THE TRANSMISSIVITY ITERATIVE PROGRAMS
ON THE PDP-9 COMPUTER
A MAN-MACHINE INTERACTIVE SYSTEM

Battelle

Pacific Northwest Laboratories
Richland, Washington 99352

1973

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

NOTICE

The report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Atomic Energy Commission, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

PACIFIC NORTHWEST LABORATORY
operated by
BATTELLE
for the
U.S. ATOMIC ENERGY COMMISSION
Under Contract AT(45-1)-1830

Printed in the United States of America
Available from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Road
Springfield, Virginia 22151
Price: Printed Copy \$5.45; Microfiche \$1.45

3 3679 00062 2771

THE TRANSMISSIVITY ITERATIVE PROGRAMS ON THE
PDP-9 COMPUTER - A MAN-MACHINE INTERACTIVE SYSTEM

by

L. E. Addison*
D. R. Friedrichs
K. L. Kipp

Water and Land Resources Department
*Mathematics Department

1973

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

A system of computer programs, The Transmissive Iterative Sequence, has been written with the purpose of providing an analytical aid in the simulation of groundwater flow. This man-machine interactive system directly involves the user making decisions in the analysis loop.

Because of the limited amount of data available, this analysis loop is highly iterative in nature and its success depends on the ability to rapidly input and output information in a form directly related to the groundwater system being analyzed. The Iterative Sequence provides these analytical capabilities in a self-contained computer package which constitutes a useful tool for modeling a groundwater system.

CONTENTS

	<u>Page</u>
SUMMARY.	i
FIGURES.	v
SYMBOLS AND DIMENSIONS	vi
INTRODUCTION	1
OPERATION OF THE ITERATIVE SYSTEM.	2
COMPUTER SYSTEM.	4
Hardware Details.	4
Software Details.	4
MONITOR PROGRAM.	7
Computer Implementation	7
Data Matrix Transfer Allocation	11
Monitor Oriented Functions.	12
Data Matrix Modification.	14
Dispatch Function	15
STREAMLINE GENERATION.	16
Computer Implementation	16
Contour Functions	16
Intercept Found for First Time.	21
Streamline Function	24
TRAVEL TIME CALCULATION.	24
Computer Implementation	24
Contours and Streamlines Generation	26
Travel Time Calculation	26
TRANSMISSIVITY CALCULATIONS.	27

CONTENTS (continued)

	<u>Page</u>
Computer Implementation.	28
ACKNOWLEDGMENTS	36
REFERENCES.	39
APPENDIX A - CALCULATION OF GRADIENT.	A-1
APPENDIX B - CALCULATION OF STREAMLINES	B-1
APPENDIX C - DATA FORMATS	C-1
APPENDIX D - COORDINATING GRAPHIC DIGITIZER, CRT DISPLAY, AND DATA MATRICES	D-1

FIGURES

<u>Figure</u>		<u>Page</u>
1	Flow Diagram of the Iterative Sequence.	3
2	Iterative Sequence Monitor-Function Relationship.	6
3	Data Flow in Monitor Data Modification.	14
4	Data Retrieval Format	17
5	Data Search Path for Contours	19
6	Example Contour Process	22
7	Graphical Picture of Contour Object Point	23
8	Convergence of Flow Tubes	27
9	Plot of Transmissivity Distance Along Line of Intersection.	29
10	Example of Transmissivity Cell Iteration.	29
11	Transmissivity Cell (Steady State).	31
12	Transmissivity Cell (Transient State)	32
13	Plot of Predicted Transmissivity vs. Distance Along Line of Intersection	33
14	Plot of Specified Transmissivity vs. Distance Along Line of Intersection	34

SYMBOLS AND DIMENSIONS

V_i	a value at point i on a nodal grid system (point i shortened form of (x_i, y_i))
n_n	horizontal nodes per segmented data section
n_v	vertical nodes per segmented data section
ϕ	overlap nodes per segmented data section
P_i	point i on a nodal grid system (point i shortened form of (x_i, y_i))
X_i	horizontal node position i in nodal grid system
Y_i	vertical node position i in nodal grid system
W_i	object point for interpolation (shortened form of (x_i, y_i))
L	vector
SIG_i	effective porosity at $STEP_i$
$PERM_i$	hydraulic conductivity $STEP_i$
$GRAD_i$	hydraulic gradient at $STEP_i$
$STEP_i$	i^{th} step along iteration path
TRN_i	transmissivity at $STEP_i$
H_i	hydraulic potential $STEP_i$
HO_i	aquifer bottom $STEP_i$
D_i	transmissivity cell width $STEP_i$
G_x	horizontal component hydraulic gradient
G_y	vertical component hydraulic gradient
t_i	time

THE TRANSMISSIVITY ITERATIVE PROGRAMS ON THE PDP-9 COMPUTER - A MAN-MACHINE INTERACTIVE SYSTEM

INTRODUCTION

The transmissivity iterative routine is a set of programs for the PDP-9 man-machine interactive computer system developed to compute the transmissivity distribution for the unconfined aquifer of the Hanford Reservation. The process is divided into four phases: 1) preparation of the necessary data and its input to the computer in a suitable form; 2) determination of the basic reference transmissivity curves on which the transmissivity distribution is based; 3) semi-automatic calculation of the transmissivity distribution; and 4) preliminary verification of that distribution which can be done through travel time calculations.

Two other reports, "Variable Thickness Transient Groundwater Flow Model - Theory and Numerical Implementation" (BNWL 1703) and "The Transmissivity Iterative Calculation Routine - Theory and Numerical Implementation" (BNWL 1706), contain the theory of the Variable Thickness Transient (VTT) groundwater flow model and the equations on which the transmissivity calculation is based. This report describes the logic, organization, and techniques of the programs which implement the transmissivity calculation. Detailed instructions for using the programs will be found in the user's manual (BNWL 1708).

HYDROLOGICAL MODELING PROGRAM COMPONENTS

OPERATION OF THE ITERATIVE SYSTEM

The basic data for a transient analysis of transmissivity consist of boundary location information based upon geological and hydrological data and measured water table or hydraulic potential distributions for the region. At least one potential surface and a gradient map are needed. The hand-contoured water table elevation and gradient maps are digitized for use in the computer by the program described in the report "Graphic Digitizer Program to Interpolate Matrix Grid Values" (BNWL 1652). The digitized surfaces are checked by the transmissivity iterative routine, which displays contours on the CRT screen. Streamlines can be calculated instantaneously for each potential surface by one of the system programs. After the streamline paths are evaluated by the investigator, improvements to the contour interpolations can be made to give streamlines that agree with other groundwater flow or geologic data available (e.g., contamination transport patterns). This is the first iterative loop in the sequence (Figure 1).

When a satisfactory potential map is obtained, the streamline program is used to locate convergence zones for the transmissivity reference lines. Then the input transmissivity data is used to calculate transmissivity distributions along these reference lines. The program allows the operator to hand-fit the best curve to the calculated transmissivity values along a given reference line. A second iterative loop is made for this step of the sequence in order to reduce the variations in calculated transmissivities at the boundary curves.

The actual transmissivity distribution is computed section by section over the region in a semi-automatic fashion based on the reference transmissivity curves. The equations for this process appear in BNWL 1706, "The Transmissivity Interactive Calculation Routine - Theory and Numerical Implementation." The resulting distribution is submitted to the VTT program via data link to the Univac 1108 computer.

Once the transmissivity distribution is known, travel time calculations can give much insight into its accuracy. A travel time represents advective transport of fluid along an instantaneous streamline, neglecting dispersion, adsorption, decay, and transient flow effects. However, these calculations can be used for order of magnitude validity checks for a system such as the Hanford Reservation. The equations for travel time calculations are given in the report mentioned above. A third iterative loop exists here since some areas of poor transmissivity values are

discovered. However, the actual VTT simulation reveals much more about the quality of the transmissivity distribution.

The system developed for the PDP-9 computer has been organized and structured to make this four-step process as easy and rapid as possible by maintaining the investigator's participation in the iterative loop, where he can bring all of his experience and knowledge to bear on the proper interpretations. In this way the man-machine interactive computer system becomes an effective analysis tool.

FIGURE 1. Flow Diagram of the Iterative Sequence

COMPUTER SYSTEM

The entire system is based upon a PDP-9 digital computer with a disc operating system and such peripherals as magnetic dectape, graphic digitizer, and interactive CRT display. This combination provides a powerful and flexible analytic tool.

Hardware Details

The PDP-9 is a small computer with 16K of core memory. Its size makes it relatively inexpensive to operate, a major factor when analyses such as those described here necessitate large amounts of computing time.

The disc provides a million words of rapid access mass storage divided as approximately 1/4 operating system and 3/4 program and data storage. The Iterative Sequence makes full use of the disc for both program storage and data storage. Without the disc the Iterative Sequence could not be effectively implemented, primarily due to the core size limitation and the slow speed of any other mass storage device.

The magnetic dectape provides an effective method of storing programs and data. Data in the dectape can be saved, making it possible to continue analysis over several days.

The graphic digitizer is used to input data into the computer. Its power lies in the fact that the data entered into the computer can be taken directly from a hand-drawn contour map.

The iterative CRT display provides an immediate output of analysis results; for example, the flow paths are displayed on the CRT as they are being calculated, so the results of the analysis are easy to interpret. With an overlay map of the system on the CRT, the progress of a flow path in relation to the boundaries of the system is visible. This type of output is much faster and more understandable than printed table output.

Software

The software system is highly disc oriented. That is, since the entire Iterative Sequence with all of its functions is too large to fit at once into a 16K core memory, it is divided into functional blocks and only one functional block

at a time is executed in the computer. The remaining functional blocks are in ready access on the disc. Also, since the data for a sizable groundwater system would quickly fill the computer, the data is are segmented and only the data for the block being computed are in the computer; the remainder is in ready access on the disc. The extensive dependence upon the disc for both storage and speed can easily be seen.

Each functional block attempts to make maximum use of the available peripherals, primarily the graphic digitizer and interactive CRT display. These are the two devices used to input data and output results; they eliminate the need for large hand-digitized tables of input data and lengthy, confusing print-outs. Through these two devices the functional blocks take their directives.

Each functional block performs one primary function, but since each block is optimized according to computer space, speed, and cross-functional interaction, it may share a secondary function with some other functional block. This provides more convenience of operation.

The functional program set is governed by a monitor-supervisory program which assigns data matrices, performs inter-matrix operations, and dispatches control to the desired function. If any function is not properly prepared, control returns to this program.

Following is a list of the programs available in the current system with a brief functional description of each. Figure 2 shows the block relationship.

Monitor-supervisor

- Assigns data to specific data positions
- Initializes data files
- Initializes digitizer
- Dispatches to functional blocks
- Performs average, difference, smoothing, filling operations

Data Digitization

- Takes data from the digitizer and stores it in the computer in a formatted data matrix

Streamline

- Displays flow paths
- Displays contours

Travel time
(hydraulic conductivity)

- Calculates travel times with constant hydraulic conductivity model
- Displays flow paths
- Displays contours

Travel time
(transmissivity)

- Calculates travel times with usable transmissivity model
- Displays flow paths
- Displays contours

Transmissivity (steady-state)

- Generates transmissivity map, steady-state model

Transmissivity (transient)

- Generates transmissivity map, transient state model

FIGURE 2. Iterative Sequence Monitor-Function Relationship

MONITOR PROGRAM

The Monitor Program of the Iterative Sequence provides a general supervisory dispatch role for the individual functional blocks. It includes the assignment and selection of data maps, cross communication between functions, arithmetic operations on data maps, and dispatch of control to a desired function. Because of the iterative nature of the system, rapid access and manipulation of functions is essential to its efficiency.

Computer Implementation

The monitor program operates by taking directives from the teletype keyboard and then completing the required action. The directives are free format fields which are delimited by either a punctuation mark or a string of three characters. Once a field has been set aside by the delimiter, the next field does not begin until a recognized alpha-numeric character which is not a delimiter is input.

The monitor signals its readiness to accept directives by typing a (>) on the teletype. At this point, directives may be entered.

For each directive the first field specifies the action to be taken, represented by one of the three-character neumonics listed in Table 1. Included in Table 1 is a brief description of the action taken by the monitor in response to that neumonic.

If characters other than the directive neumonics listed in Table 1 are entered in the first field, a (?) appears on the teletype and the monitor returns to accept another input directive.

The remaining fields in the directive consist of arguments for the instruction given in the first field. Each instruction requires different arguments which are in different fields. For the specific arguments for each instruction and their order, refer to "The Transmissivity Iterative Program User's Manual,"³ where descriptions of the instructions and their results will also be found.

Before discussing the remaining arguments for a directive and their functions, some underlying constraints on implementation should be noted.

The first item is the storage of data matrices. In general a data matrix, once it has been generated, will be stored and

TABLE 1. Directive Neumonics

1	ASG	Assign a data matrix to a specific use
2	SAV	Save a data matrix for future use
3	DEL	Delete a data matrix from a specific use
4	GEN	Generate a new data matrix
5	INT	Initialize the digitizer
6	LST	List the current data matrix usage assignments
7	TIM	Enter a time value
8	COM	Place identifying comments in a data matrix
9	AVE	Apply the average algorithm to a data matrix pair
10	DIF	Apply the difference algorithm to a data matrix pair
11	FIL	Fill the internal 0's of a data matrix
12	SMU	Apply the data smoothing algorithm to a data matrix
13	EXQ	Execute a specific Iterative Sequence Function

maintained on dectape. However, the Iterative Sequence requires that the data be on the disc to facilitate rapid access. Thus the data must be transferred from the dectape to the disc. Data are identified on dectape and disc by a directory system name and extension (the name is an identifying tag for a specific data matrix and the extension is a classification or grouping identifier). In order to know what is on the dectape a directory of the tape must be obtained from the Disc Operating System (DOS) of the PDP-9. Once the data have been transferred to the disc, they are identified on the disc by name and extension and can be located by means of the directory for the disc. The monitor performs this function of transfer between disc and dectape in either direction.

The disc is divided into five regions, each with a separate directory. The Iterative Sequence and its data matrices are contained on disc unit 2 (DK2) since it is the largest disc segment. All transfers between disc and dec-tape are between disc unit 2 and the specified dec-tape unit. Within the disc, however, data may be transferred among all disc units except unit ϕ (DK ϕ) which is reserved for the DOS. Transferring data from disc 2 (DK2) does not result in its destruction, but if new data are transferred to disc 2 (DK2) (from either dec-tape or other disc segments) with the same name and extension as data already on the disc, the old data are deleted and replaced by the new. So it is possible to save the original data on the disc, rapidly modify the data on DK2, check the results, and quickly restore the data on DK2 to the original form.

Inter-communication between the functional blocks is maintained by means of a data block which is 112 words long, the length of one disc sector. The location of this disc sector is set by the monitor and entered into the directory on disc 2 (DK2) under the name and extension of COMMON BIN. The data block is divided into three segments:

1. Disc Addresses of Functional Blocks
2. Data Matrix Names
3. Common Iterative Sequence Data

Disc addresses are the means by which the various functional blocks are loaded into core from the disc, where they were dumped after previous loading. They then can be reloaded into core memory by means of the system disc bootstrap. (The disc addresses are generated on the initial call to the Iterative Monitor program.) If when the monitor is called a program which is part of the Iterative Sequence is not on disc 2 (DK2), there is no disc address specified and it is ignored. If, however, execution of the functional block is attempted, the monitor will signal an error and ignore the directive. If at some later time after the initial calling of the monitor an additional functional block of the Iterative Sequence is transferred to DK2, it may be entered into the monitor by asking the monitor to execute itself. This checks DK2 again for the functional blocks that are available. Currently 20 locations are reserved for disc addresses for the functional blocks; these are identified in Table 2.

In the monitor program the data matrices are assigned by name to usage slots in the communication data block. Each assignment has a specific use in the various functional blocks. When the functional block requires a specific kind

TABLE 2. Functional Block Assignments

<u>Functional Block</u>	<u>Function Located at the Disc Address Specified</u>
1	Iterative Sequence Monitor
2	Stream Line
3	Transmissivity Calculation (Steady State)
4	Transmissivity Calculation (Transient)
5	Travel Time Calculation (Variable Transmissivity)
6	Travel Time Calculation (Constant Hydraulic Conductivity)
7	Graphic Digitizer Input Algorithm
8	Perspective Display of Data Matrices
9-20	Reserved for Future Use

of data, it looks at this usage table for the name of the data matrix. Use of names simplifies the directory handling chores and makes it much simpler to transfer data between devices. There currently are twenty data matrix usage assignments available, each consisting of three consecutive words. The first two words contain the 6-bit ASCII character codes for the name and the third word contains the 6-bit code for the extension. These are defined in Table 3.

The monitor refers to the usage slots in Table 3 by the three-character neumonics in the parentheses. A functional block only checks for data assignments that it requires. If a functional block does not make use of a certain data matrix it need not be assigned.

The remaining section of the data communication block is reserved for data common to all programs and currently is used for the time value between time plane 2 and time plane 1 in days. It also contains the maximum horizontal dimension in graphic digitizer units for the map on the digitizer table. These data are used for coordinating the CRT display, internal data matrices, and the graphic digitizer table. There are a total of 32 locations available for this purpose.

TABLE 3. Data Matrix Usage Assignments

<u>Data Matrix</u>	<u>Data Matrix Usage</u>
1 (MAP)	Boundary map of project for display purposes
2 (PT1)	Groundwater potential map, time plane 1
3 (PT2)	Groundwater potential map time plane 2
4 (PTA)	Average groundwater potential map, time planes 2 and 1
5 (DPT)	Difference groundwater potential map, time planes 2 and 1
6 (TRN)	Transmissivity map
7 (AQU)	Bottom aquifer map
8-20	Reserved for future use

The directives for the Iterative Monitor can be primarily divided into four major classes. These are:

1. Data Matrix Transfer-Allocation
2. Monitor Oriented Functions
3. Data Matrix Modification
4. Functional Dispatch

These classes are described below.

Data Matrix Transfer-Allocation

Four directives fall into this class:

ASG
SAV
DEL
GEN

The ASG and SAV directives require as arguments a usage tag (TAG), device (DEV), and file name (NAME) and extension (EXT). Directives using these two instructions are formatted:

ASG	TAG	DEV	NAME	EXT
SAV	TAG	DEV	NAME	EXT

The ASG directive gets a data matrix from the device (DEV), transfers it to disc unit 2 (DK2), enters that data matrix in the disc directory and makes the usage assignment to (TAG). The SAV directive takes a data matrix from disc 2 (DK2) and transfers it to the device (DEV). In both directives the data on the source device is not destroyed. If the device to which the data is being transferred already has a file with that name and extension, the old file is deleted and replaced by the new one.

The DEL directive removes a use assignment (TAG) and is formatted:

DEL	TAG
-----	-----

This simply inserts zeros into the location specified by (TAG) in the usage table.

The GEN directive sets up a new data matrix and is formatted:

GEN	TAG	NAME	EXT
-----	-----	------	-----

A data matrix is generated on disc 2 (DK2) with the name (NAME) and extension (EXT) and then assigned to use (TAG). This new matrix is filled with zeros. This directive requires additional input, however, and when this directive is given the computer will question the user for the required information. This includes the dimension of the data matrix in nodes in the horizontal and vertical directions, the number of disc sectors per data segment, and the number of nodes common to adjoining data sections. Refer to Appendix III for a discussion of these parameters and to "The Transmissivity Iterative Program User's Manual"³ for entry of these inputs.

Monitor Oriented Functions

These functions are primarily executed in the monitor and the results are passed on to the rest of the Iterative Sequence.

The first is an immediate instruction, INT, which sets the initialization of the digitizer table. When this directive is given the teletype will request the input of the origin of the data matrix on the digitizer table as well as the maximum horizontal dimension of the data distribution on the table. This information is used to coordinate the hardware and data used in the Iterative Sequence. Refer to "The Transmissivity Iterative Program User's Manual"³ for the method of entering this input.

The LST directive simply lists the current data usage assignments. The output is on the teletype and will be as shown below:

(MAP)	PROJECT BOUNDARY MAP	HANFRD MAP
(PT1)	POTENTIAL MAP TIME 1	HANFRD P72
(PT2)	POTENTIAL MAP TIME 2	HANFRD P73
* THIS IS THE NEW MAP		
(PTA)	POTENTIAL MAP (P1T1 + P1T1)/2	
(DPT)	TIME DIFFERENCE POTENTIAL (P1T2 - P1T1)	
(TRN)	TRANSMISSIVITY MAP	
(AQU)	AQUIFER MAP	
TIME	0	

The three character neumonics in () are the data usage assignment tags. Only the MAP, PT1, PT2 usage assignments have data matrices assigned to them. The PT2 assignment has some additional comments stored in the data matrix.

The COM instruction requires a data usage tag (TAG) and is formatted COM TAG. This initiates the data matrix assigned to TAG and prepares it for input of identifying comments. These comments, which are a means of placing additional identification on a data matrix, are stored directly in the data matrix in the specification block that is part of every data file. Refer to "The Transmissivity Iterative Program User's Manual"³ for a discussion of the data format. The CRT boundary map is protected so that comments cannot be entered into the display map, as it is not formatted in the same way. Refer to Appendix III for a discussion of the data format. These comments will then be listed along with the usage assignments when the LST instruction is given.

The TIM directive is formatted somewhat differently. It is entered as TIM XXXXX where XXXXX is the time value between time plane 2 and time plane 1. This number is an integer and may not be larger than 131,072 days.

Data Matrix Modification

The four directives in this class are AVE, DIF, FIL, and SMU. They perform some arithmetic operations on some data matrices.

The AVE and DIF directives are immediate; that is, they require no arguments. The data matrices specified in these directives are implied to be the potential distribution assigned to time plane 2 (PT2) and the potential distribution assigned to time plane 1 (PT1). The average potential distribution (PTA) is formed as $(PTA) = [(PT2) + (PT1)]/2$. The difference potential distribution (DPT) is formed as $(DPT) = (PT2) - (PT1)$. A data matrix must have been previously assigned to these directives' respective usage slots. They do not automatically generate data matrices to store the results.

The FIL and SMU directives require data usage slot arguments (TAG).

FIL TAG

SMU TAG

The FIL directive fills the internal zeros for the data matrix assigned to (TAG) and the SMU directive smooths the data matrix assigned to (TAG). During the FIL and SMU instruction, the data source is buffered separately from the input so all of the data receives the same amount of filling or smoothing. Figure 3 shows this.

FIGURE 3. Data Flow in Monitor Data Modification

The FIL instruction uses the following algorithm to fill the internal zeros that may occur in a data matrix as a result of errors in the input digitizing of data:

$$\begin{array}{ccccc} & & V_3 & & \\ & & & & \\ & V_1 & V_0 & V_2 & \\ & & & & \\ & & V_4 & & \end{array}$$

if $V_0 \neq 0$ there is no modification
 if $V_0 = 0$ and $V_1=V_2=V_3=V_4=0$ there is no data modification
 and $V_1 \neq 0$, or $V_2 \neq 0$, or $V_3 \neq 0$, or $V_4 \neq 0$
 then $V_0 = (V_1+V_2+V_3+V_4) / n$
 n = number of non-zero additions

The SMU instruction performs a smoothing algorithm on the data, which is of primary importance when the difference of two data matrices is formed because of the inherent numerical noise in this type of process. The data is smoothed by the following algorithm.

$$\begin{array}{ccccc} & & V_3 & & \\ & & & & \\ & V_1 & V_0 & V_2 & \\ & & & & \\ & & V_4 & & \end{array}$$

V_0 is replaced with

$$\sum_{i=0}^4 V_i / n$$

where n is the number of non-zero additions made.

Dispatch Function

There is only one directive in this class, EXQ, and it dispatches control from the Iterative Monitor to the specified functional block. The format of the directive is

EXQ FUNCTION

where FUNCTION is one of the six character neumonics for the functional blocks in the Iterative Sequence.

STREAMLINE GENERATION

This functional block performs two separate functions. It draws specified groundwater contours on the CRT display, the values of which are specified by the user at the keyboard console of the display. Simultaneously displayed on the CRT is the overlay boundary map of the system which is being analyzed. The purpose of this function is to allow the user to see the data that he has input, check for input errors, and also to give a visual reference from which to interpret results.

The second function is to generate streamlines or flow paths for the given groundwater potentials. The user selects an origin from which he desires to know a flow path. The program then begins at that origin and draws the actual flow path on the CRT screen. The path continues in the down gradient direction until a sink or boundary is found and in the up gradient direction until a source or boundary is found.

Computer Implementation

All functions performed by the Streamline functional block are applied to the data usage slot PT1, the slot allocated for the data which represents the potential at time plane 1. This is the automatic data assignment in Streamline. If another data matrix is to be evaluated by these functions in the Streamline program, that data matrix must be assigned to PT1 by the Iterative Monitor.

Contour Function

The contour function calculates lines of equal value and displays these results on the CRT. This contour value is entered from the CRT display keyboard in scientific notation format $\pm X.XXXXX\pm XX$. The contour function then searches the data and builds a visual display file. This procedure is repeated if more contours are desired.

This function reads from the disc one data section at a time (see Appendix III for discussion of data format) and systematically searches the data until an intercept of an individual contour value has been located. The coordinates are then stored in a display file. The program uses this first intercept as a starting location and searches the immediate surrounding nodes for another intercept. When a new intercept is found, this coordinate is also noted and stored in a display file. The contour is followed in this manner until a section boundary is reached or closed on

itself. The program also keeps track of all crossings for each contour line so that it will not be redrawn in future searches. Once an individual contour has been completed, the program continues to search the data section for possible additional contours of the same value until all contours have been detected and stored. The program then initializes the display file for viewing on the CRT.

During the generation of contours the program retrieves data sections in a raster scan method starting at the bottom left and moving to bottom right. It then progresses upward vertically one section and once again scans left to right, continuing in this manner until all sections have been used.

WHERE: n_h = HORIZONTAL NODES PER SECTION
 n_v = VERTICAL NODES PER SECTION
 ϕ = NUMBER OF OVERLAP NODES

= TOTAL NODES WITHIN SECTION

= TOTAL NODES MINUS OVERLAP NODES --
 NODES TO CONSIDER FOR CONTOURING.
 EXCEPTION: IF RIGHT MOST OR UPPER SECTION,
 USE ENTIRE SECTION.

FIGURE 4. Data Retrieval Format

Each time a data section is read, each word is reformatted to allow more efficient manipulation and provide room for four flags. The flags indicate the crossing of one of the four connecting lines for any point by a contour line.

The data is reduced to ≤ 14 bits from 18 bits and stored in the 0 $\rightarrow$ 13 bits in the data word.

Following is an example of two data points and an input contour value:

where: V_0 = contour value at node X_0

V_1 = contour value at node X_1

V = input contour value

$$\alpha = \frac{V - V_0}{V_1 - V_0} = \text{interpolation fraction}$$

$$X = X_0 + \alpha (X_1 - X_0)$$

Form a modified $V_i \rightarrow W_i$

$$W_i = V_i - V$$

$$\alpha = \frac{V - V_o}{V_1 - V_o} = \frac{V - V - (V_o - V)}{(V_1 - V) - (V_o - V)}$$

$$\alpha = \frac{-W_o}{W_1 - W_o} = \frac{W_o}{W_o - W_1}$$

There is an intercept IFF:

W_o and W_1 have opposite signs

Reformat Data in the File to:

$W_i = (V_i - V) * 16$ and 777760 with bit 0 a sign bit

This produces a data word structured as follows:

A parameter internal to the program is denoted as ILO. ILO is a normal part of the scan and tells the program in which direction the search is moving. The search for data within a section assumes a spiral pattern, starting at the bottom left point in the section as shown in Figure 5.

<u>DIRECTION</u>	<u>ILO</u>
Right	0
Up	1
Left	2
Down	3

FIGURE 5. Data Search Path for Contours

The four flags in the data word may be denoted as FL1, FL2, FL3, and FL4; they are stored in bits 14 → 17.

FLAGS	FL1	FL2	FL3	FL4
BITS	14	15	16	17

These flags indicate where a crossing occurred from a station point.

FLAG	DIRECTION OF INTERCEPT
FL1	Down
FL2	Left
FL3	Up
FL4	Right

The following example shows a scan pattern and the detection of an intercept for the first time.

- . Intercept on S.L. if $W_0 W_1$ is negative (i.e., W_0 and W_1 have opposite signs).
- . If intercept found, test flag bits to see if line already drawn.
- . If bit clear, draw line.
- . If bit set, continue scan.

In the above test case the intercept has not been detected before so the flag bits are set and the location of V is stored in the display file.

The intercept occurred to the right of W_0 so bit FL4 has been set. The intercept occurred to the left of W_1 so FL2 has been set for this word.

Intercept Found for First Time

Once an intercept has been found the vector (L) may take three possible directions. Following is a summary of steps to be taken:

Draw Line L: Make absolute Vector X_a

- . Mark W_0 and W_1 bits, right, left, etc....
- . Calculate X_a
- . Determine which line X_b is on
- . Calculate X_b
- . Form short vectors for $X_z \rightarrow X_b$

The directions the vectors take after an intercept has been detected are related to the scan direction ILO and can be represented by the following:

After the first vector has been detected and drawn, the following vectors depend upon the direction in which the preceding vector has entered a new box:

The object point W_0 is then moved through the matrix in this manner until the contour reaches the section boundary or closes upon itself.

Figure 6 shows the direction in which the program would move through the matrix, drawing the line segments of the contour and setting the flag bits of the data word.

FIGURE 6. Example Contour Process

Once a contour has been completed (as in Figure 4), the program returns to the original point of detection of that contour to continue the search through the matrix for any other possible contours of the same value. In this case it would return to the node labeled 1 and continue the scan to the right. Since the flags for nodes 1 and 2 have already been set for the last contour they are ignored.

The new object point W_0 is determined as the contour is followed through the matrix, as described below. The center point of Figure 7 is the present object point, and finding the next object point is dependent upon the direction the last vector has taken.

FIGURE 7. Graphical Representation of Contour Object Point

Figure 5 has been labeled according to points, lines, and quadrants; the new object point may be either 0, 1, 2 or 3. The data increments to each of these nodes from center are:

New Object Point	ΔX	ΔY
0	1	0
1	0	1
2	-1	0
3	0	-1

When a new object point has been calculated it is then positioned in the center of Figure 7 and the procedure is repeated.

Table 4 contains all of the information required for this procedure.

Streamline Function

The flow paths are generated by a gradient step technique. The gradient is calculated at a given point, namely, the point at the end of the last iteration. This gradient term is then broken into its vector components and applied to a fixed step length, providing a vector which describes the actual step from the location. This step is taken to reach the next point. This iteration is continued until the flow path is terminated.

See Appendices I and II for the technique used to calculate the gradients and the step to the next position. At the end of each iteration a display vector is calculated and put into the CRT display file. This technique is discussed in Appendix IV.

TRAVEL TIME CALCULATION

This functional block has the same capabilities as the Streamline Generation function block, namely, drawing of specific contours and calculation of a flow path. It also calculates the time of travel along the flow path.

The total travel time along the flow path is calculated by summing the travel time along the path for each step length. The length of time required to travel the distance specified by the step length is computed by a formula presented in "The Transmissivity Iterative Calculation Routine - Theory and Numerical Implementation." ² This equation uses the gradient, potential, aquifer bottom, and aquifer properties to calculate travel times. There are two equations, one for constant hydraulic conductivity and the other for a known transmissivity distribution. This version uses the transmissivity map to obtain travel time values.

Computer Implementation

The travel time functional blocks require that the following assignments be made in data usage slots:

TABLE 4. CONTOUR LOGIC PROCEDURE

<u>Intercept Line #</u>	<u>Quad of Last V.</u>	<u>Q of this V.</u>	<u>New Vector n-crossing</u>	<u>New obj. Pt.</u>	<u>ΔX_o</u>	<u>ΔY_o</u>	<u>Intercept Line New IL</u>	<u>New LQ</u>
0	30	01	1	1	0	1	3	30
0	30	01	011	1	0	1	0	30
0	30	01	010	0	1	0	1	12
0	01	30	3	3	0	-1	1	01
0	01	30	303	3	0	-1	0	01
0	01	30	300	0	1	0	3	23
1	01	12	2	2	-1	0	0	01
1	01	12	122	2	-1	0	1	01
1	01	12	121	1	0	1	2	23
1	12	01	0	0	1	0	2	12
1	12	01	010	0	1	0	1	12
1	12	01	011	1	0	1	0	30
2	12	23	3	3	0	-1	1	12
2	12	23	233	3	0	-1	2	12
2	12	23	232	2	-1	0	3	30
2	23	12	1	1	0	1	3	23
2	23	12	121	1	0	1	2	23
2	23	12	122	2	-1	0	1	01
3	23	30	0	0	1	0	2	23
3	23	30	300	0	1	0	3	23
3	23	30	303	3	0	-1	0	01
3	30	23	2	2	-1	0	0	30
3	30	23	232	2	-1	0	3	30
3	30	23	233	3	0	-1	2	12

Transmissivity Travel Time Calculations

- (PT1) Potential Time Plane 1
- (TRN) Transmissivity Distribution
- (AQU) Bottom Aquifer Distribution

Constant Hydraulic Conductivity Travel Time Calculations

- (PT1) Potential Time Plane 1

Contours and Streamlines Generation

The algorithm for generating and displaying the contours is identical to that described in the Streamline functional block. The reader is referred to those sections for a discussion of the techniques.

Travel Time Calculation

The equation for constant hydraulic conductivity and porosity is expressed as:

$$t = \frac{-\text{SIG}}{\text{PERM}} \sum_{i=1}^n \frac{\text{STEP}_i}{\text{GRAD}_i}$$

where SIG = effective porosity

PERM = hydraulic conductivity

GRAD_i = hydraulic gradient at STEP_i

STEP_i = ith step length along streamline

The variable transmissivity version requires a map of the bottom aquifer and the transmissivity distribution for the system. The algorithm used is:

$$t = -\sum_{i=1}^n \frac{\text{SIG}_i (\text{H}_i - \text{H}_0) \text{STEP}_i}{\text{TRN}_i \text{GRAD}_i}$$

SIG_i = effective porosity at STEP_i

TRN_i = transmissivity at STEP_i

H_i = hydraulic potential at $STEP_i$
 HO_i = aquifer bottom at $STEP_i$
 $STEP_i = i^{th}$ step length along streamline

TRANSMISSIVITY CALCULATIONS

This functional block provides for the generation of a transmissivity map for a given groundwater system. The map is calculated from data taken at well locations, such as groundwater potentials and transmissivity measurements. The theory of these calculations is described in "The Transmissivity Iterative Calculation Routine - Theory and Numerical Implementation."²

Instantaneous stream tubes of flow are used to generate the map. These tubes are defined by streamlines. The program assumes that the groundwater potentials are constant with respect to depth; thus it is only necessary to know the water table elevation to specify the stream tubes. Applying continuity considerations, the transmissivity can be calculated along the stream tube. The calculated transmissivity along the tube is generated in a sequential form based upon cells, each with an input equal to the output of the previous cell. The tube is generated as a string of cells.

The means of extrapolating the relatively few well transmissivity values to all of the nodes on the map is based upon the existence of regions on the map where tubes converge (Figure 8).

FIGURE 8. Convergence of Flow Tubes

A reference line is drawn across a convergence region which intersects the stream tubes. It is assumed that over a relatively short distance the transmissivity can be represented as a smooth curve. The intersected transmissivities can be plotted as a function of distance along the reference line as shown in Figure 9.

FIGURE 9. Plot of Transmissivity vs. Distance
Along Line of Intersection

With this curve, it is possible to start a tube at any node whose stream tube eventually intersects the reference line and thus calculate a transmissivity for that node using equations described in the report "The Transmissivity Iterative Calculation Routine - Theory and Numerical Implementation."² By applying this technique, a transmissivity map can be generated for an entire region on a nodal basis.

Computer Implementation

The transmissivity calculations require that the following assignments be made in the data usage slots by the monitor:

Transmissivity (Steady State)

PT1	Potential Time Plane 1
AQU	Bottom Aquifer
TRN	Transmissivity Distribution

Transmissivity (Transient)

PTA	Potential Average Time Planes 2 and 1
DPT	Potential Difference Time Planes 2 and 1
AQU	Bottom Aquifer
TRN	Transmissivity Distribution
TIME	Time Between Time Planes 2 and 1

Figure 10 shows an example of a streamtube cell iteration. The chain of cells begins at a starting point node and continues until it intersects either a source or sink, or a reference line or boundary, depending on the direction of the gradient.

FIGURE 10. Example of Transmissivity Cell Iteration

The process begins when the user puts in the point $x_0 y_0$. It is at this point that the transmissivity T_ϕ is known. The points (x'_0, y'_0) and (x''_0, y''_0) are found by taking the gradient at (x_0, y_0) and applying the initial width of the cell D_ϕ . This is expressed as

$$\Delta x = (D_\phi/2) (G_x / \text{grad})$$

$$\Delta y = (D_\phi/2) (G_y / \text{grad})$$

GRAD = magnitude of the gradient

G_x = x component of the gradient

G_y = y component of the gradient

$$x''_0 = x_0 - \Delta x$$

$$y''_0 = y_0 + \Delta y$$

$$x'_0 = x_0 + \Delta x$$

$$y'_0 = y_0 - \Delta y$$

These equations identify the starting edge for the cell. The points (x''_1, y''_1) and (x'_1, y'_1) are found by the same process used to calculate streamlines. For instance, (x''_1, y''_1) is found by generating a streamline with the step length s starting at (x''_0, y''_0) , while (x'_1, y'_1) is found in the same way, starting at (x'_0, y'_0) .

$$x''_1 = x''_0 + (G_x / \text{grad}) |_{x''_0 y''_0} * \text{STEP}$$

$$y''_1 = y''_0 + (G_y / \text{grad}) |_{x''_0 y''_0} * \text{STEP}$$

$$x'_1 = x'_0 + (G_x / \text{grad}) |_{x'_0 y'_0} * \text{STEP}$$

$$y'_1 = y'_0 + (G_y / \text{grad}) |_{x'_0 y'_0} * \text{STEP}$$

The point (x_1, y_1) is simply the midpoint of the line passing through the points (x''_1, y''_1) and (x'_1, y'_1) ,

$$x_1 = x_1'' + (x_1' - x_1''), \text{ and}$$

$$y_1 = y_1'' + (y_1' - y_1'')$$

This completely defines the cell. The transmissivity can now be calculated at the point (x_1, y_1) .

Two forms are used in calculating the transmissivity at the point (x_1, y_1) -- a steady-state version and a transient version. The steady-state version requires less data than the transient version, is simpler, and consequently much faster to operate when programmed separately. Its validity for the system must be questioned, however. The choice should be made in the initial analysis as to which version is most applicable.

The steady-state equation is expressed as follows:

$$TRN_{i+1} = TRN_i * \frac{GRAD_i * D_i}{GRAD_{i+1} * D_{i+1}}$$

This equation simply states that flow into a cell is equal to flow out of a cell. No water is stored in or released from the cell. This is portrayed graphically as Figure 11.

FIGURE 11. Transmissivity Cell (Steady State)

The transient equation is expressed as follows:

$$\text{TRN}_{i+1} = \frac{\text{TRN}_i \text{ GRAD}_i D_i}{\text{GRAD}_{i+1} D_{i+1}} + 1/2 \text{ SIG}_i \left[\left(\frac{H_{i+1,t_2} - H_{i+1,t_1}}{t_2 - t_1} \right) D_{i+1} + \left(\frac{H_{it_2} - H_{it_1}}{t_2 - t_1} \right) D_i \right] \frac{\text{STEP}_{i+1} - \text{STEP}_i}{\text{GRAD}_{i+1} D_{i+1}}$$

where:

TRN = transmissivity

GRAD_i = hydraulic gradient along stream tube

D_i = stream tube width

t_j = time j (j=1,2)

STEP_i = distance to cell i along stream tube

SIG_i = storage coefficient

This equation states that the flow into the cell is not necessarily equal to the flow out of the cell. The difference between the two is accounted for by a rate of storage or release of water in the cell. This is portrayed graphically as Figure 12.

FIGURE 12. Transmissivity Cell (Transient State)

These equations are applied at each point along the tube at the points (x_1, y_1) $1 = 1, 2, \dots, n$. This is done until the tube is terminated.

The application of this equation produces a calculated value for transmissivity at (x_1, y_1) which is predictive, depending on values of the input groundwater potentials.

The reference line is generated and stored in the computer in the following manner. A region of flow tube convergence is found by generating all of the flow tubes from well sites at which transmissivities are known. Then a convergence location is chosen and a reference line along which a smooth transmissivity curve is derived is drawn on the face of the CRT with the light pen. Once the line has been drawn those tubes which intersect that line are regenerated, terminating at the reference line. The transmissivity value at the intersection is stored in the computer. When all the intersections have been generated the computer can display a graph of those intersections, such as the one shown in Figure 13. A curve is stored in the computer by drawing a smooth curve with the light pen to fit the data. The investigator can apply his knowledge to interpolate along the curve. It is stored in the computer by a function generator which has as input the intersection point along the line and as output the transmissivity at that point. The function generator is a simple two-point, straight-line interpolation, fixed-separation type.

FIGURE 13. Plot of Predicted Transmissivity vs. Distance Along Line of Interaction

After this curve has been formed, as shown in Figure 14, it is possible to generate transmissivities for all nodes with streamlines which intersect the reference line. Because of the complex problem of locating points at the reference line to generate streamlines which would pass through each of the eligible nodes, it was decided to start at an eligible node and calculate back up the streamline to an intersection with the reference line. Then, by applying a backward calculation described in "The Transmissivity Iterative Calculation Routine - Theory and Numerical Implementation,"² it is possible to obtain the transmissivity at the beginning point (x_o, y_o). This value is then stored in the matrix of the transmissivity distribution for the area being analyzed.

These backward calculation equations for the steady-state and transient cases are:

$$\text{Steady State Case: } TRN'_o = TRN_o \frac{TRN'_{n+1}}{TRN_{n+1}}$$

$$\text{Transient Case: } TRN'_o = \frac{1}{\prod_{i=1}^{n+1} \delta_i} (TRN'_{n+1} - TRN_{n+1}) + TRN_o$$

FIGURE 14. Plot of Specified Transmissivity vs. Distance Along Line of Interaction

where

TRN_O = initial transmissivity guess at a node

TRN'_O = correct transmissivity at a node

TRN_{n+1} = calculated transmissivity at reference line

TRN'_{n+1} = transmissivity from the transmissivity curve
along the reference line

$$\delta_i = \frac{GRAD_{i+1} D_{i-1}}{GRAD_i D_i}$$

$GRAD_i$ = hydraulic gradient at step i

D_i = width of stream tube at step i

There are several ways to detect eligible nodes for a given reference line. The method chosen for implementation includes definition of a region to be searched for eligible nodes. Only nodes in this region will have a tube generated from them. If the tube leaves this defined region or does not intersect the line, that node is considered ineligible and no transmissivity is calculated for it. If the tube intersects the line, then that node is eligible and a transmissivity value is calculated for it. This region of eligibility is defined with the light pen by drawing a box on the CRT which encloses the reference line and the flow tubes that emanate from the ends of that reference line.

ACKNOWLEDGMENTS

This research was sponsored by the Advanced Technology Development Section, Research and Development Department, Atlantic Richfield Hanford Company. The authors are grateful for the assistance and direction provided by Dr. M. D. Veatch, Staff Hydrologist; D. J. Brown, Staff Geologist; and R. E. Isaacson, Manager of the Advanced Technology Development Section of Atlantic Richfield Hanford Company.

The efforts of Leila Counts, who provided editorial assistance, and Jan Greenwell, Loretta Howard and Velva Harris, who typed the many drafts and final reports, are also greatly appreciated.

REFERENCES

1. Kipp, K. L., D. B. Cearlock, A. E. Reisenauer, and C. A. Bryan, "Variable Thickness Transient Groundwater Flow Model - Theory and Numerical Implementation." BNWL-1703, December 1972.
2. Kipp, K. L. and D. B. Cearlock, "The Transmissivity Iterative Calculation Routine - Theory and Numerical Implementation." BNWL-1706, December 1972.
3. Addison, L. E., D. R. Friedrichs and K. L. Kipp, "The Transmissivity Iterative Program User's Manual." BNWL-1708, December 1972.
4. Friedrichs, D. R., "Graphic Digitizer Program to Interpolate Matrix Grid Values." BNWL-1652, December 1972.
5. Busness, K. M., "An Interactive Graphics Display System for Hydrologic Analysis." BNWL-1525, December 1970.
6. Busness, K. M., "Line Printer Interface to the PNL Hybrid Computer Facility." BNWL-B-3, August 1970.
7. Dozer, B. E., "FFTF Process Data Display Prototype." BNWL-1330, April 1970.
8. Gerhardstein, L. H., "PDP-9/Gould 4800 Device Handler and Graphics Library." BNWL-B-58, March 1971.
9. "PDP-9 User Handbook." Digital Equipment Corp., Maynard, Massachusetts, 1968.
10. Service Manual for Magnetic Disc Memory System Model No. 14120, Digital Development Corp., San Diego, California, 1969.
11. "PDP-9 Advanced Software System Monitors." Digital Equipment Corp., Maynard, Massachusetts, 1968.
12. "Utility Programs - Advanced Software System." Digital Equipment Corp., Maynard, Massachusetts, 1967.
13. Cearlock, D. B., "A Systems Approach to Management of the Hanford Groundwater Basin." BNWL-SA-3860, August 1971.
14. Gerhardstein, L. H., R. A. Burnett, D. R. Friedrichs, "PDP-9 Software System with a DDC Fixed-Head-Disc Memory." BNWL-1632, November 1971.

APPENDIX A

CALCULATION OF GRADIENT

CALCULATION OF GRADIENT

The gradient calculation is based on a Newtonian five point center differentiation formula. The formula is expressed as:

$$f'(x_0) = [f(x_{-2}) - 8f(x_{-1}) + 8f(x_1) - f(x_2)] / 12$$

This formula provides the derivative of the potential surface of a given point in one direction. Since the gradient of a two dimensional surface is expressed as:

$$\text{GRAD } [f(x,y)] = \frac{\partial f}{\partial x} \bar{u}_x + \frac{\partial f}{\partial y} \bar{u}_y$$

It is possible to find the gradient by components at a point by applying the differentiation formula in both directions. The directional component is a magnitude with the unit vector specifying the direction. These components define the gradient at a point. This algorithm then becomes for point (x_0, y_0) :

$$\text{GRAD Potential } (x_0, y_0) = \left. \frac{\partial P}{\partial x} \right|_{x_0, y_0} \bar{u}_x + \left. \frac{\partial P}{\partial y} \right|_{x_0, y_0} \bar{u}_y$$

$$\left. \frac{\partial P}{\partial x} \right|_{x_0, y_0} = [P(x_{-2}, y_0) - 8P(x_{-1}, y_0) + 8P(x_1, y_0) - P(x_2, y_0)] / 12$$

$$\left. \frac{\partial P}{\partial y} \right|_{x_0, y_0} = [P(x_0, y_{-2}) - 8P(x_0, y_{-1}) + 8P(x_0, y_1) - P(x_0, y_2)] / 12$$

The nodes used in this algorithm are located graphically:

$$\begin{array}{ccccc} & & x_0 y_2 & & \\ & & x_0 y_1 & & \\ x_{-2} y_0 & x_{-1} y_0 & x_0 y_0 & x_1 y_0 & x_2 y_0 \\ & & x_0 y_{-1} & & \\ & & x_0 y_{-2} & & \end{array}$$

An additional condition must be met. The Newtonian formula has to be applied at a given nodal position to be valid.

Most of the time the actual point where the gradient is to be evaluated (x_o, y_o) will not fall on a single specific node. However, it will be bounded by four nodes at which the derivative can be calculated.

$$\begin{array}{ccc} x_{-1}y_1 & & x_1y_1 \\ & (x_o, y_o) & \\ x_{-1}y_{-1} & & x_1y_{-1} \end{array}$$

To evaluate the gradient at this internal point (x_o, y_o) , the derivatives are evaluated at each corner position for both the x and y directional components. These values are then linearly interpolated to get the actual gradient value at the desired point. The algorithm becomes for the x directional component:

$$\text{GRAD Potential} = G_x \bar{u}_x + G_y \bar{u}_y$$

$$G_{x_1} = \left. \frac{\partial P}{\partial x} \right|_{x_1}$$

$$G_{x_2} = \left. \frac{\partial P}{\partial x} \right|_{x_2}$$

$$G_{x_3} = \left. \frac{\partial P}{\partial x} \right|_{x_3}$$

$$G_{x_4} = \left. \frac{\partial P}{\partial x} \right|_{x_4}$$

$$G_{xx_1} = G_{x_1} + (x_o - x_1) (G_{x_2} - G_{x_1})$$

$$G_{xx_2} = G_{x_3} + (x_o - x_3) (G_{x_4} - G_{x_3})$$

$$G_x = G_{xx_1} + (y_o - y_1) (G_{xx_2} - G_{xx_1})$$

The equivalent analysis is also reformed for the y directional component.

The nodal map used in this algorithm is located graphically as:

In this system, a zero groundwater potential indicates that a boundary has been reached. To allow this type of nodal pattern to approach a boundary so that calculations can continue until the boundary is included in the center square, the outer nodes are set to the value of the closest center square node. That is, if $x_6 = 0$, then x_6 is set equal to x_5 (if it is not 0). If both $x_5 = x_6 = 0$, then x_5 and x_6 are set equal to x_2 . If $x_2 = x_5 = x_6 = 0$, then the boundary has been reached. This same technique is applied in any direction as the gradient routine approaches the boundary.

APPENDIX B

CALCULATION OF STREAMLINES

CALCULATION OF STREAMLINES

The flow paths are calculated by an iterative technique, beginning at the origin and continuing to termination by either a data map boundary or a source or sink. The same algorithm is applied to each individual iteration.

x_I = current x position

x_{I+1} = next x position

y_I = current y position

y_{I+1} = next y position

GRAD = magnitude of the gradient

$\bar{G}_x$ = magnitude of the x component to the gradient

$\bar{G}_y$ = magnitude of the y component to the gradient

STEP = step length

$$GRAD = (|\bar{G}_x|^2 + |\bar{G}_y|^2)^{1/2}$$

$$x_{I+1} = x_I + (|\bar{G}_x|/GRAD) * STEP$$

$$y_{I+1} = y_I + (|\bar{G}_y|/GRAD) * STEP$$

APPENDIX C

DATA FORMATS

DATA FORMATS

DATA MAPS

A data map is a nodal digitized map of some parameter in a groundwater system. It may represent groundwater potential, bottom aquifer, transmissivity, etc. Due to the usually large number of nodes required to describe a groundwater system and the relatively small computer core in the PDP-9, the data have been formatted in a special way to make use of the disc storage available. This entails dividing the data into addressable segments on the disc. When the data in a particular segment are needed by the calculation routines they are pulled in from the disc and stored in core. Because of the speed of the disc, this swapping provides an effective means of allowing the Iterative Sequence programs to operate on almost any size groundwater system.

A data file begins with an initialization process which establishes a file on the disc containing the nodal data. This initialization includes entering the file name in the disc directory, reserving enough room to store the data, and generating a data map header block which contains the information necessary to define the configuration of the data file. Once this has been done, the data portion of the file is filled with zeros. With this completed, all programs operate on the data in the same mechanical fashion. The initialization of the data file begins by giving the file an identifying tag, including file name and extension, which is used by the Iterative Sequence to locate various data maps. This file is located on disc unit 2 (DK2) and is entered in the directory as required by the DOS. It may also be utilized by any other file-oriented program in the DOS.

The next input required is the dimensional data which specifies the configuration in which the matrix data is to be stored in the data file. The x and y dimensions in nodes of the data matrix are the first dimensional data required, and this specifies the size of the matrix. The next dimensional input is the size of a data segment which is specified in terms of the number of disc sectors in a data section. Each disc sector contains 112 words. The dimension is specified in disc sectors because the transfers are done in terms of integral units of disc sectors. An additional input parameter is the number of nodes common to adjacent data sections, or the overlap. The dimension on the overlap is primarily dictated by the numeric technique being used on the data (that is, the number of nodes surrounding the actual data point that are required to obtain the desired answer). This number becomes the dimension of the

overlap. Since one section can contain all the data required for one numerical calculation, this greatly simplifies the overhead in keeping track of the data. It also greatly reduces the number of disc-core transfers, thereby speeding up the calculation. This information is then put through an optimization algorithm to calculate the x,y dimension of the data sections which provides the least number of non-used words in a section transfer and generates the smallest number of data sections. The configuration with the minimum number of data sections describes the number of disc accesses required to evaluate a data map and the number of disc accesses is a measure of the physical time it takes to do an analysis. The optimization begins by calculating the maximum number of possible words in a data section.

$$W = 112 \times N$$

W = maximum possible number of words in data section

N = number of disc sectors per data section

In the optimization routine, the vertical dimension (V) of the data section is taken as the independent variable and the horizontal dimension (H) of the data section is taken as the dependent variable. The limits of the optimization are established on the independent variable V. The range is established on the premise $V=H$ which says the V times H = W which is a square data section. This sets the limits on V as:

$$V_{\min} = 1/2 \quad W$$

$$V_{\max} = 3 \quad V_{\min} = 3/2 \quad W$$

The optimization process then entails scanning these limits on V and using these equations to calculate the number of data sections.

$$h_i = W/v_i$$

N_{H_i} = number of horizontal data sections

$$= (H - 1 + h_i - 2 * o)/(h_i - 0 + 1)$$

N_{V_i} = number of vertical data sections

$$= (V - 1 + v_i - 2 * O) / (v_i - 0 + 1)$$

$$N_{S_i} = N_{H_i} + N_{V_i}$$

where: H = horizontal dimension of data matrix

V = vertical dimension of data matrix

O = number of nodes in overlap

This is done for all V's over the search range, and the configuration with the smallest N_{S_i} is chosen. If, for instance, two different configurations produce the same results, the one producing the largest x dimension for a data section is used because the data are stored in sequential words by x.

The manner in which the data matrix is broken up and handled is described pictorially in Figure 16. The header block occupies the first disc sector in each data file. This header block contains the input dimensional values along with some additional data which is generated in the optimization process and is formatted and accessed by the variable names and locations specified below.

Data Matrix Header Section

MX.H	Total number of horizontal nodes
MX.V	Total number of vertical nodes
MX.N	Number of disc sectors per data section
MX.D	Number of nodes in the overlap
MX.SH	Number of horizontal nodes per data section
MX.SV	Number of vertical nodes per data section
MX.NH	Number of horizontal data sections
MX.NV	Number of vertical data sections
MX.NS	Total number of data sections
MX.DS	Total number of data sections

MX.WC	Data word count per section
MX.DID	Data type
reserved	
reserved	
reserved	
reserved	

The remainder of the disc sector is used for identifying text which is entered with the Iterative Sequence Monitor.

The actual data format is in terms of one 18-bit word which can have three different interpretations: integer, scaled fix point, or logarithm. This information and any scale factor are also contained in the data matrix header block in the location labeled MX.DID.

The information is stored in the following 18-bit word format:

The TYPE is the data interpretation, which is specified by one of the following numbers:

0	Integer
1	PDP-9 scaled fix point
2	Logarithm

The S is the sign of the scale factor, 0 being positive and 1 negative. The scale factor is that number which when multiplied by the data gives the actual value of the data. All data in the matrix have been divided by the scale factor prior to storage in the matrix. The scale factor is specified as the power of 10, meaning that all scale factors are in integer units of 10.

DISPLAY MAPS

Display maps, or labeled boundary maps on which the original data are contoured and input to the data matrices, are input to the computer for use on the CRT display.

The boundary map is generated with the digitizer input routine which is operating in a CRT display output mode. This mode differs from the data mode in that the output is CRT display codes and vectors. This map is stored away in a binary code format and output to a dectape.

The monitor program can take the data from the tape and transfer them to the disc for use as a display base for the Iterative Sequence functional blocks. This map is also given a data usage assignment which each of the functional blocks requires.

When each functional block is called, the boundary is loaded into the lower portion of the core next to the resident computer operating system from the disc and displayed on the CRT as soon as the Iterative Sequence Communication Block has been interrogated. The remainder of the core between the program and the boundary map is used for holding the data matrices.

APPENDIX D

COORDINATING GRAPHIC DIGITIZER,
CRT DISPLAY, AND DATA MATRICES

COORDINATING GRAPHIC DIGITIZER, CRT DISPLAY, AND DATA MATRICES

The data matrices, graphic digitizer, and CRT display operate in different number units. The data are organized and referenced on a nodal basis by a set of (x,y) coordinates which are node numbers on the data matrix. The graphic digitizer inputs data into the computer in terms of digitizer units, binary numbers that are relative to some point on the table where the program decides to set its (0,0) reference point. This (0,0) point is set by the program and is not a hardware-fixed function of the graphic digitizer itself. So if data are to be input from the digitizer, this (0,0) point must be defined with regard to the data and it must be realized that the actual value input in terms of digitizer units is relative to that reference point. These numbers range from 1777_8 (8191_{10}) to -2000_8 (-8192_{10}) relative to the reference point. The CRT is referenced in terms of CRT display units which are absolute with the origin of the display matrix (0,0) in the lower left corner of the display. All other vectors are then specified in terms of this reference and in terms of display units. The display ranges in both horizontal and vertical directions from 0- 1777_8 (0- 1023_{10}).

Since data are input from both the graphic digitizer and the CRT, these units must be converted to nodal equivalents to reference the data matrices. Also the outputs from the various functional blocks back to the CRT for display purposes must be converted from nodal units to CRT display units. These conversions are expressed simply as:

Digitizer units to nodes

$$\text{node}_x = (\text{digitizer units } x) \frac{(\text{number of nodes in x direction})}{(\text{digitizer units equivalent to maximum x node})}$$

$$\text{node}_y = (\text{digitizer units } y) \frac{(\text{number of nodes in y direction})}{(\text{digitizer units equivalent to maximum y node})}$$

CRT units to nodes

$$\text{node}_x = (\text{display units } x) \frac{(\text{Maximum number of nodes in a single direction})}{1777_8}$$

$$\text{node}_y = (\text{display units}_y) \frac{(\text{Maximum number of nodes in a single direction})}{1777_8}$$

Nodes to CRT units

$$\text{CRT}_x = (\text{node}_x) \frac{1777_8}{(\text{Maximum number of nodes in a single direction})}$$

$$\text{CRT}_y = (\text{node}_y) \frac{1777_8}{(\text{Maximum number of nodes in a single direction})}$$

These calculations are based on the reference location for all three data planes at the point equivalent to node (0,0) in the data matrix. The origin of the nodal plane on which the data map has been drawn is concurrently the (0,0) reference point for the graphic digitizer table. The CRT's (0,0) reference point is concurrently the origin of the nodal plane on which the data map is defined. This then provides a common reference point for all three data planes.

It is also assumed that the nodal system being used is based on a square node spacing. On the graphic digitizer the scale factor or ratio of digitizer units to nodes is calculated by the number of nodes in the horizontal direction and the number of digitizer units corresponding to it. This number is input to the Iterative Monitor when the digitizer table is initialized. This assumption affects the CRT since the largest nodal direction is scaled over the entire 1024 available display units. The scale factor or ratio of display units to nodes is calculated from the largest number of nodes and this 1024 number, maximizing the coverage of the CRT screen for the best resolution. These scale factors are held constant in both the horizontal and vertical directions.

Additional effort must be employed to make sure the CRT display of streamlines is actually following the calculated path. A problem arises because the calculations end with node values which are not integer numbers. When these numbers are converted to display units, the results are in mixed numbers. However, the display requires that the vectors be specified in integer elements. This fractional part cannot be truncated or the path on the CRT will become continually less accurate in its representation of the actual calculated path. The calculations will be right, but the display will not be representative of the calculations. Since the flow path is generated with small finite elements, many iterations and vector steps are required to display the result. The problem is solved by carrying the fractional remainder when

the node is converted to display units. On the next iteration this fractional part is added to the mixed number representing the nodal position in display units. The integer part of this number is graphed on the CRT while the fractional part is held over until the next iteration. This insures that the CRT display represents the actual calculations as closely as possible.

DISTRIBUTION

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

No. of
Copies

Off-Site

3	<u>U. S. Geological Survey</u> 2100 M. Street Washington, D. C. 20037 George Debuchananne, Director M. K. Hubbert Research Geophysicist Dallas Peck
1	<u>U. S. Geological Survey</u> Menlo Park, California 94025 Jacob Rubin
1	<u>U. S. Geological Survey</u> Groundwater Branch Denver, Colorado 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> Albuquerque, New Mexico Dr. C. V. Theis
1	<u>U. S. Geological Survey</u> Water Resources Division P.O. Box 2857 Raleigh, North Carolina Ralph C. Heath District Chief

<u>No. of Copies</u>	<u>Off-Site</u>
2	<u>U. S. Bureau of Mines</u> 4800 Forbes Avenue Pittsburgh, Pennsylvania 15213 George Evans Robert W. Van Dolah
3	<u>U. S. Bureau of Mines</u> Spokane Mining Research Laboratory N. 1430 Washington St. Spokane, Washington 99201 Robert C. Bates Dan Keeley Michael M. McDonald
1	<u>U. S. Forestry Service</u> Intermountain Forest and Range Experiment Station Boise, Idaho 83702 Dr. Walter F. Meghan Research Hydrologist
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> <u>U. S. Water Conservation Laboratory</u> 4331 East Broadway Phoenix, Arizona 85040 Dr. Herman Bouwer Hydraulic Engineer

No. of
Copies

Off-Site

1	<u>USDA-ARS</u> U. S. Salinity Laboratory P.O. Box 672 Riverside, California 92502 Dr. Jan Van Schilfgaarde, Director
1	<u>USDA-ARS</u> Engineering Research Center CSU Foothills Campus Fort Collis, Colorado 80521 Dr. David A. Woolhiser Hydraulic Engineer
1	<u>USDA-ARS</u> Northwest Watershed Research Center P.O. Box 2700 Boise, Idaho 83701 Walter J. Rawls, Hydrologist
1	<u>USDA-ARS</u> P.O. Box 1096 Boise, Idaho 83701 J. L. Robins
1	<u>USDA-ARS</u> Twin Falls, Idaho M. E. Jensen
1	<u>USDA-ARS</u> Snake River Conservation Research Center Route 1, Box 186 Kimberly, Idaho 83341 Dr. Marvin E. Jensen, Director
1	<u>USDA Hydrograph Laboratory</u> <u>Plant Industry Station</u> Soils Building Beltsville, Maryland 20705 H. N. Holton, Director
1	<u>USDA-ARS</u> <u>Plant Industry Station</u> Beltsville, Maryland 20705

No. of
Copies

Off-Site

Dr. Donald L. Brakensick
Chief Engineer

1 USDA Sedimentation Laboratory
P.O. Box 30
Oxford, Mississippi 38655

A. R. Robinson, Director

1 USDA-ARS
Southern Region
Stoneville, Mississippi 38776

John C. Stephens, 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
North Appalachian Experimental Watershed
Coshocton, Ohio 43812

Dr. W. M. Edwards
Soil Scientist

1 USDA
P.O. Box 400
Chickasha, Oklahoma 73018

Harry B. Pionke

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

No. of
Copies

Off-Site

1	<u>National Academy of Science</u> Committee on Radioactive Waste Management National Research Council 2101 Constitution Avenue Washington, D. C. 20418 Dr. Cyrus Klingsberg Technical Secretary
1	<u>National Science Foundation</u> Division of Advanced Technology Application 1800 6th Street N.W. Washington, D. C. 20530 Ray Zahradnik
1	<u>U. S. Naval Civil Engineering Laboratory</u> Port Hueneme, California 93043 Dr. James J. Der
2	<u>Robert S. Kerr Water Research Center</u> P.O. Box 1198 Ada, Oklahoma 74820 Dr. J. W. Keeley Marion R. Scalf
1	<u>Alabama Geologic Survey</u> P.O. Drawer O University, Alabama 35486 Phil Lammereaux State Geologist
1	<u>Southwest Rangeland Watershed Research Center</u> 442 E. Seventh Street Tucson, Arizona 85705 Dr. R. Smith
1	<u>California Dept. of Water Resources</u> Sacramento, California Helen J. Peters

No. of
Copies

Off-Site

1	<u>Idaho Bureau of Mines & Geology</u> <u>University of Idaho</u> Moscow, Idaho 83843 Dr. Roy E. Williams
1	<u>Idaho Dept. of Water Administration</u> <u>State House, Annex 2</u> 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> <u>P.O. Box 232</u> Urbana, Illinois 61801 Keros Cartwright
1	<u>State Geological Survey of Kansas</u> <u>University of Kansas</u> Lawrence, Kansas 66044 Dr. John C. Halepaska Chief, Water Resources Section
1	<u>State of Maine</u> <u>Environmental Improvement Commission</u> Augusta, Maine 04330 Steven D. Freedman
1	<u>St. Anthony Falls Hydraulic Laboratory</u> <u>Mississippi River at 3rd Avenue East</u> Minneapolis, Minnesota 55414 Professor C. Edward Bowers
2	<u>Desert Research Institute</u> <u>Center for Water Resources Research</u> Reno, Nevada Dr. Clinton Case G. B. Maxey

No. of
Copies

Off-Site

1	<u>State of North Carolina</u> <u>Groundwater Division</u> Water and Air Resources Raleigh, North Carolina 27611 Harry M. Peek, Chief
1	<u>Water Quality Management Laboratory</u> <u>Rt. 2, Box 322A</u> Durant, Oklahoma 74701 V. L. Hauser
1	<u>State of Oregon</u> <u>516 Public Service Bldg.</u> Salem, Oregon 97310 Chris Wheeler State Engineer
1	<u>Texas Water Development Board</u> <u>Systems Engineering Division</u> P.O. Box 13087 Austin, Texas 78711 Lial F. Tischler
1	<u>Agencies for Environmental Conservation</u> <u>Montpelier, Vermont 05602</u> Dr. Martin L. Johnson Secretary
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

<u>No. of Copies</u>	<u>Off-Site</u>
1	<u>Computer Sciences Corp.</u> <u>Federal Building</u> Richland, Washington 99352 R. W. Nelson
1	<u>Sandia Laboratories</u> <u>Division 5166</u> Albuquerque, New Mexico 87115 Fred Norwood
1	Arthur M. Piper #3 Sonoma Lane Carmel, California 93921
1	<u>Alyeska Pipeline Service Co.</u> <u>2100 Travis Street</u> Houston, Texas 77002 Dr. Maurice D. Veatch
1	<u>E. I. duPont deNemours & Co., Inc.</u> <u>Savannah River, South Carolina</u> J. C. Corey
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>Dept. of Agriculture Chemistry and Soils</u> Tucson, Arizona 85721 Dr. Gordon Dutt
2	<u>University of Arizona</u> <u>Dept. of Hydrology and Water Resources</u> 200-E Old Psychology Tucson, Arizona 95721 Dr. A. K. Tyagi Dr. Chester C. Kisiel

No. of
Copies

Off-Site

2	<u>University of Arizona</u> Tucson, Arizona 95721 E. S. Simpson A. W. Warrick
2	<u>University of California</u> Berkeley, California 94720 Dr. Warren Kauffman P. A. Witherspoon Dept. of Civil Engineering
3	<u>University of California at Davis</u> Dept. of Water Sciences and Engineering Davis, California 95616 J. N. Luthin Don Nielson Dr. Verne H. Scott
1	<u>University of California at Riverside</u> College of Biological & Agricultural Sciences Riverside, California 92507 Dr. N. T. Coleman
3	<u>Stanford University</u> Department of Civil Engineering Stanford, California 94305 Paul Kruger Henry J. Ramey J. B. Franzini
1	<u>Stanford University</u> School of Earth Sciences Dept. of Geology Stanford, California 94305 E. Aguado

<u>No. of Copies</u>	<u>Off-Site</u>
1	<u>Georgia Institute of Technology</u> <u>School of Civil Engineering</u> Atlanta, Georgia 30332 Dr. J. R. Wallace
1	<u>University of Idaho</u> <u>Moscow, Idaho 83843</u> Dr. Delbert Fitzsimmons Head, Agricultural Engineering
2	<u>University of Illinois</u> <u>Urbana, Illinois 61801</u> A. Klute Professor Ven Te Chow Hydrosystems Laboratories
1	<u>Purdue University</u> <u>Dept. of Agricultural Engineering</u> Wast LaFayette, Indiana 47907 Emru D. Millette
1	<u>Iowa State University</u> <u>Ames, Iowa</u> D. Kirkham
1	<u>University of Kansas</u> <u>Water Resources Institute</u> Lawrence, Kansas 66044 Don W. Green
1	<u>University of Kentucky</u> <u>Department of Mechanical Engineering</u> Lexington, Kentucky 40506 C. T. Hahn
1	<u>Harvard University</u> <u>School of Public Health</u> Boston, Massachusetts 02138 Dr. D. W. Moeller

No. of
Copies

Off-Site

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 Geology and Geophysics 2D Pillsbury Hall Minneapolis, Minnesota 55455 Hans Olaf Pfannkuch
1	<u>University of Minnesota</u> Department of Civil Engineering Minneapolis, Minnesota 55418 H. Steforn
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 & 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

<u>No. of Copies</u>	<u>Off-Site</u>
1	<u>State University of New York at Buffalo</u> <u>Dept. of Mechanical Engineering</u> Buffalo, New York 14214 Ralph T. Cheng
1	<u>State University of New York</u> <u>Dept. of Civil Engineering</u> Buffalo, New York 14214 Dr. Ralph Rumer
2	<u>Cornell University</u> <u>School of Civil Engineering</u> B22 Bailey Hall Ithaca, New York 14850 W. H. Brutsaert Winton Covey
2	<u>Oregon State University</u> <u>Corvallis, Oregon 97331</u> Dr. Larry Boersma Professor of Soils Dr. Royal H. Brooks Professor of Agricultural Engineering
1	<u>Drexel Institute of Technology</u> <u>Philadelphia, Pennsylvania 19104</u> Irwin Remson
1	<u>Vanderbilt University</u> <u>Nashville, Tennessee 37203</u> Dr. Frank L. Parker
1	<u>Utah State University</u> <u>College of Engineering</u> <u>Dept. of Agriculture and Irrigation</u> <u>Engineering</u> Logan, Utah 84321 Dr. L. G. King

<u>No. of Copies</u>	<u>Off-Site</u>
1	<u>Utah State University</u> Utah Water Research Laboratory College of Engineering Logan, Utah 84321 J. Paul Riley
1	<u>Utah State University</u> Logan, Utah 84321 Dr. Roland W. Jeppson Associate Prof. of Civil Engineering
1	<u>Utah State University</u> Logan, Utah 84321 C. G. Clyde
1	<u>University of Wyoming</u> Laramie, Wyoming 82070 Dr. Paul A. Rechard Director, Water Resources Research Institute
2	<u>Washington State University</u> Dept. of Civil Engineering 148 Sloan Hall Pullman, Washington 99163 Dr. Donald L. Bender Dr. David Higgins
2	<u>Washington State University</u> Dept. of Agronomy & Soils Pullman, Washington 99163 Dr. Walter H. Gardner Dr. Eugene Thompson
1	<u>Washington State University</u> Albroom Hydraulic Laboratory Pullman, Washington 99163 Dr. John F. Orsborn, Head

<u>No. of Copies</u>	<u>Off-Site</u>
1	<u>University of Washington</u> Department of Geology Seattle, Washington 98105 Dr. Howard Coombs
2	<u>University of Wisconsin</u> Department of Soil Science Madison, Wisconsin 53706 W. R. Gardner P. Saffigna
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> <u>School of Civil Engineering</u> Sydney, Australia Dr. K. K. Watson
1	<u>Institut für Geophysik</u> <u>Technische Hochschule</u> Karlsplatz 13 A-1040 WIEN Österreich Austria A. E. Scheidegger
1	<u>University of British Columbia</u> <u>Vancouver, B. C., Canada</u> Dr. Allen Freeze
1	<u>Dept. of Mines and Natural Resources</u> <u>Water Resources Branch</u> 693 Taylor Avenue Winnipeg 9, Manitoba, Canada Frank W. Render

<u>No. of Copies</u>	<u>Off-Site</u>
1	<u>University of Waterloo</u> Department of Earth Science Waterloo, Ontario, Canada Richard G. Jackson
1	<u>University of Waterloo</u> Department of Civil Engineering Faculty of Engineering Waterloo, Ontario, Canada John F. Pickens
1	<u>McMaster University</u> Department of Civil Engineering Hamilton, Ontario, Canada Dr. William James
2	<u>Ontario Water Resources Commission</u> <u>135 St. Clair Avenue West</u> Toronto 7, Ontario, Canada Donald N. Jeffs Bob Ostry
1	<u>Department of the Environment</u> <u>4905 Dufferin Street</u> Downsview, Ontario, Canada Dr. Aloysius K. Lo
2	<u>Department of the Environment</u> <u>C-244, No. 8 Temporary Bldg.</u> Inland Waters Branch Ottawa, Canada R. E. Jackson Dennis Lawson
1	<u>Institut de Mecanique (I.M.G.)</u> <u>Domaine Universitaire</u> B. P. 53 Centre de Tri 38041 Grenoble-Cedex, France Georges Vachaud

No. of
Copies

Off-Site

1	<u>M. R. Engineering College</u> <u>Jaipur-302004, India</u> Dr. R. D. Verma Prof. of Civil Engineering
1	<u>University College</u> <u>Upper Merrion Street</u> <u>Dublin, 2 Ireland</u> Dr. James C. I. Dooge Professor of Civil Engineering
1	<u>Department of Mechanical Engineers,</u> <u>I.I.T. Kharagkur,</u> <u>S.E. Ry, India</u> R. G. Mokadom
3	<u>Technion-Israel Institute of Technology</u> <u>Haifa, Israel</u> J. Bear Shragga Irmay Dan Zaslavsky
1	<u>Agricultural Research Organization</u> <u>The Volcani Center</u> The Institute of Soils and Water P.O.B. 6 Bet Dagan, Israel Shlomo P. Neuman
1	<u>Food & Agriculture Organization of the UN</u> <u>Land and Water Development Division</u> Water Resources & Development Service Via Delle Terme Dl Caracalla 00100-Rome, Italy R. G. Thomas
1	<u>Kyoto University</u> <u>Kyoto, Japan</u> Yoriteru Inoue

No. of
Copies

Off-Site

1 Explorade En Produktie Laboratorium
 c/o Koninklijke/Shell
 Volmerlaan 6
 Ritswijk (Z.H.), The Netherlands

A. M. Schaaps

1 University of Canterbury, N.Z.
 Dept. of Civil Engineering
 Christ Church 1, New Zealand

Bruce W. Hunt

On-Site Hanford

1 AEC/R1 Patent Attorney

R. M. Poteat

7 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

69 Atlantic Richfield Hanford Co.

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

No. of
Copies

On-Site Hanford

3

Westinghouse Hanford Co. HEDL

R. B. Hall

29

Battelle-Northwest

D. B. Cearlock
J. P. Corley
J. R. Eliason
K. L. Kipp
D. E. Olesen
A. E. Reisenauer
D. D. Tillson
R. C. Routson
R. J. Serne
D. R. Friedrichs
S. W. Ahlstrom
R. G. Baca
A. Brandstetter
R. D. Mudd
R. W. Wallace
J. J. Fuquay
A. J. Shuckrow
L. E. Addison
Tech. Information
Tech. Pubs.

8

United Nuclear, Inc.

C. D. Corbit
F. E. Owen