONE PUMPING PERIOD---
      DO 30 INT=1,NPMP
      IF (INT.GT.1) CALL PARLOD
C        ---COMPUTE ONE TIME STEP---
      DO 10 N=1,NTIM
      KTOUT=0
C        ---LOAD NEW DELTA T---
      TINT=SUMT-(PYRSUM-PYR)
      TDEL=DMIN1(TIM(N),PYR-TINT)
      SUMT=SUMT+TDEL
      TIM(N)=TDEL
      REMN=MOD(N,NPNT)
C     **************************************************************
      CALL ITERAT
      IF (REMN.EQ.0.0.OR.N.EQ.NTIM) CALL OUTPT
      CALL VELO
```

```
      CALL MOVE
C     ************************************************************
C     ---OUTPUT ROUTINES---
      IF (REMN.EQ.0.0.OR.N.EQ.NTIM) CALL CHMOT
      IF (SUMT.GE.PYRSUM) GO TO 20
   10 CONTINUE
C     ************************************************************
C     ---SUMMARY OUTPUT---
   20 IF (KTOUT.EQ.1) GO TO 30
      CALL OUTPT
      CALL CHMOT
   30 CONTINUE
C
C     ************************************************************
      STOP
C     ************************************************************
C
C
      END
      SUBROUTINE PARLOD
      DOUBLE PRECISION TMRX,VPRM,HI,HR,HC,HK,WT,REC,RECH,TIM,AOPT,TITLE
      DOUBLE PRECISION XDEL,YDEL,S,AREA,SUMT,RHO,PARAM,TEST,TOL,PINT,HMI
     1N,PYR,PYRSUM
      DOUBLE PRECISION FCTR,TIMX,TINIT,PIES,YNS,XNS,RAT,HMX,HMY
      DOUBLE PRECISION TINT,ALPHA1,ANITP,TDEL
      INTEGER OVERRD
C     SPECIAL COMMON PRMI,PRMC,HEDA,HEDB,CHMA
      COMMON IS,IO
      COMMON /LSET/ SET(100),SEY(100)
      COMMON /GOIN/ INDT,I1,IT,KTOUT
      COMMON /PRMI/ NTIM,NPMP,NPNT,NITP,N,NX,NY,NP,NREC,INT,NNX,NNY,NMOV
     1,IMOV,ITMAX,NPNTMV,NPNTVL,NPRINP,NOBS
      COMMON /PRMC/ NODEID(20,20),INOBS(5)
      COMMON /HEDA/ THCK(20,20),PERM(20,20),ANFCTR
      COMMON /HEDB/ TMRX(20,20,2),VPRM(20,20),HI(20,20),HR(20,20),HC(20,
     120),HK(20,20),WT(20,20),REC(20,20),RECH(20,20),TIM(100),AOPT(20),T
     2ITLE(20),XDEL,YDEL,S,AREA.SUMT,RHO,PARAM,TEST,TOL,PINT,HMIN,PYR,PY
     3RSUM
      COMMON /CHMA/ PART(2,600),VX(20,20),VY(20,20),POROS,SUMTCH,TIMV,CE
     1LDIS
      COMMON /BALM/ TOTLQ,TOTLQI,TPIN,TPOUT
      COMMON /XINV/ DXINV,DYINV,ARINV,PORINV
      COMMON /CHMC/ VXBDY(20,20),VYBDY(20,20)
C     ************************************************************
      IF (INT.GT.1) GO TO 30
      WRITE (IO,730)
      READ (IS,700) TITLE
      WRITE (IO,710) TITLE
C     ************************************************************
C     ---INITIALIZE TEST AND CONTROL VARIABLES---
      TEST=0.0
      TOTLQ=0.0
```

```
      TOTLQI=0.0
      TPIN=0.0
      TPOUT=0.0
      SUMT=0.0
      SUMTCH=0.0
      INT=0
      IPRNT=0
      NCA=0
      N=0
      IMOV=0
      NMOV=0
      PYRSUM=0.0
C     **************************************************************
C     ---LOAD CONTROL PARAMETERS---
      READ (IS,720) NTIM,NPMP,NX,NY,NP,NPNT,NITP,ITMAX,NREC,NCODES,NPNTM
     1V,NPNTVL,NPRINP,NOBS
      READ (IS,800) PINT,TOL,POROS,S,TIMX,TINIT,XDEL,YDEL,CELDIS,ANFCTR
      READ (IS,420) INDT
      DO 10 I=1,INDT
   10 READ (IS,430) SET(I),SEY(I)
      NNX=NX-1
      NNY=NY-1
      DXINV=1.0/XDEL
      DYINV=1.0/YDEL
      ARINV=DXINV*DYINV
      PORINV=1.0/POROS
C     ---PRINT CONTROL PARAMETERS---
      WRITE (IO,740)
      WRITE (IO,750) NX,NY,XDEL,YDEL
      WRITE (IO,780) INDT
      DO 20 I=1,INDT
   20 WRITE (IO,790) SET(I),SEY(I)
      WRITE (IO,760) NTIM,NPMP,PINT,TIMX,TINIT
      WRITE (IO,770) S,POROS,ANFCTR
      WRITE (IO,890) NITP,TOL,ITMAX,CELDIS,NP
      WRITE (IO,900) NPNT,NPNTMV,NPNTVL,NREC,NCODES,NPRINP
      IF (NPNTMV.EQ.0) NPNTMV=999
      GO TO 40
C     **************************************************************
C     ---READ DATA TO REVISE TIME STEPS AND STRESSES FOR SUBSEQUENT
C        PUMPING PERIODS---
   30 READ (IS,1060) ICHK
      IF (ICHK.LE.0) PYRSUM=PYRSUM+PYR
      IF (ICHK.LE.0) RETURN
      READ (IS,1070) NTIM,NPNT,NITP,ITMAX,NREC,NPNTMV,NPNTVL,PINT,TIMX,T
     1INIT
      WRITE (IO,1080) INT
      WRITE (IO,1090) NTIM,NPNT,NITP,ITMAX,NREC,NPNTMV,NPNTVL,PINT,TIMX,
     1TINIT
C     **************************************************************
C     ---LIST TIME INCREMENTS---
   40 DO 50 J=1,100
```

```
      TIM(J)=0.0
   50 CONTINUE
      PYR=PINT*86400.0*365.25
      PYRSUM=PYRSUM+PYR
      TIM(1)=TINIT
      IF (S.EQ.0.0) GO TO 70
      DO 60 K=2,NTIM
   60 TIM(K)=TIMX*TIM(K-1)
      WRITE (IO,460)
      WRITE (IO,480) TIM
      GO TO 80
   70 TIM(1)=PYR
      WRITE (IO,470) TIM(1)
C     ****************************************************************
C     ---INITIALIZE MATRICES---
   80 IF (INT.GT.1) GO TO 110
      DO 90 IY=1,NY
      DO 90 IX=1,NX
      VPRM(IX,IY)=0.0
      PERM(IX,IY)=0.0
      THCK(IX,IY)=0.0
      RECH(IX,IY)=0.0
      REC(IX,IY)=0.0
      NODEID(IX,IY)=0
      TMRX(IX,IY,1)=0.0
      TMRX(IX,IY,2)=0.0
      HI(IX,IY)=0.0
      HR(IX,IY)=0.0
      HC(IX,IY)=0.0
      HK(IX,IY)=0.0
      WT(IX,IY)=0.0
      VX(IX,IY)=0.0
      VY(IX,IY)=0.0
      VXBDY(IX,IY)=0.0
      VYBDY(IX,IY)=0.0
   90 CONTINUE
C     ***********************************************************
C     ---READ IDENTIFICATION NUMBERS OF TRACER POINTS TO BE OBSERVED---
      IF (NOBS.LT.1) GO TO 110
      WRITE (IO,820)
      DO 100 I=1,NOBS
      READ (IS,830) INOBS(I)
  100 WRITE (IO,810) INOBS(I)
C     ****************************************************************
C     ---READ PUMPAGE DATA -- (X-Y COORDINATES AND RATE IN CFS)---
C        ---SIGNS : WITHDRAWAL = POS.;  INJECTION = NEG.---
  110 IF (NREC.LE.0) GO TO 130
      WRITE (IO,910)
      DO 120 I=1,NREC
      READ (IS,690) IX,IY,FCTR
      REC(IX,IY)=FCTR
  120 WRITE (IO,840) IX,IY,REC(IX,IY)
```

```
C     ****************************************************************
  130 IF (INT.GT.1) RETURN
      AREA=XDEL*YDEL
      WRITE (IO,680) AREA
      WRITE (IO,590)
      WRITE (IO,600) XDEL
      WRITE (IO,600) YDEL
C     ****************************************************************
C     ---READ TRANSMISSIVITY IN FT**2/SEC INTO VPRM ARRAY---
C        ---FCTR = TRANSMISSIVITY MULTIPLIER  --->  FT**2/SEC---
      WRITE (IO,520)
      READ (IS,540) INPUT,FCTR
      DO 170 IY=1,NY
      IF (INPUT.EQ.1) READ (IS,550) (VPRM(IX,IY),IX=1,NX)
      DO 160 IX=1,NX
      IF (INPUT.NE.1) GO TO 140
      VPRM(IX,IY)=VPRM(IX,IY)*FCTR
      GO TO 150
  140 VPRM(IX,IY)=FCTR
  150 IF (IX.EQ.1.OR.IX.EQ.NX) VPRM(IX,IY)=0.0
      IF (IY.EQ.1.OR.IY.EQ.NY) VPRM(IX,IY)=0.0
  160 CONTINUE
  170 WRITE (IO,510) (VPRM(IX,IY),IX=1,NX)
C     ****************************************************************
C     ---SET UP COEFFICIENT MATRIX --- BLOCK-CENTERED GRID---
C     ---AVERAGE TRANSMISSIVITY --- HARMONIC MEAN---
      IF (ANFCTR.NE.0.0) GO TO 180
      WRITE (IO,1050)
      ANFCTR=1.0
  180 PIES=3.1415927*3.1415927/2.0
      YNS=NY*NY
      XNS=NX*NX
      HMIN=2.0
      DO 190 IY=2,NNY
      DO 190 IX=2,NNX
      IF (VPRM(IX,IY).EQ.0.0) GO TO 190
      TMRX(IX,IY,1)=2.0*VPRM(IX,IY)*VPRM(IX+1,IY)/(VPRM(IX,IY)*XDEL+VPRM
     1(IX+1,IY)*XDEL)
      TMRX(IX,IY,2)=2.0*VPRM(IX,IY)*VPRM(IX,IY+1)/(VPRM(IX,IY)*YDEL+VPRM
     1(IX,IY+1)*YDEL)
C        ---ADJUST COEFFICIENT FOR ANISOTROPY---
      TMRX(IX,IY,2)=TMRX(IX,IY,2)*ANFCTR
C        ---COMPUTE MINIMUM ITERATION PARAMETER (HMIN)---
      IF (TMRX(IX,IY,1).EQ.0.0) GO TO 190
      IF (TMRX(IX,IY,2).EQ.0.0) GO TO 190
      RAT=TMRX(IX,IY,1)*YDEL/(TMRX(IX,IY,2)*XDEL)
      HMX=PIES/(XNS*(1.0+RAT))
      HMY=PIES/(YNS*(1.0+(1.0/RAT)))
      IF (HMX.LT.HMIN) HMIN=HMX
      IF (HMY.LT.HMIN) HMIN=HMY
  190 CONTINUE
C     ****************************************************************
```

```
C     ---READ AQUIFER THICKNESS---
      WRITE (IO,500)
      READ (IS,540) INPUT,FCTR
      DO 220 IY=1,NY
      IF (INPUT.EQ.1) READ (IS,530) (THCK(IX,IY),IX=1,NX)
      DO 210 IX=1,NX
      IF (INPUT.NE.1) GO TO 200
      THCK(IX,IY)=THCK(IX,IY)*FCTR
      GO TO 210
  200 IF (VPRM(IX,IY).NE.0.0) THCK(IX,IY)=FCTR
  210 CONTINUE
  220 WRITE (IO,490) (THCK(IX,IY),IX=1,NX)
C     ****************************************************************
C     ---READ DIFFUSE RECHARGE AND DISCHARGE---
      WRITE (IO,850)
      READ (IS,540) INPUT,FCTR
      DO 250 IY=1,NY
      IF (INPUT.EQ.1) READ (IS,550) (RECH(IX,IY),IX=1,NX)
      DO 240 IX=1,NX
      IF (INPUT.NE.1) GO TO 230
      RECH(IX,IY)=RECH(IX,IY)*FCTR
      GO TO 240
  230 IF (THCK(IX,IY).NE.0.0) RECH(IX,IY)=FCTR
  240 CONTINUE
  250 WRITE (IO,860) (RECH(IX,IY),IX=1,NX)
C     ****************************************************************
C     ---COMPUTE PERMEABILITY FROM TRANSMISSIVITY---
C     ---COUNT NO. OF CELLS IN AQUIFER---
      DO 260 IX=1,NX
      DO 260 IY=1,NY
      IF (THCK(IX,IY).EQ.0.0) GO TO 260
      PERM(IX,IY)=VPRM(IX,IY)/THCK(IX,IY)
      NCA=NCA+1
  260 VPRM(IX,IY)=0.0
C
      AAQ=NCA*AREA
      WRITE (IO,610)
      DO 270 IY=1,NY
  270 WRITE (IO,640) (PERM(IX,IY),IX=1,NX)
      WRITE (IO,620) NCA,AAQ
C     ****************************************************************
C     ---READ NODE IDENTIFICATION CARDS---
C        ---SET VERT. PERM., AND DIFFUSE RECHARGE---
C           ---SPECIFY CODES TO FIT YOUR NEEDS---
      WRITE (IO,560)
      READ (IS,540) INPUT,FCTR
      DO 290 IY=1,NY
      IF (INPUT.EQ.1) READ (IS,630) (NODEID(IX,IY),IX=1,NX)
      DO 280 IX=1,NX
  280 IF (INPUT.NE.1.AND.THCK(IX,IY).NE.0.0) NODEID(IX,IY)=FCTR
  290 WRITE (IO,570) (NODEID(IX,IY),IX=1,NX)
      WRITE (IO,920) NCODES
```

```
      IF (NCODES.LE.0) GO TO 320
      WRITE (IO,930)
      DO 310 IJ=1,NCODES
      READ (IS,870) ICODE,FCTR1,FCTR3,OVERRD
      DO 300 IX=1,NX
      DO 300 IY=1,NY
      IF (NODEID(IX,IY).NE.ICODE) GO TO 300
      VPRM(IX,IY)=FCTR1
      IF (OVERRD.NE.0) RECH(IX,IY)=FCTR3
  300 CONTINUE
      WRITE (IO,880) ICODE,FCTR1
  310 IF (OVERRD.NE.0) WRITE (IO,1100) FCTR3
  320 WRITE (IO,580)
      DO 330 IY=1,NY
  330 WRITE (IO,510) (VPRM(IX,IY),IX=1,NX)
C     ****************************************************************
C     ---READ WATER-TABLE ELEVATION---
      WRITE (IO,660)
      READ (IS,540) INPUT,FCTR
      DO 360 IY=1,NY
      IF (INPUT.EQ.1) READ (IS,650) (WT(IX,IY),IX=1,NX)
      DO 350 IX=1,NX
      IF (INPUT.NE.1) GO TO 340
      WT(IX,IY)=WT(IX,IY)*FCTR
      GO TO 350
  340 IF (THCK(IX,IY).NE.0.0) WT(IX,IY)=FCTR
  350 CONTINUE
  360 WRITE (IO,670) (WT(IX,IY),IX=1,NX)
C     ****************************************************************
C     ---SET INITIAL HEADS---
      DO 370 IX=1,NX
      DO 370 IY=1,NY
      HI(IX,IY)=WT(IX,IY)
      HC(IX,IY)=HI(IX,IY)
      HR(IX,IY)=HI(IX,IY)
  370 HK(IX,IY)=HI(IX,IY)
C
      CALL OUTPT
C     ****************************************************************
C     ---COMPUTE ITERATION PARAMETERS---
      DO 380 ID=1,20
      AOPT(ID)=0.0
  380 CONTINUE
      ANITP=NITP-1
      ALPHA1=DEXP(DLOG(1.0/HMIN)/ANITP)
      AOPT(1)=HMIN
      DO 390 IP=2,NITP
  390 AOPT(IP)=AOPT(IP-1)*ALPHA1
C
      WRITE (IO,440)
      WRITE (IO,450) AOPT
C     ****************************************************************
```

```
C     ---CHECK DATA SETS FOR INTERNAL CONSISTENCY---
      DO 410 IX=1,NX
      DO 410 IY=1,NY
      IF (THCK(IX,IY).GT.0.0) GO TO 400
      IF (TMRX(IX,IY,1).GT.0.0) WRITE (IO,940) IX,IY
      IF (TMRX(IX,IY,2).GT.0.0) WRITE (IO,950) IX,IY
      IF (NODEID(IX,IY).GT.0) WRITE (IO,960) IX,IY
      IF (WT(IX,IY).NE.0.0) WRITE (IO,970) IX,IY
      IF (RECH(IX,IY).NE.0.0) WRITE (IO,980) IX,IY
      IF (REC(IX,IY).NE.0.0) WRITE (IO,990) IX,IY
  400 IF (PERM(IX,IY).GT.0.0) GO TO 410
      IF (NODEID(IX,IY).GT.0.0) WRITE (IO,1000) IX,IY
      IF (WT(IX,IY).NE.0.0) WRITE (IO,1010) IX,IY
      IF (RECH(IX,IY).NE.0.0) WRITE (IO,1020) IX,IY
      IF (REC(IX,IY).NE.0.0) WRITE (IO,1030) IX,IY
      IF (THCK(IX,IY).GT.0.0) WRITE (IO,1040) IX,IY
  410 CONTINUE
C     ****************************************************************
      RETURN
C     ****************************************************************
C
C
C
  420 FORMAT (1I3)
  430 FORMAT (2F10.5)
  440 FORMAT (1H1,20HITERATION PARAMETERS)
  450 FORMAT (3H   ,1G20.6)
  460 FORMAT (1H1,27HTIME INTERVALS (IN SECONDS))
  470 FORMAT (1H1,15X,17HSTEADY-STATE FLOW//5X,57HTIME INTERVAL (IN SEC)
     1 FOR SOLUTE-TRANSPORT SIMULATION = ,G12.5)
  480 FORMAT (3H   ,10G12.5)
  490 FORMAT (3H   ,20F5.1)
  500 FORMAT (1H1,22HAQUIFER THICKNESS (FT))
  510 FORMAT (3H   ,20F5.2)
  520 FORMAT (1H1,30HTRANSMISSIVITY MAP (FT*FT/SEC))
  530 FORMAT (20G3.0)
  540 FORMAT (I1,G10.0)
  550 FORMAT (20G4.1)
  560 FORMAT (1H1,23HNODE IDENTIFICATION MAP//)
  570 FORMAT (1H ,20I5)
  580 FORMAT (1H1,45HVERTICAL PERMEABILITY/THICKNESS (FT/(FT*SEC)))
  590 FORMAT (1H0,10X,12HX-Y SPACING:)
  600 FORMAT (1H ,12X,10G12.5)
  610 FORMAT (1H1,24HPERMEABILTY MAP (FT/SEC))
  620 FORMAT (1H0,////10X,44HNO. OF FINITE-DIFFERENCE CELLS IN AQUIFER =
     1 ,I4//10X,28HAREA OF AQUIFER IN MODEL  = ,G12.5,10H   SQ. FT.////)
  630 FORMAT (20I1)
  640 FORMAT (3H   ,20F5.3)
  650 FORMAT (20G4.0)
  660 FORMAT (1H1,11HWATER TABLE)
  670 FORMAT (1H ,20F5.0)
  680 FORMAT (1H0,10X,19HAREA OF ONE CELL = ,G12.4)
```

```
690 FORMAT (2I2,1G8.2)
700 FORMAT (20A4)
710 FORMAT (1H0,20A4)
720 FORMAT (17I4)
730 FORMAT (1H1,66HU.S.G.S. FRONT-TRACKING MODEL FOR SOLUTE TRANSPORT
   1IN GROUND WATER)
740 FORMAT (1H0,21X,21HI N P U T     D A T A)
750 FORMAT (1H0,23X,16HGRID DESCRIPTORS//13X,30HNX    (NUMBER OF COLUM
   1NS)  =  ,I4/13X,28HNY    (NUMBER OF ROWS)     =,I6/13X,29HXDEL  (X
   2-DISTANCE IN FEET) = ,F7.1/13X,29HYDEL  (Y-DISTANCE IN FEET) = ,F7
   3.1)
760 FORMAT (1H0,23X,16HTIME  PARAMETERS//13X,40HNTIM   (MAX. NO. OF TI
   1ME STEPS)        = ,I6/13X,40HNPMP   (NO. OF PUMPING PERIODS)
   2 = ,I6/13X,39HPINT   (PUMPING PERIOD IN YEARS)      =,F10.2/13X,39
   3HTIMX   (TIME INCREMENT MULTIPLIER)    =,F10.2/13X,39HTINIT  (INIT
   4IAL TIME STEP IN SEC.)    =,F8.0)
770 FORMAT (1H0,14X,21HHYDROLOGIC PARAMETERS//13X,1HS,7X,29H(STORAGE C
   1OEFFICIENT)         =,5X,F9.6/13X,28HPOROS    (EFFECTIVE POROSITY),8X
   2,3H=  ,F8.2/13X,38HANFCTR  (RATIO OF T-YY TO T-XX)      = ,F12.6)
780 FORMAT (1H ,12X,46HINITIAL POSITION OF FRONT DEFINED BY FOLLOWING,
   1I4,8H POINTS:/40X,20HSET (=X)    SEY (=Y))
790 FORMAT (1H ,36X,F10.3,1X,F10.3)
800 FORMAT (10G5.0)
810 FORMAT (1H ,43X,I3)
820 FORMAT (1H0,4X,39HIDENTIFICATION NUMBERS OF TRACKED PTS.:)
830 FORMAT (1I3)
840 FORMAT (1H ,7X,2I4,3X,F9.4)
850 FORMAT (1H1,39HDIFFUSE RECHARGE AND DISCHARGE (FT/SEC))
860 FORMAT (1H ,1P10E10.2)
870 FORMAT (I2,2G10.2,I2)
880 FORMAT (1H0,7X,I2,7X,E10.3)
890 FORMAT (1H0,21X,20HEXECUTION PARAMETERS//13X,39HNITP   (NO. OF ITE
   1RATION PARAMETERS) = ,I4/13X,39HTOL    (CONVERGENCE CRITERIA - ADI
   2P) = ,F9.4/13X,39HITMAX  (MAX.NO.OF ITERATIONS - ADIP) = ,I4/13X,3
   34HCELDIS (MAX.CELL DISTANCE PER MOVE/24X,28HOF PARTICLES - M.O.C.)
   4    = ,F8.3/13X,32HNP     (NO. OF TRACER PARTICLES),5X,2H= ,I4)
900 FORMAT (1H0,23X,15HPROGRAM OPTIONS//13X,30HNPNT    (TIME STEP INTER
   1VAL FOR/21X,18HCOMPLETE PRINTOUT),7X,3H=  ,I4/13X,31HNPNTMV (MOVE
   2INTERVAL FOR FRONT/21X,28H     LOCATION PRINTOUT)  =  ,I4/13X,29HN
   3PNTVL (PRINT OPTION-VELOCITY/21X,24H0=NO; 1=FIRST TIME STEP;/21X,1
   47H2=ALL TIME STEPS),8X,3H=  ,I4/13X,35HNREC   (NO. OF PUMPING WELL
   5S)    = ,I5/13X,24HNCODES (FOR NODE IDENT.),9X,2H= ,I5/13X,35HNPRI
   6NP (PRINT OPTION-TRACER PTS) = ,I5/)
910 FORMAT (1H0,10X,28HLOCATION  OF  PUMPING  WELLS//11X,20HX    Y   RA
   1TE(IN CFS)/)
920 FORMAT (1H0,5X,37HNO. OF NODE IDENT. CODES SPECIFIED = ,I2)
930 FORMAT (1H0,10X,41HTHE FOLLOWING ASSIGNMENTS HAVE BEEN MADE:/5X,34
   1HCODE NO.     LEAKANCE     RECHARGE)
940 FORMAT (1H ,5X,61H*** WARNING ***   THCK.EQ.0.0 AND TMRX(X).GT.0.0
   1 AT NODE IX =,I4,6H, IY =,I4)
950 FORMAT (1H ,5X,61H*** WARNING ***   THCK.EQ.0.0 AND TMRX(Y).GT.0.0
   1 AT NODE IX =,I4,6H, IY =,I4)
```

```
  960 FORMAT (1H ,5X,61H*** WARNING ***    THCK.EQ.0.0 AND NODEID.GT.0.0
     1 AT NODE IX =,I4,6H, IY =,I4)
  970 FORMAT (1H ,5X,56H*** WARNING ***   THCK.EQ.0.0 AND WT.NE.0.0 AT N
     1ODE IX =,I4,6H, IY =,I4)
  980 FORMAT (1H ,5X,58H*** WARNING ***   THCK.EQ.0.0 AND RECH.NE.0.0 AT
     1 NODE IX =,I4,6H, IY =,I4)
  990 FORMAT (1H ,5X,58H*** WARNING ***    THCK.EQ.0.0 AND REC.NE.0.0 AT
     1 NODE IX =,I4,6H, IY =,I4)
 1000 FORMAT (1H ,5X,61H*** WARNING ***    PERM.EQ.0.0 AND NODEID.GT.0.0
     1 AT NODE IX =,I4,6H, IY =,I4)
 1010 FORMAT (1H ,5X,56H*** WARNING ***   PERM.EQ.0.0 AND WT.NE.0.0 AT N
     1ODE IX =,I4,6H, IY =,I4)
 1020 FORMAT (1H ,5X,58H*** WARNING ***   PERM.EQ.0.0 AND RECH.NE.0.0 AT
     1 NODE IX =,I4,6H, IY =,I4)
 1030 FORMAT (1H ,5X,58H*** WARNING ***    PERM.EQ.0.0 AND REC.NE.0.0 AT
     1 NODE IX =,I4,6H, IY =,I4)
 1040 FORMAT (1H ,5X,58H*** WARNING ***   PERM.EQ.0.0 AND THCK.GT.0.0 AT
     1 NODE IX =,I4,6H, IY =,I4)
 1050 FORMAT (1H0,5X,45H*** WARNING ***   ANFCTR WAS SPECIFIED AS 0.0/23
     1X,34HDEFAULT ACTION: RESET ANFCTR = 1.0)
 1060 FORMAT (1I1)
 1070 FORMAT (7I4,3G5.0)
 1080 FORMAT (1H1,5X,25HSTART PUMPING PERIOD NO. ,I2//2X,75HTHE FOLLOWIN
     1G TIME STEP, PUMPAGE, AND PRINT PARAMETERS HAVE BEEN REDEFINED:/)
 1090 FORMAT (1H0,14X,9HNTIM    = ,I4/15X,9HNPNT    = ,I4/15X,9HNITP    = ,
     1I4/15X,9HITMAX   = ,I4/15X,9HNREC    = ,I4/15X,9HNPNTMV = ,I4/15X,9H
     2NPNTVL = ,I4/,15X,9HPINT    = ,F10.3/15X,9HTIMX    = ,F10.3/15X,9HTI
     3NIT   = ,F10.3/)
 1100 FORMAT (1H ,27X,E10.3)
      END
      SUBROUTINE ITERAT
      DOUBLE PRECISION TMRX,VPRM,HI,HR,HC,HK,WT,REC,RECH,TIM,AOPT,TITLE
      DOUBLE PRECISION XDEL,YDEL,S,AREA,SUMT,RHO,PARAM,TEST,TOL,PINT,HMI
     1N,PYR,PYRSUM
      DOUBLE PRECISION B,G,W,A,C,E,F,DR,DC,TBAR,TMK,COEF,BLH,BRK,CHK,QL,
     1BRH
C     SPECIAL COMMON PRMI,PRMC,HEDA,HEDB
      COMMON IS,IO
      COMMON /PRMI/ NTIM,NPMP,NPNT,NITP,N,NX,NY,NP,NREC,INT,NNX,NNY,NMOV
     1,IMOV,ITMAX,NPNTMV,NPNTVL,NPRINP,NOBS
      COMMON /PRMC/ NODEID(20,20),INOBS(5)
      COMMON /HEDA/ THCK(20,20),PERM(20,20),ANFCTR
      COMMON /HEDB/ TMRX(20,20,2),VPRM(20,20),HI(20,20),HR(20,20),HC(20,
     120),HK(20,20),WT(20,20),REC(20,20),RECH(20,20),TIM(100),AOPT(20),T
     2ITLE(20),XDEL,YDEL,S,AREA,SUMT,RHO,PARAM,TEST,TOL,PINT,HMIN,PYR,PY
     3RSUM
      COMMON /BALM/ TOTLQ,TOTLQI,TPIN,TPOUT
      COMMON /XINV/ DXINV,DYINV,ARINV,PORINV
      DIMENSION W(20), B(20), G(20)
C     ***************************************************************
      KOUNT=0
C     ---COMPUTE ROW AND COLUMN---
```

```
C          ---CALL NEW ITERATION PARAMETER---
   10 REMN=MOD(KOUNT,NITP)
      IF (REMN.EQ.0) NTH=0
      NTH=NTH+1
      PARAM=AOPT(NTH)
C     ****************************************************************
C     ---ROW COMPUTATIONS---
      TEST=0.0
      RHO=S/TIM(N)
      BRK=-RHO
      DO 50 IY=1,NY
      DO 20 M=1,NX
      W(M)=0.0
      B(M)=0.0
      G(M)=0.0
   20 CONTINUE
      DO 30 IX=1,NX
      IF (THCK(IX,IY).EQ.0.0) GO TO 30
      COEF=VPRM(IX,IY)
      QL=-COEF*WT(IX,IY)
      A=TMRX(IX-1,IY,1)*DXINV
      C=TMRX(IX,IY,1)*DXINV
      E=TMRX(IX,IY-1,2)*DYINV
      F=TMRX(IX,IY,2)*DYINV
      TBAR=A+C+E+F
      TMK=TBAR*PARAM
      BLH=-A-C-RHO-COEF-TMK
      BRH=E+F-TMK
      DR=BRH*HC(IX,IY)+BRK*HK(IX,IY)-E*HC(IX,IY-1)-F*HC(IX,IY+1)+QL+RECH
     1(IX,IY)+REC(IX,IY)*ARINV
      W(IX)=BLH-A*B(IX-1)
      B(IX)=C/W(IX)
      G(IX)=(DR-A*G(IX-1))/W(IX)
   30 CONTINUE
C
C          ---BACK SUBSTITUTION---
      DO 40 J=2,NX
      IJ=J-1
      IA=NX-IJ
   40 HR(IA,IY)=G(IA)-B(IA)*HR(IA+1,IY)
   50 CONTINUE
C     ****************************************************************
C     ---COLUMN COMPUTATIONS---
      DO 90 IX=1,NX
      DO 60 M=1,NY
      W(M)=0.0
      B(M)=0.0
   60 G(M)=0.0
      DO 70 IY=1,NY
      IF (THCK(IX,IY).EQ.0.0) GO TO 70
      COEF=VPRM(IX,IY)
      QL=-COEF*WT(IX,IY)
```

```
      A=TMRX(IX,IY-1,2)*DYINV
      C=TMRX(IX,IY,2)*DYINV
      E=TMRX(IX-1,IY,1)*DXINV
      F=TMRX(IX,IY,1)*DXINV
      TBAR=A+C+E+F
      TMK=TBAR*PARAM
      BLH=-A-C-RHO-COEF-TMK
      BRH=E+F-TMK
      DC=BRH*HR(IX,IY)+BRK*HK(IX,IY)-E*HR(IX-1,IY)-F*HR(IX+1,IY)+QL+RECH
     1(IX,IY)+REC(IX,IY)*ARINV
      W(IY)=BLH-A*B(IY-1)
      B(IY)=C/W(IY)
      G(IY)=(DC-A*G(IY-1))/W(IY)
   70 CONTINUE
C
C        ---BACK SUBSTITUTION---
      DO 80 J=2,NY
      IJ=J-1
      IB=NY-IJ
      HC(IX,IB)=G(IB)-B(IB)*HC(IX,IB+1)
      IF (THCK(IX,IB).EQ.0.0) GO TO 80
      CHK=DABS(HC(IX,IB)-HR(IX,IB))
      IF (CHK.GT.TOL) TEST=1.0
   80 CONTINUE
   90 CONTINUE
C     *****************************************************************
      KOUNT=KOUNT+1
      IF (TEST.EQ.0.0) GO TO 120
      IF (KOUNT.GE.ITMAX) GO TO 100
      GO TO 10
C     ******************************************************************
C     ---TERMINATE PROGRAM -- ITMAX EXCEEDED---
  100 WRITE (IO,210)
      DO 110 IX=1,NX
      DO 110 IY=1,NY
  110 HK(IX,IY)=HC(IX,IY)
      CALL OUTPT
      STOP
C     ******************************************************************
C     ---SET NEW HEAD (HK)---
  120 DO 180 IY=1,NY
      DO 180 IX=1,NX
      IF (THCK(IX,IY).EQ.0.0) GO TO 180
      HR(IX,IY)=HK(IX,IY)
      HK(IX,IY)=HC(IX,IY)
C
C     ---COMPUTE LEAKAGE AND STRESSES FOR MASS BALANCE---
      IF (VPRM(IX,IY).EQ.0.0) GO TO 140
      DELQ=VPRM(IX,IY)*AREA*(WT(IX,IY)-HK(IX,IY))
      IF (DELQ.GT.0.0) GO TO 130
      TOTLQ=TOTLQ+DELQ*TIM(N)
      GO TO 140
```

```
  130 TOTLQI=TOTLQI+DELQ*TIM(N)
  140 IF (REC(IX,IY).GT.0.0) GO TO 150
      TPIN=TPIN+REC(IX,IY)*TIM(N)
      GO TO 160
  150 TPOUT=TPOUT+REC(IX,IY)*TIM(N)
  160 IF (RECH(IX,IY).GT.0.0) GO TO 170
      TPIN=TPIN+RECH(IX,IY)*AREA*TIM(N)
      GO TO 180
  170 TPOUT=TPOUT+RECH(IX,IY)*AREA*TIM(N)
  180 CONTINUE
C
      WRITE (IO,190) N
      WRITE (IO,200) KOUNT
C     ***************************************************************
      RETURN
C     ***************************************************************
C
C
C
  190 FORMAT (1H1//3X,4HN = ,1I4)
  200 FORMAT (1H ,2X,23HNUMBER OF ITERATIONS = ,1I4)
  210 FORMAT (1H0,5X,64H***   EXECUTION TERMINATED -- MAX. NO. ITERATION
     1S EXCEEDED   ***/26X,21HFINAL OUTPUT FOLLOWS:)
      END
      SUBROUTINE GENPT
      DOUBLE PRECISION TMRX,VPRM,HI,HR,HC,HK,WT,REC,RECH,TIM,AOPT,TITLE
      DOUBLE PRECISION XDEL,YDEL,S,AREA,SUMT,RHO,PARAM,TEST,TOL,PINT,HMI
     1N,PYR,PYRSUM
      DOUBLE PRECISION SP(100),ST,SI,SY,SX,SJ,REMAX,REMAY,REMAN
      INTEGER PTID
C     SPECIAL COMMON PRMI,PRMC,HEDA,HEDB,CHMA
      COMMON IS,IO
      COMMON /LSET/ SET(100),SEY(100)
      COMMON /GOIN/ INDT,I1,IT,KTOUT
      COMMON /PRMI/ NTIM,NPMP,NPNT,NITP,N,NX,NY,NP,NREC,INT,NNX,NNY,NMOV
     1,IMOV,ITMAX,NPNTMV,NPNTVL,NPRINP,NOBS
      COMMON /PRMC/ NODEID(20,20),INOBS(5)
      COMMON /HEDA/ THCK(20,20),PERM(20,20),ANFCTR
      COMMON /HEDB/ TMRX(20,20,2),VPRM(20,20),HI(20,20),HR(20,20),HC(20,
     120),HK(20,20),WT(20,20),REC(20,20),RECH(20,20),TIM(100),AOPT(20),T
     2ITLE(20),XDEL,YDEL,S,AREA,SUMT,RHO,PARAM,TEST,TOL,PINT,HMIN,PYR,PY
     3RSUM
      COMMON /CHMA/ PART(2,600),VX(20,20),VY(20,20),POROS,SUMTCH,TIMV,CE
     1LDIS
      DO 10 I=1,NP
      PART(1,I)=0.0
      PART(2,I)=0.0
   10 CONTINUE
      IF (INDT.EQ.1) GO TO 50
      I1=INDT-1
      ST=0.0
      DO 20 I=1,I1
```

```
      SP(I)=SQRT((SET(I)-SET(I+1))**2+(SEY(I)-SEY(I+1))**2)
      ST=ST+SP(I)
   20 CONTINUE
      M=0
      SJ=ST/(NP-1)
      REMAX=0.0
      REMAY=0.0
      REMAN=0.0
      DO 40 J=1,I1
      M=M+1
      PART(1,M)=SET(J)+REMAX
      PART(2,M)=SEY(J)+REMAY
      SI=((SP(J)-REMAN)/ST)*(NP-1)
      IT=SI
      SX=(SET(J+1)-PART(1,M))/SI
      SY=(SEY(J+1)-PART(2,M))/SI
      DO 30 K=1,IT
      M=M+1
      PART(1,M)=PART(1,M-1)+SX
      PART(2,M)=PART(2,M-1)+SY
   30 CONTINUE
      IF (J.EQ.I1) GO TO 40
      REMAN=SQRT((SET(J+1)-PART(1,M))**2+(SEY(J+1)-PART(2,M))**2)
      REMAN=SJ-REMAN
      REMAX=(SET(J+2)-SET(J+1))*REMAN/SP(J+1)
      REMAY=(SEY(J+2)-SEY(J+1))*REMAN/SP(J+1)
   40 CONTINUE
      PART(1,NP)=SET(INDT)
      PART(2,NP)=SEY(INDT)
      GO TO 90
   50 CONTINUE
      ST=XDEL/YDEL
      IF (ST.LE.1.0) GO TO 60
      SY=.25
      SX=.25*(1.0/ST)
      GO TO 70
   60 SX=.25
      SY=.25*ST
   70 SI=((2.0*3.1415927)/NP)
      DO 80 I=1,NP
      PART(1,I)=SET(1)+(SX*(COS(I*SI)))
   80 PART(2,I)=SEY(1)+(SY*(SIN(I*SI)))
   90 CONTINUE
      IF (NOBS.GT.0) WRITE (IO,120)
      DO 100 I=1,NOBS
  100 WRITE (IO,110) INOBS(I),PART(1,INOBS(I)),PART(2,INOBS(I))
      IF (INT.EQ.0) CALL CHMOT
C     ******************************************************************
      RETURN
C     ******************************************************************
C
C
```

```
  110 FORMAT (1H ,3X,7HPT. NO.,I3,5X,3H X=,F10.5,5X,3H Y=,F10.5)
  120 FORMAT (1H0,5X,32HINITIAL LOCATIONS OF TRACER PTS./)
      END
      SUBROUTINE VELO
      DOUBLE PRECISION TMRX,VPRM,HI,HR,HC,HK,WT,REC,RECH,TIM,AOPT,TITLE
      DOUBLE PRECISION XDEL,YDEL,S,AREA,SUMT,RHO,PARAM,TEST,TOL,PINT,HMI
     1N,PYR,PYRSUM
      DOUBLE PRECISION RATE,SLEAK,DIV
C     SPECIAL COMMON PRMI,PRMC,HEDA,HEDB,CHMA
      COMMON IS,IO
      COMMON /PRMI/ NTIM,NPMP,NPNT,NITP,N,NX,NY,NP,NREC,INT,NNX,NNY,NMOV
     1,IMOV,ITMAX,NPNTMV,NPNTVL,NPRINP,NOBS
      COMMON /PRMC/ NODEID(20,20),INOBS(5)
      COMMON /HEDA/ THCK(20,20),PERM(20,20),ANFCTR
      COMMON /HEDB/ TMRX(20,20,2),VPRM(20,20),HI(20,20),HR(20,20),HC(20,
     120),HK(20,20),WT(20,20),REC(20,20),RECH(20,20),TIM(100),AOPT(20),T
     2ITLE(20),XDEL,YDEL,S,AREA,SUMT,RHO,PARAM,TEST,TOL,PINT,HMIN,PYR,PY
     3RSUM
      COMMON /XINV/ DXINV,DYINV,ARINV,PORINV
      COMMON /CHMA/ PART(2,600),VX(20,20),VY(20,20),POROS,SUMTCH,TIMV,CE
     1LDIS
      COMMON /CHMC/ VXBDY(20,20),VYBDY(20,20)
C     **************************************************************
C     ---COMPUTE VELOCITIES AND STORE---
      VMAX=1.0E-10
      VMAY=1.0E-10
      VMXBD=1.0E-10
      VMYBD=1.0E-10
      TMV=TIM(N)
      LIM=0
C
      DO 10 IX=1,NX
      DO 10 IY=1,NY
C
      IF (THCK(IX,IY).EQ.0.0) GO TO 10
C
C        ---VELOCITIES AT NODES---
C           ---X-DIRECTION---
      GRDX=(HK(IX-1,IY)-HK(IX+1,IY))*DXINV*0.50
      IF (THCK(IX-1,IY).EQ.0.0) GRDX=(HK(IX,IY)-HK(IX+1,IY))*DXINV
      IF (THCK(IX+1,IY).EQ.0.0) GRDX=(HK(IX-1,IY)-HK(IX,IY))*DXINV
      IF (THCK(IX-1,IY).EQ.0.0.AND.THCK(IX+1,IY).EQ.0.0) GRDX=0.0
      VX(IX,IY)=PERM(IX,IY)*GRDX*PORINV
      ABVX=ABS(VX(IX,IY))
      IF (ABVX.GT.VMAX) VMAX=ABVX
C           ---Y-DIRECTION---
      GRDY=(HK(IX,IY-1)-HK(IX,IY+1))*DYINV*0.50
      IF (THCK(IX,IY-1).EQ.0.0) GRDY=(HK(IX,IY)-HK(IX,IY+1))*DYINV
      IF (THCK(IX,IY+1).EQ.0.0) GRDY=(HK(IX,IY-1)-HK(IX,IY))*DYINV
      IF (THCK(IX,IY-1).EQ.0.0.AND.THCK(IX,IY+1).EQ.0.0) GRDY=0.0
      VY(IX,IY)=PERM(IX,IY)*GRDY*PORINV*ANFCTR
      ABVY=ABS(VY(IX,IY))
```

```
      IF (ABVY.GT.VMAY) VMAY=ABVY
C
C        ---VELOCITIES AT CELL BOUNDARIES---
      GRDX=(HK(IX,IY)-HK(IX+1,IY))*DXINV
      PERMX=2.0*PERM(IX,IY)*PERM(IX+1,IY)/(PERM(IX,IY)+PERM(IX+1,IY))
      VXBDY(IX,IY)=PERMX*GRDX*PORINV
      GRDY=(HK(IX,IY)-HK(IX,IY+1))*DYINV
      PERMY=2.0*PERM(IX,IY)*PERM(IX,IY+1)/(PERM(IX,IY)+PERM(IX,IY+1))
      VYBDY(IX,IY)=PERMY*GRDY*PORINV*ANFCTR
      ABVX=ABS(VXBDY(IX,IY))
      ABVY=ABS(VYBDY(IX,IY))
      IF (ABVX.GT.VMXBD) VMXBD=ABVX
      IF (ABVY.GT.VMYBD) VMYBD=ABVY
C
   10 CONTINUE
C     *****************************************************************
C     ---PRINT VELOCITIES---
      IF (NPNTVL.EQ.0) GO TO 70
      IF (NPNTVL.EQ.2) GO TO 20
      IF (NPNTVL.EQ.1.AND.N.EQ.1) GO TO 20
      GO TO 70
   20 WRITE (IO,110)
      WRITE (IO,120)
      DO 30 IY=1,NY
   30 WRITE (IO,140) (VX(IX,IY),IX=1,NX)
      WRITE (IO,130)
      DO 40 IY=1,NY
   40 WRITE (IO,140) (VXBDY(IX,IY),IX=1,NX)
      WRITE (IO,150)
      WRITE (IO,120)
      DO 50 IY=1,NY
   50 WRITE (IO,140) (VY(IX,IY),IX=1,NX)
      WRITE (IO,130)
      DO 60 IY=1,NY
   60 WRITE (IO,140) (VYBDY(IX,IY),IX=1,NX)
   70 CONTINUE
C     *****************************************************************
C     ---COMPUTE NEXT TIME STEP---
      WRITE (IO,180)
      WRITE (IO,190) VMAX,VMAY
      WRITE (IO,200) VMXBD,VMYBD
      TDELX=CELDIS*XDEL/VMAX
      TDELY=CELDIS*YDEL/VMAY
      TDELXB=CELDIS*XDEL/VMXBD
      TDELYB=CELDIS*YDEL/VMYBD
      TIMV=AMIN1(TDELX,TDELY,TDELXB,TDELYB)
      WRITE (IO,100) TIMV
      IF (TMV.LT.TIMV) GO TO 80
      LIM=-1
      GO TO 90
   80 TIMV=TMV
      LIM=1
```

```
   90 NTIMV=TIM(N)/TIMV
      NMOV=NTIMV+1
      WRITE (IO,210) TIMV,NTIMV,NMOV
      TIMV=TIM(N)/NMOV
      WRITE (IO,160) TIM(N)
      WRITE (IO,170) TIMV
C
C     ***************************************************************
      RETURN
C     ***************************************************************
C
C
C
  100 FORMAT (1H ,20H  TIMV (CELDIS)   = ,G12.5)
  110 FORMAT (1H1,12HX VELOCITIES)
  120 FORMAT (1H ,25X,8HAT NODES/)
  130 FORMAT (1H0,25X,13HON BOUNDARIES/)
  140 FORMAT (1H ,10G12.3)
  150 FORMAT (1H1,12HY VELOCITIES)
  160 FORMAT (3H   ,11HTIM (N)  = ,1G12.5)
  170 FORMAT (3H   ,11HTIMEVELO = ,1G12.5)
  180 FORMAT (1H1,10X,29HSTABILITY CRITERIA --- M.O.C.//)
  190 FORMAT (1H0,8H VMAX = ,1PE9.2,5X,7HVMAY = ,1PE9.2)
  200 FORMAT (1H ,8H VMXBD= ,1PE9.2,5X,7HVMYBD= ,1PE9.2)
  210 FORMAT (1H0,8H TIMV = ,1PE9.2,5X,8HNTIMV = ,I5,5X,7HNMOV = ,I5/)
      END
      SUBROUTINE MOVE
      DOUBLE PRECISION TMRX,VPRM,HI,HR,HC,HK,WT,REC,RECH,TIM,AOPT,TITLE
      DOUBLE PRECISION XDEL,YDEL,S,AREA,SUMT,RHO,PARAM,TEST,TOL,PINT,HMI
     1N,PYR,PYRSUM
      INTEGER PTID
C     SPECIAL COMMON PRMI,PRMC,HEDA,HEDB,CHMA
      COMMON IS,IO
      COMMON /PRMI/ NTIM,NPMP,NPNT,NITP,N,NX,NY,NP,NREC,INT,NNX,NNY,NMOV
     1,IMOV,ITMAX,NPNTMV,NPNTVL,NPRINP,NOBS
      COMMON /PRMC/ NODEID(20,20),INOBS(5)
      COMMON /HEDA/ THCK(20,20),PERM(20,20),ANFCTR
      COMMON /HEDB/ TMRX(20,20,2),VPRM(20,20),HI(20,20),HR(20,20),HC(20,
     120),HK(20,20),WT(20,20),REC(20,20),RECH(20,20),TIM(100),AOPT(20),T
     2ITLE(20),XDEL,YDEL,S,AREA,SUMT,RHO,PARAM,TEST,TOL,PINT,HMIN,PYR,PY
     3RSUM
      COMMON /XINV/ DXINV,DYINV,ARINV,PORINV
      COMMON /CHMA/ PART(2,600),VX(20,20),VY(20,20),POROS,SUMTCH,TIMV,CE
     1LDIS
      COMMON /CHMC/ VXBDY(20,20),VYBDY(20,20)
      DIMENSION XNEW(4), YNEW(4), DIST(4)
C     ***************************************************************
      WRITE (IO,380) NMOV
      SUMTCH=SUMT-TIM(N)
      CONST1=TIMV*DXINV
      CONST2=TIMV*DYINV
C     ---MOVE PARTICLES 'NMOV' TIMES---
```

```
      DO 360 IMOV=1,NMOV
      NPTM=NP
C        ---MOVE EACH PARTICLE---
      DO 320 IN=1,NP
      IF (PART(1,IN).EQ.0.0) GO TO 320
C     ***************************************************************
C           ---COMPUTE OLD LOCATION---
      XOLD=PART(1,IN)
      IX=XOLD+0.5
      YOLD=PART(2,IN)
      IY=YOLD+0.5
      IF (THCK(IX,IY).EQ.0.0) GO TO 320
C     ***************************************************************
C           ---COMPUTE NEW LOCATION AND LOCATE CLOSEST NODE---
C           ---LOCATE NORTHWEST CORNER---
      IVX=XOLD
      IVY=YOLD
      IXE=IVX+1
      IYS=IVY+1
C     ***************************************************************
C        ---LOCATE QUADRANT, VEL. AT 4 CORNERS, CHECK FOR BOUNDARIES---
      CELDX=XOLD-IX
      CELDY=YOLD-IY
      IF (CELDX.EQ.0.0.AND.CELDY.EQ.0.0) GO TO 250
      IF (CELDX.GE.0.0.OR.CELDY.GE.0.0) GO TO 40
C              ---PT. IN NW QUADRANT---
      VXNW=VXBDY(IVX,IVY)
      VXNE=VX(IXE,IVY)
      VXSW=VXBDY(IVX,IYS)
      VXSE=VX(IXE,IYS)
      VYNW=VYBDY(IVX,IVY)
      VYNE=VYBDY(IXE,IVY)
      VYSW=VY(IVX,IYS)
      VYSE=VY(IXE,IYS)
      IF (THCK(IVX,IVY).EQ.0.0) GO TO 20
      IF (REC(IXE,IVY).EQ.0.0.AND.VPRM(IXE,IVY).LT.0.09) GO TO 10
      VXNE=VXNW
   10 IF (REC(IVX,IYS).EQ.0.0.AND.VPRM(IVX,IYS).LT.0.09) GO TO 20
      VYSW=VYNW
   20 IF (REC(IXE,IYS).EQ.0.0.AND.VPRM(IXE,IYS).LT.0.09) GO TO 240
      IF (THCK(IVX,IYS).EQ.0.0) GO TO 30
      VXSE=VXSW
   30 IF (THCK(IXE,IVY).EQ.0.0) GO TO 240
      VYSE=VYNE
      GO TO 240
C
   40 IF (CELDX.LE.0.0.OR.CELDY.GE.0.0) GO TO 100
C              ---PT. IN NE QUADRANT---
   50 VXNW=VX(IVX,IVY)
      VXNE=VXBDY(IVX,IVY)
      VXSW=VX(IVX,IYS)
      VXSE=VXBDY(IVX,IYS)
```

```
      VYNW=VYBDY(IVX,IVY)
      VYNE=VYBDY(IXE,IVY)
      VYSW=VY(IVX,IYS)
      VYSE=VY(IXE,IYS)
      IF (CELDX.EQ.0.0) GO TO 90
      IF (THCK(IXE,IVY).EQ.0.0) GO TO 70
      IF (REC(IVX,IVY).EQ.0.0.AND.VPRM(IVX,IVY).LT.0.09) GO TO 60
      VXNW=VXNE
   60 IF (REC(IXE,IYS).EQ.0.0.AND.VPRM(IXE,IYS).LT.0.09) GO TO 70
      VYSE=VYNE
   70 IF (REC(IVX,IYS).EQ.0.0.AND.VPRM(IVX,IYS).LT.0.09) GO TO 240
      IF (THCK(IXE,IYS).EQ.0.0) GO TO 80
      VXSW=VXSE
   80 IF (THCK(IVX,IVY).EQ.0.0) GO TO 240
      VYSW=VYNW
      GO TO 240
   90 IF (REC(IVX,IYS).EQ.0.0.AND.VPRM(IVX,IYS).LE.0.09) GO TO 240
      IF (THCK(IVX,IVY).EQ.0.0) GO TO 240
      VYSW=VYNW
      GO TO 240
C
  100 IF (CELDY.LE.0.0.OR.CELDX.GE.0.0) GO TO 160
C                ---PT. IN SW QUADRANT---
  110 VXNW=VXBDY(IVX,IVY)
      VXNE=VX(IXE,IVY)
      VXSW=VXBDY(IVX,IYS)
      VXSE=VX(IXE,IYS)
      VYNW=VY(IVX,IVY)
      VYNE=VY(IXE,IVY)
      VYSW=VYBDY(IVX,IVY)
      VYSE=VYBDY(IXE,IVY)
      IF (CELDY.EQ.0.0) GO TO 150
      IF (THCK(IVX,IYS).EQ.0.0) GO TO 130
      IF (REC(IVX,IVY).EQ.0.0.AND.VPRM(IVX,IVY).LT.0.09) GO TO 120
      VYNW=VYSW
  120 IF (REC(IXE,IYS).EQ.0.0.AND.VPRM(IXE,IYS).LT.0.09) GO TO 130
      VXSE=VXSW
  130 IF (REC(IXE,IVY).EQ.0.0.AND.VPRM(IXE,IVY).LT.0.09) GO TO 240
      IF (THCK(IVX,IVY).EQ.0.0) GO TO 140
      VXNE=VXNW
  140 IF (THCK(IXE,IYS).EQ.0.0) GO TO 240
      VYNE=VYSE
      GO TO 240
  150 IF (REC(IXE,IVY).EQ.0.0.AND.VPRM(IXE,IVY).LE.0.09) GO TO 240
      IF (THCK(IVX,IVY).EQ.0.0) GO TO 240
      VXNE=VXNW
      GO TO 240
C
  160 IF (CELDY.LE.0.0.OR.CELDX.LE.0.0) GO TO 230
C                ---PT. IN SE QUADRANT---
  170 VXNW=VX(IVX,IVY)
      VXNE=VXBDY(IVX,IVY)
```

```
      VXSW=VX(IVX,IYS)
      VXSE=VXBDY(IVX,IYS)
      VYNW=VY(IVX,IVY)
      VYNE=VY(IXE,IVY)
      VYSW=VYBDY(IVX,IVY)
      VYSE=VYBDY(IXE,IVY)
      IF (CELDY.EQ.0.0) GO TO 210
      IF (CELDX.EQ.0.0) GO TO 220
      IF (THCK(IXE,IYS).EQ.0.0) GO TO 190
      IF (REC(IXE,IVY).EQ.0.0.AND.VPRM(IXE,IVY).LT.0.09) GO TO 180
      VYNE=VYSE
  180 IF (REC(IVX,IYS).EQ.0.0.AND.VPRM(IVX,IYS).LT.0.09) GO TO 190
      VXSW=VXSE
  190 IF (REC(IVX,IVY).EQ.0.0.AND.VPRM(IVX,IVY).LT.0.09) GO TO 240
      IF (THCK(IXE,IVY).EQ.0.0) GO TO 200
      VXNW=VXNE
  200 IF (THCK(IVX,IYS).EQ.0.0) GO TO 240
      VYNW=VYSW
      GO TO 240
  210 IF (REC(IVX,IVY).EQ.0.0.AND.VPRM(IVX,IVY).LE.0.09) GO TO 240
      IF (THCK(IXE,IVY).EQ.0.0) GO TO 240
      VXNW=VXNE
      GO TO 240
  220 IF (REC(IVX,IVY).EQ.0.0.AND.VPRM(IVX,IVY).LE.0.09) GO TO 240
      IF (THCK(IVX,IYS).EQ.0.0) GO TO 240
      VYNW=VYSW
      GO TO 240
C
  230 IF (CELDX.EQ.0.0.AND.CELDY.LT.0.0) GO TO 50
      IF (CELDX.LT.0.0.AND.CELDY.EQ.0.0) GO TO 110
      IF (CELDX.GT.0.0.AND.CELDY.EQ.0.0) GO TO 170
      IF (CELDX.EQ.0.0.AND.CELDY.GT.0.0) GO TO 170
  240 CONTINUE
C     ****************************************************************
C        ---BILINEAR INTERPOLATION---
      CELXD=XOLD-IVX
      CELDXH=AMOD(CELXD,0.5)
      CELDX=CELDXH*2.0
      CELDY=YOLD-IVY
C     ****************************************************************
C           ---X VELOCITY---
      VXN=VXNW*(1.0-CELDX)+VXNE*CELDX
      IF (THCK(IVX,IVY).EQ.0.0.OR.THCK(IXE,IVY).EQ.0.0) VXN=VXNW+VXNE
      VXS=VXSW*(1.0-CELDX)+VXSE*CELDX
      IF (THCK(IVX,IYS).EQ.0.0.OR.THCK(IXE,IYS).EQ.0.0) VXS=VXSW+VXSE
      XVEL=VXN*(1.0-CELDY)+VXS*CELDY
      IF (THCK(IVX,IVY).EQ.0.0.AND.THCK(IXE,IVY).EQ.0.0) XVEL=VXS
      IF (THCK(IVX,IYS).EQ.0.0.AND.THCK(IXE,IYS).EQ.0.0) XVEL=VXN
C           ---Y VELOCITY---
      CELDYH=AMOD(CELDY,0.5)
      CELDY=CELDYH*2.0
      VYW=VYNW*(1.0-CELDY)+VYSW*CELDY
```

```
      IF (THCK(IVX,IVY).EQ.0.0.OR.THCK(IVX,IYS).EQ.0.0) VYW=VYNW+VYSW
      VYE=VYNE*(1.0-CELDY)+VYSE*CELDY
      IF (THCK(IXE,IVY).EQ.0.0.OR.THCK(IXE,IYS).EQ.0.0) VYE=VYNE+VYSE
      YVEL=VYW*(1.0-CELXD)+VYE*CELXD
      IF (THCK(IVX,IVY).EQ.0.0.AND.THCK(IVX,IYS).EQ.0.0) YVEL=VYE
      IF (THCK(IXE,IVY).EQ.0.0.AND.THCK(IXE,IYS).EQ.0.0) YVEL=VYW
C
      GO TO 260
  250 XVEL=VX(IX,IY)
      YVEL=VY(IX,IY)
  260 DISTX=XVEL*CONST1
      DISTY=YVEL*CONST2
C     ****************************************************************
C        ---BOUNDARY CONDITIONS---
      TEMPX=XOLD+DISTX
      TEMPY=YOLD+DISTY
      INX=TEMPX+0.5
      INY=TEMPY+0.5
      IF (THCK(INX,INY).GT.0.0) GO TO 300
C     ****************************************************************
C           ---X BOUNDARY---
      IF (THCK(INX,IY).EQ.0.0) GO TO 270
      PART(1,IN)=TEMPX
      GO TO 280
  270 BEYON=TEMPX-IX
      IF (BEYON.LT.0.0) BEYON=BEYON+0.5
      IF (BEYON.GT.0.0) BEYON=BEYON-0.5
      PART(1,IN)=TEMPX-2.0*BEYON
      INX=PART(1,IN)+0.5
      TEMPX=PART(1,IN)
C     ****************************************************************
C           ---Y BOUNDARY---
  280 IF (THCK(INX,INY).EQ.0.0) GO TO 290
      PART(2,IN)=TEMPY
      GO TO 310
C     ****************************************************************
  290 BEYON=TEMPY-IY
      IF (BEYON.LT.0.0) BEYON=BEYON+0.5
      IF (BEYON.GT.0.0) BEYON=BEYON-0.5
      PART(2,IN)=TEMPY-2.0*BEYON
      INY=PART(2,IN)+0.5
      TEMPY=PART(2,IN)
      GO TO 310
  300 PART(1,IN)=TEMPX
      PART(2,IN)=TEMPY
  310 CONTINUE
  320 CONTINUE
C     ****************************************************************
      SUMTCH=SUMTCH+TIMV
C     ****************************************************************
      IF (NOBS.LT.1) GO TO 340
      WRITE (IO,390) NOBS,IMOV
```

```
      WRITE (IO,410)
      DO 330 I=1,NOBS
  330 WRITE (IO,400) INOBS(I),PART(1,INOBS(I)),PART(2,INOBS(I))
  340 IF (IMOV.GE.NMOV) GO TO 370
      IF (MOD(IMOV,NPNTMV).EQ.0) GO TO 350
      GO TO 360
  350 CALL CHMOT
  360 CONTINUE
C     ****************************************************************
  370 RETURN
C     ****************************************************************
C
C
C
  380 FORMAT (1H0,10X,61HNO. OF PARTICLE MOVES REQUIRED TO COMPLETE THIS
     1 TIME STEP  = ,I4//)
  390 FORMAT (1H0,2X,4HNOBS,2X,2H= ,2X,I4,10X,11HIMOV      = ,2X,I4)
  400 FORMAT (1H ,3X,7HPT. NO.,1I3,5X,3H X=,1F10.5,5X,3H Y=,1F10.5)
  410 FORMAT (1H0,25HLOCATIONS OF TRACKED PTS./)
      END
      SUBROUTINE OUTPT
      DOUBLE PRECISION TMRX,VPRM,HI,HR,HC,HK,WT,REC,RECH,TIM,AOPT,TITLE
      DOUBLE PRECISION XDEL,YDEL,S,AREA,SUMT,RHO,PARAM,TEST,TOL,PINT,HMI
     1N,PYR,PYRSUM
C     SPECIAL COMMON PRMI,PRMC,HEDA,HEDB
      COMMON IS,IO
      COMMON /GOIN/ INDT,I1,IT,KTOUT
      COMMON /PRMI/ NTIM,NPMP,NPNT,NITP,N,NX,NY,NP,NREC,INT,NNX,NNY,NMOV
     1,IMOV,ITMAX,NPNTMV,NPNTVL,NPRINP,NOBS
      COMMON /PRMC/ NODEID(20,20),INOBS(5)
      COMMON /HEDA/ THCK(20,20),PERM(20,20),ANFCTR
      COMMON /HEDB/ TMRX(20,20,2),VPRM(20,20),HI(20,20),HR(20,20),HC(20,
     120),HK(20,20),WT(20,20),REC(20,20),RECH(20,20),TIM(100),AOPT(20),T
     2ITLE(20),XDEL,YDEL,S,AREA,SUMT,RHO,PARAM,TEST,TOL,PINT,HMIN,PYR,PY
     3RSUM
      COMMON /BALM/ TOTLQ,TOTLQI,TPIN,TPOUT
      DIMENSION IH(20)
C     ****************************************************************
      KTOUT=1
      TIMD=SUMT/86400.
      TIMY=SUMT/(86400.0*365.25)
C     ---PRINT HEAD VALUES---
      WRITE (IO,150)
      WRITE (IO,160) N
      WRITE (IO,170) SUMT
      WRITE (IO,180) TIMD
      WRITE (IO,190) TIMY
      WRITE (IO,200)
      DO 10 IY=1,NY
   10 WRITE (IO,210) (HK(IX,IY),IX=1,NX)
      IF (N.EQ.0) GO TO 140
C     ****************************************************************
```

```
C     ---PRINT HEAD MAP---
      WRITE (IO,150)
      WRITE (IO,160) N
      WRITE (IO,170) SUMT
      WRITE (IO,180) TIMD
      WRITE (IO,190) TIMY
      WRITE (IO,200)
      DO 30 IY=1,NY
      DO 20 IX=1,NX
   20 IH(IX)=HK(IX,IY)+0.5
   30 WRITE (IO,220) (IH(ID),ID=1,NX)
C     ****************************************************************
C     ---COMPUTE WATER BALANCE AND DRAWDOWN---
      QSTR=0.0
      PUMP=0.0
      PQIN=0.0
      PQOUT=0.0
      TPUM=0.0
      QIN=0.0
      QOUT=0.0
      QNET=0.0
      DELQ=0.0
      PCTERR=0.0
      WRITE (IO,360)
C
      DO 120 IY=1,NY
      DO 110 IX=1,NX
      IH(IX)=0.0
      IF (THCK(IX,IY).EQ.0.0) GO TO 110
      IF (REC(IX,IY).GT.0.0) GO TO 40
      PQIN=PQIN+REC(IX,IY)
      GO TO 50
   40 PQOUT=PQOUT+REC(IX,IY)
   50 IF (RECH(IX,IY).GT.0.0) GO TO 60
      PQIN=PQIN+RECH(IX,IY)*AREA
      GO TO 70
   60 PQOUT=PQOUT+RECH(IX,IY)*AREA
   70 IF (VPRM(IX,IY).EQ.0.0) GO TO 100
      DELQ=VPRM(IX,IY)*AREA*(WT(IX,IY)-HK(IX,IY))
      IF (DELQ.GT.0.0) GO TO 80
      QOUT=QOUT+DELQ
      GO TO 90
   80 QIN=QIN+DELQ
   90 QNET=QNET+DELQ
  100 DDRW=HI(IX,IY)-HK(IX,IY)
      IH(IX)=DDRW+0.5
      QSTR=QSTR+DDRW*AREA*S
  110 CONTINUE
C        ---PRINT DRAWDOWN MAP---
      WRITE (IO,370) (IH(IX),IX=1,NX)
  120 CONTINUE
      TPUM=TPUM+PQIN+PQOUT
```

```
      PUMP=TPUM*SUMT
      TOTLQN=TOTLQ+TOTLQI
      SRCS=QSTR-TPIN+TOTLQI
      SINKS=TPOUT-TOTLQ
      ERRMB=SRCS-SINKS
      DENOM=(SRCS+SINKS)*0.5
      IF (DENOM.EQ.0.0) GO TO 130
      PCTERR=ERRMB*100.0/DENOM
C        ---PRINT MASS BALANCE DATA FOR FLOW MODEL---
  130 WRITE (IO,310)
      WRITE (IO,280) TPIN
      WRITE (IO,290) TPOUT
      WRITE (IO,320) PUMP
      WRITE (IO,300) QSTR
      WRITE (IO,240) TOTLQI
      WRITE (IO,250) TOTLQ
      WRITE (IO,330) TOTLQN
      WRITE (IO,340) ERRMB
      WRITE (IO,350) PCTERR
      WRITE (IO,230)
      WRITE (IO,240) QIN
      WRITE (IO,250) QOUT
      WRITE (IO,260) QNET
      WRITE (IO,280) PQIN
      WRITE (IO,290) PQOUT
      WRITE (IO,270) TPUM
C     **************************************************************
  140 RETURN
C     **************************************************************
C
C
C
  150 FORMAT (1H1,23HHEAD DISTRIBUTION - ROW)
  160 FORMAT (1X,23HNUMBER OF TIME STEPS = ,1I5)
  170 FORMAT (8X,16HTIME(SECONDS) = ,1G12.5)
  180 FORMAT (8X,16HTIME(DAYS)    = ,1E12.5)
  190 FORMAT (8X,16HTIME(YEARS)   = ,1E12.5)
  200 FORMAT (1H )
  210 FORMAT (1H0,10F12.7)
  220 FORMAT (1H0,20I4)
  230 FORMAT (1H0,2X,33HRATE MASS BALANCE -- (IN C.F.S.) //)
  240 FORMAT (4X,29HLEAKAGE INTO AQUIFER         = ,E12.5)
  250 FORMAT (4X,29HLEAKAGE OUT OF AQUIFER       = ,E12.5)
  260 FORMAT (4X,29HNET LEAKAGE    (QNET)        = ,E12.5)
  270 FORMAT (4X,29HNET WITHDRAWAL   (TPUM)      = ,E12.5)
  280 FORMAT (4X,29HRECHARGE AND INJECTION       = ,E12.5)
  290 FORMAT (4X,29HPUMPAGE AND E-T WITHDRAWAL = ,E12.5)
  300 FORMAT (4X,29HWATER RELEASE FROM STORAGE = ,1E12.5)
  310 FORMAT (1H0,2X,38HCUMULATIVE MASS BALANCE -- (IN FT**3) //)
  320 FORMAT (4X,29HCUMULATIVE  NET  PUMPAGE   = ,1E12.5)
  330 FORMAT (4X,29HCUMULATIVE  NET  LEAKAGE   = ,1E12.5)
  340 FORMAT (1H0,7X,25HMASS BALANCE RESIDUAL  = ,G12.5)
```

```
  350 FORMAT (1H ,7X,25HERROR  (AS PERCENT)    = ,G12.5/)
  360 FORMAT (1H1,8HDRAWDOWN)
  370 FORMAT (3H   ,2015)
      END
      SUBROUTINE CHMOT
      DOUBLE PRECISION TMRX,VPRM,HI,HR,HC,HK,WT,REC,RECH,TIM,AOPT,TITLE
      DOUBLE PRECISION XDEL,YDEL,S,AREA,SUMT,RHO,PARAM,TEST,TOL,PINT,HMI
     1N,PYR,PYRSUM
      DOUBLE PRECISION ECRA,ECRD,HITA,HITX,HITY,HIND,RATYO,RTZ,RTZ1,RTZ2
     1,RTZ3
      INTEGER STAR(140)
      COMMON IS,IO
      COMMON /PRMI/ NTIM,NPMP,NPNT,NITP,N,NX,NY,NP,NREC,INT,NNX,NNY,NMOV
     1,IMOV,ITMAX,NPNTMV,NPNTVL,NPRINP,NOBS
      COMMON /PRMC/ NODEID(20,20),INOBS(5)
      COMMON /GOIN/ INDT,I1,IT,KTOUT
      COMMON /HEDA/ THCK(20,20),PERM(20,20),ANFCTR
      COMMON /HEDB/ TMRX(20,20,2),VPRM(20,20),HI(20,20),HR(20,20),HC(20,
     120),HK(20,20),WT(20,20),REC(20,20),RECH(20,20),TIM(100),AOPT(20),T
     2ITLE(20),XDEL,YDEL,S,AREA,SUMT,RHO,PARAM,TEST,TOL,PINT,HMIN,PYR,PY
     3RSUM
      COMMON /CHMA/ PART(2,600),VX(20,20),VY(20,20),POROS,SUMTCH,TIMV,CE
     1LDIS
C     ******************************************************************
      DIMENSION ALPHA(42), ANDEX(131), AMDEX(131)
      DATA ALPHA/'1','2','3','4','5','6','7','8','9','A','B','C','D','E'
     1,'F','G','H','I','J','K','L','M','N','O','P','Q','R','S','T','U','
     2V','W','X','Y','Z','-',' ','+','*','!','\','/'/
      TCHYR=SUMTCH/(86400.0*365.25)
      IF (NPRINP.EQ.1) GO TO 10
      IF (NPRINP.EQ.2.AND.IMOV.GE.NMOV) GO TO 10
      IF (NPRINP.EQ.3.AND.IMOV.GE.NMOV.AND.N.EQ.NTIM) GO TO 10
      GO TO 30
   10 WRITE (IO,360)
      DO 20 I=1,NP
   20 WRITE (IO,370) I,PART(1,I),PART(2,I)
   30 CONTINUE
      IF ((NX*XDEL).LE.(NY*YDEL)) GO TO 40
      GO TO 50
   40 NAP=NX-2
      NDP=NY-2
      MO=1
      NO=2
      RATYO=(XDEL/YDEL)
      GO TO 60
   50 NAP=NY-2
      NDP=NX-2
      MO=2
      NO=1
      RATYO=(YDEL/XDEL)
   60 DO 70 I=1,120
      NTZ=I*NAP
```

```
      IF (NTZ.GT.128) GO TO 80
   70 CONTINUE
   80 NRT=I-1
      NTZ=NRT*NAP
      NTZ3=NTZ+3
      NTZ2=NTZ+2
      NTZ1=NTZ+1
      RTZ=FLOAT(NTZ)
      ECRA=(1.0/NRT)
      ECRD=ECRA*RATYO/0.6
      NAKER=(1.0/ECRD+0.5)
      NL=NDP*NAKER
      ECRD=1.0/NAKER
      WRITE (IO,400)
      WRITE (IO,410) TCHYR
      WRITE (IO,430)
      IF (MO.EQ.1) GO TO 90
      WRITE (IO,420)
      GO TO 100
   90 WRITE (IO,390)
  100 DO 110 I=1,NTZ3
  110 ANDEX(I)=ALPHA(37)
      DO 130 I=3,NTZ3
      I2=I-2
      DO 130 J=1,NAP
      MRT=NRT/2
      IF (I2.EQ.(NRT*J-MRT)) GO TO 120
      GO TO 130
  120 ANDEX(I)=ALPHA(J+1)
      IF (MO.EQ.2) ANDEX(I)=ALPHA(NAP-J+2)
  130 CONTINUE
      WRITE (IO,380) ANDEX
      DO 140 I=2,NTZ3
  140 AMDEX(I)=ALPHA(36)
      AMDEX(1)=ALPHA(37)
      NOP=1
      IF (MOD(NRT,2).EQ.0) NOP=2
      IF (NOP.EQ.1) GO TO 150
      GO TO 170
  150 DO 160 I=1,NTZ
      HITA=(I+(.5*NRT))
      DO 160 J=1,NAP
      HIND=FLOAT(J*NRT)
      IF (HITA.GT.HIND.AND.(HITA-1.0).LT.HIND) AMDEX(I+2)=ALPHA(38)
  160 CONTINUE
      GO TO 210
  170 DO 200 I=1,NTZ
      HITA=(I+(.5*NRT))
      DO 180 J=1,NAP
      HIND=FLOAT(J*NRT)
      IF (HITA.EQ.HIND) GO TO 190
  180 CONTINUE
```

```
      GO TO 200
  190 AMDEX(I+2)=ALPHA(41)
      AMDEX(I+3)=ALPHA(42)
  200 CONTINUE
  210 WRITE (IO,380) AMDEX
      DO 350 I=1,NL
      L=I-1
      DO 220 M=1,NTZ3
  220 STAR(M)=0
      DO 250 J=1,NP
      IF (PART(NO,J).GE.(L*ECRD+1.5).AND.PART(NO,J).LT.((L+1)*ECRD+1.5))
     1 GO TO 230
      GO TO 250
  230 CONTINUE
      DO 240 M=3,NTZ2
      IF (MO.EQ.2) K=(NTZ2-M)
      IF (MO.EQ.1) K=M-3
      IF (PART(MO,J).GE.(K*ECRA+1.5).AND.PART(MO,J).LT.((K+1)*ECRA+1.5))
     1 STAR(M)=STAR(M)+1
  240 CONTINUE
  250 CONTINUE
      DO 260 M=1,NTZ3
  260 IF (STAR(M).EQ.0) ANDEX(M)=ALPHA(37)
      ANDEX(2)=ALPHA(40)
      ANDEX(NTZ3)=ALPHA(40)
      NOP=1
      IF (MOD(NAKER,2).EQ.0) NOP=2
      IF (NOP.EQ.1) GO TO 270
      GO TO 290
  270 CONTINUE
      HITY=((L*ECRD)+(.5))
      HITX=((I*ECRD)+(.5))
      DO 280 J=1,NDP
      HIND=FLOAT(J)
      IF (HITX.GT.HIND.AND.HITY.LT.HIND) GO TO 310
  280 CONTINUE
      GO TO 330
  290 CONTINUE
      DO 300 J=1,NDP
      IF (L.EQ.(J*NAKER-.5*NAKER)) GO TO 320
      IF (I.EQ.(J*NAKER-.5*NAKER)) GO TO 310
  300 CONTINUE
      GO TO 330
  310 ANDEX(1)=ALPHA(J+1)
  320 ANDEX(2)=ALPHA(38)
      ANDEX(NTZ3)=ALPHA(38)
  330 CONTINUE
      DO 340 J=2,NTZ2
      IF (STAR(J).NE.0.AND.STAR(J).LE.35) ANDEX(J)=ALPHA(STAR(J))
      IF (STAR(J).GT.35) ANDEX(J)=ALPHA(39)
  340 CONTINUE
      WRITE (IO,380) ANDEX
```

```
  350 CONTINUE                                                          H1530
      WRITE (IO,380) AMDEX                                              H1540
C     ***************************************************************  H1550
      RETURN                                                            H1560
C     ***************************************************************  H1570
C                                                                       H1580
C                                                                       H1590
C                                                                       H1600
  360 FORMAT (1H0,30H PARTICLE NO. AND X,Y LOCATION)                    H1610
  370 FORMAT (1H ,1I3,5X,2F10.5)                                        H1620
  380 FORMAT (1H ,131A1)                                                H1630
  390 FORMAT (1H ,10X,14HX-DIRECTION>>>,/)                              H1640
  400 FORMAT (1H1,10X,31H MAP VIEW OF PARTICLE LOCATIONS,//5X,34HCODE TO H1650
     1 NO. OF PTS. PER INCREMENT:/56H A=10,B=11,C=12,D=13,E=14,F=15,G=16 H1660
     2,H=17,I=18,J=19,K=20,/56H L=21,M=22,N=23,O=24,P=25,Q=26,R=27,S=28, H1670
     3T=29,U=30,V=31,/25H W=32,X=33,Y=34,Z=35,*>35/)                    H1680
  410 FORMAT (1H ,10X,7HTIME = ,1PE12.5,6H YEARS/)                      H1690
  420 FORMAT (1H ,10X,14HY-DIRECTION>>>,/)                              H1700
  430 FORMAT (1H ,10X,15HNODE NO. IN THE)                               H1710
      END                                                               H1720-
```

Attachment II
Data Input Formats

Card	Column	Format	Variable	Definition
1	1-80	20A4	TITLE	Description of problem.
2	1- 4	I4	NTIM	Maximum number of time steps in a pumping period (limit=100)*.
	5- 8	I4	NPMP	Number of pumping periods. Note that if NPMP>1, then data set 10 must be completed.
	9-12	I4	NX	Number of nodes in x direction (limit=20)*.
	13-16	I4	NY	Number of nodes in y direction (limit=20)*.
	17-20	I4	NP	Number of tracer particles (limit=600)*.
	21-24	I4	NPNT	Time-step interval (in flow equation) for printing hydraulic and particle output data.
	25-28	I4	NITP	Number of iteration parameters (usually $4 \le NITP \le 7$).
	29-32	I4	ITMAX	Maximum allowable number of iteration in ADIP solution to the flow equation (usually $100 \le ITMAX \le 200$).
	33-36	I4	NREC	Number of pumping or injection wells to be specified in a following data set.
	37-40	I4	NCODES	Number of node identification codes to be specified in a following data set (limit=10)*.
	41-44	I4	NPNTMV	Particle movement interval (IMOV) for printing particle output data. (Specify 0 to print only at end of time steps).
	45-48	I4	NPNTVL	Option for printing computed velocities (0=do not print; 1=print for first time step; 2=print for all time steps).

Card	Column	Format	Variable	Definition
2 (continued)				
	49-52	I4	NPRINP	Option for printing all particle locations (0=do not print; 1=print each NPNTMV interval; 2=print at end of time step only; 3=print at end of pumping period).
	53-56	I4	NOBS	Number of particles to track and print locations, identification numbers to be specified in a following data set (limit=5).
3	1- 5	G5.0	PINT	Pumping period in years.
	6-10	G5.0	TOL	Convergence criteria in ADIP (usually TOL$\leq$0.01).
	11-15	G5.0	POROS	Effective porosity.
	16-20	G5.0	S	Storage coefficient (set S=0 for steady flow problems).
	21-25	G5.0	TIMX	Time increment multiplier for transient flow problems. TIMX is disregarded if S=0.
	26-30	G5.0	TINIT	Size of initial time step in seconds. TINIT is disregarded if S=0.
	31-35	G5.0	XDEL	Width of finite-difference cell in x direction in feet.
	36-40	G5.0	YDEL	Width of finite-difference cell in y direction in feet.
	41-45	G5.0	CELDIS	Maximum cell distance per particle move (value between 0 and 1.0).
	46-50	G5.0	ANFCTR	Ratio of T_{yy} to T_{xx}.
4	1- 3	I3	INDT	Number of points to define initial position of front; specify INDT=1 to track movement from a point source (limit=100)*.

Data set	Number of cards	Format	Variable	Definition
1	INDT	2F10.5	SET,SEY	x and y coordinates of initial front location points.
2	Value of NOBS	I3	INOBS	Identification numbers of points for which history printout is desired (value between 1 and NP). This data set is eliminated if NOBS=0.
3	Value of NREC	2I2,G8.2	IX,IY,REC	x and y coordinates of pumping (+) or injection (-) wells, rate in ft^3/s. This data set is eliminated if NREC=0.
4	a.1	I1,G10.0	INPUT,FCTR	Parameter card† for transmissivity.
	b.Value of NY (limit = 20)*	20G4.1	VPRM	Array for temporary storage of transmissivity data, in ft^2/s. For an anisotropic aquifer, read in values of T_{xx} and the program will adjust for anisotropy by multiplying T_{yy} by ANFCTR.
5	a.1	I1,G10.0	INPUT,FCTR	Parameter card† for THCK.
	b.Value of NY (limit = 20)*	20G3.0	THCK	Saturated thickness of aquifer, in feet.
6	a.1	I1,G10.0	INPUT,FCTR	Parameter card† for RECH.
	b.Value of NY	20G4.1	RECH	Diffuse recharge (-) or discharge (+), in ft/s.
7	a.1	I1,G10.0	INPUT,FCTR	Parameter card† for NODEID.
	b.Value of NY (limit = 20)*	20I1	NODEID	Node identification matrix (used to define constant-head nodes or other boundary conditions and stresses).
8	Value of NCODES (limit = 10)*	I2,2G10.2, I2	ICODE, FCTR1, FCTR3, OVERRD	Instructions for using NODEID array. When NODEID=ICODE, program sets leakance= FCTR1, and if OVERRD is nonzero, RECH=FCTR3. Set OVERRD=0 to preserve values of RECH specified in data set 6.

Data set	Number of cards	Format	Variable	Definition
9	a.1	I1,G10.0	INPUT,FCTR	Parameter card[†] for WT.
	b.Value of NY (limit = 20)*	20G4.0	WT	Initial water-table or potentiometric elevation, or constant head in stream or source bed, in feet.
10				This data set allows time step parameters, print options, and pumpage data to be revised for each pumping period of the simulation. Data set 10 is only used if NPMP>1. The sequence of cards in data set 10 must be repeated (NPMP-1) times (that is, data set 10 is required for each pumping period after the first).
	a.1	I1	ICHK	Parameter to check whether any revisions are desired. Set ICHK=1 if data are to be revised, and the complete data set 10b and c. Set ICHK=0 if data are not to be revised for the next pumping period, and skip rest of data set 10.
	b.1	7I4,3G5.0	NTIM,NPNT, NITP,ITMAX, NREC,NPNTMV, NPNTVL,PINT, TIMX,TINIT	Ten parameters to be revised for next pumping period; the parameters were previously defined in the description of data cards 2 and 3. Only include this card if ICHK=1 in previous part a.
	c.Value of NREC	2I2,G8.2	IX,IY,REC	Revision of previously defined data set 3. Include part c only if ICHK=1 in previous part a and if NREC >0 in previous part b.

* These limits can be modified if necessary by changing the corresponding array dimensions in the COMMON statements of the program.

[†]The parameter card must be the first card of the indicated data sets. It is used to specify whether the parameter is constant and uniform, and can be defined by one value, or whether it varies in space and must be defined at each node. If INPUT=0, the data set has a constant value, which is defined by FCTR. If INPUT=1, the data set is read from cards as described by part b. Then FCTR is a multiplication factor for the values read in the data set.

Attachment III
Input Data for Test Problem

```
----------TEST PROBLEM INPUT----------
  50   1  12   11 100  10   7 200   0   1      0      0
 .05  .001  .20.0000  2.086400  100  100  1.0  1.0
  7
     10.5      3.0
     9.5      4.0
     8.8      5.0
     8.5      6.0
     8.8      7.0
     9.5      8.0
    10.5      9.0
    .1
    50.0

1   1.0

 1        1
 1        1
 1        1
 1        1
 1        1
 1        1
 1        1
 1        1
 1        1

 1      .10
1      10.0

    10.0                                  12.6
    10.0                                  12.8
    10.0                                  13.1
    10.0                                  13.5
    10.0                                  14.0
    10.0                                  13.5
    10.0                                  13.1
    10.0                                  12.8
    10.0                                  12.6
```

Attachment IV
Selected Output for Test Problem

```
U.S.G.S. FRONT-TRACKING MODEL FOR SOLUTE TRANSPORT IN GROUND WATER

----------TEST PROBLEM INPUT----------

                    I N P U T    D A T A

                      GRID DESCRIPTORS

            NX     (NUMBER OF COLUMNS)  =     12
            NY     (NUMBER OF ROWS)     =     11
            XDEL   (X-DISTANCE IN FEET) =   100.0
            YDEL   (Y-DISTANCE IN FEET) =   100.0
            INITIAL POSITION OF FRONT DEFINED BY FOLLOWING   7 POINTS:
                                        SET (=X)    SEY (=Y)
                                         10.500      3.000
                                          9.500      4.000
                                          8.800      5.000
                                          8.500      6.000
                                          8.800      7.000
                                          9.500      8.000
                                         10.500      9.000

                      TIME  PARAMETERS

            NTIM   (MAX. NO. OF TIME STEPS)        =     50
            NPMP   (NO. OF PUMPING PERIODS)        =      1
            PINT   (PUMPING PERIOD IN YEARS)       =      0.05
            TIMX   (TIME INCREMENT MULTIPLIER)     =      2.00
            TINIT  (INITIAL TIME STEP IN SEC.)     =  86400.

              HYDROLOGIC PARAMETERS

            S        (STORAGE COEFFICIENT)       =      0.000000
            POROS    (EFFECTIVE POROSITY)        =      0.20
            ANFCTR   (RATIO OF T-YY TO T-XX)     =     1.000000

                    EXECUTION PARAMETERS

            NITP   (NO. OF ITERATION PARAMETERS) =    7
            TOL    (CONVERGENCE CRITERIA - ADIP) =    0.0010
            ITMAX  (MAX.NO.OF ITERATIONS - ADIP) =  200
            CELDIS (MAX.CELL DISTANCE PER MOVE
                         OF PARTICLES - M.O.C.)  =    1.000
            NP     (NO. OF TRACER PARTICLES)     =  100

                      PROGRAM OPTIONS

            NPNT   (TIME STEP INTERVAL FOR
                    COMPLETE PRINTOUT)       =    10
            NPNTMV (MOVE INTERVAL FOR FRONT
                         LOCATION PRINTOUT)  =     0
            NPNTVL (PRINT OPTION-VELOCITY
                    0=NO; 1=FIRST TIME STEP;
                    2=ALL TIME STEPS)        =     0
            NREC   (NO. OF PUMPING WELLS)    =     0
            NCODES (FOR NODE IDENT.)         =     1
            NPRINP (PRINT OPTION-TRACER PTS) =     0
```

```
          STEADY-STATE FLOW

 TIME INTERVAL (IN SEC) FOR SOLUTE-TRANSPORT SIMULATION =  0.15779E+07

      AREA OF ONE CELL =   0.1000E+05

      X-Y SPACING:
           100.00
           100.00



   TRANSMISSIVITY MAP (FT*FT/SEC)
       0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
       0.00 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.00
       0.00 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.00
       0.00 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.00
       0.00 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.00
       0.00 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.00
       0.00 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.00
       0.00 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.00
       0.00 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.00
       0.00 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.00
       0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00



   AQUIFER THICKNESS (FT)
        0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
        0.0 50.0 50.0 50.0 50.0 50.0 50.0 50.0 50.0 50.0 50.0  0.0
        0.0 50.0 50.0 50.0 50.0 50.0 50.0 50.0 50.0 50.0 50.0  0.0
        0.0 50.0 50.0 50.0 50.0 50.0 50.0 50.0 50.0 50.0 50.0  0.0
        0.0 50.0 50.0 50.0 50.0 50.0 50.0 50.0 50.0 50.0 50.0  0.0
        0.0 50.0 50.0 50.0 50.0 50.0 50.0 50.0 50.0 50.0 50.0  0.0
        0.0 50.0 50.0 50.0 50.0 50.0 50.0 50.0 50.0 50.0 50.0  0.0
        0.0 50.0 50.0 50.0 50.0 50.0 50.0 50.0 50.0 50.0 50.0  0.0
        0.0 50.0 50.0 50.0 50.0 50.0 50.0 50.0 50.0 50.0 50.0  0.0
        0.0 50.0 50.0 50.0 50.0 50.0 50.0 50.0 50.0 50.0 50.0  0.0
        0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0



   PERMEABILTY MAP (FT/SEC)
      0.0000.0000.0000.0000.0000.0000.0000.0000.0000.0000.0000.000
      0.0000.0020.0020.0020.0020.0020.0020.0020.0020.0020.0020.000
      0.0000.0020.0020.0020.0020.0020.0020.0020.0020.0020.0020.000
      0.0000.0020.0020.0020.0020.0020.0020.0020.0020.0020.0020.000
      0.0000.0020.0020.0020.0020.0020.0020.0020.0020.0020.0020.000
      0.0000.0020.0020.0020.0020.0020.0020.0020.0020.0020.0020.000
      0.0000.0020.0020.0020.0020.0020.0020.0020.0020.0020.0020.000
      0.0000.0020.0020.0020.0020.0020.0020.0020.0020.0020.0020.000
      0.0000.0020.0020.0020.0020.0020.0020.0020.0020.0020.0020.000
      0.0000.0020.0020.0020.0020.0020.0020.0020.0020.0020.0020.000
      0.0000.0000.0000.0000.0000.0000.0000.0000.0000.0000.0000.000



          NO. OF FINITE-DIFFERENCE CELLS IN AQUIFER =   90

          AREA OF AQUIFER IN MODEL   =  0.90000E+06   SQ. FT.
```

Selected output for test problem--continued

NODE IDENTIFICATION MAP

0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	0	0	0

NO. OF NODE IDENT. CODES SPECIFIED = 1

THE FOLLOWING ASSIGNMENTS HAVE BEEN MADE:

CODE NO. LEAKANCE RECHARGE

1 0.100E+00

VERTICAL PERMEABILITY/THICKNESS (FT/(FT*SEC))

0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.10	0.00
0.00	0.10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.10	0.00
0.00	0.10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.10	0.00
0.00	0.10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.10	0.00
0.00	0.10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.10	0.00
0.00	0.10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.10	0.00
0.00	0.10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.10	0.00
0.00	0.10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.10	0.00
0.00	0.10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.10	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

WATER TABLE

0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	100.	0.	0.	0.	0.	0.	0.	0.	0.	126.	0.
0.	100.	0.	0.	0.	0.	0.	0.	0.	0.	128.	0.
0.	100.	0.	0.	0.	0.	0.	0.	0.	0.	131.	0.
0.	100.	0.	0.	0.	0.	0.	0.	0.	0.	135.	0.
0.	100.	0.	0.	0.	0.	0.	0.	0.	0.	140.	0.
0.	100.	0.	0.	0.	0.	0.	0.	0.	0.	135.	0.
0.	100.	0.	0.	0.	0.	0.	0.	0.	0.	131.	0.
0.	100.	0.	0.	0.	0.	0.	0.	0.	0.	128.	0.
0.	100.	0.	0.	0.	0.	0.	0.	0.	0.	126.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.

ITERATION PARAMETERS

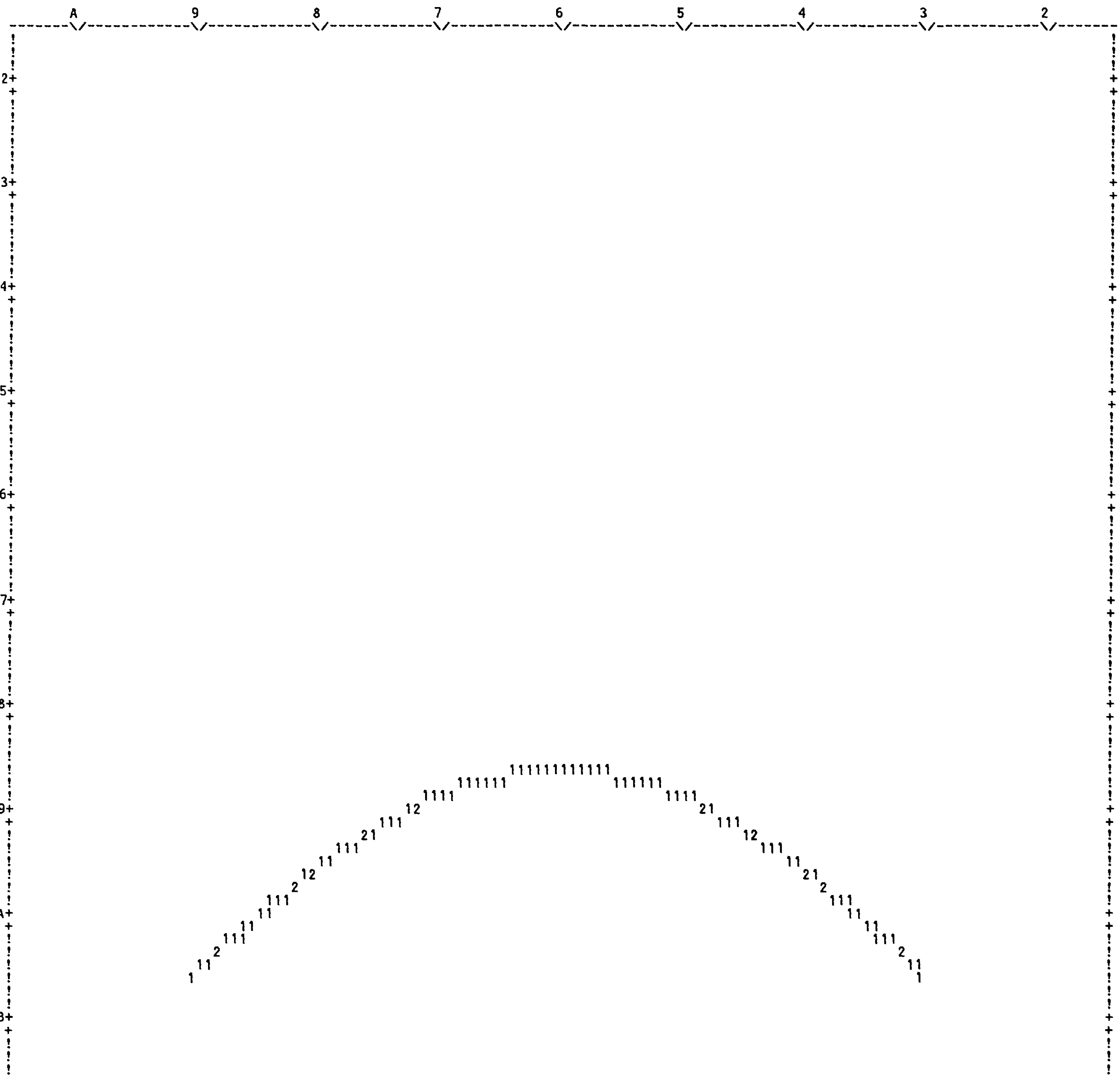
0.171347E-01
0.337467E-01
0.664638E-01
0.130900
0.257806
0.507746
1.00000
0.000000
0.000000
0.000000
0.000000

MAP VIEW OF PARTICLE LOCATIONS

CODE TO NO. OF PTS. PER INCREMENT:
A=10,B=11,C=12,D=13,E=14,F=15,G=16,H=17,I=18,J=19,K=20,
L=21,M=22,N=23,O=24,P=25,Q=26,R=27,S=28,T=29,U=30,V=31,
W=32,X=33,Y=34,Z=35,*>35

TIME = 0.00000E-01 YEARS

NODE NO. IN THE
Y-DIRECTION>>>



```
        STABILITY CRITERIA --- M.O.C.


VMAX =  8.55E-04      VMAY =  4.50E-04
VMXBD=  8.55E-04      VMYBD=  5.00E-04
 TIMV (CELDIS)    =  0.11701E+06

TIMV =  1.17E+05      NTIMV =     13     NMOV =     14

 TIM (N)  =  0.15779E+07
 TIMEVELO =  0.11271E+06
        NO. OF PARTICLE MOVES REQUIRED TO COMPLETE THIS TIME STEP  =   14

    HEAD DISTRIBUTION - ROW
    NUMBER OF TIME STEPS =      1
           TIME(SECONDS) =  0.15779E+07
           TIME(DAYS)    =  0.18262E+02
           TIME(YEARS)   =  0.50000E-01


       0    0    0    0    0    0    0    0    0    0    0    0

       0  100  103  107  110  114  117  120  123  125  126    0

       0  100  103  107  110  114  117  120  123  126  128    0

       0  100  103  107  110  114  117  121  124  128  131    0

       0  100  103  107  110  114  118  121  125  130  135    0

       0  100  103  107  110  114  118  122  126  131  140    0

       0  100  103  107  110  114  118  121  125  130  135    0

       0  100  103  107  110  114  117  121  124  128  131    0

       0  100  103  107  110  114  117  120  123  126  128    0

       0  100  103  107  110  114  117  120  123  125  126    0

       0    0    0    0    0    0    0    0    0    0    0    0

                 CUMULATIVE MASS BALANCE -- (IN FT**3)


                  RECHARGE AND INJECTION        =  0.00000E+00
                  PUMPAGE AND E-T WITHDRAWAL =  0.00000E+00
                  CUMULATIVE  NET  PUMPAGE      =  0.00000E+00
                  WATER RELEASE FROM STORAGE =  0.00000E+00
                  LEAKAGE INTO AQUIFER          =  0.52034E+07
                  LEAKAGE OUT OF AQUIFER        = -0.52033E+07
                  CUMULATIVE  NET  LEAKAGE      =  0.13400E+03

                      MASS BALANCE RESIDUAL  =   134.00
                      ERROR  (AS PERCENT)    =  0.25753E-02

                 RATE MASS BALANCE -- (IN C.F.S.)


                  LEAKAGE INTO AQUIFER          =  0.32977E+01
                  LEAKAGE OUT OF AQUIFER        = -0.32976E+01
                  NET LEAKAGE      (QNET)       =  0.86993E-04
                  RECHARGE AND INJECTION        =  0.00000E+00
                  PUMPAGE AND E-T WITHDRAWAL =  0.00000E+00
                  NET WITHDRAWAL    (TPUM)      =  0.00000E+00
```

MAP VIEW OF PARTICLE LOCATIONS

CODE TO NO. OF PTS. PER INCREMENT:
=10,B=11,C=12,D=13,E=14,F=15,G=16,H=17,I=18,J=19,K=20,
=21,M=22,N=23,O=24,P=25,Q=26,R=27,S=28,T=29,U=30,V=31,
=32,X=33,Y=34,Z=35,*>35

TIME = 5.00000E-02 YEARS

NODE NO. IN THE
Y-DIRECTION>>>