BNWL-1716
UC-70

2F

TRANSPORT MODEL
USER'S MANUAL

Battelle

Pacific Northwest Laboratories
Richland, Washington 99352

1974

Prepared for the U.S. Atomic Energy
Commission under Contract AT(45-1):1830

BNWL-1716

3 3679 00062 2797

TRANSPORT MODEL

USER'S MANUAL

by

S. W. Ahlstrom
R. G. Baca

Water and Land Resources Department

1974

This report is based on work sponsored by the
Atlantic Richfield Hanford Company under United
States Atomic Energy Commission Contract AT(45-1)-2130

BATTELLE
PACIFIC NORTHWEST LABORATORIES
RICHLAND, WASHINGTON 99352

SUMMARY

The Transport Model code consists of three main programs, TRACK, MICRO and MACRO. This document describes procedures for accessing these programs, inputting data to them, and interpreting the output that is generated.

CONTENTS

	<u>Page</u>
SUMMARY.	i
INTRODUCTION	1
THE TRACK PROGRAM - CONVECTION COMPUTATION	3
Access.	3
Input Requirements.	3
Output Interpretation	9
TRACK Subroutines	13
THE MICRO PROGRAM - TRANSPORT COMPUTATION FOR MICRO ION SYSTEMS.	14
Access.	14
Input Requirements.	14
Output Interpretation	17
MICRO Subroutines	19
THE MACRO PROGRAM - TRANSPORT COMPUTATION FOR MACRO ION SYSTEMS.	20
Access.	20
Input Requirements.	20
Output Interpretation	22
MACRO Subroutines	23
ACKNOWLEDGMENTS.	25
APPENDIX A - EXAMPLE OF TRACK INPUT AND OUTPUT	A-1
APPENDIX B - EXAMPLE OF A MICRO INPUT DECK.. . . .	B-1
APPENDIX C - EXAMPLE OF A MACRO INPUT DECK	C-1
APPENDIX D - DESCRIPTIVE FLOW CHARTS	D-1
APPENDIX E - PROGRAM LISTINGS.	E-1

TRANSPORT MODEL

USER'S MANUAL

INTRODUCTION

The numerical solution of the Transport Model equation is based on the technique known as the "particle-in-a-cell," or PIC method. (More details on this computational technique are given in BNWL-1715, "Transport Model - Theory and Numerical Implementation.") The numerical algorithm is contained in three main programs, TRACK, MICRO, and MACRO, each of which references several subprograms. The programs are run in sequence; first TRACK, followed by MACRO and/or MICRO. The output from TRACK is required input for both MACRO and MICRO. The output from MACRO is required as input by MICRO if the sorption coefficient is to be considered a function of the waste chemistry for a given problem (see Figure 1).

FIGURE 1. Transport Model Components

The execution of program TRACK establishes the desired problem size, boundaries, and grid networks and computes a movement history for the computational particles. TRACK input consists of the problem "description," which is a string of yes and no answers to a series of questions concerning whether or not certain options are desired; a set of problem "specifications" which determines the physical dimensions of the problem and specifies tolerances and limits, initial particles, and boundary cell coordinates; and a velocity field. The velocity field can be obtained from field measurements; however, it is usually compiled by either the VTT or PST computer programs (BNWL-1703 and BNWL-1713).

Program MACRO calculates a numerical solution to the macro ion transport equation. The particle movement history, available on an output tape created by TRACK, provides part of the input necessary for the execution of this program. Other input consists of the problem description, the problem specifications, initial concentrations of the macro ions, and other required parameters such as the dispersion coefficients and aquifer thickness. The computed macro ion distribution is stored on tape for use as input to MICRO.

A numerical solution to the micro ion transport equation is provided by the execution of program MICRO. The required input to MICRO consists of a particle movement history, dispersion coefficients, aquifer thickness, a micro ion distribution which is used to calculate sorption coefficients, an initial concentration distribution of the micro ions of interest, the problem "description," and the problem "specifications." MICRO can be used to simulate the transport of several micro ions from one set of particle movement and macro ion concentration data.

Descriptive flow charts for TRACK, MICRO and MACRO are contained in Appendix D.

THE TRACK PROGRAM - ADVECTION COMPUTATION

ACCESS

The TRACK program code is currently available as the first file on magnetic tape U5122(ARHCO). The file consists of main program TRACK and subroutines DARCY, HOUND, THICK, DCOMP, VELMAX, CARDRD, and TAPERD.

INPUT REQUIREMENTS

The required input for TRACK is arranged in four basic categories: 1) descriptive or control parameters, 2) problem specifications, 3) initial particle coordinates and boundary cell designations, and 4) velocity data. The exact form of the input deck will vary, depending on the problem options chosen by the user. One example of a typical input arrangement is shown in Appendix A.

Descriptive Parameters

The first card in the data deck should always be a title card. The title card is followed by a card with a string of YES or NO responses to questions concerning the program options. These answers are punched in a series of right-justified A6 formats. All option combinations are allowed.

The first question to be answered by the user is: "Is the flow field steady?" If the answer is YES, only one set of data containing either the velocity field or the data necessary to calculate the velocity field will need to be read, regardless of the number of time planes to be computed. If the flow is not steady, a new set of data must be input for each time plane.

In the case of steady flow, the time step size is determined by the following function:

$$\Delta t = \min \left\{ \begin{array}{l} X_{\text{cell}} / V_{x,\text{max}} \\ Y_{\text{cell}} / V_{y,\text{max}} \end{array} \right\}$$

where

$X_{\text{cell}}, Y_{\text{cell}}$ = computation cell dimensions

$V_{x,\text{max}}, V_{y,\text{max}}$ = maximum x and y velocity components

If the flow field is time varying, the time step increments are forced to match those for which velocity data are available. Time interpolation of velocity data is not performed.

The second question asks, "Is the media thickness to be considered variable?" If the answer is YES, an aquifer thickness distribution will be computed from the potential distribution and the aquifer bottom map. In the event the flow field is not steady, the thickness will be recomputed for each time plane. The aquifer thickness is subsequently written to the TRACK output tape for use by the MACRO or MICRO program.

Question three is: "Are the dispersion coefficients variable?" A YES answer to this question will cause dispersion coefficient array to be calculated from a user-supplied correlation relating the dispersion coefficient to fluid velocity and/or media permeability. These arrays are written to the output tape for use by MACRO or MICRO.

The next question reads: "Is the velocity or potential data on cards?" The response to this question tells the program where to look (cards or magnetic tape) for the velocity data.

The fifth question concerns the form of the velocity data: "Is the input data in terms of potentials?" If the response is YES, the program will attempt to read potential, permeability and porosity data and then compute a velocity field using Darcy's Law. Otherwise, it will attempt to read the velocity field directly.

The response to the sixth question controls the generation of printed output: "Is the nonessential printing to be suppressed?" A negative response will cause the program to print the complete velocity field and the coordinates of each of the particles for every time plane. In addition, depending upon the options used, the potential, permeability and porosity arrays as well as the dispersion coefficient and media thickness distributions may be written. For the average problem it is recommended that the YES answer be used in order to prevent a large amount of unnecessary printed output. However, in some instances (such as during the debug phase of the problem) it may be helpful if the printing is not suppressed.

The final question is: "Does the effective permeability vary with time?" This option was included to allow the computation of velocity fields for unsaturated flow conditions. The YES response is required for unsaturated flow and the NO response for saturated systems.

Problem Specifications

The parameters which specify the problem geometry and set computational tolerances and limits are entered on the next card in the data deck. These are input to the program by NAMELIST \$PARAM. A description of the NAMELIST parameters follows:

ITMAX	maximum permissible number of interactions for convergence of the algorithm which computes the velocity integral
NTIMES	number of time planes to be computed plus one; e.g., if computation of 12 time planes is desired, NTIMES = 13
NODESX	number of nodes in the x-direction
NODESY	number of nodes in the y-direction
NICPTS, NBCPTS	number of initial condition particles and number of boundary condition particles (The particles which carry the numerical solution in the PIC technique are divided into two categories, initial condition particles and boundary condition particles. Initial condition particles are used to specify the quality of the groundwater at $t = 0$ and boundary condition particles are used to specify the quality of inflows into the groundwater system. This distinction will be explained further in the next section.)
KOUNT	number of boundary condition cells (Cells are considered to be the rectangular areas bounded at their vertices by four adjacent nodes. The relationships between the nodal and cellular indexing systems are detailed in Figure 2. The total number of cells in the system is defined by $(NODESX-1)*(NODESY-1)$. The cells are subdivided into two groups, computational and boundary cells. This distinction will be clarified in the next section.
DELX	cell dimension in the x-direction
DELY	cell dimension in the y-direction

FIGURE 2. Nodal and Cellular Indexing System

Initial Particle Coordinates and Boundary Cell Designation

A PIC technique computes the advective portion of a mass transport by use of hypothetical particles which trace the fluid movement. In order to simulate the inflow of fluid at some internal or external boundary, it is necessary to somehow provide for a constant source of particles. This is accomplished in this model through the use of so-called boundary cells and boundary condition particles. Although boundary condition particles are always initially located within a boundary cell, they may flow out of the boundary cells during a given time step and become computational particles. At the beginning of each time step a new set of particles is placed at the starting coordinates of the original boundary condition particles, thereby creating a continuous source of particles at these locations. Particles which are not initially

contained in boundary cells are referred to as initial condition particles.

Besides serving as an originating area for particles, boundary cells are also used to eliminate particles from the computation when they reached an outflow boundary. The problem matrix is usually outlined with a row of boundary cells to catch any particle before it leaves the system. If a particle is within a boundary cell at the end of a time step, it is considered to have been advected out of the system and is subsequently eliminated from all further calculations. Exiting particles may be recycled into the system as inflow particles if they are needed to replace moving particles at a source location. An example of a computational matrix with both types of cells is shown in Figure 3. Quality changes caused by advection, diffusion and source terms are computed by the MACRO and MICRO programs; this program only computes the locations of the particles.

FIGURE 3. Sample Computational Matrix

The initial coordinates of particles within the system are input by specifying the coordinates of the first and last particles in the line and their index numbers. The program uses this information to compute the coordinates of the intermediate particles at even intervals. The variables used for specifying the coordinates of a line of particles are WORD, NODEF, XF, YF, NODEL, XL, and YL. WORD contains a sentinel which designates the end of a block of data. If WORD contains the right-justified characters LAST, the program will consider the current data read complete and will continue on with the execution of the program. NODEF is the index number of the first particle in a line of particles and XF and YF are the coordinates of that particle. NODEL is the index number of the last particle in a line of particles and XL and YL are the coordinates of that particle. For each line of particles that is to be input, values for these variables are entered on one card in a A6,I4,2F10.4,I5,2F10.4 format. Node (1,1) is always considered to have the coordinates (0.0, 0.0).

The starting coordinate data for all of the initial condition points are added to the deck one line at a time, followed by a card with the characters LAST in columns 3-6. The initial coordinate data for the boundary condition particles are then added, again followed by a LAST card. The particle index numbers must range sequentially from one to the total number of particles present. Initial condition particles must be numbered first, then boundary condition particles. See Appendix A for an example of this portion of the input.

The next piece of information required by TRACK is a designation of the boundary cell index numbers. These designations are made in blocks by specifying the nodal coordinates of the lower left hand corner of the starting and ending cells in the block. The required input is read into the variables WORD, IFIRST, JFIRST, ILAST, JLAST in a A6,I9,3I5 format. WORD is the same sentinel described previously. IFIRST and JFIRST are the nodal coordinates of the first cell in a block and ILAST and JLAST are the nodal coordinates of the last cell in a block.

For example, if (IFIRST, JFIRST) is (1,1) and (ILAST, JLAST) is (3,3), then the following cells would be designated as boundary cells: (1,1), (1,2), (1,3), (2,1), (2,2), (2,3), (3,1), (3,2), and (3,3). If (IFIRST, JFIRST) is (1,1) and (ILAST, JLAST) is (1,3), then only (1,1), (1,2), and (1,3) would be boundary cells. If (IFIRST, JFIRST) is (1,1) and (ILAST, JLAST) is (3,1), then only (1,1), (2,1), and (3,1) would be boundary cells. An example of this portion of the input is also included in Appendix A.

Velocity Data

The input format for the velocity matrices, VX and VY, can vary greatly depending upon the control options chosen (see "Descriptive Parameters" above). There are 6! variations of the control options, which prohibits listing all of the possible combinations which might arise. In this case it is most convenient to first decide on the options to be used, then obtain a program listing and trace the desired path through the input portions of the code, noting applicable input formats as they are encountered. The two appropriate sections of code for this purpose can be found under the comment headings: "READ OR COMPUTE THE INITIAL VELOCITY FIELD" and "READ OR COMPUTE THE CURRENT VELOCITY FIELD." If velocity data are to be input from tape, this tape must be assigned to logical unit 4.

The VX matrix contains the x-direction component of velocity and the VY matrix contains the y-direction velocity components. Other arrays which may be input under the heading of velocity data are the groundwater potential distribution, PHI; the permeability distribution, PERM; the aquifer bottom map, H; and the porosity, PORS. The user should note that velocities or velocity related data are defined in this model only at the nodes, not at the centers of the cells. Hence, all velocity data will have a dimension of NODESX by NODESY.

All of the velocity data distributions naturally occur as two-dimensional arrays; to increase the ease with which these arrays can be manipulated, however, they are stored in the TRACK program as extended one-dimensional arrays. The one-dimensional node index, IJ, of the node (I,J) can be compared by

$$IJ = (J-1)*NODESX + I$$

All velocity data must be converted to the extended one-dimensional form prior to being input from tape to TRACK. For example, in a system which is 10 by 10 nodes the one-dimensional index of node (5,5) would be

$$(5-1)*10 + 5 = 45$$

OUTPUT INTERPRETATION

Internally Generated Diagnostics

Program TRACK performs several checks on the input data and on the progress of the solution. If any of these checks are not satisfied, an error message will be printed and in most cases

the run will be terminated. These internal diagnostics are listed below with brief explanations of their meanings.

"ONE OF THE INPUT VARIABLES
IS LT OR EQ TO ZERO"

The problem as specified has zero size or has less than zero particles within it. This message will occur if DELX, DELY, NODESX, or NODESY are less than or equal to zero or if NICPTS or NBCPTS are less than zero. This error will terminate the run.

"MAX IS GT NODS"

The total number of nodes is greater than the number of locations that have been allocated in core. The amount of core reserved can be increased by increasing NODS in the PARAMETER statement at the beginning of the main program and in each subroutine. This error will terminate the run.

"LIMIT IS GT NPTS"

The total number of particles is greater than the number of locations that have been allocated in core. The amount of reserved space can be increased by increasing NPTS in the PARAMETER statement at the beginning of the main program and each subroutine. This error will terminate the run.

"NODEL IS LT NODEF"

The index number of the first particle in a line of particles has been assigned a value larger than the index number of the last particle in the line. This is a terminal error.

"NCOUNT IS NE TO NICPTS" or
"NCOUNT IS NE TO NBCPTS"

The internal count of initial condition particles or boundary condition particles does not match NICPTS or NBCPTS which were specified previously by the NAMELIST read. This error is usually caused by failure to index the particles sequentially. Either of these errors will terminate the run.

"CARD ERROR IN BOUNDARY CELL
READ IN"

The indexing used to specify a block of boundary cells is somehow in error.

"THE RECOMMENDED MAX TIME
STEP SIZE IS TOO SMALL.
TIMECK = _____."

For some reason the time step size is unreasonably small. Check the other input data. This error is terminal.

"WARNING: THE TIME STEP SIZE
EXCEEDS THE RECOMMENDED
MAXIMUM. RECOMMENDED TIME
STEP SIZE = _____.
ACTUAL TIME STEP SIZE
= _____."

The program has a built-in check to help determine if there is some possibility that numerical holes may appear in the solution. It does this by computing the time it would take the fastest moving particle to pass through one cell on the shortest possible trajectory and comparing the result with the actual time step being used. For problems with a steady flow field, TRACK will automatically set the time step size to the maximum allowed unless steps are taken to delete this part of the code. For unsteady flow problems the time step size is controlled by the intervals for which the velocity data are available. If the actual step size is larger than the recommended maximum this message will be written. The message does not terminate the run because the error may or may not have a significant numerical effect, depending on how the problem has been set up. This decision is left to the user's discretion.

"ERROR: TRACK IS OUTSIDE
THE GRID SYSTEM"

This message is generated by subroutine HOUND. Somehow a particle has been moved outside the area of interest without being eliminated by the ring of boundary cells. This is usually caused by an error in the velocity field or by a time step that is too long. The run is not terminated by this error.

"SPECIFIED INPUT IS
INDEXED INCORRECTLY"

Generated by subroutine CARDRD. The message indicates that some part of the velocity data being input on cards has an indexing error. This error is terminal.

"TAPE READ ERROR. UNIT
= _____, L = _____."

This indicates that a failure has occurred in the NTRAN tape manipulating routine. UNIT refers to the logical unit number and L indicates the NTRAN error code. The user is referred to the NTRAN manual for an explanation of error codes. This error will terminate the run.

Tape Output

The primary output from TRACK is to magnetic tape in binary form for future use by MICRO and MACRO. The tape output for each time plane consists of the boundary cell index numbers, IJBDY; the time plane number, NN; the number of particles, LIMIT; and the time step size, DELT; as well as the particle coordinates XMOVE and YMOVE. Dispersion coefficient arrays as well as aquifer thickness data may appear on this tape if so directed by the choice of program options. The output tape must be assigned to logical unit 3.

Printed Output

The first line of printed output is always the problem title. This is followed by a list of the program options being used and the problem specifications. Next, the data used by the program to establish the initial particle coordinates are printed so that they may be checked if desired. All remaining printed output is used to determine if the solution has proceeded in a reasonable fashion. If the nonessential printing is suppressed, the only information that will be written during

each time plane is the time plane number, the number of particles, and the time step size. If the total number of particles decreases during a time plane, a set of numbers labeled LIMSFT (which are used internally in the program as part of the book-keeping procedures) will also be printed. In a physical sense these numbers represent the excess particles which have left the system. The first number to the right of LIMSFT is the total number of excess particles and the remaining values are particle index numbers. A sample of printed output from TRACK is shown in Appendix A.

TRACK SUBROUTINES

Main program TRACK references seven subprograms, the functions of which are explained below.

DARCY	computes a velocity field from potential, permeability and porosity data using Darcy's Law.
HOUND	interpolates nodal velocities to get point velocities using a combined bi-linear, bi-quadratic, and bi-cubic interpolation scheme based on Newton's divided difference method.
THICK	computes an aquifer thickness array from potential data and an aquifer bottom map.
DCOMP	computes dispersion coefficient arrays from velocity and/or permeability data. The user must supply an appropriate correlation relating these variables for each system considered by replacing lines 19 and 20 in DCOMP.
VELMAX	finds the maximum nodal velocity which is subsequently used to calculate the maximum recommended time step size.
CARDRD	reads velocity data from cards.
TAPERD	reads velocity data from magnetic tape.

THE MICRO PROGRAM - TRANSPORT COMPUTATION FOR MICRO IONS

ACCESS

The MICRO Program code is currently available as the second file on magnetic tape U5122 (ARHCO). The file consists of the main program MICRO and subroutines AVRGE, INPUT, TIMER, SKDSR, and SKDCS.

INPUT REQUIREMENTS

The required input for MICRO is arranged into five basic categories: 1) descriptive or control parameters; 2) problem specifications; 3) initial concentrations; 4) input from the TRACK program; and 5) input from the MACRO program. The format of the input deck remains basically the same, regardless of which control options are chosen. An example of a typical deck for a MICRO run is shown in Appendix B.

Descriptive Parameters

The first card in the data deck should always be a title card. The title card is followed by a string of YES or NO responses to questions about the program control options to be used. These answers are punched in a series of right-justified A6 formats. Several of the control options chosen must match those picked for the corresponding TRACK run but there are no other restrictions on the selection of descriptive parameters.

The first question is: "Does this solution consider dispersion?" If dispersion is not important for the problem under consideration, NO is answered and the part of the code which computes the dispersion contribution will be ignored.

The next question asks: "Are the final results to be output to tape?" A YES answer to this option will cause the concentration distribution at the end of each time plane to be written to tape. This output tape can be used to generate contour maps of the results or for further analysis.

The third question controls the amount of printed output generated: "Is the nonessential printing to be suppressed?" A negative response will cause the program to print the initial concentration field, the particle coordinates, the dispersion coefficient arrays, and the aquifer thickness, in addition to the concentration matrix. Unless a great deal of debug data is desired, the YES response is recommended.

The fourth question reads: "Does the problem consider a variable K_d ?" This question asks if the sorption coefficient for the problem being considered is variable with respect to time and space. If the answer to this question is YES, program MICRO will attempt to read a set of macro ion concentrations for each cell and for each time plane from the output tape of program MACRO.

The next three questions are the same as the three that were asked by TRACK: "Is the flow field steady?" "Are the dispersion coefficients variable?" "Does the problem consider a variable aquifer thickness?" Answers to these questions must be the same as those used in the corresponding TRACK run in order to insure that the tape input records will match.

The final question asks: "Is the implicit algorithm to be used?" The code contains two different algorithms for computing the dispersion contribution. One is a standard explicit finite difference scheme and the other is a partially implicit, alternating-direction approach. The implicit scheme is more stable, but it requires slightly more computation time.

Problem Specifications

The parameters which specify the problem geometry and which set computational limits are input to the program by NAMELIST \$SPECS. Those parameters must be consistent with the problem specifications for the corresponding TRACK run. Other constants found in the transport equation are input by NAMELIST \$PRMS. A description of the NAMELIST \$SPECS variables follows:

NFREQ	print frequency variable which specifies how often the concentration matrix is to be printed (If NFREQ = 2, every other time plane will be printed. If NFREQ = 3, every third time plane would be written, etc.)
NTIMES	number of time planes to be computed plus one (e.g., if computation of 12 time steps is desired, NTIMES = 13)
NTOPX	number of cells in the x-direction
NTOPY	number of cells in the y-direction
KOUNT	number of boundary cells
DELX	cell dimension in the x-direction
DELY	cell dimension in the y-direction

The parameters of NAMELIST \$PRMS are:

DIFUSX	dispersion coefficient in the x-direction (This variable is significant only if the dispersion coefficient is constant; otherwise a dispersion array is read from the TRACK input tape.)
DIFUSY	dispersion coefficient in the y-direction (This variable is significant only if the dispersion coefficient is constant; otherwise a dispersion array is read from the TRACK input tape.)
ZKD	sorption coefficient (This variable is significant only if the sorption coefficient is constant; otherwise it will be computed from macro ion data.)
DBULK	bulk density of the soil
ZLAMB	radioactive decay constant

Initial Concentrations

The initial concentration must be assigned to each cell before a solution can be computed. This is accomplished by reading these values from cards in blocks in much the same way boundary cells are designated. The input variable list is WORD, CO, I1, J1, I2, J2, NEXT. These parameters are read in an AG, E9.4, 515 format, one data block per card. WORD contains a termination sentinel, as explained in "Initial Particle Coordinate and Boundary Cell Designation" on page 6. (I1, J1) are the nodal coordinates of the first cell in the block and (I2, J2) are the nodal coordinates of the last cell in the block. CO is the concentration of the cells specified by I1, J1, I2, J2. NEXT contains an integer constant which tells the main program what time plane to return to the concentration input subroutine in order to read new values for selected cells. If the concentration is not set for any given cell, it will automatically be set to zero by the program. A LAST card must follow this set of data.

All concentration values are defined at the centers of the cells. Consequently, the concentration related arrays will have dimensions of NTOPX by NTOPY or (NODESX-1) by (NODESY-1). In MICRO, as in TRACK, all two dimensional arrays are stored in the program as extended one dimensional arrays. The one-dimensional cell index, IJ, for cell (I,J) can be calculated by

$$IJ = (J-1)*NTOPX + I$$

For example, in a 10 cell by 10 cell system the one-dimensional index of cell (4,7) would be

$$(7-1) * 10 + 4 = 64$$

MICRO Input From TRACK

A primary source or input to MICRO is the particle movement history from program TRACK which is on magnetic tape. This tape must be assigned to logical unit 3.

MICRO Input From MACRO

If the sorption coefficient is variable, the output tape from MACRO must be input to MICRO. This tape must be assigned to logical unit 7.

OUTPUT INTERPRETATION

Internally Generated Diagnostics

Program MICRO performs several checks on the input information from TRACK and also on the progress of the solution. If any of these checks are not satisfied, an error message will be printed and the run will terminate. These internal diagnostics are listed below with brief explanations of their meanings.

"STABILITY PARAMETERS"

The computational algorithm for the calculation of the dispersion contribution is unstable for this particular set of problem specifications. The two variables printed are STAB1 and STAB2.
 $STAB1 = D_x \Delta t / \Delta x^2$ and $STAB2 = D_y \Delta t / \Delta y^2$.^x If STAB1 or STAB2 is greater than 0.3 and less than 0.7, this message will be printed and the diffusion contribution will be computed by the implicit scheme regardless of the options chosen. If STAB1 or STAB2 is greater than 0.7 the run will terminate.

"L = _____."

NTRAN read or write failure. See the NTRAN manual for an explanation of the error codes. This is a terminal error.

"TIME PLANE READ IS
MISMATCHED _____, _____."

Program MICRO is attempting to read some block of information from the TRACK tape that does not match what is expected. This error is usually caused by mismatching of the appropriate control options in the MICRO run and its corresponding TRACK run. This error will cause the run to terminate.

"SPECIFIED INPUT EXCEEDS
THE MAXIMUM"

Generated by subroutine INPUT. This indicates that more concentrations have been specified than there are cells. Check the indexing on the input cards.

"SPECIFIED INPUT IS
INDEXED INCORRECTLY"

Generated by subroutine INPUT. The last cell in the data block has been assigned an index lower than the first cell in the block. Indexing has probably been specified backwards.

Tape Output

The choice of the appropriate control option will cause the solution matrix to be written to tape each time plane. This output is written in the following format:

```
COMMON N, CIJ(NCEL)
```

```
CALL NTRAN (13,2,MAX,N,L)  [UNIVAC 1108 Tape I/O  
                             Routine]
```

where: N is the time plane number

MAX is the total number of cells plus one (NCEL + 1)

CIJ is the solution matrix

The output is directed to logical unit 13.

Printed Output

The first line of printed output is the problem heading. This is followed by a list of the program options selected, the problem specifications, and the transport related parameters. If the nonessential printing is suppressed, the remainder of the output consists of the solution matrix printed with a

frequency controlled by NFRFQ. The matrix is printed from top to bottom with the horizontal dimension folded in groups of 10, if necessary. Each row of the matrix is numbered appropriately.

MICRO SUBROUTINES

AVRGE	assigns each cell a concentration which is the average of the concentrations at all the particles within the cell.
INPUT	inputs the initial microion concentration distribution and, if desired, resets the concentration of any cell at some user-selected time during the execution of the program.
TIMER	computes the total accumulated run time and the time since this subroutine was last called.
SKDSR	calculates saturated sorption coefficients for strontium as a function of macroion concentrations.
SKDCS	calculates saturated sorption coefficients for cesium as a function of macroion concentrations.

THE MACRO PROGRAM - TRANSPORT COMPUTATION FOR MACRO IONS

ACCESS

The MACRO Program code is currently available as the third file on magnetic tape U5122 (ARHCO). The file consists of main program MACRO and subroutines AVRGE, INPUT, EQUILB, and NEWTIT.

INPUT REQUIREMENTS

The required input for MACRO is arranged in four basic categories: 1) descriptive or control parameters, 2) problem specifications, 3) initial concentrations, and 4) input data from TRACK. The format of the input deck remains basically the same, regardless of which control options are chosen. An example of a typical deck for a MACRO run is shown in Appendix C.

Descriptive Parameters

The first card in the data deck should always be a title card. The title card is followed by a string of YES or NO responses to questions about the program control options which are to be used. These answers are punched on a series of right-justified A6 formats. The last two control options chosen must match those picked for the corresponding track run. There are no other restrictions on the selection of descriptive parameters.

The first question is: "Does this solution consider dispersion?" If dispersion is not important for the problem under consideration, the answer is NO and the part of the code which computes the dispersion contribution will be omitted.

The next question asks: "Are the final results to be output to tape?" A YES answer to this question will cause the concentration distribution of each component at the end of each time plane to be written to tape. This output tape is used by program MICRO to compute sorption coefficients. A YES response should be used for all runs except for preliminary debugging.

The third question controls the amount of printed output generated: "Is the nonessential printing to be suppressed?" A negative response will cause the program to print the initial concentration field, the particle coordinates, the dispersion coefficient arrays, and the concentration matrices.

The final two questions to be answered are the same as the two asked by TRACK: "Is the flow field steady?" "Are

the dispersion coefficients variable?" Answers to these questions must be the same as those for the corresponding questions for the coupled TRACK run in order to insure that the tape input formats will match.

Problem Specifications

The parameters which specify the problem geometry and set computational limits are input to the program by NAMELIST \$SMACS. This NAMELIST is identical to NAMELIST \$SPECS which was described in "MICRO Program, Problem Specifications" on page 16. Other constants used by program MACRO are input by NAMELIST \$PMAC\$. The variables in this NAMELIST are:

DIFUSX	dispersion coefficient in the x-direction (This variable is significant only if the dispersion coefficient is constant; otherwise a dispersion array is read from the TRACK input tape.)
DIFUSY	dispersion coefficient in the y-direction (This variable is significant only if the dispersion coefficient is constant; otherwise a dispersion array is read from the TRACK input tape.)
DBULK	bulk density of the soil
C	chloride ion concentration
AN	nitrate ion concentration

Initial Concentrations

The initial concentrations of the chemical species considered by MACRO are input in a format identical to that used to input initial microion concentrations. This format is detailed in "MICRO Program, Initial Concentrations" on page 17. The concentration distributions are input in series, one species at a time. The data for each component must be followed by a LAST card. The input sequence must have the following order: fluid phase concentrations of Mg^{++} , Na^+ , K^+ , Ca^{++} , HCO_3^- , SO_4^{--} , and $CaSO_4$ (aq); pH; and soil phase concentrations of exchangeable Ca, Na, M, K, as well as the concentrations of the minerals $CaSO_4$ and $CaCO_3$.

Dry Soil Conditions

Subroutine START has been included in order to simulate conditions which would arise when an influent solution contacts dry

soil. This simulation is valid only for a ten-segment soil column. Input data and formats required for use of this starting condition are explained in BNWL-1718, "One Dimensional Model of the Movement of Trace Radioactive Solutes through Soil Columns: The PERCOL Model." If this option is chosen, line 173 in main program MACRO must be replaced by the statement CALL START.

MACRO Input From TRACK

The primary portion of input to MACRO is the particle movement history from the output tape written by TRACK. This tape must be assigned to logical unit 3.

OUTPUT INTERPRETATION

Internally Generated Diagnostics

Program MACRO performs several checks on the input data from TRACK and also on the progress of the solution. If any of these checks are not satisfied, an error message will be printed and the run will terminate. The diagnostic messages generated by MACRO and their interpretation are identical to those used by MICRO, page 18.

In addition to these diagnostics, subroutine EQUILB may generate the following messages:

"ROOT DID NOT CONVERGE
WITHIN MAXIMUM NUMBER OF
ITERATIONS, PROGRAM
RETURNING TO NEXT SET
OF DATA"

The Newton-Raphson iteration scheme, which is used to solve some of the nonlinear chemical equilibrium expressions, has not converged.

"OVERFLOW IN COMPUTATION,
RETURNING TO THE NEXT
SET OF DATA"

The slope of the curve for which a root is being computed by the Newton-Raphson scheme has become zero within machine accuracy.

Tape Output

The primary purpose of program MACRO is creation of an output tape to be used by MICRO for computing sorption coefficients. If option 2 is YES, at the end of every time plane the concentration of each species in each cell is written in binary form on the output tape, starting with the lowest numbered cell and proceeding sequentially to the highest numbered cell. This output tape must be assigned to logical unit 7.

Printed Output

If the user elects to suppress the nonessential printing, the output from MACRO is minimal. Only the problem title, specifications and other constants, and the initial concentrations are printed. The printing should not be suppressed during preliminary runs so that a solution matrix for each component will be printed and the progress of the solution inspected. For final runs, the printing should always be suppressed or a great deal of unnecessary printed output will be generated.

MACRO SUBROUTINES

AVRGE	assigns each cell a set of concentrations which is the average of the sets of concentrations associated with all the particles within the cell.
INPUT	inputs the concentrations of all the chemical components and, if desired, resets the concentration of any component in any cell at a user-selected time during the program execution.
EQUILB	calculates the multicomponent chemical equilibria for each cell.
NEWTIT	a Newton-Raphson iteration sequence called by EQUILB to solve nonlinear equilibrium expressions.

ACKNOWLEDGMENTS

This research was sponsored by the Physical and Life Sciences Technology Section, Research and Development Department, Atlantic Richfield Hanford Company. The authors are grateful for the support and assistance provided by D. J. Brown, Staff Geologist, and R. E. Isaacson, Manager of the Physical and Life Sciences Technology Section of Atlantic Richfield Hanford Company.

APPENDIX A

EXAMPLE OF TRACK INPUT AND OUTPUT

APPENDIX A

A sample input deck for a TRACK run is listed on the following pages for a problem that was designed to check out a portion of TRACK's capabilities. Notice that all velocity data were input from a tape which has been assigned to logical unit 4 (F). The input deck listing is immediately followed by a copy of the output generated by this run.

```
! RUN,S 8,CUB620,,SWA,600,5100          S. AHLSTROM  PSB2 EX2138
! ASG A=$TRACK(AHLS)
! X ASG,W B=(AHLS,LR,AT)
! ASG F=U2599(AHLS)
! XQT CUR
IN A
TRW A
TOC
! XQT TRACK

      TEST CONVECTION PATTERNS
      NO      NO      NO      NO      NO      YES      NO
$PARAM ITMAX=10, NTIMES=65, NODESX=40, NODESY=40, NICPTS=1472,
      NBCPTS=4, KOUNT=153, TOL=.03, DELX=1000.0, DELY=1000.0,
$END
0001    20250.0    19250.0    0032    35750.0    19250.0
0033    20250.0    19750.0    0064    35750.0    19750.0
0065    20250.0    20250.0    0096    35750.0    20250.0
0097    20250.0    20750.0    0128    35750.0    20750.0
0129    15250.0    21250.0    0170    35750.0    21250.0
0171    15250.0    21750.0    0212    35750.0    21750.0
0213    15250.0    22250.0    0254    35750.0    22250.0
0255    15250.0    22750.0    0296    35750.0    22750.0
0297    15250.0    23250.0    0338    35750.0    23250.0
0339    15250.0    23750.0    0380    35750.0    23750.0
0381    15250.0    24250.0    0422    35750.0    24250.0
0423    15250.0    24750.0    0464    35750.0    24750.0
0465    15250.0    25250.0    0506    35750.0    25250.0
0507    15250.0    25750.0    0548    35750.0    25750.0
0549    15250.0    26250.0    0590    35750.0    26250.0
0591    15250.0    26750.0    0632    35750.0    26750.0
0633    15250.0    27250.0    0674    35750.0    27250.0
0675    15250.0    27750.0    0716    35750.0    27750.0
0717    15250.0    28250.0    0758    35750.0    28250.0
0759    15250.0    28750.0    0800    35750.0    28750.0
0801    15250.0    29250.0    0842    35750.0    29250.0
0843    15250.0    29750.0    0884    35750.0    29750.0
```

0885	15250.0	30250.0	0926	35750.0	30250.0
0927	15250.0	30750.0	0968	35750.0	30750.0
0969	15250.0	31250.0	1010	35750.0	31250.0
1011	15250.0	31750.0	1052	35750.0	31750.0
1053	15250.0	32250.0	1094	35750.0	32250.0
1095	15250.0	32750.0	1136	35750.0	32750.0
1137	15250.0	33250.0	1178	35750.0	33250.0
1179	15250.0	33750.0	1220	35750.0	33750.0
1221	15250.0	34250.0	1262	35750.0	34250.0
1263	15250.0	34750.0	1304	35750.0	34750.0
1305	15250.0	35250.0	1346	35750.0	35250.0
1347	15250.0	35750.0	1388	35750.0	35750.0
1389	15250.0	36250.0	1430	35750.0	36250.0
1431	15250.0	36750.0	1472	35750.0	36750.0
LAST					
1473	25250.0	23250.0	1474	25750.0	23250.0
1475	25250.0	23750.0	1476	25750.0	23750.0
LAST					
01	01	01	39		
02	01	39	01		
02	39	39	39		
39	02	39	38		
26	24	26	24		
LAST					
FIN					

* * * * *

TEST CONVECTION PATTERNS

* * * * *

* * * * *

PROBLEM DESCRIPTION

IS THE FLOW FIELD STEADY?	NO
IS THE MEDIA THICKNESS VARIABLE?	NO
ARE THE DISPERSION COEFFICIENTS VARIABLE?	NO
IS THE VELOCITY OR POTENTIAL DATA ON CARDS?	NO
IS THE INPUT DATA IN TERMS OF POTENTIAL?	NO
IS THE NON-ESSENTIAL PRINTING TO BE SUPPRESSED?	YES
DOES THE PERMEABILITY VARY WITH TIME?	NO

PROBLEM SPECIFICATIONS

MAXIMUM NUMBER OF ITERATIONS FOR EACH TRAJECTORY	10
NUMBER OF TIME PLANES TO BE COMPUTED	65
NUMBER OF NODES IN THE X-DIRECTION	40
NUMBER OF NODES IN THE Y-DIRECTION	40
NUMBER OF INITIAL CONDITION POINTS	1472
NUMBER OF BOUNDARY CONDITION POINTS	4
NUMBER OF BOUNDARY CELLS	153
ALLOWABLE DEVIATION FOR EACH TRAJECTORY CALCULATION ..	.3000-01
CELL DIMENSION IN THE X-DIRECTION	.1000+04
CELL DIMENSION IN THE Y-DIRECTION	.1000+04

120250.000019250.0000	3235750.000019250.0000
3320250.000019750.0010	6435750.000019750.0010
6520250.000020250.0000	9635750.000020250.0000
9720250.000020750.0000	12835750.000020750.0000
12915249.999921250.0000	17035750.000021250.0000
17115249.999921750.0000	21235750.000021750.0000
21315249.999922250.0000	25435750.000022250.0000
25515249.999922750.0000	29635750.000022750.0000
29715249.999923250.0000	33835750.000023250.0000
33915249.999923750.0000	38035750.000023750.0000
38115249.999924250.0000	42235750.000024250.0000
42315249.999924750.0000	46435750.000024750.0000
46515249.999925250.0010	50635750.000025250.0010
50715249.999925750.0000	54835750.000025750.0000
54915249.999926250.0000	59035750.000026250.0000
59115249.999926750.0010	63235750.000026750.0010
63315249.999927250.0000	67435750.000027250.0000
67515249.999927750.0000	71635750.000027750.0000
71715249.999928250.0000	75835750.000028250.0000
75915249.999928750.0000	80035750.000028750.0000
80115249.999929250.0000	84235750.000029250.0000
84315249.999929750.0000	88435750.000029750.0000
88515249.999930250.0000	92635750.000030250.0000
92715249.999930750.0000	96835750.000030750.0000
96915249.999931250.0000	101035750.000031250.0000
101115249.999931750.0000	105235750.000031750.0000
105315249.999932250.0010	109435750.000032250.0010
109515249.999932750.0000	113635750.000032750.0000
113715249.999933250.0000	117835750.000033250.0000
117915249.999933750.0010	122035750.000033750.0010
122115249.999934250.0000	126235750.000034250.0000
126315249.999934750.0000	130435750.000034750.0000
130515249.999935250.0010	134635750.000035250.0010
134715249.999935750.0000	138835750.000035750.0000
138915249.999936250.0000	143035750.000036250.0000
143115249.999936750.0000	147235750.000036750.0000
147325250.001023250.0000	147425750.000023250.0000
147525250.001023750.0000	147625750.000023750.0000

TIME PLANE = 1 NUMBER OF POINTS = 1472 TIME STEP SIZE = .5000-00
LIMSFT 2 1473 1474

TIME PLANE = 2 NUMBER OF POINTS = 1472 TIME STEP SIZE = .5000-00

TIME PLANE = 3 NUMBER OF POINTS = 1470 TIME STEP SIZE = .5000-00

LIMSFT 2 361 362

TIME PLANE = 4 NUMBER OF POINTS = 1470 TIME STEP SIZE = .5000-00

TIME PLANE = 5 NUMBER OF POINTS = 1472 TIME STEP SIZE = .5000-00

TIME PLANE = 6 NUMBER OF POINTS = 1472 TIME STEP SIZE = .5000-00

TIME PLANE = 7 NUMBER OF POINTS = 1472 TIME STEP SIZE = .5000-00

TIME PLANE = 8 NUMBER OF POINTS = 1470 TIME STEP SIZE = .5000-00
LIMSFT 2 282 1337

TIME PLANE = 9 NUMBER OF POINTS = 1468 TIME STEP SIZE = .5000-00
LIMSFT 2 1386 1428

TIME PLANE = 10 NUMBER OF POINTS = 1466 TIME STEP SIZE = .5000-00
LIMSFT 2 1344 1427

TIME PLANE = 11	NUMBER OF POINTS = 1464	TIME STEP SIZE = .5000-00
LIMSFT 2	1302 1426	
<hr/>		
TIME PLANE = 12	NUMBER OF POINTS = 1461	TIME STEP SIZE = .5000-00
LIMSFT 3	282 263 324	
<hr/>		
TIME PLANE = 13	NUMBER OF POINTS = 1459	TIME STEP SIZE = .5000-00
LIMSFT 2	282 322	
<hr/>		
TIME PLANE = 14	NUMBER OF POINTS = 1457	TIME STEP SIZE = .5000-00
LIMSFT 2	321 356	
<hr/>		
TIME PLANE = 15	NUMBER OF POINTS = 1455	TIME STEP SIZE = .5000-00
LIMSFT 2	357 397	
<hr/>		
TIME PLANE = 16	NUMBER OF POINTS = 1456	TIME STEP SIZE = .5000-00
<hr/>		
TIME PLANE = 17	NUMBER OF POINTS = 1458	TIME STEP SIZE = .5000-00
<hr/>		
TIME PLANE = 18	NUMBER OF POINTS = 1456	TIME STEP SIZE = .5000-00
LIMSFT 1	1457	
<hr/>		
TIME PLANE = 19	NUMBER OF POINTS = 1456	TIME STEP SIZE = .5000-00
<hr/>		
TIME PLANE = 20	NUMBER OF POINTS = 1455	TIME STEP SIZE = .5000-00
LIMSFT 0		

TIME PLANE = 21 NUMBER OF POINTS = 1455 TIME STEP SIZE = .5000-00

TIME PLANE = 22 NUMBER OF POINTS = 1455 TIME STEP SIZE = .5000-00

TIME PLANE = 23 NUMBER OF POINTS = 1455 TIME STEP SIZE = .5000-00

TIME PLANE = 24 NUMBER OF POINTS = 1456 TIME STEP SIZE = .5000-00

TIME PLANE = 25 NUMBER OF POINTS = 1454 TIME STEP SIZE = .5000-00
LIMSFT 2 479 481

A-6

TIME PLANE = 26 NUMBER OF POINTS = 1450 TIME STEP SIZE = .5000-00
LIMSFT 4 480 1165 1207 1249

TIME PLANE = 27 NUMBER OF POINTS = 1441 TIME STEP SIZE = .5000-00
LIMSFT 9 1122 1123 1124 1164 1165 1205 1206 1246 1247

TIME PLANE = 28 NUMBER OF POINTS = 1425 TIME STEP SIZE = .5000-00
LIMSFT 16 361 401 1080 1081 1082 1083 1084 1122 1123 1161 1162 1163 1200

-----1201 1239 1240

TIME PLANE = 29 NUMBER OF POINTS = 1403 TIME STEP SIZE = .5000-00
LIMSFT 22 1036 1037 1038 1039 1040 1041 1078 1079 1080 1081 1115 1116 1117

-----LIMSFT 1191 1225 1228-----1153 1154 1155 1188 1189 1190

TIME PLANE = 30 NUMBER OF POINTS = 1375 TIME STEP SIZE = .5000-00

LIMSFT 28 1036 1037 1072 1073 1074 1075 1105 1106 1107 1139 1140 1141 1172 1173 1174 1175
LIMSFT 1208 1209 1243 1244 1284 1285 1324 1325 1363 1176 1205 1206

TIME PLANE = 31 NUMBER OF POINTS = 1343 TIME STEP SIZE = .5000-00
LIMSFT 32 401 994 995 996 998 999 1000 1036 1070 1071 1072 1100 1101 1102 1130 1131
LIMSFT 1161 1162 1163 1164 1186 1189 1192 1222 1223 1261 1262 1299 1336 1132 1133 1160

TIME PLANE = 32 NUMBER OF POINTS = 1327 TIME STEP SIZE = .5000-00
LIMSFT 16 995 996 997 1030 1088 1089 1091 1116 1118 1119 1143 1166 1168 1196 1270 1305

TIME PLANE = 33 NUMBER OF POINTS = 1318 TIME STEP SIZE = .5000-00
LIMSFT 9 1026 1027 1058 1059 1060 1084 1085 1109 1136

TIME PLANE = 34 NUMBER OF POINTS = 1317 TIME STEP SIZE = .5000-00
LIMSFT 1 994

TIME PLANE = 35 NUMBER OF POINTS = 1319 TIME STEP SIZE = .5000-00

TIME PLANE = 36 NUMBER OF POINTS = 1318 TIME STEP SIZE = .5000-00
LIMSFT 1 1318

TIME PLANE = 37 NUMBER OF POINTS = 1317 TIME STEP SIZE = .5000-00
LIMSFT 1 602

WARNING: THE TIME STEP SIZE EXCEEDS THE RECOMMENDED MAXIMUM.
RECOMMENDED MAXIMUM TIME STEP SIZE = .4176-00
ACTUAL TIME STEP SIZE = .5000-00

TIME PLANE = 38 NUMBER OF POINTS = 1319 TIME STEP SIZE = .5000-00
WARNING: THE TIME STEP SIZE EXCEEDS THE RECOMMENDED MAXIMUM.
RECOMMENDED MAXIMUM TIME STEP SIZE = .3711-00
ACTUAL TIME STEP SIZE = .5000-00

TIME PLANE = 39 NUMBER OF POINTS = 1319 TIME STEP SIZE = .5000-00
WARNING: THE TIME STEP SIZE EXCEEDS THE RECOMMENDED MAXIMUM.
RECOMMENDED MAXIMUM TIME STEP SIZE = .3466-00
ACTUAL TIME STEP SIZE = .5000-00

TIME PLANE = 40 NUMBER OF POINTS = 1319 TIME STEP SIZE = .5000-00
WARNING: THE TIME STEP SIZE EXCEEDS THE RECOMMENDED MAXIMUM.
RECOMMENDED MAXIMUM TIME STEP SIZE = .3581-00
ACTUAL TIME STEP SIZE = .5000-00

TIME PLANE = 41 NUMBER OF POINTS = 1321 TIME STEP SIZE = .5000-00
WARNING: THE TIME STEP SIZE EXCEEDS THE RECOMMENDED MAXIMUM.
RECOMMENDED MAXIMUM TIME STEP SIZE = .3437-00
ACTUAL TIME STEP SIZE = .5000-00

TIME PLANE = 42 NUMBER OF POINTS = 1321 TIME STEP SIZE = .5000-00
WARNING: THE TIME STEP SIZE EXCEEDS THE RECOMMENDED MAXIMUM.
RECOMMENDED MAXIMUM TIME STEP SIZE = .3647-00
ACTUAL TIME STEP SIZE = .5000-00

TIME PLANE = 43 NUMBER OF POINTS = 1322 TIME STEP SIZE = .5000-00

WARNING: THE TIME STEP SIZE EXCEEDS THE RECOMMENDED MAXIMUM.

RECOMMENDED MAXIMUM TIME STEP SIZE = .4060-00

ACTUAL TIME STEP SIZE = .5000-00

TIME PLANE = 44 NUMBER OF POINTS = 1324 TIME STEP SIZE = .5000-00

WARNING: THE TIME STEP SIZE EXCEEDS THE RECOMMENDED MAXIMUM.

RECOMMENDED MAXIMUM TIME STEP SIZE = .4798-00

ACTUAL TIME STEP SIZE = .5000-00

TIME PLANE = 45 NUMBER OF POINTS = 1328 TIME STEP SIZE = .5000-00

TIME PLANE = 46 NUMBER OF POINTS = 1331 TIME STEP SIZE = .5000-00

TIME PLANE = 47 NUMBER OF POINTS = 1334 TIME STEP SIZE = .5000-00

TIME PLANE = 48 NUMBER OF POINTS = 1336 TIME STEP SIZE = .5000-00

TIME PLANE = 49 NUMBER OF POINTS = 1336 TIME STEP SIZE = .5000-00

TIME PLANE = 50 NUMBER OF POINTS = 1336 TIME STEP SIZE = .5000-00

TIME PLANE = 51 NUMBER OF POINTS = 1337 TIME STEP SIZE = .5000-00

TIME PLANE = 52 NUMBER OF POINTS = 1337 TIME STEP SIZE = .5000-00

TIME PLANE = 53 NUMBER OF POINTS = 1338 TIME STEP SIZE = .5000-00

TIME PLANE =	54	NUMBER OF POINTS =	1338	TIME STEP SIZE =	.5000-00
TIME PLANE =	55	NUMBER OF POINTS =	1336	TIME STEP SIZE =	.5000-00
LIMSFT	2 894 934				
TIME PLANE =	56	NUMBER OF POINTS =	1336	TIME STEP SIZE =	.5000-00
TIME PLANE =	57	NUMBER OF POINTS =	1338	TIME STEP SIZE =	.5000-00
TIME PLANE =	58	NUMBER OF POINTS =	1338	TIME STEP SIZE =	.5000-00
TIME PLANE =	59	NUMBER OF POINTS =	1336	TIME STEP SIZE =	.5000-00
LIMSFT	2 611 640				
TIME PLANE =	60	NUMBER OF POINTS =	1336	TIME STEP SIZE =	.5000-00
TIME PLANE =	61	NUMBER OF POINTS =	1338	TIME STEP SIZE =	.5000-00
TIME PLANE =	62	NUMBER OF POINTS =	1338	TIME STEP SIZE =	.5000-00
TIME PLANE =	63	NUMBER OF POINTS =	1339	TIME STEP SIZE =	.5000-00
TIME PLANE =	64	NUMBER OF POINTS =	1339	TIME STEP SIZE =	.5000-00

APPENDIX B

EXAMPLE OF A MICRO INPUT DECK

APPENDIX B

A sample input deck for a MICRO run is listed on the following pages. This input was for one portion of a tank leak simulation.

```

1 RUN,S 8,CUR620,,SWA,900,10000 S. AHLSTROM PSB2 EX2138
1 PRL TEL INEQ00
1 ASG A=MICRO(RACA)
1 X ASG R=U5342(AHLS)
1 X ASG,W C=(AHLS,LR,AT)
1 XQT CUR
IN A
TRW A
TCC
1 F FR5,* AVRGF,AVRGF
-6,6
1 F FR5,* INPUT,INPUT
-7,7
1 F FR5,* SKDSR,SKDSR
-6,6
1 F FR5,* MICRO,MICRO
-40,40
1 F FR5,* SKDCS,SKDCS
-6,6
1 XQT MICRO
TANK LEAK SIMULATION - MICRO ION NON-SORPTIVE CASE, SEG 4
YES YES YES NO YES NO NO
$SDECS NREFQ=1, NTIMFS=16,NTOPX=41, NTOFY=48, KOINT=88, DFLX=3.0,
DFLY=2.0,
$END
$PRMS DIFUSX=0.02, DIFUSY=0.02, ZKD=0.0, DBULK=1.68, 7LAMB=.00192,
$END
. 5097-02 2 32 5 32
. 5094-02 6 32 6 32
. 5083-02 7 32 7 32
. 5407-02 8 22 8 32
. 5629-02 9 32 9 32
. 5013-02 10 32 10 32

```

•4936-02	11	32	11	32
•4971-02	12	32	12	32
•4853-02	13	32	13	32
•2063-01	2	33	5	33
•2062-01	6	33	6	33
•2061-01	7	33	7	33
•2267-01	8	33	8	33
•2367-01	9	33	9	33
•2326-01	10	33	10	33
•2394-01	11	33	11	33
•2430-01	12	33	12	33

•2579-01	13	33	13	33
•2047-01	2	34	5	34
•2046-01	6	34	6	34
•2045-01	7	34	7	34
•2051-01	8	34	8	34
•2969-01	9	34	9	34
•0110-01	10	34	10	34
•1147+00	11	34	11	34
•1147+00	12	34	12	34
•1063+00	13	34	13	34
•2213-00	2	35	5	35
•2213-00	6	35	6	35
•2213-00	7	35	7	35
•2213-00	8	35	8	35
•2224-00	9	35	9	35
•2283-00	10	35	10	35
•2280-00	11	35	11	35
•2402-00	12	35	12	35
•2160-00	13	35	13	35
•5210-00	2	36	5	36
•5210-00	6	36	6	36
•5211-00	7	36	7	36
•5211-00	8	36	8	36
•5210-00	9	36	9	36
•5209-00	10	36	10	36
•5204-00	11	36	11	36
•5200-00	12	36	12	36
•5145-00	13	36	13	36
•1070-02	14	36	14	36
•8159-00	2	37	5	37
•8159-00	6	37	6	37
•8163-00	7	37	7	37
•8171-00	8	37	8	37
•8178-00	9	37	9	37
•8173-00	10	37	10	37
•8198-00	11	37	11	37
•8216-00	12	37	12	37
•8138-00	13	37	13	37
•1714-02	14	37	14	37
•8970-00	2	38	5	38
•8970-00	6	38	6	38
•8986-00	7	38	7	38

•8998-00	8	38	8	38
•8984-00	9	38	9	38
•8977-00	10	38	10	38
•9007-00	11	38	11	38
•9353-00	12	38	12	38
•9392-00	13	38	13	38
•1972-02	14	38	14	38
•9600-00	2	39	5	39
•9600-00	6	39	6	39
•9635-00	7	39	7	39
•9654-00	8	39	8	39
•9660-00	9	39	9	39
•9656-00	10	39	10	39
•9663-00	11	39	11	39
•9682-00	12	39	12	39
•9602-00	13	39	13	39
•9579-02	14	39	14	39
•6843-02	15	39	15	39
•6064-02	16	39	16	39
•5218-02	17	39	17	39
•4310-02	18	39	18	39
•1396-02	19	39	19	39
•9739-00	2	40	5	40
•9740-00	6	40	6	40
•9784-00	7	40	7	40
•9782-00	8	40	8	40
•9776-00	9	40	9	40
•9779-00	10	40	10	40
•9770-00	11	40	11	40
•9762-00	12	40	12	40
•9681-00	13	40	13	40
•1657-00	14	40	14	40
•1460-00	15	40	15	40
•1260-00	16	40	16	40
•1055+00	17	40	17	40
•8522-01	18	40	18	40
•3066-01	19	40	19	40
•2193-02	20	40	20	40
•9745-00	2	41	5	41
•9743-00	6	41	6	41
•9759-00	7	41	7	41
•9757-00	8	41	8	41
•9749-00	9	41	9	41
•9744-00	10	41	10	41
•9729-00	11	41	11	41
•9727-00	12	41	12	41
•9686-00	13	41	13	41
•4650-00	14	41	14	41
•4260-00	15	41	15	41
•3534-00	16	41	16	41
•2692-00	17	41	17	41
•1913-00	18	41	18	41
•1129+00	19	41	19	41
•4216-01	20	41	20	41
•9738-00	2	42	5	42
•9737-00	6	42	6	42

•9723-00	7	42	7	42
•9723-00	8	42	8	42
•9724-00	9	42	9	42
•9723-00	10	42	10	42
•9723-00	11	42	11	42
•9724-00	12	42	12	42
•9706-00	13	42	13	42
•6790-00	14	42	14	42
•6120-00	15	42	15	42
•5607-00	16	42	16	42
•4215-00	17	42	17	42
•2615-00	18	42	18	42
•1004-00	19	42	19	42
•7216-01	20	42	20	42
•5206-02	21	42	21	42
•9742-00	2	43	5	43
•9733-00	6	43	6	43
•9717-00	7	43	7	43
•9718-00	8	43	8	43
•9720-00	9	43	9	43
•9718-00	10	43	10	43
•9718-00	11	43	11	43
•9721-00	12	43	12	43
•9702-00	13	43	13	43
•7794-00	14	43	14	43
•6954-00	15	43	15	43
•6276-00	16	43	16	43
•4800-00	17	43	17	43
•2858-00	18	43	18	43
•1409-00	19	43	19	43
•7311-01	20	43	20	43
•1002-01	21	43	21	43
•9718-00	2	44	5	44
•9702-00	6	44	6	44
•9677-00	7	44	7	44
•9679-00	8	44	8	44
•9696-00	9	44	9	44
•9690-00	10	44	10	44
•9693-00	11	44	11	44
•9700-00	12	44	12	44
•9697-00	13	44	13	44
•9332-00	14	44	14	44
•7899-00	15	44	15	44
•7293-00	16	44	16	44
•6822-00	17	44	17	44
•5849-00	18	44	18	44
•2934-00	19	44	19	44
•1471-00	20	44	20	44
•8016-01	21	44	21	44

.6274-02	22	44	22	44
.9868-00	2	45	5	45
.9629-00	6	45	6	45
.9624-00	7	45	7	45
.9610-00	8	45	8	45
.9612-00	9	45	9	45
.9613-00	10	45	10	45
.9619-00	11	45	11	45
.9621-00	12	45	12	45
.9608-00	13	45	13	45
.9364-00	14	45	14	45
.8319-00	15	45	15	45
.7797-00	16	45	16	45
.7094-00	17	45	17	45
.6463-00	18	45	18	45
.3913-00	19	45	19	45
.1963-00	20	45	20	45
.8883-01	21	45	21	45
.8776-02	22	45	22	45
.9845-00	2	46	5	46
.9492-00	6	46	6	46
.9490-00	7	46	7	46
.9486-00	8	46	8	46
.9485-00	9	46	9	46
.9487-00	10	46	10	46
.9489-00	11	46	11	46
.9486-00	12	46	12	46
.9483-00	13	46	13	46
.9408-00	14	46	14	46
.9122-00	15	46	15	46
.8777-00	16	46	16	46
.8806-00	17	46	17	46
.5650-00	18	46	18	46
.3974-00	19	46	19	46
.3010-00	20	46	20	46
.1473-00	21	46	21	46
.1967-01	22	46	22	46
.9813-00	2	47	5	47
.9424-00	6	47	6	47
.9421-00	7	47	7	47
.9417-00	8	47	8	47
.9409-00	9	47	9	47
.9407-00	10	47	10	47
.9402-00	11	47	11	47
.9391-00	12	47	12	47
.9381-00	13	47	13	47
.9400-00	14	47	14	47
.9350-00	15	47	15	47
.9242-00	16	47	16	47

.9239-00	17	47	17	47
.8802-00	18	47	18	47
.7439-00	19	47	19	47
.3980-00	20	47	20	47
.1995-00	21	47	21	47
.3080-01	22	47	22	47
.9788-00	2	48	5	48
.9357-00	6	48	6	48
.9353-00	7	48	7	48
.9345-00	8	48	8	48
.9309-00	9	48	9	48
.9305-00	10	48	10	48
.9303-00	11	48	11	48
.9296-00	12	48	12	48
.9297-00	13	48	13	48
.9392-00	14	48	14	48
.9371-00	15	48	15	48
.9483-00	16	48	16	48
.9439-00	17	48	17	48
.9114-00	18	48	18	48
.8032-00	19	48	19	48
.5452-00	20	48	20	48
.3722-00	21	48	21	48
.2032-00	22	48	22	48

LAST
FIN

APPENDIX C

EXAMPLE OF A MACRO INPUT DECK

APPENDIX C

A sample input deck for a MACRO run is shown below. This was set up for one of the verification runs done during the debugging stage of MACRO development.

```
! RUN,0 8,CUR620,,SWA,200,5100          $. AHLSTROM PSR2 EX2138
! ASG A=$MACRO(RACA)
!X ASG R=111717(AHLS)
!X ASG,W D=(AHLS,LR,AT)
! XQT CUR
IN A
TRW A
TOC
! XQT MACRO
NA, K BREAKTHROUGH CURVES                SWA      2/24/72
      YES YES YES YES NO
$SMACS NREQ=8, NTIMES=280, NTOPIX=12, NTOPIY=3, KOINT=26, DFLX=1.5
      DFLY=10.0
$END
$PMACS DIFUSX=0.01, DIFUSY=0.01, DRULK=1.55, C=0.0, AN=0.013
$END
      0.0000+00      1      1      12      3
LAST
      1.1000-02      1      2      1      2
LAST
      2.0000-03      1      2      1      2
LAST
      0.0000+00      1      1      12      3
LAST
      0.0000+00      1      1      12      3
LAST
      0.0000+00      1      1      12      3
LAST
      0.0000+00      1      1      12      3
LAST
      7.0000+00      1      2      1      2
LAST
      0.0000+00      1      1      12      3
LAST
      2.4250-05      2      2      11      2
LAST
      0.0000+00      1      1      12      3
LAST
      2.4250-05      2      2      11      2
LAST
      0.0000+00      1      1      12      3
LAST
      0.0000+00      1      1      12      3
LAST
! FIN
```

X

APPENDIX D

DESCRIPTIVE FLOW CHARTS

TRACK PROGRAM FLOWCHART

MICRO PROGRAM FLOWCHART

MACRO PROGRAM FLOWCHART

APPENDIX E
PROGRAM LISTINGS

TRACK: A PROGRAM TO CALCULATE THE POINT MOVEMENT HISTORY FOR USE
IN THE METHOD OF CHARACTERISTICS NUMERICAL SOLUTION OF THE
MICRO AND MACRO TRANSPORT MODELS.

WRITTEN BY 1) S.W. AHLSTROM
2) R.G. BACA
3) A. BRANDSETTER

PROGRAMED BY 1) S.W. AHLSTROM
2) R.G. BACA
3) D. FRIEDRICHS

GENERALIZED VERSION - MARCH 30, 1972

BATTELLE-NORTHWEST: WATER RESOURCES SYSTEMS SECTION

PARAMETER NODS=2000, NEX=250, NPTS=4000
COMMON/TRK/NODESX,NODESY,NN,LIMIT,DELT,DELX,DELY,MAX,EPS,SUPRES,YE
15,PHI(NODS),PERM(NODS),PORS(NODS),STOREX(100),STOREY(100),H(NODS),
2VDIS(NODS)
DIMENSION VX(NODS),VY(NODS),VXX(NODS),VYY(NODS),XMOVE(NPTS),YMOVE(
1NPTS),NFLAG(NEX),XBOUND(NEX),YBOUND(NEX),INDEX(NEX),IJBURY(NEX),
2TITLE(13)
NAMELIST/PARAM/ ITMAX,NTIMES,NODESX,NODESY,NICPTS,NBCPTS,KOUNT,
1TOL,DELX,DELY
REAL LAST
DATA LAST,YES,EPS/6H LAST,6H YES,0.00001/
ERROR=0.0
IPOINT=0
VMAXX=0.0
VMAXY=0.0

READ THE PROBLEM TITLE FROM CARDS

READ (5,830) (TITLE(I),I=1,13)
WRITE (6,840)
WRITE (6,830) (TITLE(I),I=1,13)
WRITE (6,840)

READ THE PROBLEM DESCRIPTION FROM CARDS

QUESTIONS: (YES OR NO)
STEADY.....IS THE FLOW FIELD STEADY?
VARTHK.....DOES THE MEDIA HAVE A VARIABLE THICKNESS?
VARDIS.....ARE THE DISPERSION COEFFICIENTS VARIABLE?
VCARDS.....IS THE VELOCITY OR POTENTIAL DATA ON CARDS?
POTEN.....IS THE INPUT DATA IN TERMS OF POTENTIALS?
SUPRES.....IS THE NON-ESSENTIAL PRINTING TO BE SUPRESSED?
VARPER.....DOES THE PERMEABILITY VARY WITH TIME?

READ (5,830) STEADY,VARTHK,VARDIS,VCARDS,POTEN,SUPRES,VARPER
WRITE (6,840)
WRITE (6,850) STEADY,VARTHK,VARDIS,VCARDS,POTEN,SUPRES,VARPER
WRITE (6,840)

READ PROBLEM SPECIFICATIONS FROM CARDS

NOMENCLATURE:

ITMAX.....MAXIMUM NUMBER OF ITERATIONS FOR EACH TRAJECTORY
 NTIMES.....NUMBER OF TIME PLANES TO BE COMPUTED
 NODESX.....NUMBER OF NODES IN THE X-DIRECTION
 NODESY.....NUMBER OF NODES IN THE Y-DIRECTION
 NICPTS.....NUMBER OF INITIAL CONDITION POINTS
 NBCPTS.....NUMBER OF BOUNDARY CONDITION POINTS
 KOUNT.....NUMBER OF BOUNDARY CELLS
 TOL.....ALLOWABLE DEVIATION FOR EACH TRAJECTORY CALCULATION
 DELX.....CELL DIMENSION IN X-DIRECTION
 DELY.....CELL DIMENSION IN Y-DIRECTION

READ (5,PARAM)
 WRITE (6,860) ITMAX,NTIMES,NODESX,NODESY,NICPTS,NBCPTS,KOUNT,TOL,DELX,DELY
 WRITE (6,840)
 WRITE (6,1080)
 LIMIT=NICPTS+NBCPTS
 MAX=NODESX*NODESY
 NTOPIX=NODESX-1
 NTOPIY=NODESY-1
 IF (DELX.LE.0.0.OR.DELY.LE.0.0.OR.NODESX.LE.0.0.OR.NODESY.LE.0.0.OR.NICPTS.LT.0.0.OR.NBCPTS.LT.0) WRITE (6,870)
 IF (DELX.LE.0.0.OR.DELY.LE.0.0.OR.NODESX.LE.0.0.OR.NODESY.LE.0.0.OR.NICPTS.LT.0.0.OR.NBCPTS.LT.0) STOP
 IF (MAX.GT.NODES) WRITE (6,880)
 IF (MAX.GT.NODES) STOP
 IF (LIMIT.GT.NPTS) WRITE (6,890)
 IF (LIMIT.GT.NPTS) STOP

INPUT AND CHECK BASIC GRID SYSTEM

NCOUNT=0
 ASSIGN 110 TO MSPOT
 ASSIGN 190 TO NSPOT

READ THE INITIAL POINT AND BOUNDARY POINT ARRAYS

NOMENCLATURE:

WORD.....* LAST * IS A SENTINAL DESIGNATING THE END OF A SET OF INPUT DATA
 NODEF.....NUMBER OF FIRST POINT IN A ROW OR COLUMN OF POINTS
 XF,YF.....COORDINATES OF THE FIRST POINT
 NODEL.....NUMBER OF LAST POINT IN A ROW OR COLUMN OF POINTS
 XL,YL.....COORDINATES OF THE LAST POINT

100 READ (5,900) WORD,NODEF,XF,YF,NODEL,XL,YL
 IF (WORD.EQ.LAST) GO TO 180
 WRITE (6,900) WORD,NODEF,XF,YF,NODEL,XL,YL
 IF (NODEF.EQ.0.AND.NODEL.EQ.0) GO TO 180
 IF (NODEL.EQ.0) NODEL=NODEF
 IF (NODEL.LT.NODEF) WRITE (6,910)

```

IF (NODEL.LT.NODEF) STOP
R=SQRT((XL-XF)**2+(YL-YF)**2)
DELR=R/FLOAT(NODEL-NODEF)
CHEAT=0.0
IF (ABS(XL-XF).LT.1.E-10) CHEAT=1.E-10
THETA=ATAN((YL-YF)/(XL-XF+CHEAT))
GO TO MSPOT, (110,140)
110 IF (SUPRES.EQ.YES) GO TO 120
WRITE (6,920) NODEF,NODEL
120 DO 130 I=NODEF,NODEL
XMOVE(I)=XF+DELR*FLOAT(I-NODEF)*COS(THETA)
YMOVE(I)=YF+DELR*FLOAT(I-NODEF)*SIN(THETA)
IF (SUPRES.EQ.YES) GO TO 130
WRITE (6,930) I,XMOVE(I),YMOVE(I)
130 CONTINUE
GO TO 170
140 CONTINUE
IF (SUPRES.EQ.YES) GO TO 150
WRITE (6,940) NODEF,NODEL
150 DO 160 I=NODEF,NODEL
MSTEP=I-NODEF+NCOUNT+1
XBOUND(MSTEP)=XF+DELR*FLOAT(I-NODEF)*COS(THETA)
YBOUND(MSTEP)=YF+DELR*FLOAT(I-NODEF)*SIN(THETA)
IF (SUPRES.EQ.YES) GO TO 160
WRITE (6,930) MSTEP,XBOUND(MSTEP),YBOUND(MSTEP)
160 CONTINUE
170 NCOUNT=NCOUNT+(NODEL-NODEF+1)
GO TO 100
180 CONTINUE
GO TO NSPOT, (190,200)
190 CONTINUE
IF (NCOUNT.NE.NICPTS) WRITE (6,950)
IF (NCOUNT.NE.NICPTS) STOP
NCOUNT=0
ASSIGN 140 TO MSPOT
ASSIGN 200 TO NSPOT
GO TO 100
200 CONTINUE
IF (NCOUNT.NE.NBCPTS) WRITE (6,960)
IF (NCOUNT.NE.NBCPTS) STOP
DO 210 I=1,NBCPTS
XMOVE(NICPTS+I)=XBOUND(I)
YMOVE(NICPTS+I)=YBOUND(I)
INDEX(I)=NICPTS+I
210 CONTINUE
NCOUNT=0

C
C READ THE BOUNDARY CELL I,J COORDINATES FROM CARDS
C
C NOMENCLATURE:
C IFIRST,JFIRST...I,J COORDINATES OF THE FIRST BOUNDARY CELL IN A
C ROW OR COLUMN OF CELLS
C ILAST,JLAST.....I,J COORDINATES OF THE LAST BOUNDARY CELL IN A
C ROW OR COLUMN OF CELLS
C

```



```

220  READ (5,970) WORD,IFIRST,JFIRST,ILAST,JLAST
    IF (WORD.EQ.LAST) GO TO 260
    IF (IFIRST.LE.0.AND.JFIRST.LE.0) GO TO 260
    IF ((ILAST.LT.IFIRST.AND.ILAST.GT.0).OR.(JLAST.LT.JFIRST.AND.JLAST
1.GT.0)) WRITE (6,980)
    IF ((ILAST.LT.IFIRST.AND.ILAST.GT.0).OR.(JLAST.LT.JFIRST.AND.JLAST
1.GT.0)) STOP
    IF (ILAST.LE.0) ILAST=IFIRST
    IF (JLAST.LE.0) JLAST=JFIRST
    DO 250 J=JFIRST,JLAST
    DO 250 I=IFIRST,ILAST
    IJ=(J-1)*NTOPX+I
    IF (NCOUNT.LE.0) GO TO 240
    DO 230 ISKIP=1,NCOUNT
    IF (IJ.EQ.IJBDRY(ISKIP)) GO TO 250
230  CONTINUE
240  CONTINUE
    NCOUNT=NCOUNT+1
    IJBDRY(NCOUNT)=IJ
250  CONTINUE
    GO TO 220
260  IF (NCOUNT.NE.KOUNT) WRITE (6,990) NCOUNT
    IF (NCOUNT.NE.KOUNT) KOUNT=NCOUNT
    IF (SUPRES.EQ.YES) GO TO 280
    WRITE (6,1000)
    DO 270 IPLAY=1,KOUNT
    IJXY=IJBDRY(IPLAY)
    IJY=((IJXY-EP3)/NTOPX)+1
    IJX=IJXY-(IJY-1)*NTOPX
    IF (IJX.LE.0) IJX=NTOPX
    WRITE (6,1010) IJX,IJY
270  CONTINUE
280  CONTINUE
    CALL NTRAN (3,1,KOUNT,IJBDRY,M1)
290  IF (M1.EQ.-1) GO TO 290
    IF (M1.LT.0) WRITE (6,1110) M1
    IF (M1.LT.0) STOP

C
C
C  READ OR COMPUTE THE INITIAL VELOCITY FIELD

    IF (VCARDS.NE.YES) GO TO 350
    IF (STEADY.NE.YES) READ (5,1020) DELT
    IF (POTEN.EQ.YES) GO TO 330
    CALL CARDRD (VX)
    CALL CARDRD (VY)
    IF (VARDIS.EQ.YES) CALL CARDRD (PERM)
    IF (SUPRES.EQ.YES) GO TO 320
    WRITE (6,1040)
    DO 310 MJ=1,NODESY
    NLINE=NODESY-MJ
    LINUM=NLINE+1
    DO 300 MI=1,NODESX
    STOREX(MI)=VX(NLINE*NODESX+MI)
    STOREY(MI)=VY(NLINE*NODESX+MI)
300  CONTINUE

```

```

WRITE (6,1050) LINUM
WRITE (6,1060) (STOREX(IJ),IJ=1,NODESX)
WRITE (6,1060) (STOREY(IJ),IJ=1,NODESX)
310 CONTINUE
320 CONTINUE
GO TO 340
330 CALL CARDRD (PHI)
CALL CARDRD (PERM)
CALL CARDRD (PORS)
CALL DARCY (VX,VY)
340 IF (VARTHK.EQ.YES) CALL CARDRD (H)
GO TO 410
350 IF (STEADY.NE.YES) CALL TAPERD (4,1,DELT)
IF (POTEN.EQ.YES) GO TO 390
CALL TAPERD (4,MAX,VX)
CALL TAPERD (4,MAX,VY)
IF (VARDIS.EQ.YES) CALL TAPERD (4,MAX,PERM)
IF (SUPRES.EQ.YES) GO TO 380
WRITE (6,1040)
DO 370 MJ=1,NODESY
NLINE=NODESY-MJ
LINUM=NLINE+1
DO 360 MI=1,NODESX
STOREX(MI)=VX(NLINE+NODESX+MI)
STOREY(MI)=VY(NLINE+NODESX+MI)
360 CONTINUE
WRITE (6,1050) LINUM
WRITE (6,1060) (STOREX(IJ),IJ=1,NODESX)
WRITE (6,1060) (STOREY(IJ),IJ=1,NODESX)
370 CONTINUE
380 CONTINUE
GO TO 400
390 CALL TAPERD (4,MAX,PHI)
CALL TAPERD (4,MAX,PERM)
CALL TAPERD (4,MAX,PORS)
CALL DARCY (VX,VY)
400 IF (VARTHK.EQ.YES) CALL TAPERD (4,MAX,H)
410 MAXIN=NTOPX*NTOPY
C
C   COMPUTE THE MEDIA THICKNESS IF APPLICABLE
C
IF (VARTHK.NE.YES) GO TO 430
CALL THICK
CALL NTRAN (3,1,MAXIN,H,M1)
420 IF (M1.EQ.-1) GO TO 420
IF (M1.LT.0) WRITE (6,1110) M1
IF (M1.LT.0) STOP
C
C   COMPUTE THE DISPERSION COEFFICIENTS IF APPLICABLE
C
430 IF (VARDIS.NE.YES) GO TO 460
CALL DCOMP (VX)
CALL NTRAN (3,1,MAXIN,VDIS,M1)
440 IF (M1.EQ.-1) GO TO 440
IF (M1.LT.0) WRITE (6,1110) M1

```

```

      IF (M1.LT.0) STOP
      CALL DCOMP (VY)
      CALL NTRAN (3,1,MAXIN,VDIS,M1)
450   IF (M1.EQ.-1) GO TO 450
      IF (M1.LT.0) WRITE (6,1110) M1
      IF (M1.LT.0) STOP
C
C      COMPUTE AND CHECK RECOMMENDED TIME STEP SIZE
C
460   CALL VELMAX (VX,VMAXX,DELT,1)
      CALL VELMAX (VY,VMAXY,DELT,2)
      IF (DELT.LT.EPS) TIMECK=DELT
      IF (DELT.LT.EPS) TIMECK=DELT
      IF (DELT.LT.EPS.AND.DELT.LT.EPS) WRITE (6,1070) TIMECK
      IF (DELT.LT.EPS.AND.DELT.LT.EPS) STOP
      IF (DELT.LE.DELT.AND.DELT.GT.EPS) TIMECK=DELT
      IF (DELT.GT.DELT.AND.DELT.GT.EPS) TIMECK=DELT
      IF (STEADY.EQ.YES) DELT=TIMECK
      IF (TIMECK.LT.DELT) WRITE (6,1090) TIMECK,DELT
C
C      WRITE INITIAL TIME PLANE DATA TO TAPE
C
      NN=0
      CALL NTRAN (3,1,3,NN,M2)
470   IF (M2.EQ.-1) GO TO 470
      IF (M2.LT.0) WRITE (6,1120) M2
      IF (M2.LT.0) STOP
      CALL NTRAN (3,1,LIMIT,XMOVE,M3)
480   IF (M3.EQ.-1) GO TO 480
      IF (M3.LT.0) WRITE (6,1130) M3
      IF (M3.LT.0) STOP
      CALL NTRAN (3,1,LIMIT,YMOVE,M4)
490   IF (M4.EQ.-1) GO TO 490
      IF (M4.LT.0) WRITE (6,1140) M4
      IF (M4.LT.0) STOP
      IF (SUPRES.EQ.YES) GO TO 500
      WRITE (6,1030) NN,LIMIT,DELT
      WRITE (6,930) (I,XMOVE(I),YMOVE(I),I=1,LIMIT)
500   CONTINUE
      LIMITP=LIMIT.
C
C      ENTER TIME PLANE LOOP
C
      DO 820 N=2,NTIMES
      IF (STEADY.NE.YES) GO TO 520
      IF (N.GT.2) GO TO 590
      DO 510 I=1,MAX
      VXX(I)=VX(I)
      VYY(I)=VY(I)
510   CONTINUE
      GO TO 590
520   IF (N.LE.2) GO TO 540
      DO 530 I=1,MAX
      VX(I)=VXX(I)
      VY(I)=VYY(I)

```

```

530 CONTINUE
540 CONTINUE
C
C READ OR COMPUTE THE CURRENT VELOCITY FIELD
C
IF (VCARDS.NE.YES) GO TO 560
READ (5,1020) DELT
IF (POTEN.EQ.YES) GO TO 550
CALL CARDRD (VXX)
CALL CARDRD (VYY)
GO TO 580
550 CALL CARDRD (PHI)
CALL DARCY (VXX,VYY)
GO TO 580
560 CALL TAPERD (4,1,DELT)
IF (POTEN.EQ.YES) GO TO 570
CALL TAPERD (4,MAX,VXX)
CALL TAPERD (4,MAX,VYY)
GO TO 580
570 CALL TAPERD (4,MAX,PHI)
IF (VARPER.EQ.YES) CALL TAPERD (4,MAX,PERM)
IF (VARPER.EQ.YES) CALL TAPERD (4,MAX,PORS)
CALL DARCY (VXX,VYY)
580 CONTINUE
VMAXX=0.0
VMAXY=0.0
CALL VELMAX (VXX,VMAXX,DELT,1)
CALL VELMAX (VYY,VMAXY,DELT,2)
IF (DELT.LT.EPS) TIMECK=DELT
IF (DELT.LT.EPS) TIMECK=DELT
IF (DELT.LT.EPS.AND,DELT.LT.EPS) WRITE (6,1070) TIMECK
IF (DELT.LT.EPS.AND,DELT.LT.EPS) STOP
IF (DELT.LE,DELT.AND,DELT.GT.EPS) TIMECK=DELT
IF (DELT.GT,DELT.AND,DELT.GT.EPS) TIMECK=DELT
IF (TIMECK.LT,DELT) WRITE (6,1090) TIMECK,DELT
C
C COMPUTE CURRENT COORDINATES OF THE PARTICLES
C
C
590 DO 630 NPOINT=1,LIMIT
ITER=0
XX=XMOVE(NPOINT)
YY=YMOVE(NPOINT)
C
C PREDICTOR EQUATIONS
C
CALL HOUND (XX,YY,VX,VXSTAR)
CALL HOUND (XX,YY,VY,VYSTAR)
XBAR=XX+DELT*VXSTAR
YBAR=YY+DELT*VYSTAR
C
C CORRECTOR EQUATIONS
C
600 CALL HOUND (XBAR,YBAR,VXX,VXBAR)
CALL HOUND (XBAR,YBAR,VYY,VYBAR)
ITER=ITER+1

```

```

XXX=XX+DELTA*(VXSTAR+VXBAR)/2.
YYY=YY+DELTA*(VYSTAR+VYBAR)/2.
ERROR=SQRT(((XXX-XBAR)**2)/XXX+((YYY-YBAR)**2)/YYY)
IF (ERROR.LE.TOL.OR.ITER.GE.ITMAX) GO TO 610
XBAR=XXX
YBAR=YYY
GO TO 600
610 ITER=0
IX=IFIX(XXX/DELX+EPS)+1
JY=IFIX(YYY/DELY+EPS)+1
IJ=(JY-1)*NTOPX+IX
DO 620 L=1,KOUNT
IF (IJBDRY(L),NE,IJ) GO TO 620
IPOINT=IPOINT+1
INDEX(IPOINT)=NPOINT
GO TO 630
620 CONTINUE
XMOVE(NPOINT)=XXX
YMOVE(NPOINT)=YYY
630 CONTINUE
C
C DETERMINE THE NECESSARY POINT REPLACEMENTS
C
IPTS=IPOINT
IPOINT=0
LIMSFT=0
IF (NBCPTS,EQ,0) GO TO 650
IF (IPTS.LT,NBCPTS) GO TO 680
DO 640 I=1,NBCPTS
J=INDEX(I)
XMOVE(J)=XBOUND(I)
YMOVE(J)=YBOUND(I)
640 CONTINUE
650 IF (IPTS.LE,NBCPTS) GO TO 720
I1=NBCPTS+1
DO 670 I=I1,IPTS
LIMIT=LIMIT-1
NPOINT=INDEX(I)
IF (NPOINT.GT,LIMIT) GO TO 670
LIMSFT=LIMSFT+1
NFLAG(LIMSFT)=NPOINT
DO 660 J=NPOINT,LIMIT
XMOVE(J)=XMOVE(J+1)
YMOVE(J)=YMOVE(J+1)
660 CONTINUE
670 CONTINUE
GO TO 720
680 IF (IPTS.LT,1) GO TO 700
DO 690 I=1,IPTS
J=INDEX(I)
XMOVE(J)=XBOUND(I)
YMOVE(J)=YBOUND(I)
690 CONTINUE
700 IFF=IPTS+1
DO 710 I=IFF,NBCPTS

```

```

LIMIT=LIMIT+1
XMOVE(LIMIT)=XBOUND(I)
YMOVE(LIMIT)=YBOUND(I)
INDEX(I)=LIMIT
710 CONTINUE
C
C WRITE CURRENT TIME PLANE DATA TO TAPE
C
720 NN=N-1
WRITE (6,1030) NN,LIMIT,DELT
CALL NTRAN (3,1,3,NN,M2)
730 IF (M2.EQ.-1) GO TO 730
IF (M2.LT.0) WRITE (6,1120) M2
IF (M2.LT.0) STOP
CALL NTRAN (3,1,LIMIT,XMOVE,M3)
740 IF (M3.EQ.-1) GO TO 740
IF (M3.LT.0) WRITE (6,1130) M3
IF (M3.LT.0) STOP
CALL NTRAN (3,1,LIMIT,YMOVE,M4)
750 IF (M4.EQ.-1) GO TO 750
IF (M4.LT.0) WRITE (6,1140) M4
IF (M4.LT.0) STOP
IF (LIMIT.GE.LIMITP) GO TO 780
CALL NTRAN (3,1,1,LIMSFT,M2)
760 IF (M2.EQ.-1) GO TO 760
IF (M2.LT.0) WRITE (6,1120) M2
IF (M2.LT.0) STOP
IF (LIMSFT.EQ.0) WRITE (6,1100) LIMSFT
IF (LIMSFT.EQ.0) GO TO 780
CALL NTRAN (3,1,LIMSFT,NFLAG,M2)
770 IF (M2.EQ.-1) GO TO 770
IF (M2.LT.0) WRITE (6,1120) M2
IF (M2.LT.0) STOP
WRITE (6,1100) LIMSFT,(NFLAG(J),J=1,LIMSFT)
780 LIMITP=LIMIT
IF (STEADY.EQ.YES.OR.VARDIS.NE.YES) GO TO 810
CALL OCOMP (VXX)
CALL NTRAN (3,1,MAXIN,VDIS,M1)
790 IF (M1.EQ.-1) GO TO 790
IF (M1.LT.0) WRITE (6,1110) M1
IF (M1.LT.0) STOP
CALL OCOMP (VYY)
CALL NTRAN (3,1,MAXIN,VDIS,M1)
800 IF (M1.EQ.-1) GO TO 800
IF (M1.LT.0) WRITE (6,1110) M1
IF (M1.LT.0) STOP
810 IF (SUPRES.EQ.YES) GO TO 820
WRITE (6,930) (I,XMOVE(I),YMOVE(I),I=1,LIMIT)
820 CONTINUE
END FILE 3
STOP
C
C
C
830 FORMAT (13A6)

```

```

840  FORMAT (/ ,16(0*0,4X),/)
850  FORMAT (20X,0 PROBLEM DESCRIPTION 0//0 IS THE FLOW FIELD STEADY?..
1.....0,A6/0 IS THE MEDIA THICKNESS VARIABLE?...
2.....0,A6/0 ARE THE DISPERSION COEFFICIENTS VARIABLE?...
3.....0,A6/0 IS THE VELOCITY OR POTENTIAL DATA ON CARDS?.....
40,A6/0 IS THE INPUT DATA IN TERMS OF POTENTIAL?.....0,A6/0 I
5S THE NON-ESSENTIAL PRINTING TO BE SUPRESSED?.....0,A6/0 DOES THE
6PERMEABILITY VARY WITH TIME?.....0,A6)
860  FORMAT (20X,0 PROBLEM SPECIFICATIONS0//0 MAXIMUM NUMBER OF ITERATI
1ONS FOR EACH TRAJECTORY .....0,IS/0 NUMBER OF TIME PLANES TO BE CO
2MPUTED 0,17(0.0),IS/0 NUMBER OF NODES IN THE X-DIRECTION 0,19(0.0)
3,IS/0 NUMBER OF NODES IN THE Y-DIRECTION 0,19(0.0),IS/0 NUMBER OF
4INITIAL CONDITION POINTS 0,19(0.0),IS/0 NUMBER OF BOUNDARY CONDITI
5ON POINTS 0,18(0.0),IS/0 NUMBER OF BOUNDARY CELLS 0,29(0.0),IS/0 A
6ALLOWABLE DEVIATION FOR EACH TRAJECTORY CALCULATION .. 0,E10.4/0 CE
7LL DIMENSION IN THE X-DIRECTION 0,20(0.0),E11.4/0 CELL DIMENSION I
8N THE Y-DIRECTION 0,20(0.0),E11.4)
870  FORMAT (0 0,0 ONE OF THE INPUT VARIABLES IS LT OR EQ TO ZERO0)
880  FORMAT (/ ,0 MAX IS GT NODS 0)
890  FORMAT (/ ,0 LIMIT IS GT NPTS0)
900  FORMAT (A6,I4,2F10.4,IS,2F10.4)
910  FORMAT (0 0,0 NODEL IS LT NODEF0)
920  FORMAT (0 ...INITIAL CONDITION POINTS:  NODES RANGE FROM0,I4,5H
1TO,I4)
930  FORMAT (3(I6,3X,E10.4,3X,E10.4,6X))
940  FORMAT (0 ...BOUNDARY CONDITION POINTS:  NODES RANGE FROM0,I4,5H
1 TO,I4)
950  FORMAT (0 0,0 NCOUNT IS .NE. TO NICPTS0)
960  FORMAT (0 0,0 NCOUNT IS .NE. TO NBCPTS0)
970  FORMAT (A6,I4,3I5)
980  FORMAT (/ ,0 CARD ERROR IN BOUNDARY CELL READ IN 0)
990  FORMAT (1X,0NCOUNT NOT EQUAL TO KOUNT, NCOUNT = 0,IS)
1000  FORMAT (1H1,/,5X,0...BOUNDARY ARRAY...0,/,12X,0I0,5X,0J0,/)
1010  FORMAT (9X,2(I4,2X))
1020  FORMAT (10I10)
1030  FORMAT (/,0 TIME PLANE = 0IS,4X,0NUMBER OF POINTS =0,IS,4X,0 TIME
1 STEP SIZE =0,E10.4)
1040  FORMAT (1H1,/,5X,0...VELOCITY FIELD:  VX,VY...0,/)
1050  FORMAT (/,0 J = 0,IS)
1060  FORMAT (1P10E12.5)
1070  FORMAT (/ ,0 THE RECOMMENDED MAX TIME STEP SIZE IS TOO SMALL.  TIM
1ECK = 0,E10.4)
1080  FORMAT (1H1)
1090  FORMAT (/ ,0 WARNING: THE TIME STEP SIZE EXCEEDS THE RECOMMENDED MA
1XIMUM.0/0 RECOMMENDED MAXIMUM TIME STEP SIZE = 0,E10.4/0 ACTUAL TI
2ME STEP SIZE = 0,E12.4)
1100  FORMAT (1X,0 LIMSFT0,20I5)
1110  FORMAT (1X,0 M1 = 0,I4)
1120  FORMAT (1X,0 M2 = 0,I4)
1130  FORMAT (1X,0 M3 = 0,I4)
1140  FORMAT (1X,0 M4 = 0,I4)
      END

```

```

SUBROUTINE DARCY (VX,VY)
C
C THIS SUBROUTINE COMPUTES THE VELOCITY DISTRIBUTION FROM THE
C POTENTIAL, PERMEABILITY, AND POROSITY DISTRIBUTIONS.
C
PARAMETER NODS=2000, NEX=250, NPTS=4000
COMMON/TRK/NODESX,NODESY,NN,LIMIT,DELT,DELX,DELY,MAX,EPS,SUPRES,YE
1S,PHI(NODS),PERM(NODS),PORS(NODS),STOREX(100),STOREY(100)
DIMENSION VX(NODS),VY(NODS)
IF (SUPRES.EQ.YES) GO TO 100
WRITE (6,200)
WRITE (6,210) (PHI(I),I=1,MAX)
WRITE (6,220)
WRITE (6,210) (PERM(I),I=1,MAX)
WRITE (6,230)
WRITE (6,210) (PORS(I),I=1,MAX)
100 DO 160 J=1,NODESY
DO 160 I=1,NODESX
IJN=(J-1)*NODESX+I
IF (I.EQ.1) GO TO 110
IF (I.EQ.NODESX) GO TO 120
VX(IJN)=-((PERM(IJN)/PORS(IJN))*(PHI(IJN+1)-PHI(IJN-1)))/(2.0*DELX)
GO TO 130
110 VX(IJN)=-((PERM(IJN)/PORS(IJN))*(PHI(IJN+1)-PHI(IJN))/DELX
GO TO 130
120 VX(IJN)=-((PERM(IJN)/PORS(IJN))*(PHI(IJN)-PHI(IJN-1))/DELX
130 IF (J.EQ.1) GO TO 140
IF (J.EQ.NODESY) GO TO 150
VY(IJN)=-((PERM(IJN)/PORS(IJN))*(PHI(IJN+NODESX)-PHI(IJN-NODESX)))/(
12.0*DELY)
GO TO 160
140 VY(IJN)=-((PERM(IJN)/PORS(IJN))*(PHI(IJN+NODESX)-PHI(IJN))/DELY
GO TO 160
150 VY(IJN)=-((PERM(IJN)/PORS(IJN))*(PHI(IJN)-PHI(IJN-NODESX))/DELY
160 CONTINUE
IF (SUPRES.EQ.YES) GO TO 190
WRITE (6,240)
DO 180 MJ=1,NODESY
NLINE=NODESY-MJ
LINUM=NLINE+1
DO 170 MI=1,NODESX
STOREX(MI)=VX(NLINE*NODESX+MI)
STOREY(MI)=VY(NLINE*NODESX+MI)
170 CONTINUE
WRITE (6,250) LINUM
WRITE (6,210) (STOREX(IJ),IJ=1,NODESX)
WRITE (6,210) (STOREY(IJ),IJ=1,NODESX)
180 CONTINUE
190 CONTINUE
RETURN
C
C
C
200 FORMAT (/,25X,0 POTENTIAL DISTRIBUTION 0,/)

```



```

210  FORMAT (1P10E12.5)
220  FORMAT (/ ,25X,0 PERMEABILITY DISTRIBUTION 0,/)
230  FORMAT (/ ,25X,0 POROSITY DISTRIBUTION 0,/)
240  FORMAT (1H1,///,5X,0...VELOCITY FIELD: VX,VY...0,/)
250  FORMAT (///,0 J = 0,15)
      END

```

```

SUBROUTINE HOUND (X,Y,TERM,FX,Y)
C
C THIS SUBROUTINE INTERPOLATES NODAL VELOCITIES TO POINT VELOCITIES
C USING A BI-LINEAR, BI-QUADRATIC, OR BI-CUBIC INTERPOLATION SCHEME
C BASED ON NEWTONS DIVIDED DIFFERENCE METHOD
C
PARAMETER NODS=2000, NEX=250, NPTS=4000
COMMON/TRK/NODESX,NODESY,NN,LIMIT,DELT,DELX,DELY,MAX,EPS,SUPRES,YE
1S,PHI(NODS),PERM(NODS),PORS(NODS),STOREX(100),STOREY(100)
DIMENSION TERM(NODS)
II=IFIX(X/DELX+EPS)+1
JJ=IFIX(Y/DELY+EPS)+1
IF (II.LT.1.OR.JJ.LT.1.OR.II.GT.NODESX.OR.JJ.GT.NODESY) GO TO 120
IJ=(JJ-1)*NODESX+II
NXLES1=NODESX-1
NXLES2=NODESX-2
NYLES1=NODESY-1
NYLES2=NODESY-2
X0=FLOAT(II-2)*DELX
X1=FLOAT(II-1)*DELX
X2=FLOAT(II)*DELX
Y0=FLOAT(JJ-2)*DELY
Y1=FLOAT(JJ-1)*DELY
Y2=FLOAT(JJ)*DELY
B1=DELY
B2=2.*DELY**2
B3=6.*DELY**3
C1=(Y-Y0)
C2=(Y-Y0)*(Y-Y1)
C3=(Y-Y0)*(Y-Y1)*(Y-Y2)
IF (II.EQ.1.OR.JJ.EQ.1.OR.II.EQ.NXLES1.OR.JJ.EQ.NYLES1) GO TO 100
GO TO 110
C
C BI-LINER INTERPOLATION ...
C
100 ALPHA=(Y-Y1)/(Y2-Y1)
BETA=(X-X1)/(X2-X1)
FX,Y=ALPHA*(1.-BETA)*TERM(IJ+NODESX)+ALPHA*BETA*TERM(IJ+NODESX+1)+(
11.-ALPHA)*(1.-BETA)*TERM(IJ)+(1.-ALPHA)*BETA*TERM(IJ+1)
GO TO 130
C
C BI-QUADRATIC INTERPOLATION ...
C
110 FA=TERM(IJ-NODESX-1)+C1*(TERM(IJ-1)-TERM(IJ-NODESX-1))/B1+C2*(TERM
1(IJ+NODESX-1)-2.*TERM(IJ-1)+TERM(IJ-NODESX-1))/B2
FB=TERM(IJ-NODESX)+C1*(TERM(IJ)-TERM(IJ-NODESX))/B1+C2*(TERM(IJ+NO
1DESX)-2.*TERM(IJ)+TERM(IJ-NODESX))/B2
FC=TERM(IJ-NODESX+1)+C1*(TERM(IJ+1)-TERM(IJ-NODESX+1))/B1+C2*(TERM
1(IJ+NODESX+1)-2.*TERM(IJ+1)+TERM(IJ-NODESX+1))/B2
FX,Y=FA+(X-X0)*(FB-FA)/DELX+(X-X0)*(X-X1)*(FA-2.*FB+FC)/(2.*DELX**2
1)
IF (II.EQ.2.OR.JJ.EQ.2.OR.II.EQ.NXLES2.OR.JJ.EQ.NYLES2) GO TO 130
C
C BI-CUBIC INTERPOLATION ...

```

C

```

FD=TERM(IJ-NODESX+2)+C1*(TERM(IJ+2)-TERM(IJ-NODESX+2))/B1+C2*(TERM
1(IJ+NODESX+2)-2.*TERM(IJ+2)+TERM(IJ-NODESX+2))/B2+C3*(TERM(IJ+2*NO
2DESX+2)-3.*TERM(IJ+NODESX+2)+3.*TERM(IJ+2)-TERM(IJ-NODESX+2))/B3
FA=FA+C3*(TERM(IJ+2*NODESX-1)-3.*TERM(IJ+NODESX-1)+3.*TERM(IJ-1)-T
1ERM(IJ-NODESX-1))/B3
FB=FB+C3*(TERM(IJ+2*NODESX)-3.*TERM(IJ+NODESX)+3.*TERM(IJ)-TERM(IJ
1-NODESX))/B3
FC=FC+C3*(TERM(IJ+2*NODESX+1)-3.*TERM(IJ+NODESX+1)+3.*TERM(IJ+1)-T
1ERM(IJ-NODESX+1))/B3
FXY=FA+(X-X0)*(FB-FA)/DELX+(X-X0)*(X-X1)*(FA-2.*FB+FC)/(2.*DELX**2
1)+(X-X0)*(X-X1)*(X-X2)*(FD-3.*FC+3.*FB-FA)/(6.*DELX**3)
GO TO 130

```

120
130

```

WRITE (6,140)
RETURN

```

C
C
C

140

```

FORMAT (/ ,5X,0...ERROR: TRACK IS OUTSIDE GRID SYSTEM ...0)
END

```

```

SUBROUTINE THICK
C
C THIS SUBROUTINE COMPUTES AN AQUIFER THICKNESS FROM POTENTIAL DATA
C AND AN AQUIFER BOTTOM MAP.
C
PARAMETER NODS=2000, NEX=250, NPTS=4000
COMMON/TRK/NODESX,NODESY,NN,LIMIT,DELT,DELX,DELY,MAX,EPS,SUPRES,YE
1S,PHI(NODS),PERM(NODS),PORS(NODS),STOREX(100),STOREY(100),H(NODS),
2VDIS(NODS)
DO 100 I=1,MAX
H(I)=PHI(I)-H(I)
100 CONTINUE
NTOPX=NODESX-1
NTOPY=NODESY-1
MAXIN=NTOPX*NTOPY
DO 120 J=1,NTOPY
DO 110 K=1,NTOPX
IJN=(J-1)*NODESX+K
IJ=(J-1)*NTOPX+K
H(IJ)=(H(IJN)+H(IJN+1)+H(IJN+NODESX)+H(IJN+NODESX+1))/4.0
110 CONTINUE
120 CONTINUE
IF (SUPRES.EQ.YES) GO TO 130
WRITE (6,140)
WRITE (6,150) (H(I),I=1,MAXIN)
130 CONTINUE
RETURN
C
C
C
140 FORMAT (35X,' MEDIA THICKNESS ARRAY, H#)
150 FORMAT (10E12.4)
END

```

SUBROUTINE OCOMP (V)

THIS SUBROUTINE COMPUTES DISPERSION COEFFICIENT ARRAYS FROM VELOCITIES AND/OR PERMEABILITY DATA.

PARAMETER NODS=2000, NEX=250, NPTS=4000
COMMON/TRK/NODESX,NODESY,NN,LIMIT,DELT,DETX,DELY,MAX,EPS,SUPRES,YE
IS,PHI(NODS),PERM(NODS),PORS(NODS),STOREX(100),STOREY(100),H(NODS),
2VDIS(NODS)

DIMENSION V(1)

NTOPX=NODESX-1

NTOPY=NODESY-1

MAXIN=NTOPX*NTOPY

DO 110 J=1,NTOPY

DO 100 I=1,NTOPX

IJN=(J-1)*NODESX+I

IJ=(J-1)*NTOPX+I

VCEN=(V(IJN)+V(IJN+1)+V(IJN+NODESX)+V(IJN+NODESX+1))/4.0

VDIS(IJ)=0.099*ABS(VCEN)+0.0028

VDIS(IJ)=1.55*VDIS(IJ)

100 CONTINUE

110 CONTINUE

IF (SUPRES.EQ.YES) GO TO 120

WRITE (6,130)

WRITE (6,140) (VDIS(I),I=1,MAXIN)

120 CONTINUE

RETURN

130 FORMAT (35X,' DISPERSION COEFFICIENT ARRAY, VDIS')

140 FORMAT (10E12.4)

END

```

SUBROUTINE VELMAX (V,VMAX,TIME,L)
C
C THIS SUBROUTINE FINDS THE MAXIMUM VELOCITY WHICH IS THEN USED TO
C CALCULATE THE TIME STEP SIZE FOR USE IN MICRO, MACRO AND TRACK
C
PARAMETER NODS=2000, NEX=250, NPTS=4000
COMMON/TRK/NODESX,NODESY,NN,LIMIT,DELT,DELX,DELY,MAX,EPS,SUPRES,YE
1S,PHI(NODS),PERM(NODS),PORS(NODS),STOREX(100),STOREY(100)
DIMENSION V(NODS)
DO 100 I=1,MAX
IF (ABS(V(I)).GT,VMAX) VMAX=ABS(V(I))
100 CONTINUE
IF (VMAX.EQ.0.0) GO TO 120
IF (L.GT.1) GO TO 110
TIME=DELX/VMAX
GO TO 120
110 TIME=DELY/VMAX
120 CONTINUE
RETURN
END

```

```

SUBROUTINE CARDRD (X)
C
C THIS SUBROUTINE INPUTS VELOCITY DATA FROM CARDS.
C
COMMON/TRK/NODESX,NODESY
DIMENSION X(1)
REAL LAST
DATA LAST/6H LAST/
C
C NOMENCLATURE:
C   WORD.....SENTINAL WORD TERMINATING INPUT
C   PARAM.....PARAMETER WHICH IS TO BE READ
C   I1,J1.....I,J COORDINATES OF THE FIRST NODE IN A ROW OR COLUMN
C               OF NODES
C   I2,J2.....I,J COORDINATES OF THE LAST NODE IN A ROW OR COLUMN
C               OF NODES
C
100 READ (5,150) WORD,PARAM,I1,J1,I2,J2
WRITE (6,150) WORD,PARAM,I1,J1,I2,J2
IF (WORD.EQ.LAST) GO TO 140
IF (I1.EQ.0.AND.J1.EQ.0) GO TO 140
IF ((I2.GT.0.AND.I1.GT.I2).OR.(J2.GT.0.AND.J1.GT.J2)) WRITE (6,160
1)
IF ((I2.GT.0.AND.I1.GT.I2).OR.(J2.GT.0.AND.J1.GT.J2)) STOP
IF (I2.NE.0) GO TO 110
I2=I1
110 IF (J2.NE.0) GO TO 120
J2=J1
120 DO 130 J=J1,J2
DO 130 I=I1,I2
IJ=(J-1)*NODESX+I
X(IJ)=PARAM
130 CONTINUE
GO TO 100
140 RETURN
C
C
C
150 FORMAT (A6,E9.4,4I5)
160 FORMAT (0 0,0 SPECIFIED INPUT IS INDEXED INCORRECTLY0)
END

```

```

SUBROUTINE TAPERD (NUNIT,NSIZE,NSTORE)
C
C THIS SUBROUTINE INPUTS VELOCITY DATA FROM MAGNETIC TAPE.
C
CALL NTRAN (NUNIT,2,NSIZE,NSTORE,L)
100 IF (L.EQ.-1) GO TO 100
IF (L.LT.0) WRITE (6,110) NUNIT,L
IF (L.LT.0) STOP
RETURN
C
C
C
110 FORMAT (' TAPE READ ERROR, UNIT = ',I2,' L = ',I2)
END

```


MICRO: A PROGRAM FOR THE NUMERICAL SOLUTION OF THE TIME DEPENDENT TWO DIMENSIONAL, MASS TRANSPORT EQUATION WITH SOURCE TERMS PRESENT. THE SOLUTION ALGORITHM IS BASED ON THE METHOD OF CHARACTERISTICS AND IS APPLICABLE TO CASES OF LOW OR ZERO DISPERSION. THE MASS TRANSPORT PARTIAL DIFFERENTIAL EQUATION IS CAST AS:

$$P(C+B*A)/PT = (1/H)*P(DX*H*P(C)/PX)/PX \\ + (1/H)*P(DY*H*P(C)/PY)/PY \\ = U*P(C)/PX - V*P(C)/PY - L*(C+B*A)$$

WHERE:

P()/PT.....PARTIAL DERIVATIVE W.R.T. TIME
P()/PX, P()/PY..PARTIAL DERIVATIVE W.R.T. SPACE COORDS
C(X,Y,T).....LIQUID PHASE CONCENTRATION
A(X,Y,T).....SORBED CONCENTRATION
DX, DY.....DISPERSION COEFFICIENTS
U(X,Y,T).....FLOW VELOCITY IN X-DIRECTION
V(X,Y,T).....FLOW VELOCITY IN Y-DIRECTION
B.....SOIL TO SOLUTION RATIO
L.....DECAY CONSTANT
H.....AQUIFER THICKNESS

WRITTEN BY 1) R.G. BACA
2) A. BRANDSTETTER
3) S.W. AHLSTROM

PROGRAMMED BY 1) R.G. BACA
2) D. FRIEDRICHS
3) S.W. AHLSTROM

VERSION 1 - MAY 30, 1971
VERSION 2 - JAN 31, 1972

BATTELLE-NORTHWEST: WATER RESOURCES SYSTEMS SECTION

REFERENCES: 1) A.O. GARDER, ET. AL.:@NUMERICAL CALCULATION OF MULTIDIMENSIONAL MISCIBLE DISPLACEMENT BY THE METHOD OF CHARACTERISTICS@, SOC. PET. ENGRS. JOUR. (MARCH 1964) P.26
2) G.F. PINDER, ET. AL.:@A NUMERICAL TECHNIQUE FOR CALCULATING THE TRANSIENT POSITION OF THE SALTWATER FRONT@, WATER RESOURCES RESEARCH, VOL.6, NO.3 (JUNE 1970) P.875

PARAMETER NPTS=1500, NCEL=1100, NBDY=250
COMMON/MIC/NPL,LIMIT,DELT,DELX,DELY,KOUNT,NTOPX,NTOPY,SUPRES,MAXIN
COMMON/LAR/XMOVE(NPTS),YMOVE(NPTS),NN,CIJ(NCEL),CSTAR(NCEL),CPOINT
1(NPTS),IJBORY(NBDY),DELC(NCEL),STORE(100),VKD(NCEL),VKDP(NCEL)
DIMENSION TITLE(13),NFLAG(NBDY),H(NCEL),VDISX(NCEL),VDISY(NCEL)
DIMENSION EP(100),UP(100),AA(100),BB(100),DD(100),EE(100),

```

1TER(NCEL)
NAMELIST/SPECS/ NFREQ,NTIMES,NTOPX,NTOPY,KOUNT,DELX,DELY
NAMELIST/PRMS/ DIFUSX,DIFUSY,ZKO,DBULK,ZLAMB
REAL LAST
DATA LAST,YES,EPS/6H LAST,6H YES,0.00001/

C
C
C   READ THE PROBLEM TITLE FROM CARDS (1)

C
C   CALL ETIME
C   CALL ETIMEF (TSTRT)
C   READ (5,1070) (TITLE(I),I=1,13)
C   WRITE (6,1270)
C   WRITE (6,1080)
C   WRITE (6,1070) (TITLE(I),I=1,13)

C
C   READ THE PROBLEM DESCRIPTION FROM CARDS (2)

C
C   QUESTIONS: (YES OR NO)
C   DISPER.....DOES THE PROBLEM CONSIDER DISPERSION?
C   TAPEOP.....ARE THE FINAL RESULTS TO BE OUTPUT TO TAPE?
C   SUPRES.....IS THE NON-ESSENTIAL OUTPUT TO BE SUPRESSED?
C   VARYKD.....DOES THE PROBLEM CONSIDER A VARYING KD?
C   STEADY.....IS THE FLOW FIELD STEADY?
C   VARDIS.....IS THE DISPERSION COEFFICIENT VARIABLE?
C   VARTHK.....DOES THE PROBLEM CONSIDER VARING AQUIFER THICKNESS?
C   SOLVEI.....IS THE IMPLICIT ALGORITHM TO BE USED?

C
C   READ (5,1070) DISPER,TAPEOP,SUPRES,VARYKD,STEADY,VARDIS,VARTHK,SOL
1VEI
C   WRITE (6,1080)
C   WRITE (6,1090) DISPER,TAPEOP,SUPRES,VARYKD,STEADY,VARDIS,VARTHK,SO
1LVEI

C
C   READ THE PROBLEM SPECIFICATION FROM CARDS (3)

C
C   NOMENCLATURE:
C   NFREQ.....PRINT FREQUENCY
C   NTIMES.....NUMBER OF TIME PLANES
C   NTOPIX.....NUMBER OF CELLS IN X-DIRECTION
C   NTOPIY.....NUMBER OF CELLS IN Y-DIRECTION
C   KOUNT.....NUMBER OF BOUNDARY CELLS
C   DELX.....CELL DIMENSION IN X-DIRECTION
C   DELY.....CELL DIMENSION IN Y-DIRECTION

C
C   READ (5,SPECS)
C   WRITE (6,1080)
C   WRITE (6,1100) NTIMES,NFREQ,NTOPX,NTOPY,KOUNT,DELX,DELY

C
C   READ THE INPUT PARAMETERS FROM CARDS (4)

C
C   NOMENCLATURE:
C   DIFUSX.....DISPERSION COEFFICIENT, X-DIRECTION
C   DIFUSY.....DISPERSION COEFFICIENT, Y-DIRECTION
C   ZKO.....APPROXIMATE SORPTION CONSTANT
C   DBULK.....BULK DENSITY

```

```

IF (L4.LT.0) STOP
IF (NPL.NE.(N-1)) WRITE (6,1260) NPL,N
IF (NPL.NE.(N-1)) STOP
NDIF=LIMITP-LIMIT
IF (NDIF.LE.0) GO TO 330
CALL NTRAN (3,2,1,LIMSFT,L2)
280 IF (L2.EQ.-1) GO TO 280
IF (L2.LT.0) WRITE (6,1220) L2
IF (L2.LT.0) STOP
IF (LIMSFT.EQ.0) GO TO 330
CALL NTRAN (3,2,LIMSFT,NFLAG,L2)
290 IF (L2.EQ.-1) GO TO 290
IF (L2.LT.0) WRITE (6,1220) L2
IF (L2.LT.0) STOP
IF (LIMSFT.GE.NDIF) GO TO 300
LIMITP=LIMITP-(NDIF-LIMSFT)
300 DO 320 K=1,LIMSFT
JK=NFLAG(K)
LIMITP=LIMITP-1
DO 310 JL=JK,LIMITP
CPOINT(JL)=CPOINT(JL+1)
310 CONTINUE
320 CONTINUE
330 IF (VARDIS.NE.YES.OR.STEADY.EQ.YES) GO TO 370
CALL NTRAN (3,2,MAXIN,VDISX,L1)
340 IF (L1.EQ.-1) GO TO 340
IF (L1.LT.0) WRITE (6,1210) L1
IF (L1.LT.0) STOP
IF (SUPRES.EQ.YES) GO TO 350
WRITE (6,1200) (VDISX(I),I=1,MAXIN)
350 CALL NTRAN (3,2,MAXIN,VDISY,L1)
360 IF (L1.EQ.-1) GO TO 360
IF (L1.LT.0) WRITE (6,1210) L1
IF (L1.LT.0) STOP
IF (SUPRES.EQ.YES) GO TO 390
WRITE (6,1200) (VDISY(I),I=1,MAXIN)
370 IF (SUPRES.EQ.YES) GO TO 390
WRITE (6,1140) NPL,LIMIT,DELT
IF (LIMSFT.LE.0) GO TO 380
WRITE (6,1130) (NFLAG(J),J=1,LIMSFT)
380 WRITE (6,1150) (I,XMOVE(I),YMOVE(I),I=1,LIMIT)
390 CONTINUE
CALL TIMER (TSTRT,TNEW)

C
C   FIND THE AVERAGE CONC. IN EACH CELL AFTER POINT MOVEMENT
C
CALL AVRGE
DO 400 J=1,NTOPY
IJ=(J-1)*NTOPX+1
CSTAR(IJ)=CSTAR(IJ+1)
400 CONTINUE

C
C   COMPUTE DECAY LOSSES
C
IF (ZLAMB.LT.1.0E-08) GO TO 430

```

```

DO 420 IJ=1,MAXIN
DO 410 K=1,KOUNT
IF (IJ.EQ.IJBDRY(K)) GO TO 420
410 CONTINUE
CSTAR(IJ)=CSTAR(IJ)*EXP(-ZLAMB*DELT)
420 CONTINUE
C
C    CHECK TO SEE IF KD IS VARIABLE.  IF SO, COMPUTE NEW KD.
C
430 IF (ZKD.LT.EPS) GO TO 480
IF (VARYKD.NE.YES) GO TO 450
IF (NTAG.GT.0.AND.NCHK.LT.11) GO TO 440
CALL SKDSR (NCHK,NTAG,NREF)
IF (NTAG.LT.1) GO TO 450
440 NCHK=NCHK+1
450 CONTINUE
C
C    COMPUTE CONCENTRATION CHANGE RESULTING FROM SORPTION AND DECAY
C
DO 470 IJ=1,MAXIN
DO 460 KK=1,KOUNT
IF (IJ.EQ.IJBDRY(KK)) GO TO 470
460 CONTINUE
DELC(IJ)=(ZBETA/(1.+ZBETA*VKD(IJ)))*(VKDP(IJ)*CIJ(IJ)-VKD(IJ)*CSTAR(IJ))
470 CONTINUE
480 CONTINUE
C
C    CHECK TO SEE IF DISPERSION IS PRESENT.  IF SO COMPUTE CONC. CHANGE
C
IF (DISPER.NE.YES) GO TO 860
C
C    IMPLICIT ALGORITHM FOR SOLVING THE DIFFUSION TERMS
C
IF (SOLVEI.NE.YES) GO TO 770
DO 490 K=1,MAXIN
VKDP(K)=CSTAR(K)
490 CONTINUE
500 DO 620 J=1,NTOPY
IF (J.GT.1.AND.VARDIS.NE.YES) GO TO 530
DO 510 I=1,NTOPX
IJ=(J-1)*NTOPX+I
IF (VARDIS.EQ.YES) DIFUSX=VDISX(IJ)
PX=DIFUSX*DELT/(2.0*DELY**2)
PY=DIFUSX*DELT/(2.0*DELX**2)
AA(I)=DIFUSX*DELT/(2.0*DELX**2)
EE(I)=AA(I)
BB(I)=-(2.0*AA(I)+VKD(IJ)+1)
510 CONTINUE
EE(1)=2.0*EE(1)
AA(NTOPX)=2.0*AA(NTOPX)
EP(1)=EE(1)/BB(1)
MAXM=NTOPX-1
DO 520 I=2,MAXM
EP(I)=EE(I)/(BB(I)-AA(I)*EP(I-1))

```

```

520 CONTINUE
530 IJ=(J-1)*NTOPX+1
    IF (VARDIS.EQ.YES) PX=VDISY(IJ)*DELTA/(2.0*DELY**2)
    IF (VARDIS.EQ.YES) PY=VDISX(IJ)*DELTA/(2.0*DELX**2)
    IF (J.NE.1) GO TO 540
    DD(1)=-CSTAR(IJ)-2.0*PX*(CSTAR(IJ+NTOPX)-CSTAR(IJ))
    GO TO 560
540 IF (J.NE.NTOPY) GO TO 550
    DD(1)=-CSTAR(IJ)-2.0*PX*(CSTAR(IJ-NTOPX)-CSTAR(IJ))
    GO TO 560
550 DD(1)=-CSTAR(IJ)-PX*(CSTAR(IJ+NTOPX)-2.0*CSTAR(IJ)+CSTAR(IJ-NTOPX))
1)
560 DP(1)=DD(1)/BB(1)
    DO 600 I=2,NTOPX
    IJ=(J-1)*NTOPX+I
    IF (VARDIS.EQ.YES) PX=VDISY(IJ)*DELTA/(2.0*DELY**2)
    IF (VARDIS.EQ.YES) PY=VDISX(IJ)*DELTA/(2.0*DELX**2)
    IF (J.NE.1) GO TO 570
    DD(I)=-CSTAR(IJ)-2.0*PX*(CSTAR(IJ+NTOPX)-CSTAR(IJ))
    GO TO 590
570 IF (J.NE.NTOPY) GO TO 580
    DD(I)=-CSTAR(IJ)-2.0*PX*(CSTAR(IJ-NTOPX)-CSTAR(IJ))
    GO TO 590
580 DD(I)=-CSTAR(IJ)-PX*(CSTAR(IJ+NTOPX)-2.0*CSTAR(IJ)+CSTAR(IJ-NTOPX))
1)
590 DP(I)=(DD(I)-AA(I)*DP(I-1))/(BB(I)-AA(I)*EP(I-1))
600 CONTINUE
    TEP(IJ)=DP(NTOPX)
    DO 610 I=1,MAXM
    TEP(IJ-I)=DP(NTOPX-I)-EP(NTOPX-I)*TEP(IJ-I+1)
610 CONTINUE
620 CONTINUE
    DO 630 K=1,MAXIN
    CSTAR(K)=TEP(K)
630 CONTINUE
    DO 750 I=1,NTOPX
    IF (I.GT.1.AND.VARDIS.NE.YES) GO TO 660
    DO 640 J=1,NTOPY
    IJ=(J-1)*NTOPX+I
    IF (VARDIS.EQ.YES) DIFUSY=VDISY(IJ)
    PY=DIFUSY*DELTA/(2.0*DELX**2)
    PX=DIFUSY*DELTA/(2.0*DELY**2)
    AA(J)=DIFUSY*DELTA/(2.0*DELY**2)
    EE(J)=AA(J)
    BB(J)=-(2.0*AA(J)+VK0(IJ)+1)
640 CONTINUE
    EE(1)=2.0*EE(1)
    AA(NTOPY)=2.0*AA(NTOPY)
    EP(1)=EE(1)/BB(1)
    MAXM=NTOPY-1
    DO 650 J=2,MAXM
    EP(J)=EE(J)/(BB(J)-AA(J)*EP(J-1))
650 CONTINUE
660 IJ=I
    IF (VARDIS.EQ.YES) PY=VDISX(IJ)*DELTA/(2.0*DELX**2)

```

```

      IF (VARDIS.EQ.YES) PX=VDISY(IJ)*DELTA/(2.0*DELY**2)
      IF (I.NE.1) GO TO 670
      DD(1)=-CSTAR(IJ)-PY*(CSTAR(IJ+1)-CSTAR(IJ))
      GO TO 690
670    IF (I.NE.NTOPX) GO TO 680
      DD(1)=-CSTAR(IJ)
      GO TO 690
680    DD(1)=-CSTAR(IJ)-PY*(CSTAR(IJ+1)-2.0*CSTAR(IJ)+CSTAR(IJ-1))
690    DP(1)=DD(1)/BB(1)
      DO 730 J=2,NTOPY
      IJ=(J-1)*NTOPX+I
      IF (VARDIS.EQ.YES) PY=VDISX(IJ)*DELTA/(2.0*DELX**2)
      IF (VARDIS.EQ.YES) PX=VDISY(IJ)*DELTA/(2.0*DELY**2)
      IF (I.NE.1) GO TO 700
      DD(J)=-CSTAR(IJ)-PY*(CSTAR(IJ+1)-CSTAR(IJ))
      GO TO 720
700    IF (I.NE.NTOPX) GO TO 710
      DD(J)=-CSTAR(IJ)
      GO TO 720
710    DD(J)=-CSTAR(IJ)-PY*(CSTAR(IJ+1)-2.0*CSTAR(IJ)+CSTAR(IJ-1))
720    DP(J)=(DD(J)-AA(J)*DP(J-1))/(BB(J)-AA(J)*EP(J-1))
730    CONTINUE
      TEP(IJ)=DP(NTOPY)
      DO 740 J=1,MAXM
      NMIN=J*NTOPX
      NOTH=(J-1)*NTOPX
      TEP(IJ-NMIN)=DP(NTOPY-J)-EP(NTOPY-J)*TEP(IJ-NOTH)
740    CONTINUE
750    CONTINUE
      DO 760 K=1,MAXIN
      TEP(K)=TEP(K)+DELC(K)
      DELC(K)=TEP(K)-VKDP(K)
      CSTAR(K)=TEP(K)-DELC(K)
760    CONTINUE
      GO TO 860

C
C      EXPLICIT ALGORITHM FOR SOLVING THE DIFFUSION TERMS
C
770    DELCX=0.0
      DELCY=0.0
      DO 850 J=1,NTOPY
      DO 840 I=1,NTOPX
      IJ=(J-1)*NTOPX+I
      IF (VARDIS.EQ.YES) DIFUSX=VDISX(IJ)
      IF (VARDIS.EQ.YES) DIFUSY=VDISY(IJ)
      STAB1=(DELTA/(1.0+ZBETA*VKD(IJ)))*(DIFUSX/DELX**2)
      STAB2=(DELTA/(1.0+ZBETA*VKD(IJ)))*(DIFUSY/DELY**2)
      IF (STAB1.GT.0.3.OR.STAB2.GT.0.3) WRITE (6,1170) STAB1,STAB2
      IF (STAB1.GT.0.7.OR.STAB2.GT.0.7) STOP
      IF (STAB1.GT.0.3.OR.STAB2.GT.0.3) GO TO 500
      IF (DIFUSX.EQ.0.0) GO TO 800
      IF (I.EQ.1) GO TO 780
      IF (I.EQ.NTOPX) GO TO 790
      DELCX=(1./(1.+ZBETA*VKD(IJ)))*(DIFUSX*DELTA/DELX**2)*(CSTAR(IJ+1)-2
1.0*CSTAR(IJ)+CSTAR(IJ-1))

```

```

      GO TO 800
780  DELCX=(1./((1.+ZBETA*VKD(IJ)))*(DIFUSX*DELX**2)*2.0*(CSTAR(IJ+
      11)-CSTAR(IJ))
      GO TO 800
790  DELCY=(1./((1.+ZBETA*VKD(IJ)))*(DIFUSX*DELX**2)*2.0*(CSTAR(IJ-
      11)-CSTAR(IJ))
800  IF (DIFUSY.EQ.0.0) GO TO 830
      IF (J.EQ.1) GO TO 810
      IF (J.EQ.NTOPY) GO TO 820
      DELCY=(1./((1.+ZBETA*VKD(IJ)))*(DIFUSY*DELT/DELY**2)*(CSTAR(IJ+NTOP
      1X)-2.0*CSTAR(IJ)+CSTAR(IJ-NTOPX))
      GO TO 830
810  DELCY=(1./((1.+ZBETA*VKD(IJ)))*(DIFUSY*DELT/DELY**2)*2.0*(CSTAR(IJ+
      1NTOPX)-CSTAR(IJ))
      GO TO 830
820  DELCY=(1./((1.+ZBETA*VKD(IJ)))*(DIFUSY*DELT/DELY**2)*2.0*(CSTAR(IJ-
      1NTOPX)-CSTAR(IJ))
830  IF (ZKU.LT.EPS.AND.VARYKD.NE.YES) DELC(IJ)=0.0
      DELC(IJ)=DELC(IJ)+DELCX+DELCY
840  CONTINUE
850  CONTINUE
C
C      CHECK TO SEE IF AQUIFER THICKNESS IS VARIABLE
C      IF SO, COMPUTE THE CONCENTRATION CHANGE
C
860  IF (VARTHK.NE.YES) GO TO 950
      DELCX=0.0
      DELCY=0.0
      DO 940 J=1,NTOPY
      DO 930 I=1,NTOPX
      IJ=(J-1)*NTOPX+I
      IF (H(IJ).LT.0.0) GO TO 930
      IF (VARDIS.EQ.YES) DIFUSX=VDISX(IJ)
      IF (VARDIS.EQ.YES) DIFUSY=VDISY(IJ)
      FACH=DELT/(H(IJ)*(1+ZBETA*VKD(IJ)))
      DCOEX=DIFUSX/(4.0*DELX**2)
      DCOEY=DIFUSY/(4.0*DELY**2)
      IF (I.EQ.1) GO TO 870
      IF (I.EQ.NTOPX) GO TO 880
      DELCX=FACH*DCOEX*(H(IJ+1)-H(IJ-1))*(CSTAR(IJ+1)-CSTAR(IJ-1))
      GO TO 890
870  DELCX=4.0*FACH*DCOEX*(H(IJ+J)-H(IJ))*(CSTAR(IJ+1)-CSTAR(IJ))
      GO TO 890
880  DELCX=4.0*FACH*DCOEX*(H(IJ)-H(IJ-1))*(CSTAR(IJ)-CSTAR(IJ-1))
890  IF (J.EQ.1) GO TO 900
      IF (J.EQ.NTOPY) GO TO 910
      DELCY=FACH*DCOEY*(H(IJ+NTOPX)-H(IJ-NTOPX))*(CSTAR(IJ+NTOPX)-CSTAR(
      1IJ-NTOPX))
      GO TO 920
900  DELCY=4.0*FACH*DCOEY*(H(IJ+NTOPX)-H(IJ))*(CSTAR(IJ+NTOPX)-CSTAR(IJ
      1))
      GO TO 920
910  DELCY=4.0*FACH*DCOEY*(H(IJ)-H(IJ-NTOPX))*(CSTAR(IJ)-CSTAR(IJ-NTOPX
      1))
920  DELC(IJ)=DELC(IJ)+DELCX+DELCY

```

```

930  CONTINUE
940  CONTINUE
950  CONTINUE
    DO 970 IJ=1,MAXIN
    DO 960 K=1,KOUNT
    IF (IJ.EQ.IJBDRY(K)) GO TO 970
960  CONTINUE
    CIJ(IJ)=CSTAR(IJ)+DELC(IJ)
970  CONTINUE
    DO 990 NP=1,LIMIT
    I=IFIX(XMOVE(NP)/DELX)+1
    J=IFIX(YMOVE(NP)/DELY)+1
    IJ=(J-1)*NTOPX+I
    DO 980 KK=1,KOUNI
    IF (IJ.NE.IJBORY(KK)) GO TO 980
    CPOINT(NP)=CIJ(IJ)
    GO TO 990
980  CONTINUE
    CPOINT(NP)=CPOINT(NP)+DELC(IJ)
    IF (CPOINT(NP).LT.2.0) CPOINT(NP)=0.0
990  CONTINUE
C
    CALL TIMER (TSTRT,TNEW)
C    WRITE SOLUTION FOR CURRENT TIME PLANE TO TAPE AND/OR PRINTER
C
    IF (TAPEOP.NE.YES) GO TO 1010
    NN=N-1
    NMAX=MAXIN+1
    CALL NTRAN (13,1,NMAX,NN,LS)
1000 IF (LS.EQ.-1) GO TO 1000
    IF (LS.LT.0) WRITE (6,1250) LS
    IF (LS.LT.0) STOP
1010 CONTINUE
    IF (KEEPER.EQ.NFREQ) GO TO 1020
    KEEPER=KEEPER+1
    GO TO 1050
1020 NN=N-1
    WRITE (6,1180) NN,TIMTOT
    DO 1040 MJ=1,NTOPY
    NLINE=NTOPY-MJ
    LINUM=NLINE+1
    DO 1030 MI=1,NTOPX
    STORE(MI)=CIJ(NLINE+NTOPX+MI)
1030 CONTINUE
    WRITE (6,1190) LINUM
    WRITE (6,1200) (STORE(IJ),IJ=1,NTOPX)
1040 CONTINUE
    KEEPER=1
1050 CONTINUE
    LIMITP=LIMIT
C
C    CHECK TO SEE IF NEW BOUNDARY CONCENTRATIONS ARE TO BE READ
C
    IF (N.NE.NEXT) GO TO 1060
    CALL INPUT (NEXT)

```



```

1060 CONTINUE
    END FILE 13
    STOP

```

```

C
C   FORMAT STATEMENTS
C

```

```

1070 FORMAT (13A6)
1080 FORMAT (/ ,16(0*0,4X),/)
1090 FORMAT (20X,0 PROBLEM DESCRIPTION0,//0 DOES THE PROBLEM CONSIDER D
1100 1 DISPERSION?.....0,A6/0 ARE THE FINAL RESULTS TO BE OUTP
2 UT TO TAPE?.....0,A6/0 IS THE NON-ESSENTIAL PRINTING TO BE S
3 UPRESSED?.....0,A6/0 DOES THE PROBLEM REQUIRE A VARIABLE KD?...
4 .....0,A6/0 IS THE FLOW FIELD STEADY?.....
5 .....0,A6/0 IS THE DISPERSION COEFFICIENT VARIABLE?.....
6 .....0,A6/0 DOES THE PROBLEM CONSIDER VARYING AQUIFER THICKNESS?...0,A
7 6/0 IS AN IMPLICIT ALGORITHM USED?.....0,A6)
1100 FORMAT (15X,0 PROBLEM SPECIFICATION0,//0 NUMBER OF TIME PLANES....
1 .....0,I5,/0 PRINT FREQUENCY0,19(0.0),I5,/0 NUMBER OF CELLS IN
2 X-DIRECTION.....0,I5,/0 NUMBER OF CELLS IN Y-DIRECTION.....0,I5,/0
3 NUMBER OF BOUNDARY CELLS.....0,I5,/0 CELL DIMENSION IN X-DIRE
4 CTION.....0,E10.4,/0 CELL DIMENSION IN Y-DIRECTION.....0,E10.4)
1110 FORMAT (18X,0 INPUT PARAMETERS 0,//0 DISPERSION COEFFICIENT, X-DIR
1 ECTION.....0,E10.4,/0 DISPERSION COEFFICIENT, Y-DIRECTION.....0,E10.
2 4,/0 APPROXIMATE SORPTION CONSTANT.....0,E10.4/0 BULK DENSITY
3 .....0,E10.4,/0 DECAY CONSTANT.....
4 .....0,E10.4)
1120 FORMAT (1H1,0 BOUNDARY CELL NUMBERS0)
1130 FORMAT (1X,20I5)
1140 FORMAT (/ ,0 TIME PLANE = 0,I5,4X,0 NUMBER OF POINTS = 0,I4,4X,0 TI
1 ME STEP SIZE =0,E10.4)
1150 FORMAT (3(16,3X,E10.4,3X,E10.4,6X))
1160 FORMAT (1H1,5X,0...FINAL SOLUTION...0,/)
1170 FORMAT (/ ,0 STABILITY PARAMETERS0,2F12.5)
1180 FORMAT (/ ,9X,0 TIME STEP0,I4,10X,0 REAL TIME0,E12.4)
1190 FORMAT (/ ,0 J = 0,I4)
1200 FORMAT (1X,1P10E12.4)
1210 FORMAT (1X,0 L1 = 0,I4)
1220 FORMAT (1X,0 L2 = 0,I4)
1230 FORMAT (1X,0 L3 = 0,I4)
1240 FORMAT (1X,0 L4 = 0,I4)
1250 FORMAT (1X,0 L5 = 0,I4)
1260 FORMAT (/ ,0 TIME PLANE READ IS MISMATCHED 0,2I5)
1270 FORMAT (1H1)
    END

```

SUBROUTINE AVRGE

C
C
C
C

THIS SUBROUTINE ASSIGNS EACH CELL A CONCENTRATION WHICH IS THE AVERAGE CONCENTRATION OF ALL POINTS WITHIN THE CELL.

```

PARAMETER NPTS=1500, NCEL=1100, NBOY=250
COMMON/MIC/NPL,LIMIT,DELT,DELX,DELY,KOUNT,NTOPX,NTOPY,SUPRES,MAXIN
COMMON/LAR/XMOVE(NPTS),YMOVE(NPTS),NN,CIJ(NCEL),CSTAR(NCEL),CPOINT
1(NPTS),IJBORY(NBOY),DELC(NCEL),STORE(100),VKD(NCEL)
DIMENSION ICS(NCEL)
DO 100 IJ=1,MAXIN
  ICS(IJ)=0
100 CONTINUE
  DO 110 NP=1,LIMIT
    I=IFIX(XMOVE(NP)/DELX)+1
    J=IFIX(YMOVE(NP)/DELY)+1
    IJ=(J-1)*NTOPX+I
    ICS(IJ)=ICS(IJ)+1
    IF (ICS(IJ).LE.1) CSTAR(IJ)=CPOINT(NP)
    IF (ICS(IJ).GT.1) CSTAR(IJ)=CSTAR(IJ)+CPOINT(NP)
110 CONTINUE
    DO 120 IJ=1,MAXIN
      IF (ICS(IJ).LT.1) GO TO 120
      DENOM=FLOAT(ICS(IJ))
      CSTAR(IJ)=CSTAR(IJ)/DENOM
120 CONTINUE
  RETURN
END

```

```

C      SUBROUTINE INPUT (NEXT)
C
C      THIS SUBROUTINE SETS UP THE INITIAL MICRO ION CONCENTRATION
C      DISTRIBUTION, AND IF DESIRED, RE-SETS THESE CONCENTRATIONS AT SOME
C      GIVEN TIME DURING THE EXECUTION OF THE MICRO ION PROGRAM
C
C      PARAMETER NPTS=1500, NCEL=1100, NBDY=250
C      COMMON/MIC/NPL,LIMIT,DELT,DELY,DELY,KOUNT,NTOPX,NTOPY,SUPRES,MAXIN
C      COMMON/LAR/XMOVE(NPTS),YMOVE(NPTS),NN,CIJ(NCEL),CSTAR(NCEL),CPOINT
C      1(NPTS),IJBDRY(NBDY),DELC(NCEL),STORE(100),VKO(NCEL)
C      REAL LAST
C      DATA LAST,YES,EPS,/6H LAST,6H YES,0.00001/
C      NCOUNT=0
C
C      READ THE INITIAL OR THE NEW MICRO ION DISTRIBUTION OF THE CELLS
C      FROM CARDS
C
C      NOMENCLATURE:
C      WORD.....@LAST@ IS A SENTINAL DESIGNATING THE END OF A SET OF
C      INPUT DATA
C      CO.....INITIAL OR NEW CONCENTRATION
C      I1,J1.....I,J COORDINATES OF THE FIRST CELL IN A ROW OR COLUMN
C      OF CELLS
C      I2,J2.....I,J COORDINATES OF THE LAST CELL IN A ROW OR COLUMN
C      OF CELLS
C
100  READ (5,200) WORD,CO,I1,J1,I2,J2,NEXT
      IF (WORD.EQ.LAST) GO TO 150
      IF (I1.EQ.0.AND.J1.EQ.0) GO TO 150
      IF ((I2.GT.0.AND.I1.GT.I2).OR.(J2.GT.0.AND.J1.GT.J2)) GO TO 140
      IF (I2.NE.0) GO TO 110
      I2=I1
110  IF (J2.NE.0) GO TO 120
      J2=J1
120  DO 130 J=J1,J2
      DO 130 I=I1,I2
      IJ=(J-1)*NTOPX+I
      CIJ(IJ)=CO
      CSTAR(IJ)=CO
130  CONTINUE
      NCOUNT=NCOUNT+(I2-I1+1)*(J2-J1+1)
      IF (NCOUNT.LE.MAXIN) GO TO 100
      WRITE (6,210)
      STOP
140  WRITE (6,220)
      STOP
C
C      ASSIGN THE POINTS IN EACH CELL THE SAME CONCENTRATION AS THE CELL
C
150  NCOUNT=0
      DO 160 NPOINT=1,LIMIT
      IPX=IFIX(XMOVE(NPOINT)/DELY+EPS)+1
      JPY=IFIX(YMOVE(NPOINT)/DELY+EPS)+1
      IJXY=(JPY-1)*NTOPX+IPX

```

```

      CPOINT(NPOINT)=CIJ(IJXY)
160  CONTINUE
      IF (SUPRES.EQ.YES) GO TO 190
      WRITE (6,230)
      DO 180 MJ=1,NTOPY
      NLINE=NTOPY-MJ
      LINUM=NLINE+1
      DO 170 MI=1,NTOPX
      STORE(MI)=CIJ(NLINE*NTOPX+MI)
170  CONTINUE
      WRITE (6,240) LINUM
      WRITE (6,250) (STORE(IJ),IJ=1,NTOPX)
180  CONTINUE
190  CONTINUE
      RETURN
C
200  FORMAT (A6,E9.4,S15)
210  FORMAT (/ ,5X,0,...SPECIFIED INPUT EXCEEDS MAXIMUM...0,/)
220  FORMAT (/ ,5X,0,...SPECIFIED INPUT IS INDEXED INCORRECTLY...0,/)
230  FORMAT (1H1,/,/,5X,0,...INITIAL CONCENTRATION FIELD...0,/)
240  FORMAT (/,/,0 J=0,15,/)
250  FORMAT (1P10E12.4)
      END

```

```

SUBROUTINE TIMER (TSTRT,TNEW)
C
C PRINT THE TOTAL ACCUMULATED TIME AND THE TIME SINCE THIS
C ROUTINE WAS LAST CALLED.
C TNEW -(AT ENTRY) TIME ROUTINE WAS LAST CALLED
C TSTRT=INITIAL TIME PROGRAM STARTED
C
CALL ETIMEF (SEC)
TOLD=TNEW
TNEW=SEC
SEC1=(TNEW-TOLD)
SEC2=(TNEW-TSTRT)
WRITE (6,100) SEC1,SEC2
RETURN
C
C
100 FORMAT (/,'X',* TIME USED:*,F10.3,* SECONDS.*,*
11LATED TIME:*,F10.3,* SECONDS.*,/)
END
ACCUMU

```

```

SUBROUTINE SKDSP (NCHK,NTAG,NREF)
C
C THIS SUBROUTINE CALCULATES THE SATURATED SORPTION COEFFICIENTS FOR
C STRONTIUM AS A FUNCTION OF MACRO ION CONCENTRATION
C
PARAMETER NPTS=1500, NCEL=1100, NBDY=250
COMMON/MIC/NPL,LIMIT,DELT,DELY,DELY,KOUNT,NTOPX,NTOPY,SUPRES,MAXIN
COMMON/LAR/XMOVE(NPTS),YMOVE(NPTS),NN,CIJ(NCEL),CSTAR(NCEL),CPOINT
1(NPTS),IJBDY(NBDY),DELC(NCEL),STORE(100),VKD(NCEL),VKDP(NCEL)
DIMENSION CMAC(8)
DATA YES/6H YES/
NREF=NREF+1
DO 100 I=1,MAXIN
VKDP(I)=VKD(I)
100 CONTINUE
IF (NCHK.GT.1) NCHK=NCHK-1
DO 170 NTIM=1,NCHK
CALL NTRAN (7,2,1,NION,L)
110 IF (L.EQ.-1) GO TO 110
DO 160 J=1,MAXIN
DO 120 K=1,KOUNT
IF (J.EQ.IJBDY(K)) GO TO 160
120 CONTINUE
CALL NTRAN (7,2,NION,CMAC,L)
130 IF (L.EQ.-1) GO TO 130
IF (NTAG.GT.0.AND.NTIM.LT.NCHK) GO TO 160
IF (CMAC(4).LT.0.005) GO TO 140
A=CMAC(2)-0.11
B=40.0*CMAC(3)-1.0
C=CMAC(4)-0.0275
VKD(J)=1.9337-60.72*C-0.5287*B+17.831*B*C-3.263*A+126.52*A*C+5.43*
1A*B-222.864*A*B*C
GO TO 150
140 A=CMAC(2)-0.11
B=40.0*CMAC(3)-1.0
C=CMAC(4)-0.0035
VKD(J)=4.325-683.3*C-1.67*B+493.3*C*B-11.0*A+3259.26*A*C+11.94*A*B
1-1000.0*A*B*C
150 CONTINUE
IF (VKD(J).LT.0.0) VKD(J)=0.0
IF (SUPRES.EQ.YES) GO TO 160
WRITE (6,210) NN,NREF,(CMAC(IJ),IJ=1,NION),VKD(J)
160 CONTINUE
170 CONTINUE
DO 180 I=1,MAXIN
COMP=ABS(VKD(I)-VKDP(I))
IF (COMP.GT.0.005) GO TO 190
180 CONTINUE
NTAG=1
NCHK=1
GO TO 200
190 CONTINUE
NTAG=0
NCHK=1

```

```
200  CONTINUE  
      RETURN  
C  
210  FORMAT (1X,0FROM KDSR0,2I6,8E10.3,5X,F10.5)  
      END
```

```

C      SUBROUTINE SKDCS (NCHK,NTAG,NREF)
C
C      THIS SUBROUTINE CALCULATES THE SATURATED SORPTION COEFFICIENTS FOR
C      CESIUM AS A FUNCTION OF MACRO ION CONCENTRATION
C
      PARAMETER NPTS=1500, NCEL=1100, NBDY=250
      COMMON/MIC/NPL,LIMIT,DELT,DELX,DELY,KOUNT,NTOPX,NTOPY,SUPRES,MAXIN
      COMMON/LAR/XMOVE(NPTS),YMOVE(NPTS),NN,CIJ(NCEL),CSTAR(NCEL),CPOINT
1(NPTS),IJBDRY(NBDY),DELC(NCEL),STORE(100),VKD(NCEL),VKDP(NCEL)
      DIMENSION CMAC(8)
      DATA YES/6H YES/
      NREF=NREF+1
      DO 100 I=1,MAXIN
      VKDP(I)=VKD(I)
100    CONTINUE
      IF (NCHK.GT.1) NCHK=NCHK-1
      DO 180 NTIM=1,NCHK
      CALL NTRAN (7,2,1,NION,L)
110    IF (L.EQ.-1) GO TO 110
      DO 170 J=1,MAXIN
      DO 120 K=1,KOUNT
      IF (J.EQ.IJBDRY(K)) GO TO 170
120    CONTINUE
      CALL NTRAN (7,2,NION,CMAC,L)
130    IF (L.EQ.-1) GO TO 130
      IF (NTAG.GT.0.AND.NTIM.LT.NCHK) GO TO 170
      IF (CMAC(3).LT.0.01) GO TO 140
      IF (CMAC(3).GT.0.154) GO TO 150
      B=CMAC(3)-.082
      A=CMAC(2)-.125
      C=CMAC(4)-.080
      VKD(J)=44.7125-112.7*C-458.85*B+1460.81*B*C-27.83*A+497.6*A*C+307.
19*B*A-6977.5*A*B*C
      GO TO 160
140    B=CMAC(3)-.0055
      VKD(J)=174.125-458.93*C-21416.7*B+53571.4*C*B-331.67*A+1500.0*A*C+
162592.6*A*B-111111.0*A*B*C
      GO TO 160
150    VKD(J)=9.6437-312.49*(CMAC(3)-.1605)
160    CONTINUE
      IF (VKD(J).LT.0.0) VKD(J)=0.0
      IF (SUPRES.EQ.YES) GO TO 170
      WRITE (6,220) NN,NREF,(CMAC(IJ),IJ=1,NION),VKD(J)
170    CONTINUE
180    CONTINUE
      DO 190 I=1,MAXIN
      COMP=ABS(VKD(I)-VKDP(I))
      IF (COMP.GT.0.005) GO TO 200
190    CONTINUE
      NTAG=1
      NCHK=1
      GO TO 210
200    CONTINUE
      NTAG=0

```



```
      NCHK=1
210  CONTINUE
      RETURN
C
220  FORMAT (1X,0FROM KDCS0,216,8E10.3,5X,F10.5)
      END
```

MACRO: A PROGRAM FOR THE NUMERICAL SOLUTION OF THE TIME DEPENDENT
TWO DIMENSIONAL, MASS TRANSPORT EQUATION WITH REACTION
TERMS PRESENT. THE SOLUTION ALGORITHM IS BASED ON THE
METHOD OF CHARACTERISTICS AND IS APPLICABLE TO CASES OF
LOW OR ZERO DISPERSION. THE MASS TRANSPORT PARTIAL
DIFFERENTIAL EQUATION IS CAST AS:

$$P(C)/PT = DX*P(P(C)/PX)/PX + DY*P(P(C)/PY)/PY \\ -U*P(C)/PX - V*P(C)/PY + R$$

WHERE:

P()/PT.....PARTIAL DERIVATIVE W.R.T. TIME
P()/PX, P()/PY..PARTIAL DERIVATIVE W.R.T. SPACE COORDS
C(X,Y,T).....LIQUID PHASE CONCENTRATION
DX, DY.....DISPERSION COEFFICIENTS
U(X,Y,T).....FLOW VELOCITY IN X-DIRECTION
V(X,Y,T).....FLOW VELOCITY IN Y-DIRECTION
R.....REACTION EXPRESSION

WRITTEN BY 1) R.G. BACA
2) A. BRANDSTETTER
3) R.J. SERNE
4) S.W. AHLSTROM

PROGRAMMED BY 1) R.G. BACA
2) D. FRIEDRICHS
3) R.J. SERNE
4) S.W. AHLSTROM

VERSION 1 - MAY 30, 1971
VERSION 2 - JAN 31, 1972

BATTELLE-NORTHWEST: WATER RESOURCES SYSTEMS SECTION

REFERENCES: 1) A.O. GARDER, ET. AL.: NUMERICAL CALCULATION
OF MULTIDIMENSIONAL MISCIBLE DISPLACEMENT
BY THE METHOD OF CHARACTERISTICS,
SOC. PET. ENGRS. JOUR. (MARCH 1964) P.26
2) G.F. PINDER, ET. AL.: A NUMERICAL TECHNIQUE
FOR CALCULATING THE TRANSIENT POSITION OF
THE SALTWATER FRONT,
WATER RESOURCES RESEARCH, VOL.6, NO.3
(JUNE 1970) P.875

PARAMETER NPTS=1500, NCEL=1100, NBDY=200, NION=8, NIONS=6
COMMON/MIC/NPL,LIMIT,DELT,DELX,DELY,KOUNT,NTOPX,NTOPY,SUPRES,MAXIN
COMMON /LARGE/ XMOVE(NPTS),YMOVE(NPTS),CIJ(NCEL,NION),CPOINT(NPTS,
1(NION),CSTAR(NCEL,NION),CSOIL(NCEL,NIONS),DELC(NCEL,NION),IJBORY(NB
2DY),STORE(100)
DIMENSION TITLE(13),Z(NION),ZT(NION),NFLAG(NBDY),VDISX(NCEL),VDISY
1(NCEL),TIJ(NION)
NAMelist/SMACS/NFREQ,NTIMES,NTOPX,NTOPY,KOUNT,DELX,DELY
NAMelist/PMACS/ DIFUSX,DIFUSY,DBULK,C,AN,ASR
REAL LAST


```

100  IF (L1.EQ.-1) GO TO 100
      IF (L1.LT.0) WRITE (6,740) L1
      IF (L1.LT.0) STOP
      IF (SUPRES.EQ.YES) GO TO 101
      WRITE (6,650)
      WRITE (6,660) (IJBDY(I),I=1,KOUNT)
C
C      READ THE DISPERSION COEFFICIENT ARRAYS IF APPLICABLE.
C
101  IF (VARDIS.NE.YES) GO TO 110
      CALL NTRAN(3,2,MAXIN,VDISX,L1)
102  IF (L1.EQ.-1) GO TO 102
      IF (L1.LT.0) WRITE(6,740) L1
      IF (L1.LT.0) STOP
      IF (SUPRES.EQ.YES) GO TO 103
      WRITE (6,730) (VDISX(I),I=1,MAXIN)
103  CALL NTRAN(3,2,MAXIN,VDISY,L1)
104  IF (L1.EQ.-1) GO TO 104
      IF (L1.LT.0) WRITE(6,740) L1
      IF (L1.LT.0) STOP
      IF (SUPRES.EQ.YES) GO TO 110
      WRITE (6,730) (VDISY(I),I=1,MAXIN)
C
C      READ THE INITIAL TIME PLANE DATA FROM TRACK
C
110  CALL NTRAN(3,2,3,NPL,L2)
120  IF (L2.EQ.-1) GO TO 120
      IF (L2.LT.0) WRITE (6,750) L2
      IF (L2.LT.0) STOP
      CALL NTRAN (3,2,LIMIT,XMOVE,L3)
130  IF (L3.EQ.-1) GO TO 130
      IF (L3.LT.0) WRITE (6,760) L3
      IF (L3.LT.0) STOP
      CALL NTRAN (3,2,LIMIT,YMOVE,L4)
140  IF (L4.EQ.-1) GO TO 140
      IF (L4.LT.0) WRITE (6,770) L4
      IF (L4.LT.0) STOP
      IF (NPL.NE.0) WRITE (6,780) NPL,N
      IF (NPL.NE.0) STOP
      IF (SUPRES.EQ.YES) GO TO 150
      WRITE (6,670) NPL,LIMIT,DELT
      WRITE (6,680) (I,XMOVE(I),YMOVE(I),I=1,LIMIT)
150  MAXIN=NTOPX*NTOPY
      B=2.65*DBULK/(2.65-DBULK)
      B=B*914.0
      DO 160 K=1,NION
      DO 160 IJ=1,MAXIN
      CIJ(IJ,K)=0.0
      CSTAR(IJ,K)=0.0
      DELC(IJ,K)=0.0
160  CONTINUE
      DO 170 NP=1,LIMIT
      CPOINT(NP,K)=0.0
170  CONTINUE
180  CONTINUE

```

```

DO 200 K=,NIONS
DO 190 IJ=1,MAXIN
CSOIL(IJ,K)=0.0
190 CONTINUE
200 CONTINUE
C
C   READ INITIAL AND BOUNDARY CONCENTRATIONS FROM CARDS (5+)
C
CALL INPUT(NEXT)
WRITE (6,690)

C
C   ENTER TIME PLANE LOOP
C
LIMITP=LIMIT
TIMTOT=0.0
KEEPER=1
DO 590 N=2,NTIMES

C
C   READ THE CURRENT TIME PLANE
C
LIMSFT=0
CALL NTRAN (3,2,3,NPL,L2)
210 IF (L2.EQ.-1) GO TO 210
IF (L2.LT.0) WRITE (6,750) L2
IF (L2.LT.0) STOP
TIMTOT=TIMTOT+DELT
CALL NTRAN (3,2,LIMIT,XMOVE,L3)
220 IF (L3.EQ.-1) GO TO 220
IF (L3.LT.0) WRITE (6,760) L3
IF (L3.LT.0) STOP
CALL NTRAN (3,2,LIMIT,YMOVE,L4)
230 IF (L4.EQ.-1) GO TO 230
IF (L4.LT.0) WRITE (6,770) L4
IF (L4.LT.0) STOP
IF (NPL.NE.(N-1)) WRITE (6,780) NPL,N
IF (NPL.NE.(N-1)) STOP
NMAX=MAXIN+1
NDIF=LIMITP-LIMIT
IF(NDIF.LE.0) GO TO 275
CALL NTRAN(3,2,1,LIMSFT,L2)
262 IF(L2.EQ.-1) GO TO 262
IF(L2.LT.0) WRITE(6,750) L2
IF(L2.LT.0) STOP
IF(LIMSFT.EQ.0) GO TO 275
CALL NTRAN(3,2,LIMSFT,NFLAG,L2)
264 IF(L2.EQ.-1) GO TO 264
IF(L2.LT.0) WRITE(6,750) L2
IF(L2.LT.0) STOP
IF (LIMSFT.GE.NDIF) GO TO 240
LIMITP=LIMITP-(NDIF-LIMSFT)
240 DO 270 K=1,LIMSFT
JK=NFLAG(K)
LIMITP=LIMITP-1
DO 260 MN=1,NION
DO 250 JL=JK,LIMITP

```

```

      CPOINT(JL,MN)=CPOINT(JL+1,MN)
250  CONTINUE
260  CONTINUE
270  CONTINUE
275  IF(VAROIS.NE.YES.OR.STEADY.EQ.YES) GO TO 280
      CALL NTRAN(3,2,MAXIN,VDISX,L1)
276  IF (L1.EQ.-1) GO TO 276
      IF(L1.LT.0) WRITE(6,740) L1
      IF(L1.LT.0) STOP
      IF (SUPRES.EQ.YES)GO TO 277
      WRITE (6,730) (VDISX(I),I=1,MAXIN)
277  CALL NTRAN(3,2,MAXIN,VDISY,L1)
278  IF (L1.EQ.-1) GO TO 278
      IF(L1.LT.0) WRITE(6,740) L1
      IF(L1.LT.0) STOP
      IF (SUPRES.EQ.YES)GO TO 300
      WRITE (6,730) (VDISY(I),I=1,MAXIN)
280  IF (SUPRES.EQ.YES) GO TO 300
      WRITE (6,670) NPL,LIMIT,DELT
      IF (LIMSFT.LE.0) GO TO 290
      WRITE (6,660) (NFLAG(J),J=1,LIMSFT)
290  WRITE (6,680) (I,XMOVE(I),YMOVE(I),I=1,LIMIT)
300  CONTINUE
      DO 310 KN=1,NION
C
C      FIND THE AVERAGE CONC. IN EACH CELL AFTER POINT MOVEMENT
C
      CALL AVRGE (KN)
      DO 305 J=1,NTOPY
      JZ=(J-1)*NTOPX+1
      CSTAR(JZ,KN)=CSTAR(JZ+2,KN)
305  CONTINUE
310  CONTINUE
C
C      EQUILIBRATE THE CELLS AND COMPUTE THE CONCENTRATIONS AT THE END
C      OF THE CURRENT TIME STEP
C
      NOMENCLATURE:
C      CIJ(IJ,1).....MG++
C      CIJ(IJ,2).....NA+
C      CIJ(IJ,3).....K+
C      CIJ(IJ,4).....CA++
C      CIJ(IJ,5).....PCO3-
C      CIJ(IJ,6).....SO4--
C      CIJ(IJ,7).....CASO4(AQ)
C      CIJ(IJ,8).....PH
C      C.....CL-
C      AN.....NO3-
C      ASR.....SR++
C      CSOIL(IJ,1).....CAX
C      CSOIL(IJ,2).....NAX
C      CSOIL(IJ,3).....MGX
C      CSOIL(IJ,4).....KX
C      CSOIL(IJ,5).....CASO4
C      CSOIL(IJ,6).....CACO3

```

```

C
DO 350 IJ=1,MAXIN
DO 320 KK=1,KOUNT
IF (IJ.EQ.IJHDY(KK)) GO TO 350
320 CONTINUE
DO 330 KN=1,NION
Z(KN)=CSTAR(IJ,KN)
ZT(KN)=CSTAR(IJ,KN)
330 CONTINUE
CALL EQUILB (Z(1),Z(2),Z(3),Z(4),Z(5),Z(6),Z(7),Z(8),C,AN,ASR,CSOI
1L(IJ,1),CSOIL(IJ,2),CSOIL(IJ,3),CSOIL(IJ,4),CSOIL(IJ,5),CSOIL(IJ,6
2),R,N)
DO 340 KN=1,NION
DELC(IJ,KN)=Z(KN)-ZT(KN)
340 CONTINUE
350 CONTINUE
C
C CHECK TO SEE IF DISPERSION IS PRESENT. IF SO COMPUTE CONC. CHANGE
C
IF (DISPER.NE.YES) GO TO 450
STABLE=DELT*(DIFUSX/DELX**2+DIFUSY/DELY**2)
IF (DISPER.EQ.YES.AND.STABLE.GT.0.5) WRITE (6,700) STABLE
IF (DISPER.EQ.YES.AND.STABLE.GT.0.5) STOP
DELCX=0.0
DELCY=0.0
DO 440 KN=1,NION
DO 430 J=1,NTOPY
DO 420 I=1,NTOPX
IJ=(J-1)*NTOPX+I
IF (VARDIS.EQ.YES) DIFUSX=VDISX(IJ)
IF (VARDIS.EQ.YES) DIFUSY=VDISY(IJ)
IF (DIFUSX.EQ.0.0) GO TO 380
IF (I.EQ.1) GO TO 360
IF (I.EQ.NTOPX) GO TO 370
DELCX=(DIFUSX*DELT/DELX**2)*(CSTAR(IJ+1,KN)-2.*CSTAR(IJ,KN)+CSTAR(
1IJ-1,KN))
GO TO 380
360 DELCX=(DIFUSX*DELT/DELX**2)*(CSTAR(IJ+1,KN)-CSTAR(IJ,KN))*2.0
GO TO 380
370 DELCX=(DIFUSX*DELT/DELX**2)*(CSTAR(IJ-1,KN)-CSTAR(IJ,KN))*2.0
380 IF (DIFUSY.EQ.0.0) GO TO 410
IF (J.EQ.1) GO TO 390
IF (J.EQ.NTOPX) GO TO 400
DELCY=(DIFUSY*DELT/DELY**2)*(CSTAR(IJ+NTOPX,KN)-2.*CSTAR(IJ,KN)+CS
1TAR(IJ-NTOPX,KN))
GO TO 410
390 DELCY=(DIFUSY*DELT/DELY**2)*(CSTAR(IJ+NTOPX,KN)-CSTAR(IJ,KN))*2.0
GO TO 410
400 DELCY=(DIFUSY*DELT/DELY**2)*(CSTAR(IJ-NTOPX,KN)-CSTAR(IJ,KN))*2.0
410 DELC(IJ,KN)=DELC(IJ,KN)+DELCX+DELCY
420 CONTINUE
430 CONTINUE
440 CONTINUE
450 CONTINUE
DO 460 IJ=1,MAXIN

```

```

DO 460 KN=1,NION
CIJ(IJ,KN)=CSTAR(IJ,KN)+DELC(IJ,KN)
460 CONTINUE
DO 500 NP=1,LIMIT
I=IFIX(XMOVE(NP)/DELX)+1
J=IFIX(YMOVE(NP)/DELY)+1
IJ=(J-1)*NTOPX+I
DO 480 KK=1,KOUNT
IF (IJ.NE.IJBDRY(KK)) GO TO 480
DO 470 KN=1,NION
CPOINT(NP,KN)=CIJ(IJ,KN)
470 CONTINUE
GO TO 500
480 CONTINUE
DO 490 KN=1,NION
CPOINT(NP,KN)=CPOINT(NP,KN)+DELC(IJ,KN)
IF (CPOINT(NP,KN).LT.0.0) CPOINT(NP,KN)=0.0
490 CONTINUE
500 CONTINUE
C
C   WRITE SOLUTION FOR CURRENT TIME PLANE TO TAPE AND/OR PRINTER
C
IF (TAPEOP.NE.YES) GO TO 530
NN=N-1
NZ=NION
CALL NTRAN(7,1,1,NZ,L1)
501 IF(L1.EQ.-1) GO TO 501
IF(L1.LT.0) WRITE(6,740) L1
IF(L1.LT.0) STOP
DO 520 IJ=1,MAXIN
DO 510 KK=1,KOUNT
IF (IJ.EQ.IJBDRY(KK)) GO TO 520
510 CONTINUE
DO 515 KN=1,NION
TIJ(KN)=CIJ(IJ,KN)
515 CONTINUE
CALL NTRAN(7,1,NION,TIJ,L1)
516 IF(L1.EQ.-1) GO TO 516
IF(L1.LT.0) WRITE(6,740) L1
IF(L1.LT.0) STOP
520 CONTINUE
530 IF (KEEPER.EQ.NFREQ) GO TO 540
KEEPER=KEEPER+1
GO TO 580
540 NN=N-1
IF(SUPRES.EQ.YES) GO TO 580
DO 570 KN=1,NION
WRITE (6,710) NN,TIMTOT,KN
DO 560 MJ=1,NTOPY
NLINE=NTOPY-MJ
LINUM=NLINE+1
DO 550 MI=1,NTOPX
STORE(MI)=CIJ(NLINE*NTOPX+MI,KN)
550 CONTINUE
WRITE (6,720) LINUM

```



```

WRITE (6,730) (STORE(IJ),IJ=1,NTOPX)
560 CONTINUE
570 CONTINUE
KEEPER=1
590 CONTINUE
LIMITP=LIMIT

C
C CHECK TO SEE IF NEW BOUNDARY CONCENTRATIONS ARE TO BE READ
C
IF (N.NE.NEXT) GO TO 590
CALL INPUT (NEXT)
590 CONTINUE
REWIND 3
END FILE 7
REWIND 7
CALL CHAIN(3)
STOP

C
C FORMAT STATEMENTS
C
C
600 FORMAT (13A6)
610 FORMAT (/ ,16(0*0,4X),/)
620 FORMAT (20X,0 PROBLEM DESCRIPTION0,/,0 DOES THE PROBLEM CONSIDER D
1 ISPERSION?0,12(0.0),A6,/,0 ARE THE FINAL RESULTS TO BE OUTPUT TO TA
2 PE?.....0,A6,/,0 IS THE NON-ESSENTIAL PRINTING TO BE SUPRESSED?...0
3 ,A6,/,0 IS THE FLOW FIELD STEADY?0,24(0.0),A6,/,0 ARE THE DISPERSION
4 COEFFICIENTS VARIABLE?.....0,A6)
630 FORMAT (15X,0 PROBLEM SPECIFICATION0,/,0 NUMBER OF TIME PLANES....
1 .....0,15,/,0 PRINT FREQUENCY0,19(0.0),15,/,0 NUMBER OF CELLS IN
2 X-DIRECTION.....0,15,/,0 NUMBER OF CELLS IN Y-DIRECTION.....0,15,/,0
3 NUMBER OF BOUNDARY CELLS.....0,15,/,0 CELL DIMENSION IN X-DIRE
4 CTION.....0,E10.4,/,0 CELL DIMENSION IN Y-DIRECTION.....0,E10.4)
640 FORMAT (18X,0 INPUT PARAMETERS 0,/,0 DISPERSION COEFFICIENT, X-DIR
1 ECTION.....0,E10.4,/,0 DISPERSION COEFFICIENT, Y-DIRECTION.....0,E10.
2 4,/,0 BULK DENSITY0,27(0.0),E10.4,/,0 CHLORIDE ION CONCENTRATION.....
3 .....0,E10.4,/,0 NITRATE ION CONCENTRATION.....0,E10.4,/,0
4 STRONTIUM ION CONCENTRATION.....0,E10.4)
650 FORMAT (1H1,0 BOUNDARY CELL NUMBERS0)
660 FORMAT (1X,20I5)
670 FORMAT (/ ,0 TIME PLANE = 0,15,4X,0 NUMBER OF POINTS = 0,14,4X,0 TI
1 ME STEP SIZE = 0,E10.4)
680 FORMAT (1X,3(15,3X,E10.4,3X,E10.4,6X))
690 FORMAT (1H1,5X,0...FINAL SOLUTION...0,/)
700 FORMAT (/ ,5X,0 STABILITY PARAMETER = 0,1PE11.3,/)
710 FORMAT (/,9X,0 TIME PLANE0,14,15X,0 REAL TIME = 0,E10.3,10X,0 ION
1 NUMBER0,13)
720 FORMAT (/ ,0 J = 0,14)
730 FORMAT (1X,1P10E12.4)
740 FORMAT (1X,0 L1 = 0,14)
750 FORMAT (1X,0 L2 = 0,14)
760 FORMAT (1X,0 L3 = 0,14)
770 FORMAT (1X,0 L4 = 0,14)
780 FORMAT (/ ,0 TIME PLANE READ IS MISMATCHED 0,2I5)
790 FORMAT(1H1)

```

SUBROUTINE EQUILB (F,S,AP,A,HCO3,G,CASO,APH,H,AN,ASR,ET,SAT,CT,RP,
1XXT,CAL,B,N)

```

C
C   THIS SUBROUTINE EQUILBRATES THE MACRO ION SYSTEM EACH TIME STEP
C
100  A11=A
C      IS GYPSUM PRESENT
      IF (XXT) 110,110,120
110  U=SQRT(2.0*(A+F+G+ASR)+0.5*(S+H+HCO3+AP+AN))
      AA=EXP(-9.366*(U/(U+1.)-.2*U*U))
C      IS THE SOLUBILITY PRODUCT EXCEEDED
      IF (2.4E-5-A*G*AA) 120,180,180
C      SOLVE ALL EQUILIBRIA FOR GYPSUM
120  X=0.0
      U=SQRT(2.0*(A+F+G+ASR)+0.5*(S+H+HCO3+AP+AN))
      BB=A+G
      EX=(9.366*(U/(1.+U)-.2*U*U))
      CC=A*G-(2.4E-5)*EXP(EX)
      R=SQRT(BB*BB-4.0*CC)
      X=(-BB+R)/2.0
      CAS1=4.897E-3-CASO
      DEL=B*XXT-CAS1
C      IS THERE ENOUGH GYPSUM TO SATISFY AQUEOUS COMPLEX AND IONS
      IF (DEL=X) 130,210,210
C      NOT ENOUGH GYPSUM TO REACH EQUILIBRIUM SO DISSOLVE ALL AND SOLVE A
130  X=XXT*B
      XXT=0.0
      CAS1=0.0
      A=A+X
      G=G+X
      U=SQRT(2.0*(A+F+G+ASR)+0.5*(S+H+HCO3+AP+AN))
      AA=EXP(-9.366*(U/(U+1.)-.2*U*U))
140  BB=-(4.9E-03+AA*A+AA*G)
      CC=AA*A*G-4.9E-3*CASO
      XXXX=BB*BB-4.0*AA*CC
      IF (XXXX) 150,150,160
150  X1=0.0
      GO TO 170
160  X1=(-BB-SQRT(XXXX))/(2.0*AA)
170  CASO=CASO+X1
      A=A-X1
      G=G-X1
      GO TO 220
C      IS SO4 ION PRESENT (GYPSUM NOT AND SOLUBILITY PRODUCT NOT EXCEEDED)
180  IF (G) 200,200,190
C      IS CA ION PRESENT (GYPSUM NOT AND SOLUBILITY PRODUCT NOT EXCEEDED)
C      IF CA AND SO4 IONS PRESENT EQUILIBRATE WITH AQUEOUS COMPLEX
190  IF (A) 200,200,140
C      IS CASO4 COMPLEX PRESENT IF YES THEN EQUILIBRATE WITH IONS
200  IF (CASO) 220,220,140
210  A=A+X
      G=G+X
      XXT=XXT-X/B
      CASO=CASO+CAS1

```

```

      XXT=XXT-CAS1/B
220  A2=A
      SF1=S
C    CHECK K AND NA SPECIES IF ONE SPECIES NOT PRESENT NO EXCHANGE
      IF ((RP.EQ.0.0.OR.S.EQ.0.0).AND.(AP.EQ.0.0.OR.SAT.EQ.0.0)) GO TO 3
130
      P=1.15
      FK=8.23
      IF (AP.GE.4.0E-07) GO TO 230
      ERROR=5.0E-06
      GO TO 240
230  ERROR=1.200E-06
240  KAR=0
      NUMB=100
C    INITIAL SOLUTION TO START NEWTON-RAPHSON AND CHECK CONSTRAINTS
      X=.5*AP/B-5.0E-09
      IF (X.GE.SAT) X=.6*SAT-5.0E-10
      IF (-X.GT.RP) X=-RP+1.0E-11
      IF (-X.GT.S/B) X=-S/B+1.0E-11
      ION=3
      GO TO 260
250  X=X+.1*X
260  X70=X
270  S1=S+B*X
      AP1=AP-B*X
      SAT1=SAT-X
      RP1=RP+X
C    NO CHEMICALS CAN BE NEGATIVE
      IF (S1.LT.0.0.OR.SAT1.LT.0.0.OR.RP1.LT.0.0.OR.AP1.LT.0.0) GO TO 28
10
      GO TO 290
280  X=.5*X70
      X70=X
      GO TO 270
290  FX=(S1/AP1)*(RP1/SAT1)**P=FK
C    K=NA ION EXCHANGE EQUATION
      FXP=(FX+FK)*(P/RP1+P/SAT1+B/S1+B/AP1)
      CALL NEWTIT (X,FX,FXP,ERROR,NUMB,KAR,ION,F,S,AP,A,HCO3,ET,SAT,CT,R
1P,XXT,CAL,B)
      IF (NUMB.GE.90) WRITE (6,700) ION,FX,X,FXP,S1,AP1,SAT1,RP1
      GO TO (270,320,300,320,250,310), KAR
300  WRITE (6,710)
      GO TO 690
310  WRITE (6,720)
      GO TO 690
C    FINAL ANSWERS
320  Y=X
      S=S+B*Y
      AP=AP-B*Y
      SAT=SAT-Y
      RP=RP+Y
330  INDEX=8
      SF2=S
C    CHECK NA AND CA SPECIES IF ONE NOT PRESENT NO EXCHANGE
      IF ((ET.EQ.0.0.OR.S.EQ.0.0).AND.(A.EQ.0.0.OR.SAT.EQ.0.0)) GO TO 40

```

```

10
  ERROR=1.000E-06
  NUMB=100
  KAR=0
  ION=2
  P=.494
  FK=2.24
C  INITIALIZE NEWTON-RAPHSON AND CHECK CONSTRAINTS
  X=.1*S/B-5.0E-09
  IF (X.GE.ET) X=.8*ET-5.0E-10
  IF (-X.GT.SAT/2.) X=-SAT/2.+1.0E-11
  IF (-X.GT.A/B) X=-A/B+1.0E-11
  GO TO 350
340  X=X+.1*X
350  X70=X
360  A1=A+B*X
  S1=S-2.*B*X
  ET1=ET-X
  SAT1=SAT+2.*X
C  NO CHEMICALS CAN BE NEGATIVE
  IF (A1.LT.0.0.OR.S1.LT.0.0.OR.ET1.LT.0.0.OR.SAT1.LT.0.0) GO TO 370
  GO TO 380
370  X=.5*X70
  X70=X
  GO TO 360
380  U=SQRT(2.0*(A+F+G+ASR)+0.5*(S+H+HCO3+AP+AN))
  AA=EXP(-2.344*(U/(U+1.)-.2*U*U))
C  NA-CA ION EXCHANGE EQUATION
  FX=AA*(A1/(S1*S1))*(SAT1/ET1)**P-FK
  FXP=(FX+FK)*((2.*P)/SAT1+P/ET1+B/A1+(4.*B)/S1)
  CALL NEWTIT (X,FX,FXP,ERROR,NUMB,KAR,ION,F,S,AP,A,HCO3,ET,SAT,CT,R
1P,XYT,CAL,B)
  IF (NUMB.GE.90) WRITE (6,700) ION,FX,X,FXP,S1,A1,SAT1,ET1
  GO TO (360,390,300,390,340,310), KAR
C  FINAL ANSWERS
390  Y=X
  A=A+B*Y
  S=S-2.*B*Y
  ET=ET-Y
  SAT=SAT+2.*Y
400  A3=A
  IF (CT.LT.1.01E-11) CT=0.0
C  CHECK MG-CA SPECIES IF ONE NOT PRESENT NO EXCHANGE
  IF ((ET.EQ.0.0.OR.F.EQ.0.0).AND.(A.EQ.0.0.OR.CT.EQ.0.0)) GO TO 470
  P=1.23
  FK=1.26
  ERROR=1.000E-06
  NUMB=100
  KAR=0
  ION=1
C  INITIALIZE NEWTON-RAPHSON AND CHECK CONSTRAINTS
  X=.1*F/B-5.0E-09
  IF (X.GE.ET) X=.8*ET-5.0E-10
  IF (-X.GT.CT) X=-CT+1.0E-11
  IF (-X.GT.A/B) X=-A/B+1.0E-11

```

```

GO TO 420
410 X=X+.1*X
420 X70=X
430 A1=A+B*X
    F1=F-B*X
    ET1=ET-X
    CT1=CT+X
C    NO CHEMICALS CAN BE NEGATIVE
    IF (A1.LT.0.0.OR.F1.LT.0.0.OR.ET1.LT.0.0.OR.CT1.LT.0.0) GO TO 440
    GO TO 450
440 X=.5*X70
    X70=X
    GO TO 430
C    MG-Ca ION EXCHANGE
450 FX=(A1/F1)*(CT1/ET1)**P-FK
    FXP=(FX+FK)*(P/ET1+P/CT1+B/A1+B/F1)
    CALL NEWTIT (X,FX,FXP,ERROR,NUMB,KAR,ION,F,S,AP,A,HCO3,ET,SAT,CT,P
1P,XXT,CAL,B)
    IF (NUMB,GE.90) WRITE (6,700) ION,FX,X,FXP,F1,A1,CT1,ET1
    GO TO (430,460,300,460,410,310), KAR
C    FINAL ANSWERS
460 Y=X
    A=A+B*Y
    F=F-B*Y
    ET=ET-Y
    CT=CT+Y
470 A4=A
    U=SQRT(2.0*(A+F+G+ASR)+0.5*(S+H+HCO3+AP+AN))
    AA=EXP(-7.034*(U/(U+1.))-0.2*U*U)
    Z3=4.676E-10-A*HCO3**2*AA
C    IS CALCIUM BICARBONATE SOLUBILITY EXCEEDED
C    YES SOLVE NEW EQUILIBRIUM
C    NO CHECK TO SEE IF CALCIUM CARBONATE PRESENT
    IF (Z3) 480,550,550
480 ERROR=1.0E-11
    KAR=0
    NUMB=100
    ION=4
C    INITIALIZE NEWTON-RAPHSON AND CHECK CONSTRAINTS
    XJ=SQRT(4.676E-10/(AA*A))
    X=(XJ-HCO3)/2.
    IF (-X,GT.A) X=-(.93*A-5.0E-07)
    GO TO 500
490 X=X+.1*X
500 X70=X
510 CA1=A+X
    HC1=HCO3+2.*X
    IF (CA1.LT.0.0.OR.HC1.LT.0.0) GO TO 520
    GO TO 530
520 X=.5*X70
    X70=X
    GO TO 510
C    CALCIUM BICARBONATE SOLUBILITY EQUATION
530 FX=CA1*HC1**2-4.676E-10/AA
    FXP=HC1**2+4.*CA1*HC1

```

```

CALL NEWTIT (X,FX,FXP,ERROR,NUMB,KAR,ION,F,S,AP,A,HCO3,ET,SAT,CT,R
IP,XYT,CAL,H)
IF (NUMB.GE.90) WRITE (6,700) ION,FX,X,FXP,CAL,HCO1
GO TO (510,540,300,540,490,310), KAR
C
FINAL ANSWERS
540 A=A+X
HCO3=HCO3+2.*X
CAL=CAL-X/B
IF (A.LT.0.0.OR.HCO3.LT.0.0) GO TO 690
GO TO 580
C
SOLUBILITY PRODUCT NOT EXCEEDED
C
IF NOT SKIP TO NEXT STEP
C
IF CALCIUM CARBONATE PRESENT DISSOLVE IT UP AND CALCULATE NEW EQUIL
550 IF (CAL.GT.0.0) GO TO 560
GO TO 580
560 XM=CAL*B
AOLD=A
HCOLD=HCO3
COLD=CAL
A=A+XM
HCO3=HCO3+2.*XM
CAL=0.0
QU=SQRT(2.0*(A+F+G+ASR)+.5*(S+H+HCO3+AP+AN))
QA=EXP(-7.034*QU/(QU+1))-0.2*QU*QU
Z3=4.676E-10-A*HCO3**2*QA
IF (Z3) 570,580,580
570 A=AOLD
HCO3=HCOLD
CAL=COLD
GO TO 480
C
ALL EQUATIONS AT EQUILIBRIUM SEPERATELY NOW ITERATE UNTIL ALL APPR
C
SIMULTANEOUS EQUILIBRIUM
580 DEL=A-A11
IF (DEL+1.0E-5) 100,590,590
590 IF (DEL-1.0E-5) 600,600,100
600 DEL=A-A2
IF (DEL+1.0E-5) 100,610,610
610 IF (DEL-1.0E-5) 620,620,100
620 DEL=A-A3
IF (DEL+1.0E-5) 100,630,630
630 IF (DEL-1.0E-5) 640,640,100
640 DEL=A-A4
IF (DEL+1.0E-5) 100,650,650
650 IF (DEL-1.0E-5) 660,660,100
660 DEL=S-SF1
IF (ABS(DEL)-1.0E-05) 670,670,100
670 DEL=S-SF2
IF (ABS(DEL)-1.0E-05) 680,680,100
C
SYSTEM AT EQUILIBRIUM
C
IF BICARBONATE IS PRESENT PH IS DETERMINED THRU EQUILIBRIUM CONSTR
680 IF (HCO3.LE.0.0) RETURN
U=SQRT(2.0*(A+F+G+ASR)+0.5*(S+H+HCO3+AP+AN))
APH=6.65-0.5*ALOG10(A)+1.018*(U/(1.0+U)-0.2*U**2)
690 RETURN
C

```

700 FORMAT (1I6,7E15.5)
710 FORMAT (96H2ROOT DID NOT CONVERGE WITHIN MAXIMUM NUMBER OF ITERATI
10NS,PROGRAM RETURNING TO NEXT SET OF DATA)
720 FORMAT (54H0OVERFLOW IN COMPUTATION,RETURNING TO NEXT SET OF DATA)
 END

SUBROUTINE NEWTIT (X,FX,FXP,E,N,K,ION,F,S,AP,A,HCO3,ET,SAT,CT,RP,X
 1XT,CAL,B)

```

C
C   THIS SUBROUTINE CALCULATES THE ROOT OF AN EQUATION USING THE
C   NEWTON-RAPHSON ITERATION SCHEME
C
      IF (K.GT.0) GO TO 100
      J=K
      K=1
      NMAX=N
      N=0
100   T=FX
      IF (J.EQ.0) GO TO 110
      IF (ABS(X).LT.1.E-34) GO TO 120
      T=T/X
110   IF (.NOT.ABS(T).GT.F) GO TO 240
120   IF (N.EQ.NMAX) GO TO 250
      IF (.NOT.ABS(FXP).GT.0.) GO TO 270
      T=X
      X=X-FX/FXP
      GO TO (130,160,190,220), ION
130   IF (F.EQ.0.) GO TO 140
      IF (B*X.GE.F) X=.95*F/B
      IF (ET.EQ.0.) GO TO 150
140   IF (X.GE.ET) X=.95*ET
150   IF (-X.GT.ET) X=-CT+1.0E-11
      IF (-X.GT.A/B) X=-A/B+1.0E-11
      GO TO 230
160   IF (S.EQ.0.) GO TO 170
      IF (2.*B*X.GE.S) X=.95*S/(2.*B)
      IF (ET.EQ.0.) GO TO 180
170   IF (X.GE.ET) X=.95*ET
180   IF (-X.GT.SAT/2.) X=-SAT/2.+1.0E-11
      IF (-X.GT.A/B) X=-A/B+1.0E-11
      GO TO 230
190   IF (AP.EQ.0.) GO TO 200
      IF (B*X.GE.AP) X=.95*AP/B
      IF (SAT.EQ.0.) GO TO 210
200   IF (X.GE.SAT) X=.95*SAT
210   IF (-X.GT.RP) X=-RP+1.0E-11
      IF (-X.GT.S/B) X=-S/B+1.0E-11
      GO TO 230
220   IF (-X.GT.A) X=-.95*A
      IF (-X.GT.HCO3) X=-.45*HCO3
230   N=N+1
      CALL OVERFL (I)
      IF (I.EQ.1) GO TO 280
      IF (.NOT.ABS(T-X).GT.0.) GO TO 260
      RETURN
240   K=2
      RETURN
250   K=3
      RETURN
260   K=4
  
```


SUBROUTINE INPUT (NEXT)

THIS SUBROUTINE SETS UP THE INITIAL MACRO ION CONCENTRATION DISTRIBUTION, AND IF DESIRED, RE-SETS THESE CONCENTRATIONS AT SOME GIVEN TIME DURING THE EXECUTION OF THE MACRO ION PROGRAM

PARAMETER NPTS=1500, NCEL=1100, NBDY=200, NION=8, NIONS=6
COMMON/MIC/NPL,LIMIT,DELT,DELX,DELY,KOUNT,NTOPX,NTOPY,SUPRES,MAXIN
COMMON /LARGE/ XMOVE(NPTS),YMOVE(NPTS),CIJ(NCEL,NION),CPOINT(NPTS,
1NION),CSTAR(NCEL,NION),CSOIL(NCEL,NIONS),DELC(NCEL,NION),IJBORY(NB
2DY),STORE(100)

REAL LAST
DATA LAST,YES,EPS,/6H LAST,6H YES,0.00001/
NCOUNT=0
ASSIGN 130 TO MSKIP
ASSIGN 180 TO NSKIP
NMORE=NION+NIONS

READ THE INITIAL OR THE NEW MACRO ION DISTRIBUTION OF THE CELLS FROM CARDS

NOMENCLATURE:

WORD.....@LAST@ IS A SENTINAL DESIGNATING THE END OF A SET OF
INPUT DATA
CO.....INITIAL OR NEW CONCENTRATION
I1,J1.....I,J COORDINATES OF THE FIRST CELL IN A ROW OR COLUMN
OF CELLS
I2,J2.....I,J COORDINATES OF THE LAST CELL IN A ROW OR COLUMN
OF CELLS

DO 200 ION=1,NMORE
IF (ION.GT.NION) ASSIGN 140 TO MSKIP
IF (ION.GT.NION) ASSIGN 200 TO NSKIP
100 READ (5,240) WORD,CO,I1,J1,I2,J2,NEXT
WRITE (6,240) WORD,CO,I1,J1,I2,J2,NEXT
IF (WORD.EQ.LAST) GO TO 170
IF (I1.EQ.0.AND.J1.EQ.0) GO TO 170
IF ((I2.GT.0.AND.I1.GT.I2).OR.(J2.GT.0.AND.J1.GT.J2)) GO TO 160
IF (I2.NE.0) GO TO 110
I2=I1
110 IF (J2.NE.0) GO TO 120
J2=J1
120 DO 150 J=J1,J2
DO 150 I=I1,I2
IJ=(J-1)*NTOPX+I
GO TO MSKIP, (130,140)
130 CIJ(IJ,ION)=CO
CSTAR(IJ,ION)=CO
GO TO 150
140 CSOIL(IJ,ION-NION)=CO
150 CONTINUE
NCOUNT=NCOUNT+(I2-I1+1)*(J2-J1+1)
IF (NCOUNT.LE.MAXIN) GO TO 100
WRITE (6,250)

```

      STOP
160  WRITE (6,260)
      STOP
C
C   ASSIGN THE POINTS IN EACH CELL THE SAME CONCENTRATION AS THE CELL
C
170  NCOUNT=0
      GO TO NSKIP, (180,200)
180  DO 190 NPOINT=1,LIMIT
      IPX=IFIX(XMOVE(NPOINT)/DELX+EPS)+1
      JPY=IFIX(YMOVE(NPOINT)/DELY+EPS)+1
      IJXY=(JPY-1)*NTOPX+IPX
      DO 190 IONS=1,NION
      CPOINT(NPOINT,IONS)=CIJ(IJXY,IONS)
190  CONTINUE
200  CONTINUE
      IF (SUPRES.EQ.YES) GO TO 230
      WRITE (6,270)
      DO 220 ION=1,NION
      WRITE (6,280) ION
      DO 220 MJ=1,NTOPY
      NLINE=NTOPY-MJ
      LINUM=NLINE+1
      DO 210 MI=1,NTOPX
      STORE(MI)=CIJ(NLINE*NTOPX+MI,ION)
210  CONTINUE
      WRITE (6,290) LINUM
      WRITE (6,300) (STORE(IJ),IJ=1,NTOPX)
220  CONTINUE
230  CONTINUE
      RETURN
C
240  FORMAT (A6,E9.4,5I5)
250  FORMAT (/ ,5X,0,...SPECIFIED INPUT EXCEEDS MAXIMUM...0,/)
260  FORMAT (/ ,5X,0,...SPECIFIED INPUT IS INDEXED INCORRECTLY...0,/)
270  FORMAT (1H1,/,5X,0,...INITIAL CONCENTRATION FIELD...0,/)
280  FORMAT (/,17X,0 ION NUMBER 0,I4)
290  FORMAT (/ ,0 J=0,I5,/)
300  FORMAT (1P10E12.4)
      END

```

SUBROUTINE AVRGE (KN)

THIS SUBROUTINE ASSIGNS EACH CELL A CONCENTRATION WHICH IS THE AVERAGE CONCENTRATION OF ALL POINTS WITHIN THE CELL.

PARAMETER NPTS=1500, NCEL=1100, NBDY=200, NION=8, NIONS=6
COMMON/MIC/NPL,LIMIT,DELT,DELX,DELY,KOUNT,NTOPX,NTOPY,SUPRES,MAXIN
COMMON /LARGE/ XMOVE(NPTS),YMOVE(NPTS),CIJ(NCEL,NION),CPOINT(NPTS,
1NION),CSTAR(NCEL,NION),CSOIL(NCEL,NIONS),DELC(NCEL,NION),IJBURY(NB
2DY)

DIMENSION ICS(NCEL)

DO 100 IJ=1,MAXIN

ICS(IJ)=0

100 CONTINUE

DO 110 NP=1,LIMIT

I=IFIX(XMOVE(NP)/DELX)+1

J=IFIX(YMOVE(NP)/DELY)+1

IJT=(J-1)*NTOPX+I

ICS(IJT)=ICS(IJT)+1

IF(ICS(IJT).LE.1) CSTAR(IJT,KN)=CPOINT(NP,KN)

IF(ICS(IJT).GT.1) CSTAR(IJT,KN)=CSSTAR(IJT,KN)+CPOINT(NP,KN)

110 CONTINUE

DO 120 IJ=1,MAXIN

IF(ICS,IJ,0) GO TO 120

DENOM=FLOAT(ICS(IJ))

CSSTAR(IJ,KN)=CSSTAR(IJ,KN)/DENOM

120 CONTINUE

RETURN

END

DISTRIBUTION

<u>No. of Copies</u>	<u>Off-Site</u>
1	<u>Chicago Patent Group</u> A. A. Churm
215	<u>Energy Research and Development Administration</u> <u>Technical Information Center</u>
2	<u>Energy Research and Development Administration</u> <u>Headquarters</u> Waste Management and Transportation Division Germantown, Maryland 20767 Owen P. Gornley Chief, Waste Facilities Branch Robert W. Ramsey, Jr. Chief, Development Branch
2	<u>Nuclear Regulatory Commission</u> <u>Directorate of Regulation</u> 7920 Norfolk Avenue Bethesda, Maryland 20014 Ronald L. Bullard Chief, Environmental Specialist Branch W. Gamill Chief, Site Analysis Branch
2	<u>Energy Research and Development Administration</u> <u>Division of Production & Material Management</u> Washington, D. C. 20545 William L. Lennemann Chief, Chemical Process James W. Pollock
2	<u>Energy Research and Development Administration</u> <u>Idaho Falls, Idaho 83401</u> Dr. Adrian H. Dahl Chief, Environmental Science Branch B. L. Schmalz

No. of
Copies

Off-Site

2	<u>U. S. Geological Survey</u> 2100 M. Street Washington, D. C. 20037 George Debuchananne, Director M. K. Hubbert Research Geophysicist
1	<u>U. S. Geological Survey</u> Menlo Park, California 94025 Jacob Rubin
1	<u>U. S. Geological Survey</u> Groundwater Branch Denver, Colorado 80225 R. W. Stallman
1	<u>U. S. Geological Survey</u> Water Resources Division Bldg. 53, Room H-2910 Lakewood, Colorado 80225 J. D. Bredehoeft
1	<u>U. S. Geological Survey</u> Water Resources Division P.O. Box 917 Council Bluffs, Iowa 51501 O. J. Ramsvick Supervisory Hydrologist
1	<u>U. S. Geological Survey</u> Water Resources Division P.O. Box 2857 Raleigh, North Carolina 27202 Ralph C. Heath District Chief
1	<u>U. S. Bureau of Mines</u> 4800 Forbes Avenue Pittsburgh, Pennsylvania 15213 George Evans

No. of
Copies

Off-Site

2	<u>U. S. Bureau of Mines</u> Spokane Mining Research Laboratory N. 1430 Washington St. Spokane, Washington 99201 Robert C. Bates Michael M. McDonald
1	<u>Environmental Protection Agency</u> Crystal Mall Washington, D. C. 20460 Arnold Joseph
1	<u>Environmental Protection Agency</u> Office of Research & Monitoring Edison Water Quality Office Edison, New Jersey 08817 Richard Field
1	<u>National Water Commission</u> 800 N. Quincy Arlington, Virginia 22209 Dr. John S. Gladwell
1	<u>USDA-ARS</u> Engineering Research Center CSU Foothills Campus Fort Collins, Colorado 80521 Dr. David A. Woolhiser Hydraulic Engineer
1	<u>USDA-ARS</u> P.O. Box 1096 Boise, Idaho 83701 J. L. Robins
1	<u>USDA-ARS</u> Snake River Conservation Research Center Route 1, Box 186 Kimberly, Idaho 83341 Dr. Marvin E. Jensen, Director

No. of
Copies

Off-Site

- 1 USDA Hydrograph Laboratory
 Plant Industry Station
 Soils Building
 Beltsville, Maryland 20705
- H. N. Holton, Director
- 1 USDA Sedimentation Laboratory
 P.O. Box 30
 Oxford, Mississippi 38655
- A. R. Robinson, Director
- 1 USDA-ARS
 North Central Watershed Research Center
 P.O. Box 916
 Columbia, Missouri 65201
- Dr. C. R. Amerman
 Hydraulic Engineer
- 1 USDA-ARS
 Northeast Watershed Research Center
 Pennsylvania State University
 111 Research Building 3
 University Park, Pennsylvania 16802
- Dr. A. S. Rogowski
 Soil Scientist
- 1 National Academy of Science
 U. S. National Committee for the IHD
 2101 Constitution Avenue
 Washington, D. C. 20418
- Dr. Leo Heindl
 Executive Secretary
- 1 National Academy of Science
 Committee on Radioactive Waste Management
 National Research Council
 2101 Constitution Avenue
 Washington, D. C. 20418
- Dr. Cyrus Klingsberg
 Technical Secretary

<u>No. of Copies</u>	<u>Off-Site</u>
1	<u>National Science Foundation</u> Division of Advanced Technology Application 1800 Sixth Street N.W. Washington, D. C. 20530 Ray Zahradnik
1	<u>Robert S. Kerr Water Research Center</u> P.O. Box 1198 Ada, Oklahoma 74820 Dr. J. W. Keeley
1	<u>Alabama Geologic Survey</u> P.O. Box Drawer 0 University, Alabama 35486 Phil Lammereaux State Geologist
1	<u>Idaho Bureau of Mines & Geology</u> University of Idaho Moscow, Idaho 83843 Dr. Roy E. Williams
1	<u>Idaho Department of Water Administration</u> State House, Annex 2 Boise, Idaho 83707 Keith R. Higginson, Director
1	<u>Idaho Water Resources Board</u> Boise, Idaho R. R. Lee
1	<u>Illinois State Water Survey</u> P.O. Box 232 Urbana, Illinois 61801 Keros Cartwright
1	<u>State Geological Survey of Kansas</u> University of Kansas Lawrence, Kansas 66044 Dr. John C. Halepaska Chief, Water Resources Section

<u>No. of Copies</u>	<u>Off-Site</u>
1	<u>State of Maine</u> Environmental Improvement Commission Augusta, Maine 04330 Steven D. Freedman
1	<u>St. Anthony Falls Hydraulic Laboratory</u> Mississippi River at Third Avenue East Minneapolis, Minnesota 55414 Professor C. Edward Bowers
1	<u>Desert Research Institute</u> Center for Water Resources Research Reno, Nevada Dr. Clinton Case
1	<u>State of North Carolina</u> Groundwater Division Water and Air Resources Raleigh, North Carolina 27611 Harry M. Peek, Chief
1	<u>Water Quality Management Laboratory</u> Rt. 2, Box 322A Durant, Oklahoma 74701 V. L. Hauser
1	<u>State of Oregon</u> 516 Public Service Bldg. Salem, Oregon 97310 Chris Wheeler State Engineer
1	<u>Agencies for Environmental Conservation</u> Montpelier, Vermont 05602 Dr. Martin L. Johnson Secretary

<u>No. of Copies</u>	<u>Off-Site</u>
1	<u>State of Washington</u> <u>Department of Ecology</u> P.O. Box 829 Olympia, Washington 98504 John R. Raymond
1	<u>Washington State Water Research Center</u> <u>Pullman, Washington 99163</u> James Crosby, III Geologist
1	<u>Atomic Industrial Forum, Inc.</u> <u>475 Park Avenue South</u> New York, New York 10016 Gerald V. Halvorsen Environmental Projects Manager
1	<u>Computer Sciences Corporation</u> <u>Federal Building</u> Richland, Washington 99352 R. W. Nelson
1	<u>Sandia Laboratories</u> <u>Division 5166</u> Albuquerque, New Mexico 87115
1	<u>University of Arizona</u> <u>Office of Arid Lands Studies</u> 1201 East Speedway Blvd. Tucson, Arizona 95719 Dr. Ken Faster
1	<u>University of Arizona</u> <u>Department of Hydrology and Water Resources</u> 200-E Old Psychology Tucson, Arizona 95721 Dr. A. K. Tyagi

<u>No. of Copies</u>	<u>Off-Site</u>
1	<u>University of Arizona</u> Tucson, Arizona 95721 E. S. Simpson
2	<u>University of California</u> Berkeley, California 94720 Dr. Warren Kauffman P. A. Witherspoon Department of Civil Engineering
1	<u>University of California at Davis</u> Department of Water Sciences and Engineering Davis, California 95616 J. N. Luthin
1	<u>University of California at Riverside</u> College of Biological & Agricultural Sciences Riverside, California 92507 Dr. N. T. Coleman
1	<u>Stanford University</u> Department of Civil Engineering Stanford, California 94305 Paul Kruger
1	<u>Stanford University</u> School of Earth Sciences Department of Ecology Stanford, California 94305 E. Aguado
1	<u>Georgia Institute of Technology</u> School of Civil Engineering Atlanta, Georgia 30332 Dr. J. R. Wallace

<u>No. of Copies</u>	<u>Off-Site</u>
2	<u>University of Illinois</u> Urbana, Illinois 61801 A. Klute Professor Ven Te Chow Hydrosystems Laboratories
1	<u>Purdue University</u> Dept. of Agricultural Engineering West LaFayette, Indiana 47907 Emru D. Millette
1	<u>Iowa State University</u> Ames, Iowa D. Kirkham
1	<u>University of Kentucky</u> Department of Mechanical Engineering Lexington, Kentucky 40506 C. T. Hahn
1	<u>Harvard University</u> School of Public Health Boston, Massachusetts 02138 Dr. D. W. Moeller
1	<u>Massachusetts Institute of Technology</u> 77 Massachusetts Avenue Room 48-263 Cambridge, Massachusetts 02139 Dr. Donald Harleman
1	<u>University of Minnesota</u> Department of Civil Engineering Minneapolis, Minnesota 55418 H. Stefor

No. of
Copies

Off-Site

2	<u>University of Nebraska</u> School of Civil Engineering Lincoln, Nebraska 68503 Mark J. Hammer Warren Viessman, Jr.
1	<u>Princeton University</u> School of Engineering/Applied Science Department of Civil and Geological Engineering Princeton, New Jersey 08540 Dr. Robert W. Cleary
2	<u>Princeton University</u> Department of Civil and Geological Engineering Princeton, New Jersey 08540 R. J. M. DeWiest G. F. Pinder
1	<u>New Mexico State University</u> Department of Agronomy P. O. Box 3Q Las Cruces, New Mexico 88001 P. J. Wierenga
1	<u>New Mexico Institute of Mining and Technology</u> Department of Geosciences Socorro, New Mexico 87801 Lynn W. Gelhar
1	<u>State University of New York at Buffalo</u> Department of Mechanical Engineering Buffalo, New York 14214 Ralph T. Cheng

No. of
Copies

Off-Site

1	<u>State University of New York at Buffalo</u> Department of Civil Engineering Buffalo, New York 14214 Dr. Ralph Rumer
1	<u>Cornell University</u> School of Civil Engineering B22 Bailey Hall Ithaca, New York 14850 W. H. Brutsaert
1	<u>Oregon State University</u> Corvallis, Oregon 97331 Dr. Larry Boersma Professor of Soils
1	<u>Drexel Institute of Technology</u> Philadelphia, Pennsylvania 19104 Irwin Remson
1	<u>Vanderbilt University</u> Nashville, Tennessee 37203 Dr. Frank L. Parker
1	<u>Utah State University</u> Utah Water Research Laboratory College of Engineering Logan, Utah 84321 J. Paul Riley
1	<u>University of Wyoming</u> Laramie, Wyoming 82070 Dr. Paul A. Rechard Director, Water Resources Research
1	<u>Washington State University</u> Department of Civil Engineering 148 Sloan Hall Pullman, Washington 99163 Dr. Donald L. Bender

No. of
Copies

Off-Site

2	<u>Washington State University</u> Department of Agronomy and Soils Pullman, Washington 99163 Dr. Walter H. Gardner Dr. Gaylon S. Campbell
1	<u>Washington State University</u> Albrook Hydraulic Laboratory Pullman, Washington 99163 Dr. John F. Orsborn, Head
1	<u>Washington State University</u> Agricultural Engineering Department Pullman, Washington 99163 Dr. L. G. King
1	<u>University of Wisconsin</u> Department of Soil Science Madison, Wisconsin 53706 W. R. Gardner
1	<u>Water Conservation and Irrigation Commission</u> P. O. Box 220R Royal Exchange Sydney, 2000, Australia Frans R. Kalf
1	<u>University of New South Wales</u> School of Civil Engineering Sydney, Australia Dr. K. K. Watson
1	<u>University of Waterloo</u> Department of Civil Engineering Faculty of Engineering Waterloo, Ontario, Canada John Pickens

No. of
Copies

Off-Site

- 1 McMaster University
 Department of Civil Engineering
 Hamilton, Ontario, Canada
- Dr. William James
- 1 Institut de Mecanique (I.M.G.)
 Domaine Universitaire
 B. P. 53
 Centre de Tri 38041
 Grenoble-Cedex, France
- Georges Vachaud
- 1 Technion-Israel Institute of Technology
 Haifa, Israel
- J. Bear
- 1 Agricultural Research Organization
 The Volcani Center
 The Institute of Soils and Water
 P. O. B. 6
 Bet Dagan
- Shlomo P. Neuman
- 1 Kyoto University
 Kyoto, Japan
- Yoriteru Inoue

On-Site Hanford

- 8 AEC Richland Operations Office
- O. J. Elgert
 J. L. Rhoades
 P. J. Holsted (2)
 E. B. Jackson
 P. G. Rhoades
 R. B. Goranson
 B. J. Melton

No. of
Copies

On-Site Hanford

Atlantic Richfield Hanford Co.

16
G. E. Backman
G. L. Borsheim
D. J. Brown
L. E. Bruns
R. A. Deju
R. E. Gephart
W. M. Harty
R. E. Isaacson (2)
L. M. Knights
C. W. Malody
B. J. McMurray
G. A. Nicholson
H. P. Shaw
P. W. Smith
ARHCO File

1 Westinghouse Hanford Co. HEDL

R. B. Hall

18 Battelle-Northwest

D. B. Cearlock (3)
J. R. Eliason
K. L. Kipp
D. E. Olesen
A. E. Reisenauer
R. C. Routson
R. J. Serne
D. R. Friedrichs
S. W. Ahlstrom
R. G. Baca
A. Brandstetter
R. W. Wallace
J. J. Fuquay
A. J. Shuckrow
Tech. Information
Tech. Pubs.

3 United Nuclear, Inc.

C. D. Corbit (2)
F. E. Owen