PREPROCESSOR AND POSTPROCESSOR COMPUTER PROGRAMS FOR A RADIAL-FLOW, FINITE-ELEMENT MODEL

By Amleto A. Pucci, Jr. and Daryll A. Pope

U.S. GEOLOGICAL SURVEY

Open-File Report 87-680

West Trenton, New Jersey

1987

DEPARTMENT OF THE INTERIOR

DONALD PAUL HODEL, Secretary

U.S. GEOLOGICAL SURVEY

Dallas L. Peck, Director

For additional information
write to:

U.S. Geological Survey
Mountain View Office Park
810 Bear Tavern Rd
Suite 206
West Trenton, New Jersey 08628

Copies of this report
can be purchased from:

U.S. Geological Survey
Books and Open-file Reports
Box 25425, Federal Center, Bldg. 810
Denver, Colorado 80225

CONTENTS

Page

Abstract.. 1
Introduction.. 1
Background... 1
Purpose and scope.. 1
Radial-flow model... 2
Preprocessor program 4
Preprocessor concepts...................................... 7
Preprocessor algorithms.................................... 8
Division of subregions................................. 8
Node numbering... 8
Division into triangles...............................14
Preprocessor input and output.............................14
General input requirements............................15
"P" operation mode and output 15
"R" operation mode and output..........................15
Postprocessor and modified program.............................16
Postprocessor concepts....................................16
Postprocessor algorithms..................................18
Calculation of observation-well drawdowns.............18
Calculation of the Gaussian norm......................18
General postprocessor input and output description.......18
Postprocessor input...................................19
Postprocessor output..................................20
Summary..20
Selected references..21

APPENDIXES

Appendix 1. MESH Code..25
2. Formats for input files..................................37
Section A. BASIC DATA FILE...........................37
Section B. PROPERTIES................................39
Section C. OB.WELL...................................41
Section D Observation well data file.................42
3. Example of BASIC DATA FILE for the six subregion example mesh shown in figure 3.................................43
4. Data files used for example simulation shown in figure 4....44
Section A. BASIC DATA FILE for generating figure 4.....44
Section B. PROPERTIES data file for region shown in figure 4...............................47
Section C. OB.WELL data file used to generate figure 5...............................48
Section D. Observed field-data file used to generate figure 5...............................48
5. Example of OUTPUT file from MESH P mode for example shown in figure 3.....................................49
6. RADFLOW-S Code...53

ILLUSTRATIONS

Page

Figure 1. Conceptual model for a radially-symmetric ground-water flow system.. 3
2. An example of a finite-element mesh for a radial ground-water flow region.................................... 5
3. Schematic diagrams of transformation between global and local reference frames, and the division of a subregion into triangles.. 9
4. Sample problem region with subregions, lithology, and final calibrated hydrogeologic properties....................11
5. Plot of simulated and observed drawdowns for sample problem shown in figure 2..................................17

TABLES

Table 1. Global node numbers, and number of column and row increments for each subregion in figure 3(a).....................13
2. Connectivity matrix for the subregion mesh shown in Figure 3(a).......................................13

CONVERSION FACTORS AND ABBREVIATIONS

For use of readers who prefer to use metric (International System) units, conversion factors for the inch-pound terms used in this report are listed below:

Multiply Inch-Pound Unit	By	To Obtain Metric Unit
	Length	
inch (in.)	25.4	millimeter (mm)
foot (ft)	0.3048	meter (m)
	Area	
square mile (mi^2)	2.590	square kilometer (km^2)
	Volume	
gallon (gal)	.003785	cubic meter (m^3)
	Flow	
foot per day (ft/d)	0.3048	meter per day (m/d)
foot squared per day (ft^2/d)	0.09294	meter squared per day (m^2/d)
gallon per minute (gal/min)	0.000063	cubic meter per second (m^3/s)
gallon per minute per foot [(gal/min)/ft]	0.2070	liter per second per meter [(L/s)/m]

PREPROCESSOR AND POSTPROCESSOR COMPUTER PROGRAMS FOR A RADIAL-FLOW, FINITE-ELEMENT MODEL

By Amleto A. Pucci, Jr. and Daryll A. Pope

ABSTRACT

Preprocessing and postprocessing computer programs that enhance the utility of the U.S. Geological Survey radial-flow model have been developed. The preprocessor program (1) generates a triangular finite-element mesh from minimal data input, (2) produces graphical displays and tabulations of data for the mesh, and (3) prepares an input data file to use with the radial-flow model.

The postprocessor program is a version of the radial-flow model, which was modified to (1) produce graphical output for simulation and field results, (2) generate a statistic for comparing the simulation results with observed data, and (3) allow hydrologic properties to vary in the simulated region.

Examples of the use of the processor programs for a hypothetical aquifer test are presented. Instructions for the data files, format instructions, and a listing of the preprocessor and postprocessor source codes are given in the appendixes.

INTRODUCTION

Background

Reilly (1984) developed a Galerkin Finite-Element Model (RADFLOW) to simulate ground-water flow in a radially-symmetric aquifer. The model can be used to simulate the response of an aquifer to pumping (Lindner and Reilly, 1983). Finite-element and node data are input manually, and output data are created as arrays by the RADFLOW model. This approach can be tedious and can introduce errors into the spatial discretization (finite-element mesh). In this report, new procedures are introduced to minimize data-entry errors and output manipulation.

Purpose and Scope

The purpose of this report is to present the preprocessor and postprocessor algorithms and computer programs that have been written for the radial, ground-water flow model, and to demonstrate the use of these computer programs. The preprocessor, called MESH, is a Fortran computer code, that contains algorithms for generating a triangular mesh, and creates and formats input data files for the ground-water flow model. The postprocessor program, named RADFLOW-S, is a modification of the Reilly (1984) code. Post-processing operations produce graphical displays and statistical evaluations of water-level changes in observation wells in the simulated region. This report includes (1) a general explanation of the preprocessor and postprocessor concepts and algorithms; (2) a general summary of the MESH, and RADFLOW-S codes, as well as a listing of these codes; and(3) descriptions of how to use the preprocessors and postprocessors with input and output examples.

RADIAL FLOW MODEL

The user needs to have some familiarity with the geometry of the radial-flow regions, and the assumptions and requirements imposed on the preprocessing by the original radial-flow model to understand the preprocessor program. The flow region is radially symmetric, with a discharging well at the center of the region. Pumping is assumed to have little effect on water levels at large distances from the pumping well. This condition is approximated in the numerical simulation as a constant-head boundary at the lateral boundary of the flow region. The lower boundary of the flow region must be a no-flow boundary. The upper boundary may be either an impermeable boundary (no-flow boundary), or a free surface (water-table condition) (Reilly, 1984). Because of the radial symmetry, the hydrogeologic units in the radial-flow system are modeled as layers with uniform hydrologic properties. A conceptual model of such a region is shown in figure 1. Reilly (1984) describes the model assumptions and the geometric configuration of the region in more detail in the original program documentation.

The original radial-flow model code of Reilly (1984), requires the user to manually discretize the radial-aquifer region into three-node triangular elements, each with uniform hydraulic properties. The computer code simulates the radial flow in the discretized region by generating a system of algebraic equations derived from the Galerkin finite-element approximation of the differential equations of two-dimensional, radially-symmetric ground-water flow (Reilly, 1984; p. 5, eq. 13). Input data required to solve the system of equations include a number for each triangular finite-element, the finite-element node numbers, the location of each node in the region, and the hydraulic properties associated with each element.

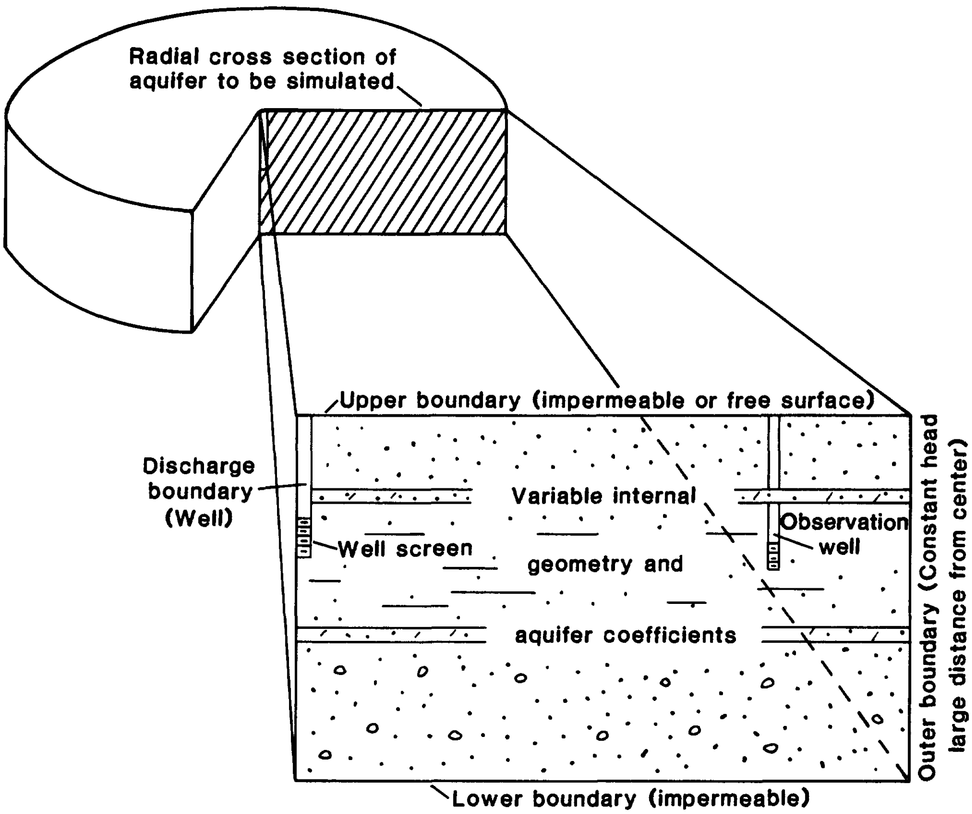

Figure 1.--Conceptual model for a radially symmetric ground-water flow system. (Modified from Linder and Reilly, 1983, fig. 6, p. 21)

PREPROCESSOR PROGRAM

The preprocessor program, MESH, automatically generates the finite-elements for the radial-flow finite-element program, RADFLOW-S. It generates a finite-element mesh from a minimum amount of information supplied by the user. Figure 2 is a representative example of a ground-water flow region that has been divided into triangular elements using MESH. The user inputs data that (1) defines zones of different hydraulic properties; (2) identifies sources or sinks, and the flow boundaries of the radial ground-water region; (3) creates a finite-element mesh of the dimensions needed to meet the required simulation accuracy; (4) generates triangular shapes of reasonable form that do not cause numerical difficulties; and (5) automatically numbers nodes for the region. Reilly (1984) gives a description of constraints in designing a mesh for a radial flow system, which is beyond the scope of this report. Pinder and Gray (1977), Steinmueller (1974) and many other authors discuss numerical stability and design of meshes used for finite-element analysis. A listing of the preprocessor code is given in appendix 1.

page 7 follows

Preprocessor Concepts

The automatic, mesh-generating algorithm uses eight-node, quadratic, rectangular, finite elements. These rectangular finite elements are a family of two-dimensional finite elements that are different from the triangular finite elements that are generated for the numerical computations of the radial-flow model. Each rectangular finite element is defined by the program user as a subregion, which represents a radial section in the ground-water flow region. The horizontal and vertical extent of the subregions are determined from the hydrogeologic units in the radial-flow region. Generally, one or more rows of subregions are assigned to each hydrogeologic unit.

Examples of how rectangular elements are used to represent subregions is schematically shown for the simple region in figure 3(a). This region is divided into six rectangular, finite-elements. The node positions for the quadratic, rectangular elements in figure 3(a) are in the global reference frame, r,z, where "r" represents the radial distance from the well, and "z" the vertical elevation in relation to datum. In this example, the rectangular element that represents subregion 5 in the global reference frame has a range in "r" from 0 feet to 200 feet, and for "z" the range is 0 feet to 100 feet.

The mesh-generating program uses a local reference frame in the algorithm that divides each rectangular element representing a subregion into triangular finite elements. Zienkiewicz (1977) and Segerlind (1976) cover the concept of local reference frame in detail. The rectangular element, which represents the subregion in the r, z (global) reference frame, is transformed to a square element in the u, v (local) reference frame, where both u and v range from -1 to 1. This reference frame is centered within the boundaries of each subregion, as shown in figure 3(b). Once a subregion is divided, the coordinates are transformed back to the global coordinates that are required to solve the numerical equation for radial flow (Reilly, 1984; p.5, eq. 13). The local reference frame is used for the dividing operations because of the convenience of the identical geometry for each subregion. The tranformation is done automatically using quadratic rectangular basis functions. The equations to transform a point "k" from the local reference frame coordinates, u,v, to the global reference frame, r,z, are:

$$R_i = \sum_{j=1}^{8} N_j \times R_j,$$

and

$$Z_i = \sum_{j=1}^{8} N_j \times Z_j,$$

where R_j, Z_j are the global coordinates for the eight nodes, j=1,...,8, for the quadratic, rectangular, finite element,

R_i, Z_i are global coordinates for point k transformed to the global reference frame,

N_j are the quadratic, rectangular basis functions for the eight nodes, and j=1,...,8 for the quadratic, rectangular, finite element in the local reference frame in counter-clockwise order, beginning with the lower left corner node.

Figure 4 shows a radial-flow region of 4 hydrogeologic units, 50 subregions, and 181 global nodes. These subregions were divided into the 480 finite-element, 273 finite-element node mesh in figure 2. (For sake of clarity, the eight nodes in each rectangular element that represents a subregion are referred to as global nodes. The nodes the program generates in the triangular mesh are called the finite-element nodes.) In this example, alternating layers of sand and clay are delineated into rows of subregions using rectangular elements. The subregion discretization is much coarser than the generated finite-element mesh. The algorithm requires the identification of a primary hydrogeologic unit, and in this example it is the lower sand unit. Note that the discretization intervals between nodes along the horizontal sides of each rectangular finite element is centered in the logarithmic scale, and is not centered in a linear scale. This consideration adds to the numerical stability of the radial-flow model solver (Reilly, 1984).

Preprocessor Algorithms

Division of Subregions

The preprocessor algorithm proceeds by dividing a subregion into simple rectangles, then into triangular finite elements as in fig. 3(b) through 3(d). These operations are completed for each subregion before numbering finite-element nodes and then proceeding to the next subregion. The algorithm requires: (1) the global locations of the eight nodes of the rectangular, finite element that represent each subregion, and (2) the number of row and column increments to be made for each subregion. The transformed subregion in figure 3(c) is divided into eight simple rectangles, by specifying five column increments along the u-axis and three row increments along the v-axis. The intersection of each row and column in the divided subregion is the location of a finite-element node, which is given a node number and assigned to a triangular finite element in subsequent operations in the code. Zienkiewicz and Phillips (1971) originated this algorithm, which Segerlind (1976) also used.

Node Numbering

The node numbering algorithm begins when the user discretizes the flow region into subregions. The radial-flow solver requires the lowest triangular finite-element node numbers to be along the constant-head boundary (Reilly, 1984). Because the automatic finite-element node-numbering algorithm is based on the sequence of subregions, constraints are made on the numbering of subregions by the user. The numbering algorithm requires the lowest subregion numbers to be in sequence, top to bottom, along the constant-head boundary. As an example, subregion 1 and 2 are the two subregions on the constant-head boundary in figure 3(a). A number for each subregion, and a global node number for each of the eight nodes in the

page 11 follows

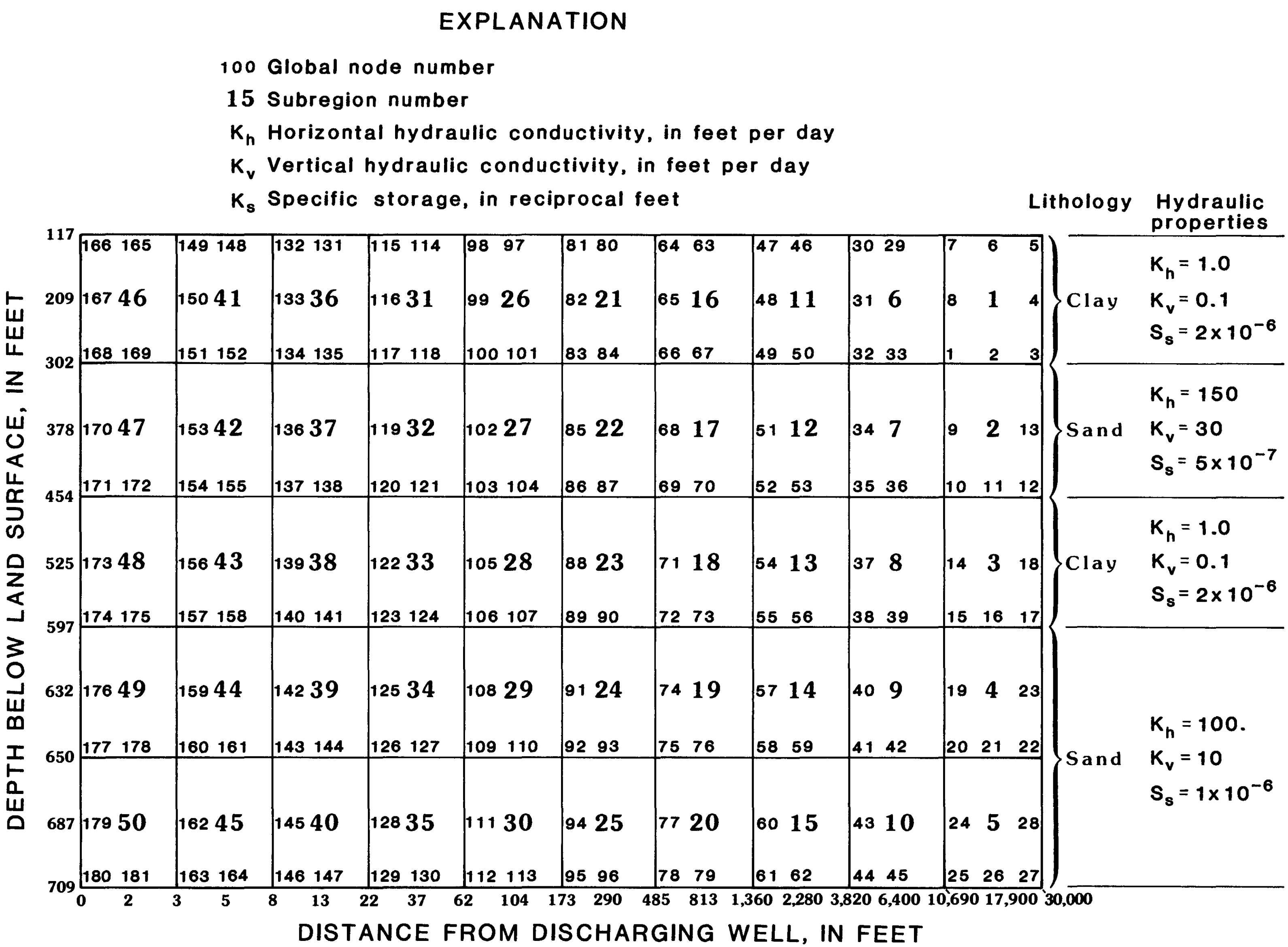

Figure 4.--Sample problem region with subregions, lithology, and final calibrated hydrogeologic properties.

rectangular element representing that subregion, are assigned by the user. Table 1 illustrates the assignment of global node numbers for the subregions in figure 3(a). Global node numbers can be in any order and are not the same as the triangular finite-element numbers to be generated by the algorithm.

The algorithm determines the number of finite-element nodes created in a subregion from the number of row and column increments specified by the user in each subregion. Each subregion must be divided into the same number of rows, if the subregions share a boundary along a vertical side, or divided into the same number of columns, if the shared boundary is along a horizontal side. Columns 2 and 3 in table 1 list these values for the example given in figures 3(a)-3(e). For example, subregion 1 has three rows of increments, and subregion 2 has five rows of increments.

The user must assign finite-element node numbers to finite-element nodes on the constant-head boundary. The number of finite-element nodes on this boundary is equal to the total number of row increments for each subregion along the constant-head boundary minus the number of shared rows. For the region shown in figure 3(a), three-row increments are assigned for subregion 1, and five-row increments are assigned for subregion 2; these subregions share one boundary. Therefore, seven finite-element nodes are located along this boundary (5 + 3 row increments - 1 row shared between subregion 1 and 2). For the example in figure 3, nodes 1 through 7 are assigned on the constant-head boundary.

Automatic numbering is done by the code after a subregion is divided into columns and rows, before creating the triangular finite elements in the subregion, and before proceeding to the next subregion to repeat the series of operations. Information on the spatial arrangement of the subregions is needed for the algorithm to avoid renumbering the finite-element nodes along shared subregion boundaries. This information is stored in the "connectivity matrix".

The numerical function of the connectivity matrix is demonstrated by example. Assembly of the connectivity matrix for the region in figure 3(a) is shown in table 2. This region consists of six subregions, numbered 1 through 6. Each subregion can have one neighboring subregion on each of its four sides. The orientation of each of the four sides is numbered 1 through 4. Side 1 is below, and sides 2, 3, and 4 are numbered counter-clockwise. The connectivity matrix identifies the subregion number on each shared side for each subregion, as shown in table 2 for figure 3(a). A zero is used where the subregion side is not shared. As indicated by table 2, subregion 5 in figure 3(a) is bounded by subregion 3 to the right (side 2), and by subregion 6 below (side 1); sides 3 and 4 are not shared.

The algorithm assigns node numbers to the corners of the simple rectangles formed from the intersection of the columns and rows in each divided subregion, row by row, from left to right, and top to bottom. The algorithm searches the connectivity matrix for each subregion to determine if nodes along its boundaries have been numbered previously. For subregion 5 in figure 3(a) and in table 2, the numbering algorithm proceeds through the subregions in numerical order and the search of the connectivity matrix determines that subregion 3 is divided and numbered, and that the

Table 1.--Global node numbers, and number of column and row increments for each subregion in figure 3(a).

Subregion number	Number of row increments	Number of column increment	Global node numbers associated with each subregion							
1	3	3	1	2	3	4	5	6	7	8
2	5	3	10	11	12	13	3	2	1	9
3	3	3	17	18	1	8	7	14	15	16
4	5	3	20	21	10	9	1	18	17	19
5	3	5	25	26	17	16	15	22	23	24
6	5	5	28	29	20	19	17	26	25	27

Table 2.--Connectivity matrix for the subregions shown in figure 3(a).

Subregion number	Shared boundary subregion number			
	Side 1	Side 2	Side 3	Side 4
1	2	0	0	3
2	0	0	1	4
3	4	1	0	5
4	0	2	3	6
5	6	3	0	0
6	0	4	5	0

nodes shared by subregion 5 and subregion 3 are numbered. Therefore, these finite-element nodes are not renumbered. However, because subregion 6 has not been divided and numbered, the nodes it shares along the boundary with subregion 5 are assigned. Node numbers on each boundary of the subregion are stored in an array. Segerlind (1976) and Collins (1973) discuss node numbering and illustrate the use of the connectivity matrix in more detail.

Division into Triangles

After numbering finite-element nodes and before the next subregion is divided, the algorithm divides each simple rectangle formed from the intersection of columns and rows in each subregion into two triangles as illustrated in figure 3(d). The creation of triangles, proceeds left-to-right, top-to-bottom, through all the simple rectangles in each subregion. The algorithm determines the proper numbering for diagonals from the simple rectangle, assigns the node sequence for each triangular element to produce the consistent counter-clockwise order needed to assemble the matrices in the numerical radial-flow equation (Reilly, 1984; p. 5, eq. 13), and gives each triangle a finite-element number. Reilly (1984) discusses the node-numbering conventions. Global coordinates for each node are calculated for each triangular finite-element node using the transformation operation described in the section on preprocessor concepts. The bandwidth is determined for each triangular-element. If it is the largest bandwidth in the region, it is recorded and printed in the final tabulation of results.

In summary, the preprocessor algorithms generate the finite-element mesh, one subregion at a time. The algorithm to divide each subregion into simple rectangles uses the geometry of a local reference frame. The user determines the resolution of the mesh by specifying the number of row and column increments. The number of simple rectangles produced for each subregion is determined from the number of assigned rows and columns for each subregion. The corners of these simple rectangles become the finite-element nodes, and each node is numbered. Each simple rectangle is divided into two triangular finite elements, and then each triangular element is numbered. The global locations of the finite-element nodes are determined by transforming the coordinates from local to global coordinates. These finite-element-mesh data are assembled with hydrogeologic data as described in the following section for input to the radial-flow model.

Preprocessor Input and Output

The MESH program can be used in two different modes, the "P" and "R" modes. The "P" mode is used to produce a plot of the finite-element mesh at a graphics terminal or to create a plotfile, which can be plotted. It also produces a descriptive output file called OUTPUT, which can be used instead of, or in conjunction with, a plotter to evaluate the generated finite-element mesh. It is not necessary to use either a graphics terminal or plotter to run the MESH program. Once the mesh geometry has been evaluated by the user, the "R" mode can be used to automatically create the properly formatted input data file to run RADFLOW-S.

General Input Requirements

The MESH program requires input from the BASIC DATA FILE in either the "P" or "R" mode. Following the order that data are input, the BASIC DATA FILE includes data on (1) the number of subregions and total number of global nodes in the modeled region, (2) the radial distance from the well head to the global nodes, (3) the vertical distance of the global nodes relative to a datum elevation, (4) subregion numbers and the connectivity matrix data, (5) the number of subregions and finite-element node numbers along the constant-head boundary, (6) the number of row increments for each subregion along the constant-head boundary, and (7) the global node numbers in each subregion and the number of row and column increments for each subregion.

A description of the input-data variables and formats for the BASIC DATA FILE is given in appendix 2. Sample input data, which produced the finite-element mesh shown in figure 3(e) from the subregions shown in figure 3(a), is presented in appendix 3. The BASIC DATA FILE for the region in figure 4 is presented in appendix 4, section A. This data set was used to produce the 480 finite-element mesh shown in figure 2.

"P" Operation Mode and Output

In "P" operation, the program prompts the user to choose the output device. The program uses the graphics device utility (GDIU) to facilitate graphical output. A graphics terminal compatible to the TAB 132/15-G[1] or TEKTRONICS 4010 terminals or plotting devices at the users installation may be chosen, or a plotfile can be created for input to another graphics plotter. Visual examination of the graphical output of the finite-element mesh can quickly show problems in the global-mesh data. Errors in the input data set can be corrected by referring to the OUTPUT file descriptions and by examining the graphical output.

The descriptive file OUTPUT is created in the "P" operation mode. The OUTPUT file includes (1) a tabulation of the connectivity matrix for all the subregions, (2) a list of the number of row and column increments in which each subregion is divided, (3) the finite-element node numbers for each subregion, and (4) the finite-element numbers and the associated finite-element node numbers, and the global coordinates of these finite-element nodes. The program prompts the user at the terminal to enter the maximum length of the "r"and "z" axes, and a choice of either an arithmetic or logarithmic plotting scale along the "r" axis. An example of the OUTPUT file containing mesh data for the simple region shown in figures 3(e) is given in appendix 5.

"R" Operation Mode and Output

The PROPERTIES data file and the BASIC DATA FILE are required input to run MESH in the "R" mode. The BASIC DATA FILE was previouly described.

---

[1] "Use of brand names in this report is for identification purposes only and does not constitute endorsement by the U.S. Geological Survey."

PROPERTIES data are formatted in four sections. The first data section contains control data for setting up the simulation input data file. It contains (1) a simulation identifier, (2) identification of the hydraulic properties of the primary hydrogeologic unit in the region, (3) the number of hydrogeologic units that are distinct from the primary hydrogeologic unit, (4) the number of finite-element nodes along the screen in the discharging well (must be more than 1), and (5) the number of pumping periods. The second data section specifies the finite-element node numbers along the screen of the discharging well, and, therefore, is determined after evaluating the output on the mesh from "P" mode operation. The third data section assigns the hydraulic properties for those hydrogeologic units that the user has identified as distinct from the primary hydrogeologic unit. For the example shown in figure 4, one primary hydrogeologic unit and three distinct units were identified. The fourth data section gives information on the test-well discharge rate and duration.

A description of the PROPERTIES data-file input variables and their formats is given in appendix 2, section B. Further discussion of the function of these hydraulic properties to simulate a radial-flow system is described in the original RADFLOW report (Reilly, 1984).

MESH produces the RADFLOW.INPUT data file as output, which is used by the RADFLOW-S radial-flow model. The RADFLOW.INPUT data file is identical to the input file documented by Reilly (1984). The RADFLOW.INPUT file must be manually modified as described in the original report to simulate an aquifer test of an unconfined system (Reilly, 1984).

POSTPROCESSOR and MODIFIED PROGRAM

This section summarizes the concepts and modifications that are in the RADFLOW-S version but are not included in the original code radial-flow model. The original RADFLOW model determined water levels at each finite-element node at each time step in a summary output file. RADFLOW-S retains the summary output file as part of the output from the radial-flow simulations. The modified code is presented in appendix 6.

Postprocessor Concepts

The summary output file from the original model consists of a printout of large matrices from simulations, which can be can be unwieldly. To improve this, the innovation used in the RADFLOW-S program was to focus on only that portion of the computed results that are typically used in the analysis--that is, the computed drawdown in observation wells within the radial-flow region.

The postprocessor performs two functions: (1) it creates standard logarithmic plots from the observation-well water-level changes from the finite-element simulation and from the field data, and (2) it evaluates a Gaussian-norm statistic that compares the simulation results and field data at each observation well. An example of the logarithmic plot produced by the program is given in figure 5. Modifications to the hydrogeologic data and mesh can be made as iterative feedback to improve the match of simulation and field data.

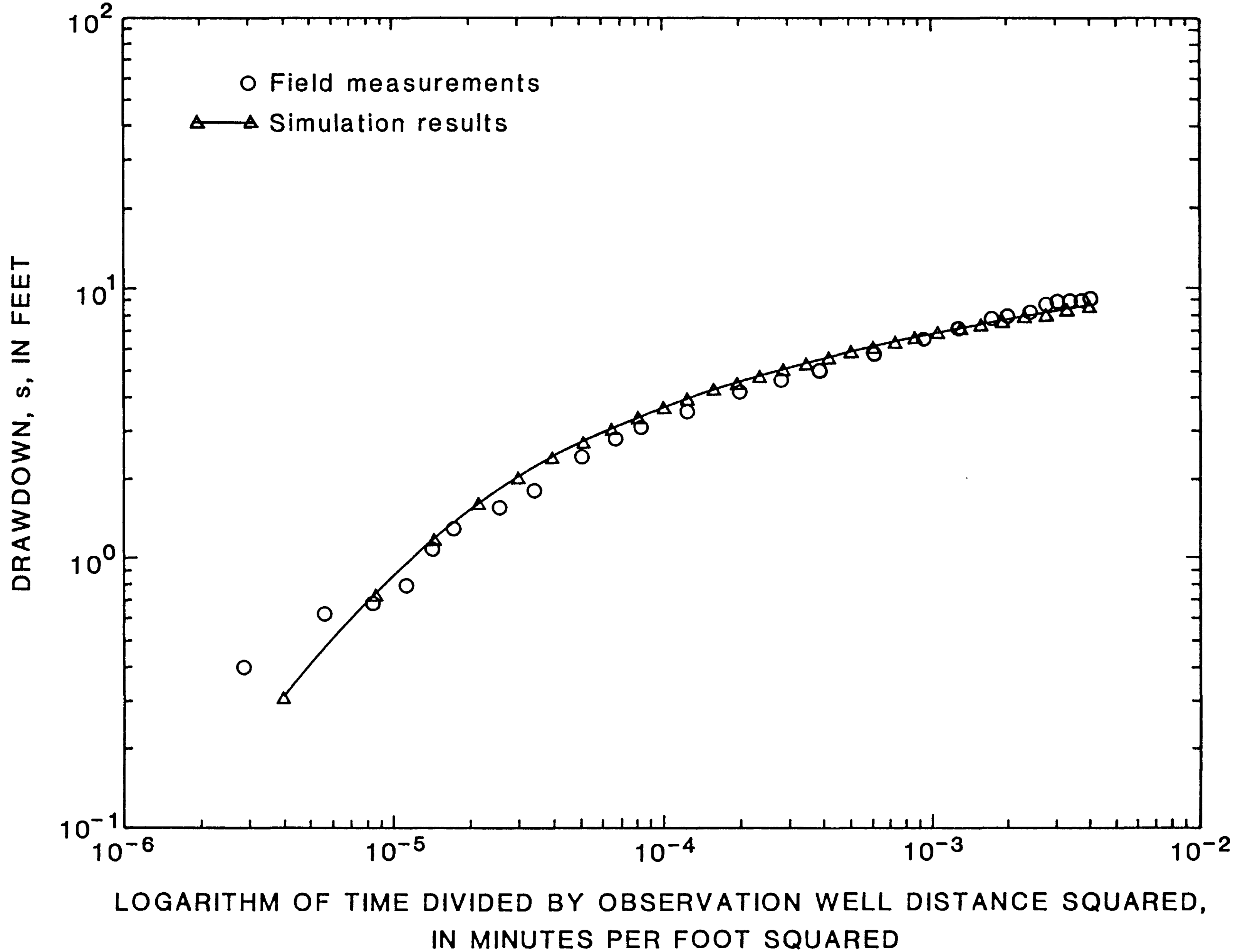

Figure 5.--Plot of simulated and observed drawdowns for example simulation shown in figure 2.

Postprocessor Algorithms

Calculation of Observation-well Drawdowns

As in the original model, the RADFLOW-S model calculates the ground-water potential, or head, for each triangular finite-element node in the mesh for each time step of the simulation period (Reilly, 1984). The postprocessor program uses triangular basis functions of the same type as in the numerical solution to the radial-flow equation (Reilly, 1984; p.4, eq. 7) to interpolate heads in observation wells from the node heads. The method of numerical interpolation using linear-triangular basis functions is basic and is described in detail in Segerlind (1976) and Zienkiewicz (1977).

Calculation of the Gaussian Norm

The Gaussian-norm statistic, "e", is a general comparison of the adequacy of the simulated drawdown and the observed drawdown for each observation for the entire period of simulation. It is expressed by the formula:

$$e = \sum_{i=1}^{n} \frac{h_i - h'_i}{h_i}$$

where h_i = is the observed head at the observation well point, i, at time t.

h'_i= is the calculated head at the observation well point, i, at time t.

n = number of observed head values at the observation well point.

For example, where the observed and calculated heads are approximately the same value, the Gaussian-norm statistic will approach zero, indicating that the field and simulation results are in good agreement. This comparison is done once for each observation well. Because the simulated values usually are calculated for times that are different from the times of the observed field data, the algorithm does a linear interpolation to determine a simulated value at the same time as the field data.

General Postprocessor Input and Output Description

The program, RADFLOW-S, is a modified version of the radial-flow finite-element program code, RADFLOW, with the addition of postprocessing subroutines, which are called PTPLOT, PMPLT, and EPSILON. PTPLOT and PMPLOT subroutines generate logarithmic graphs of water-level changes versus time, and EPSILON calculates the Gaussian-norm statistic. PTPLOT prompts the user for input to control the plotting routines for water-level plots of each observation well, and calls the PMPLOT and EPSILON subroutines. PMPLOT prompts the user to name the input file that contains the drawdown data for each observation well, then reads these data files and generates plots of corresponding field and simulated drawdown data. More detailed descriptions of these subroutines and data files follows.

Postprocessor Input

The RADFLOW-S program reads data both through user responses to prompts at the terminal and by reading data files. Subroutine PTPLOT prompts the user at the terminal for (1) the name of the file containing the field water-level data, (2) the headings for the logarithmic water-level graphs, and (3) axes limits for the graph. The graph axes limits must be given as log values that span the range of the data as shown in figure 5. The x-axis is in units of time per length squared, (time per square of the radial distance of the observation well from the pumped well), and the y-axis is in feet. Subroutine PTPLOT calls the GDIU and prompts the user to specify an output device to either output directly to a graphics terminal, or to create a plotfile for a plotter. The program plots the simulated water-level data in the order that the observation wells are included in the OB.WELL data file; therefore, the user follows the same order when responding to prompts for field data and plotting control.

The three datafiles required to run the program are: (1) the RADFLOW.INPUT data file, previously described in the section on the 'R' option for the preprocessor, MESH; (2) the OB.WELL data file, which contains data on the number of observation wells and their screen locations; and (3) user generated data files containing measured field data for each of the observation wells.

The OB.WELL data file lists the number of observation wells in the region, the location of the midpoint of the screen of each observation well, and the finite-element number in the mesh that contains the observation-well screen. The location of the observation well(s) in the finite-element mesh is determined by the user after running and reviewing the graphics output and the output tabulation file OUTPUT generated by MESH in "P" mode. The observation well location is assumed to be the midpoint of the total screen length. Drawdowns in up to five observation wells can be calculated. The requirement for observation-well data to run RADFLOW-S can be overridden by creating an empty data file. Further descriptions of the data file variables and the data formats are presented in appendix 1, section C. The OB.WELL data file used to create the plot shown in figure 5 is presented in appendix 4, section C.

Data files of field data on each observation well must be separate and must be given a label by the user. Each data file lists the number of water-level measurements for the observation well, the distance of the observation well from the pumped well, a prepumping reference water level (optional), and the water-level measurement and recorded time at each observation well. A dummy reference water-level value is used in place of a prepumping water-level if water-level measurements are entered as the drawdown (or recovery) value. Appendix 2, section D describes the input formats for these data. The data file for the observation well example shown in figure 5 is given in appendix 4, section D.

Postprocessor Output

Postprocessor output of logarithmic graphs of water-level changes may be plotted to a terminal screen or to another device by creating a plot file. The value of the Gaussian-norm statistic is shown on the user's terminal at the completion of the logarithm plot for each observation well. The code creates a summary output file in the format originally documented by Reilly (1984).

Summary

Preprocessing and postprocessing computer programs have been developed that simplify and enhance the utility of the U.S. Geological Survey radial-flow model. Principals of automatic mesh generation and data manipulation have been used in the programming. Features of the preprocessor include: (1) reduced user input for generating the finite-element mesh for the radial-flow model, (2) use of global and local reference-frame geometry to discretize the flow region, (3) a capability for the user to simply describe the spatial variation of hydrologic properties (horizontal and vertical hydraulic conductivity, and specific storage or specific yield) of subregions in the flow region, (4) automatic division of the flow region into triangular finite-elements, (5) automatic numbering of the finite-element nodes in the discretized region, and (6) generation of an input file to run the radial-flow model, RADFLOW-S.

The postprocessing program is part of the simulation code for the radial-flow system. The postprocessor features (1) interpolation of the water-level changes at observation wells for the simulated region; (2) plotting simulated water-level changes and field data from observation wells together for as many as five observation wells; and (3) calculation of a Gaussian-norm statistic, which compares the simulation to field results.

In addition to the program descriptions, discussion of the concepts in these processors and examples of their use for a hypothetical aquifer test are presented. Appendixes provide format instructions for the data files, example data files, and a listing of the preprocessor and postprocessor source codes.

SELECTED REFERENCES

Collins, R.J., 1973, Bandwidth reduction by automatic renumbering: International Journal for Numerical Methods in Engineering, Vol. 6, p. 345-356.

Lindner, J.B., and Reilly, T.E., 1983, Analysis of Three Tests of the unconfined aquifer in Southern Nassau County, Long Island New York: U.S. Geological Survey Water-Resources Investigations Report 82-4021, 46 p.

Pinder, G.F., and Gray, W.G., 1977, Finite element simulation in surface and subsurface hydrology: Academic Press, New York, 295 p.

Reilly, T.E., 1984, A Galerkin finite-element flow model for the transient response of a radially symmetric aquifer: U.S. Geological Survey Water-Supply Paper 2198, 60 p.

Segerlind, L.J., 1976, Applied Finite Element Analysis: John Wiley & Sons, New York, 422 p.

Steinmueller, G., 1974, Restrictions in the Application of Automatic Mesh Generation Schemes by 'Isoparametric' Co-ordinates: International Journal for Numerical Methods in Engineering, Vol. 8, p. 289-294.

Zienkiewicz, O.C., 1977, The Finite Element Method: 3rd. Ed., McGraw-Hill Book Company (UK) Limited, London, U.K. 787 p.

Zienkiewicz, O.C., and Phillips, D.V., 1971, An Automatic Mesh Generation Scheme for Plane and Curved Surfaces by 'Isoparametric' coordinates: International Journal for Numerical Methods in Engineering, v. 3, p. 519-528.

APPENDIXES

page 25 follows

```
C******************************************************************************
C                                  APPENDIX 1
C
C                                   MESH Code
C
C
C*******************************************************************************
C*  MESH.F77
C*  WRITTEN BY AMLETO A. PUCCI AND DARYLL A. POPE, USGS TRENTON, NJ.  6/19/87
C*
C*  Program MESH.F77 is a finite-element mesh generator and a pre-
C*  processor program for running RADFLOW-S.F77, a Galerkin Finite-Element
C*  model for simulating transient responses to aquifer tests. MESH.F77
C*  automatically partitions second-order, quadrilateral, finite elements, or
C*  "subregions", into linear triangular elements which may define a
C*  multilayer, radially-symmetric, ground-water aquifer. It also can be used
C*  in a pre-processor mode to create an input data-file for running
C*  RADFLOW-S, which is a modified version of a program by Thomas Reilly
C*  (1984).
C******************************************************************************
c*
C*  The program employs DISSPLA graphics , and is compatible with TAB graphics
C*  terminals, and ZETA type plotters on the Prime system.
C*
C*
C*  Mesh Generating Mode:
C*  The program reads a data file which is needed to define the simulation
C*  domain. The input data includes the number of quadrilateral subregions,
C*  the number of regional nodes, the coordinates of the regional nodes,
C*  connectivity data (or data which describes how the sub-regions are posi-
C*  tioned next to each other), the number of subregions which appear at the
C*  constant-head boundary of the domain, the number of nodes on the constant-
C*  head boundary, the number of rows in each subregion, and information on
C*  how each subregion is to be divided into triangular elements. An
C*  annotated input file for mesh generating appears in Appendix 1.
C*
C*  The program outputs a descriptor file, OUTPUT, which describes the
C*  division of each subregion into triangular elements, and the numbering
C*  and positions of each triangular element node. Graphics output can also be
C*  used to examine the mesh.
C*
C*  RADFLOW-S Input generation:
C*  The program reads the same data file which is used to generate the mesh,
C*  and a second data file which must be titled "PROPERTIES" . The second data
C*  file of parameters needed to simulate an aquifer test as described in
C*  Appendixl.B of the supplemental data for running MESH.F77 and in WSP 2198.
C*
C*
```

```
C***************************************************************************
C*
C****************************************************************************
C*
C*                  DEFINITION OF SELECTED PROGRAM VARIABLES
C*
C*     NB      -counter used for assigning finite element nodes
C*     NBW     -counter for calculating maximum bandwidth
C*     NELBW   -number of the triangular element causing the maximum bandwidth
C*     NEL     -counter for the total number of elements generated
C*     O2      -logical for output control
C*     INRG    -number of quadrilateral subregions used to define the problem
C*              domain
C*     INBP    -total number of global nodes used to discretized the domain
C*              into quadrilateral subregions
C*     XP,YP   -the global coordinates of the regional nodes
C*     JT      -connnectivity data for each subregion
C*     NRC     -number of subregions which appear on the constant-head boundary
C*              of the domain
C*     TNR     -total number of element nodes which are located on the
C*              constant-head boundary
C*     NRB     -the number of constant-head nodes which appear in each
c*              subregion
C*     NNRB    -the node numbers on each side of a subregion, defined
C*              NNRB(subregion,side,node number)
C*     NELR    -counter for the number of elements in a subregion
C*     NROWS   -number of rows of nodes to divide a subregion into
C*     NCOL    -number of columns of nodes to divide a subregion into
C*     NDN     -global node numbers used to define the quadrilateral subregions
C*     N       -second order quadrilateral shape functions associated with
C*              each regional node
C*     XC,YC   -the x and y coordinates of all the generated regional nodes
C*     ICOMP   -array of digital data which determines the orientation of
C*              contiguous subregions
C*     NN      -temporary array used for numbering element nodes in a subregion
C*     XE,YE,  -the X and Y coordinates, and the node numbers of the four
C*     NE       node elementary quadrilateral which is divided into two
C*              triangles
C*     NTF     -logical file used to determine if the node has been assigned
C*     NODXY   -file of the X and Y positions for each node used for RADFLOW
C*     ELM     -node numbers for each triangular element
C***************************************************************************
C*
      PROGRAM MESH
      INTEGER NDN(8),NN(21,21),NNRB(50,4,21),JT(50,4),LB(3),NE(25)
     1 ,NR(4),NRB(10),NRC,TNR,NELR,REGION(50,32),ELM(800,3)
     1 ,JA(3),NELRC(50)
      REAL XP(200),YP(200),N(8),YC(21,21),XC(21,21),XE(25)
     1,YE(25),XRG(9),YRG(9),NODXY(1000,2)
      CHARACTER FILES*1,DATAFILE*32,ANS*1
      LOGICAL NTF(50)
      LOGICAL O2
      COMMON NODXY,ELM
      COMMON REGION,NELRC
```

```
C*******************************************************************************
*
C       DATA INITIALIZATION
C*******************************************************************************
*****
      DATA NE/25*0/
      DATA ((NN(I,J), I=1,21),J=1,21)/441*0/
      DATA ((NODXY(I,J),I=1,1000),J=1,2)/2000*0/
      DATA NTF/50 * .FALSE./
      NB=0
      NBW=0
      NELBW=0
      NEL=0
      O2 = .FALSE.
      CALL TNOUA('ENTER DATA FILE: ',INTS(17))
      READ(1,10) DATAFILE
10    FORMAT(A32)
      OPEN(UNIT=40,FILE=DATAFILE,STATUS='OLD')
      OPEN(UNIT=90,FILE='OUTPUT',STATUS='NEW')
C
C**************************************************************************
C INPUT FROM SCREEN FOR CONTROL OF FILES TO BE OUTPUT
C**************************************************************************
20    WRITE(1,30) 'DO YOU WISH TO PLOT THE MESH OR TO GENERATE'
      WRITE(1,30) 'THE INPUT FILES FOR USE IN RADFLOW.F77?'
      CALL TNOUA('ENTER P FOR PLOTTING OR R FOR RADFLOW:',INTS(38))
30    FORMAT(A48)
      READ(1,40) FILES
40    FORMAT(A1)
      IF (FILES .EQ. 'P') THEN
        O2 = .NOT. O2
       ELSE IF (FILES .EQ. 'R') THEN
        GO TO 50
       ELSE
        GO TO 20
50    ENDIF
C*********************************************************************
C   INPUT AND OUTPUT OF TITLE,CONTROL CARD,GLOBAL COORDINATES,AND
C   CONNECTIVITY DATA
C**********************************************************************
      READ(40,60) INRG,INBP
60    FORMAT (2I4)
      READ(40,70)(XP(I),I=1,INBP)
      READ(40,70)(YP(I),I=1,INBP)
70    FORMAT(8F10.5)
      DO 80 I=1,INRG
80     READ(40,90)NRG,(JT(NRG,J),J=1,4)
90    FORMAT (5I3)
      WRITE(90,100)
100   FORMAT(//1X,17HCONNECTIVITY DATA/
     1,39H                        SHARED BOUNDARY/1X,
     142HSUBREGION           1      2      3      4  /)
```

```
C100   FORMAT(//1X,39HCONNECTIVITY DATA       SHARED BOUNDARY/1X,
C     143HSUB REGION            1      2      3      4  )
      DO 110 I=1,INRG
110    WRITE(90,120)I,(JT(I,J),J=1,4)
120   FORMAT(2X,I3,16X,4(I2,5X))
C***************************************************************************
C   INPUT OF NODE NUMBERS ALONG CONSTANT-HEAD BOUNDARY
C***************************************************************************
      READ(40,130) NRC,TNR
130   FORMAT(2I3)
      DO 140 I=1,NRC
140    READ(40,150) NRB(I)
150   FORMAT (I3)
      NB=TNR
      K=2
      DO 160 I=1,NRC
       IM=I-1
       NNRB(I,2,1)=NNRB(IM,2,NRB(IM))
       DO 160 J=2,NRB(I)
        NNRB(I,2,J)=K
        K=K+1
160   CONTINUE
      NNRB(1,2,1)=1
C**************************************************************************
C  LOOP ON THE SUBREGIONS TO GENERATE THE ELEMENTS
C**************************************************************************
      DO 420 KK=1,INRG
       NELR=0
       READ (40,170)NRG,NROWS,NCOL,(NDN(I),I=1,8)
170    FORMAT(11I4)
      IF (NRG .NE. 1) THEN
      WRITE(90,175)
175   FORMAT(/30X,41H*NOTE: COORDINATES ARE MULTIPLIED BY .001)
      ENDIF
       WRITE(90,180) NRG,NROWS,NCOL,(NDN(I),I=1,8)
180    FORMAT(1H1///1X,
     164H****  ARRANGEMENT OF GLOBAL NODE AND FINITE ELEMENT NODE NUMBERS
     1,13H IN SUBREGION
     1,I2,6H  ****//22HNUMBER OF INCREMENTS =//10X,I2,11H ROWS   AND,4X
     1,I2,8H COLUMNS//10X,19HGLOBAL NODE NUMBERS,10X,8I5)
C**************************************************************************
C  GENERATION OF THE ELEMENT NODAL COORDINATES
C**************************************************************************
       DO 190 I=1,8
        II=NDN(I)
        XRG(I)=XP(II)
190     YRG(I)=YP(II)
       XRG(9)=XP(1)
       YRG(9)=YP(1)
       TR=NROWS-1
       DETA=2./TR
       TR=NCOL-1
       DSI=2./TR
       DO 200 I=1,NROWS
```

```
      TR=I-1
      ETA=1.-TR*DETA
      DO 200 J=1,NCOL
       TR=J-1
       SI=-1.+TR*DSI
       N(1)=-0.25*(1.-SI)*(1.-ETA)*(SI+ETA+1.)
       N(2)=0.5*(1.-SI**2)*(1.-ETA)
       N(3)=0.25*(1.+SI)*(1.-ETA)*(SI-ETA-1.)
       N(4)=0.50*(1.+SI)*(1.-ETA**2)
       N(5)=0.25*(1.+SI)*(1.+ETA)*(SI+ETA-1.)
       N(6)=0.50*(1.-SI**2)*(1.+ETA)
       N(7)=0.25*(1.-SI)*(1.+ETA)*(ETA-SI-1.)
       N(8)=0.50*(1.-SI)*(1.-ETA**2)
       XC(I,J)=0.0
       YC(I,J)=0.0
       DO 200 K=1,8
        XC(I,J)=XC(I,J)+XRG(K)*N(K)
200     YC(I,J)=YC(I,J)+YRG(K)*N(K)
C*********************************************************************
C  GENERATION OF THE SUBREGION NODE NUMBERS
C*********************************************************************
       KN1=1
      KS1=1
      KN2=NROWS
      KS2=NCOL
      DO 270 I=1,4
       NRT=JT(NRG,I)
       IF(NRT.EQ.0.OR.NRT.GT.NRG)GO TO 270
       DO 210 J=1,4
210     IF(JT(NRT,J).EQ.NRG)NRTS=J
       K=NCOL
       IF(I.EQ.2.OR.I.EQ.4)K=NROWS
       JL=1
       DO 260 J=1,K
        GO TO(220,230,240,250),I
220     NN(NROWS,J)=NNRB(NRT,NRTS,JL)
        KN2=NROWS-1
        GO TO 260
230     NN(J,NCOL)=NNRB(NRT,NRTS,JL)
        KS2=NCOL-1
        GO TO 260
240     NN(1,J)=NNRB(NRT,NRTS,JL)
        KN1=2
        GO TO 260
250     NN(J,1)=NNRB(NRT,NRTS,JL)
        KS1=2
260     JL=JL+1
270    CONTINUE
       IF(NRG .GT. NRC)GO TO 290
       KS2=NCOL-1
       DO 280 I=1,NROWS
280     NN(I,NCOL)=NNRB(NRG,2,I)
```

```
290     DO 300 I=KN1,KN2
         DO 300 J=KS1,KS2
         NB=NB+1
300      NN(I,J)=NB
C************************************************************************
C  STORAGE OF THE BOUNDARY NUMBERS
C**************************************************************************
        DO 310 I=1,NCOL
         NNRB(NRG,1,I)=NN(NROWS,I)
310      NNRB(NRG,3,I)=NN(1,I)
        DO 320 I=1,NROWS
         NNRB(NRG,2,I)=NN(1,I)
320      NNRB(NRG,4,I)=NN(I,1)
C ************************************************************************
C  OUTPUT OF THE SUBREGION NODE NUMBERS
C****************************************************************************
        WRITE(90,330)
330     FORMAT(//1X,27HFINITE-ELEMENT NODE NUMBERS/)
        DO 340 I=1,NROWS
340      WRITE(90,350)(NN(I,J),J=1,NCOL)
350     FORMAT(1X,20I5)
C*************************************************************************
C  DIVISION INTO TRIANGULAR ELEMENTS
C**************************************************************************
       WRITE (90,355)
355    FORMAT (//5X,49HELEMENT NUMBERS, FINITE-ELEMENT NODE NUMBERS, AND
      117H NODE COORDINATES)
        WRITE(90,360)
360    FORMAT(//3X,17HNEL  NODE NUMBERS,9X,4HX(1),8X,4HY(1),8X,4HX(2),8X,
      14HY(2),8X,4HX(3),8X,4HY(3)     )
        K=1
        DO 370 I=1,NROWS
         DO 370 J=1,NCOL
          XE(K)=XC(I,J)
          YE(K)=YC(I,J)
          NE(K)=NN(I,J)
370      K=K+1
        L=NROWS-1
        DO 410 I=1,L
         DO 410 J=2,NCOL
         NR(1)=NCOL *I+J-1
         NR(2)=NCOL *I+J
         NR(3)=NCOL *(I-1)+J
         NR(4)=NCOL *(I-1)+J-1
         DO 410 IJ=1,2
          NEL=NEL+1
          NELR=NELR+1
          REGION(KK,NELR)=NEL
          J1=NR(1)
          J2=NR(IJ+1)
          J3=NR(IJ+2)
          JA(1)=J1
          JA(2)=J2
          JA(3)=J3
```

```
         LB(1)=IABS(NE(J1)-NE(J2))+1
         LB(2)=IABS(NE(J2)-NE(J3))+1
         LB(3)=IABS(NE(J1)-NE(J3))+1
         DO 380 IK =1,3
          IF(LB(IK).LE.NBW) GO TO 380
            NBW=LB(IK)
            NELBW=NEL
380         CONTINUE
C*****************************************************************************
*
C      CREATION OF THE ARRAY NODXY(1000,2)
C       FOR USE IN OUTPUT TO RADFLOW
C*****************************************************************************
*
         IF (NTF(KK) .EQV. .TRUE.) GO TO 390
          DO 390 II=1,3
           NODXY(NE(JA(II)),1)=XE(JA(II)) * 1000.
           NODXY(NE(JA(II)),2)=YE(JA(II)) * 1000.
390       CONTINUE
C********     WRITING DATA TO FILE OUTPUT
      WRITE(90,400)NEL,NE(J1),NE(J2),NE(J3),XE(J1),YE(J1),XE(J2),YE(J2)
     1,XE(J3),YE(J3)
400   FORMAT(1X,4I5,3X,6F12.4)
C********     CREATION OF ARRAY ELM(800,3) NEEDED TO PRODUCE 'RADFLOW.INPUT'
         ELM(NEL,1)=NE(J1)
         ELM(NEL,2)=NE(J2)
         ELM(NEL,3)=NE(J3)
410    CONTINUE
       NTF(KK) = .NOT. NTF(KK)
       NELRC(KK)=NELR
C*****************************************************************************
*
C      OUTPUT OF SELECTED FILES
C*****************************************************************************
*
420   CONTINUE
C
C  FINISH WRITING TO FILE 'OUTPUT'
C
      WRITE(90,430) NBW,NELBW
430   FORMAT(///1X,21HBANDWIDTH QUANTITY IS,I4,22H CALCULATED IN ELEMENT
     *,I4)
      CLOSE(90)
      CLOSE(40)
C*****************************************************************************
*
      IF (O2 .EQV. .TRUE.) THEN
        CALL PLOT(NEL,NB)
       ELSE
        CALL RADIN(NEL,NB,TNR)
       ENDIF
      STOP
      END
      SUBROUTINE RADIN(NEL,NB,TNR)
```

```
      INTEGER NND(50),DIFFK,REGION(50,32),ELM(800,3),NELRC(50),RN,RD
     *,NCH,NQ,NDIF,NSCON,NTS,TNR
      REAL NODXY(1000,2),PPR,PPZ,S,SY,PR,PZ,Q,QRCH,DELT,TMAXF,TSM
      COMMON NODXY,ELM
      COMMON REGION,NELRC
C
      OPEN(UNIT=80,FILE='PROPERTIES',STATUS='OLD')
      OPEN(UNIT=100,FILE='RADFLOW.INPUT',STATUS='NEW')
C********  READ BASIC VARIABLES FROM 'PROPERTIES' AND WRITE TO 'RADFLOW.INPUT'
      READ(80,10) TITLE
      WRITE(100,10) TITLE
10    FORMAT (T1,A25)
      READ(80,20) PPR,PPZ,IPP
      WRITE(100,20) PPR,PPZ,IPP
20    FORMAT(F10.0,F10.2,I10)
      READ(80,30) S,SY
      WRITE(100,30) S,SY
30    FORMAT(F10.9,3X,F10.9)
      READ(80,40) NQ,NDIF
40    FORMAT(T16,2I5)
      WRITE(100,50) NEL,NB,TNR,NQ,NDIF
50    FORMAT(5I5)
      DO 70 I=1,NQ
       READ(80,60) NND(I)
       WRITE(100,60) NND(I)
60     FORMAT(I5)
70    CONTINUE
C******   READ VALUES FOR SUBREGIONS OF DIFFERING HYDRAULIC CONDUCTIVITIES
C******   FROM 'PROPERTIES' AND WRITE TO 'RADFLOW.INPUT'
      READ(80,80)DIFFK
80    FORMAT (I3)
      DO 120 I=1,DIFFK
        READ(80,90)PR,PZ,B,DIFFS
90      FORMAT (F10.5,F10.5,I4,F10.9)
        DO 120 J=1,B
         READ(80,100) RN
100      FORMAT(I4)
         C=NELRC(RN)
         DO 120 K=1,C
          WRITE(100,110) REGION(RN,K),PR,PZ,DIFFS
110       FORMAT (I10,F10.5,F10.5,5X,F10.9)
120    CONTINUE
C********      WRITE ELEMENT DATA TO 'RADFLOW.INPUT'
      DO 130 I=1,NEL
130    WRITE(100,140) I,ELM(I,1),ELM(I,2),ELM(I,3)
140   FORMAT (4I5)
      NSCON=1
C********      WRITE NODAL COORDINATES TO FILE 'RADFLOW.INPUT'
      DO 170 I=1,NB
       IF (I .LE. TNR) THEN
        WRITE(100,150) I,NODXY(I,1),NODXY(I,2),NSCON
150     FORMAT (I10,F10.0,F10.0,10X,I3)
       ELSE
        WRITE(100,160) I,NODXY(I,1),NODXY(I,2)
```

```
160      FORMAT (I10,F10.0,F10.0)
        ENDIF
170   CONTINUE
C********     READ PUMPING DATA FROM 'PROPERTIES' AND WRITE TO 'RADFLOW.INPUT'
      READ(80,180) Q,QRCH,DELT,TMAXF,TSM,NTS
      WRITE(100,180) Q,QRCH,DELT,TMAXF,TSM,NTS
180   FORMAT (5F10.3,I10)
      CLOSE(80)
      CLOSE(100)
      RETURN
      END
      SUBROUTINE PLOT(NEL,NB)
C     A. PUCCI AND D. POPE                VERSION 3/2/87
C
C***********************************************************************
C     DISPLA PROGRAM TO PLOT TRIANGULAR MESH
C***********************************************************************
C
C     NEL=NUMBER OF ELEMENTS, NB=NUMBER OF NODES
C     XD,YD ARE ARRAYS USED REPEATEDLY IN PLOTTING EACH ELEMENT BORDER
C
C     NODXY CONTAINS THE X AND Y COORDINATES OF EACH NODE
C     ELM   CONTAINS THE NUMBERS OF THE NODES WHICH ARE AT THE CORNERS OF
C           EACH ELEMENT
C
      INTEGER ELM(800,3)
      REAL NODXY(1000,2),XD(4),YD(4),XMAX,YMAX
      CHARACTER DEV*1,ANS*1,SCALE*1,RESP1*1,RESP2*2,PEN2*4,PEN3*4
       CHARACTER*1   OPT         /*  DEVICE OPTION CODE
       CHARACTER*50  DES         /*  DEVICE DESCRIPTION
       CHARACTER*32  PALET       /*  DEVICE PALET
       LOGICAL       FLAGS(3)    /*  DEVICE OPTION FLAGS
C GDIU                           /*    1-PLOT FILE REQUIRED
C GDIU                           /*    2-DEVICE IS NOT A GRAPHIC TERMINAL
C GDIU                           /*    3-DEVICE HAS GRAPHIC INPUT
       REAL          SIZE(2)     /*  OPTIONAL MAX WIDTH AND HEIGHT OF PLOT
       CHARACTER*128 PLTFIL      /*  NAME OF PLOT FILE, IF NEEDED
       INTEGER       LUPLOT      /*  LOGICAL UNIT TO OPEN PLOT FILE ON
       INTEGER       NDEV        /*  DEVICE NUMBERS
       CHARACTER*8   COLOR       /*  PLOT COLOR
      COMMON NODXY,ELM
C     NEL = TOTAL NUMBER OF ELEMENTS
C
      WRITE(1,*) 'LENGTH OF X-AXIS?'
      WRITE(1,*) 'MAXIMUM RADIAL DISTANCE FOR DOMAIN.'
      READ(1,10) XMAX
10    FORMAT(F6.0)
      WRITE(1,*) 'LENGTH OF Y-AXIS?'
      WRITE(1,*) 'MAXIMUM DEPTH BELOW SURFACE'
      READ(1,10) YMAX
      WRITE(1,*) 'DO YOU WANT THE RADIAL DISTANCE ON A LOG OR AN'
      WRITE(1,*) 'ARITHMETIC SCALE?  (ENTER L OR A)'
      READ(1,40) SCALE
```

```
40    FORMAT(A1)
      IF (SCALE .EQ. 'L') THEN
       XMAX = LOG(XMAX)
       DO 20 I = 1,NB
        IF (NODXY(I,1) .NE. 0) THEN
        NODXY(I,1) = LOG(NODXY(I,1))
       ENDIF
20    CONTINUE
      ENDIF
C
C*****************************************************************************
C   SUBROUTINE G_QRDV PRESENTS MENU OF AVAILABLE DEVICES
C*****************************************************************************
C
      LUPLT = 51
C
      NDEV = 0
30    CALL G_QRDV(NDEV)
C
      CALL G_DVDF(OPT,DES,PALET,FLAGS,SIZE,-NDEV,IRTN)
      IF (IRTN .NE. NDEV) STOP 'BAD DEVICE SELECTED'
      IF (FLAGS(1)) GO TO 90
C
      CALL G_INIT(NDEV,1,1,LUPLT,IERR)
      IF(IERR.NE.0) STOP 'ERROR TRYING TO INITIALIZE GRAPHIC DEVICE'
C
      CALL G_GRMO(NDEV,1,1,LUPLT,IERR)
      IF(IERR .NE. 0)STOP 'ERROR TRYING TO GO TO GRAPHIC MODE'
C
      CALL PAGE(10.,10.)
      CALL AREA2D(9.,9.)
      CALL XNAME('DISTANCE FROM WELL',18)
      CALL YNAME('DEPTH BELOW SURFACE',19)
      CALL GRAF(0.,500.,XMAX,YMAX,-50.,0.)
      DO 70 I=1,NEL
       DO 60 JJ=1,3
        XD(JJ)=NODXY(ELM(I,JJ),1)
60      YD(JJ)=NODXY(ELM(I,JJ),2)
       XD(4)=NODXY(ELM(I,1),1)
       YD(4)=NODXY(ELM(I,1),2)
       CALL CURVE(XD,YD,4,0)
70    CONTINUE
C
      CALL ENDPL(0)
        CALL G_TXMO(NDEV,1,1,LUPLT,IERR)
        IF(IERR.NE.0)STOP 'ERROR AT END OF PLOT'
         CALL G_DNPL(NDEV,1,1,LUPLT,IERR)
        IF(IERR.NE.0)STOP 'ERROR AT END OF PLOT'
      CALL TNOUA('DO YOU WANT ANOTHER PLOT, Y OR N?  : ',INTS(34))
      READ(1,80) ANS
80    FORMAT(A1)
      IF (ANS .EQ. 'Y') THEN
       GO TO 30
      ENDIF
```

```
      GO TO 100
C
90    OPEN(LUPLT,FILE='FEM.PLOT',STATUS='NEW')
C
      CALL G_INIT(NDEV,1,1,LUPLT,IERR)
      IF(IERR.NE.0) STOP 'ERROR TRYING TO INITIALIZE GRAPHIC DEVICE'
C
      CALL G_GRMO(NDEV,1,1,LUPLT,IERR)
      IF(IERR .NE. 0)STOP 'ERROR TRYING TO GO TO GRAPHIC MODE'
C
      CALL PAGE(30.,30.)
      CALL AREA2D(28.,28.)
      CALL XNAME('DISTANCE FROM WELL',18)
      CALL YNAME('DEPTH BELOW SURFACE',19)
      CALL GRAF(0,50,XMAX,YMAX,-50,0)
      DO 120 I=1,NEL
       DO 110 JJ=1,3
        XD(JJ)=NODXY(ELM(I,JJ),1)
110      YD(JJ)=NODXY(ELM(I,JJ),2)
       XD(4)=NODXY(ELM(I,1),1)
       YD(4)=NODXY(ELM(I,1),2)
       CALL CURVE(XD,YD,4,0)
120    CONTINUE
C
      IF (PALET .NE. ' ') THEN
       PEN2 = PALET (9:12)
       PEN3 = PALET (13:16)
      ENDIF
C
C     LABEL TRIANGULAR ELEMENTS
      CALL TNOUA('DO YOU WANT TO LABEL THE ELEMENT NUMBERS? ',INTS(42))
      READ(1,125) RESP1
      IF (RESP1 .EQ. 'N') GO TO 135
125    FORMAT(A1)
      CALL HEIGHT(0.08)
      IF (PALET .NE. ' ') CALL SETCLR(PEN2)
      DO 130 J=1,NEL
       X = (NODXY(ELM(J,1),1)+NODXY(ELM(J,2),1)+NODXY(ELM(J,3),1))/3.0
       Y = (NODXY(ELM(J,1),2)+NODXY(ELM(J,2),2)+NODXY(ELM(J,3),2))/3.0
130    CALL RLINT(J,X ,Y )
C     LABEL NODE NUMBERS ON PLOT
135    CALL TNOUA('DO YOU WANT TO LABEL THE NODE NUMBERS? ',INTS(39))
      READ(1,125) RESP2
      IF (RESP2 .EQ. 'N') GO TO 150
      CALL HEIGHT('RESET')
      CALL HEIGHT(0.05)
      IF (PALET .NE. ' ') CALL SETCLR(PEN3)
      CALL MSHIFT(0.0,-0.05)
      DO 140 K=1,NB
       IF (SCALE .EQ. 'A') THEN
        XSP=NODXY(K,1)+5.
       ELSE
        XSP = NODXY(K,1) + .4
       ENDIF
```

```
      YSP=NODXY(K,2)+3.
140   CALL RLINT(K,XSP,YSP)
150   CALL ENDPL(0)
      CALL G_TXMO(NDEV,1,1,LUPLT,IERR)
       IF(IERR.NE.0)STOP 'ERROR AT END OF PLOT'
        CALL G_DNPL(NDEV,1,1,LUPLT,IERR)
       IF(IERR.NE.0)STOP 'ERROR AT END OF PLOT'
     CLOSE(51)
100  RETURN
     END
```

APPENDIX 2

Formats for input files

SECTION A

BASIC DATA FILE

Group 1

Card	Columns	Format	Variable		Definition
1	1-4	I4	INRG	=	Number of subregions in mesh
	5-8	I4	INBP	=	Number of global-mesh nodes

Group 2: Coordinates of boundary points

Number of cards depends on number of boundary points

Card	Columns	Format	Variable		Definition
--	1-80	8(F10.5)	XP	=	X-coordinates of global-mesh nodes. Number of values = INBP. Origin is pumping well (feet/1000).
	1-80	8(F10.5)	YP	=	Y-coordinates of global-mesh nodes. Number of values = INBP. Origin is land surface (feet/1000).

Group 3: Connectivity data

INRG number of cards

Card	Columns	Format	Variable		Definition
1-INRG	1-3	I3	NRG	=	subregion number.
	4-15	4(I3)	JT(NRG,K)	=	Number of the subregion which shares side K with subregion NRG, K=1 to 4.

Group 4: Data on subregions along constant head boundary

TNR + 1 number of cards

Card	Columns	Format	Variable		Definition
1	1-3	I3	NRC	=	Number of subregions along the constant-head boundary.

	4-6	I3	TNR	=	Total number of rows along the constant-head boundary.
2-INRG	1-3	I3	NRB(I)	=	Number of rows in each subregion in order from top to bottom. I=1 to NRC.

Group 5: subregion Data

Card	Columns	Format	Variable		Definition
INRG number of cards					
1-INRG	1-4	I4	NRG	=	subregion number.
	5-8	I4	NROWS	=	Number of rows in subregion.
	9-12	I4	NCOL	=	Number of columns in subregion.
	13-44	8(I4)	NDN(I)	=	Numbers of the boundary points which define the subregion, I=1 to 8.

SECTION B

PROPERTIES

GROUP 1: Title and problem setup

Card	Columns	Format	Variable		Definition
1	1-80	20A4	TITLE	=	Any title the user wishes.
2	1-10	F10.0	PPR	=	Primary radial hydraulic conductivity in feet per day (ft/d).
	11-20	F10.0	PPZ	=	Primary vertical hydraulic conductivity (ft/d).
	21-30	I10	IPP	=	Number of pumping periods.
3	1-10	F10.0	S	=	Coefficient of compressive storage (Ss) in 1/ft.
	11-20	F10.0	SY	=	Specific Yield (unitless).
4	16-20	I5	NQ	=	Number of nodes associated with well screen.
	21-25	I5	NDIF	=	Number of elements having different hydraulic conductivities than the primary ones on card 2.

Group 2: Nodes along the well screen

NQ number of cards

Card	Columns	Format	Variable		Definition
--	1-5	I5	NND(I)	=	Node along well screen.

Group 3: Data to assign secondary hydrologic properties to user defined sets of subregions

Card	Columns	Format	Variable		Definition
1	1-3	I3	DIFFK	=	Number of subsets of subregions to be assigned secondary properties.

Subgroups: This section will be repeated for each subset of subregions

1	1-10	F10.0	PR	=	Radial hydraulic conductivity assigned to all elements contained in the subregions in this subset (ft/d).
	11-20	F10.0	PZ	=	Vertical hydraulic conductivity (ft/d).
	21-24	I4	B	=	Number of subregions in this subset of subregions.
	25-34	F10.9	DIFFS	=	Coefficient of compressive storage (Ss) in $feet^{-1}$.
2-B	1-4	I4	RN	=	subregion number to be included in this subset.

Group 4: Pumping period information used in RADFLOW-S.F77

Card	Columns	Format	Variable		Definition
	1-10	F10.0	Q	=	Pumping rate in cubic feet per second (ft^3/sec).
	11-20	F10.0	QRCH	=	Recharge rate (ft/d).
	21-30	F10.0	DELT	=	Initial time step (Days).
	31-40	F10.0	TMAXF	=	Maximum length of pumping Period (Days).
	41-50	F10.0	TSM	=	Time step multiplier (each time step after DELT is multiplied by TSM.
	51-60	I10	NTS	=	Number of time steps in Pumping period.

For more information on RADFLOW options for pumping period simulations see Reilly (1984 pg. 27).

SECTION C

OB.WELL

Group 1: Number of observation wells you have data for

Card	Columns	Format	Variable		Definition
1	1-4	I4	NNO	=	Number of observation wells you wish to use.

Group 2: Location of midpoint of screen for each observation well

NNO number of cards

Card	Columns	Format	Variable		Definition
--	2-7	F6.0	XBAR(I)	=	X-coordinate (in feet) of the midpoint of the screen for each observation well, I=1 to NNO.
--	9-14	F6.0	YBAR(I)	=	Y-coordinate (in feet) of the midpoint of the screen for each observation well, I=1 to NNO.

Group 3: Element numbers that contain the screen for each observation well

NNO number of cards

Card	Columns	Format	Variable		Definition
--	1-4	I4	NNE(I)	=	Element number that contains the midpoint of the screen for each observation well, I=1 to NNO.

SECTION D

Observation well data file

Group 1

One card

Card	Columns	Format	Variable		Definition
1	2-4	I3	N	=	Number of observed data points in file.
	5-9	F5.0	R	=	Distance of observed well from pumping well (in feet).
	11-15	F5.1	RE	=	Reference elevation. Will be zero if the data is drawdown data. If the data is water level data this should be the elevation of the measuring point of the well.

Group 2: Time-drawdown or Time-water-level data

N number of cards

Card	Columns	Format	Variable		Definition
--	1-10	F10.0	T(I)	=	Time of measurement in minutes. I=1,N.
	11-20	F10.3	S(I)	=	Drawdown or water-level data in feet, I=1 to N.

APPENDIX 3

Example of BASIC DATA FILE for the six subregion example mesh shown in figure 3.

The description of the input data file format is given in appendix 2, section A.

```
6 29
    .4          .5      .6          .6       .6       .5       .4       .4
    .4          .4      .5          .6       .6       .3       .2       .2
    .2          .3      .2          .2       .3       .1      0.0      0.0
   0.0          .1     0.0         0.0       .1
    .1          .1      .1          .05     0.0      0.0      0.0       .05
    .15         .2      .2          .2       .15     0.0      0.0       .05
    .1          .1      .15         .2       .2      0.0      0.0       .05
    .1          .1      .15         .2       .2
 1  2  0  0  3
 2  0  0  1  4
 3  4  1  0  5
 4  0  2  3  6
 5  6  3  0  0
 6  0  4  5  0
  2  7
  3
  5
 1  3   3   1   2   3    4   5   6   7   8
 2  5   3  10  11  12   13   3   2   1   9
 3  3   3  17  18   1    8   7  14  15  16
 4  5   3  20  21  10    9   1  18  17  19
 5  3   5  25  26  17   16  15  22  23  24
 6  5   5  28  29  20   19  17  26  25  27
```

APPENDIX 4

Data files used for example simulation shown in figure 4.

SECTION A

BASIC DATA FILE for generating figure 4.

The description of the input data file format is given in appendix 2, section A.

50	181						
10.69	17.9	30.	30.	30.	17.9	10.69	10.69
10.69	10.69	17.9	30.	30.	10.69	10.69	17.9
30.	30.	10.69	10.69	17.9	30.	30.	10.69
10.69	17.9	30.	30.	6.40	3.82	3.82	3.82
6.40	3.82	3.82	6.40	3.82	3.82	6.40	3.82
3.82	6.40	3.82	3.82	6.40	2.28	1.36	1.36
1.36	2.28	1.36	1.36	2.28	1.36	1.36	2.28
1.36	1.36	2.28	1.36	1.36	2.28	.813	.485
.485	.485	.813	.485	.485	.813	.485	.485
.813	.485	.485	.813	.485	.485	.813	.290
.173	.173	.173	.290	.173	.173	.290	.173
.173	.290	.173	.173	.290	.173	.173	.290
.104	.062	.062	.062	.104	.062	.062	.104
.062	.062	.104	.062	.062	.104	.062	.062
.104	.037	.022	.022	.022	.037	.022	.022
.037	.022	.022	.037	.022	.022	.037	.022
.022	.037	.013	.008	.008	.008	.013	.008
.008	.013	.008	.008	.013	.008	.008	.013
.008	.008	.013	.005	.003	.003	.003	.005
.003	.003	.005	.003	.003	.005	.003	.003
.005	.003	.003	.005	.002	0.0	0.0	0.0
.002	0.0	0.0	.002	0.0	0.0	.002	0.0
0.0	.002	0.0	0.0	.002			
0.302	0.302	0.302	0.209	0.117	0.117	0.117	0.209
0.378	0.454	0.454	0.454	0.378	0.525	0.597	0.597
0.597	0.525	0.632	0.650	0.650	0.650	0.632	0.687
0.709	0.709	0.709	0.687	0.117	0.117	0.209	0.302
0.302	0.378	0.454	0.454	0.525	0.597	0.597	0.632
0.650	0.650	0.687	0.709	C.709	0.117	0.117	0.209
0.302	0.302	0.378	0.454	0.454	0.525	0.597	0.597
0.632	0.650	0.650	0.687	0.709	0.709	0.117	0.117
0.209	0.302	0.302	0.378	0.454	0.454	0.525	0.597
0.597	0.632	0.650	0.650	0.687	0.709	0.709	0.117
0.117	0.209	0.302	0.302	0.378	0.454	0.454	0.525
0.597	0.597	0.632	0.650	0.650	0.687	0.709	0.709
0.117	0.117	0.209	0.302	0.302	0.378	0.454	0.454
0.525	0.597	0.597	0.632	0.650	0.650	0.687	0.709
0.709	0.117	0.117	0.209	0.302	0.302	0.378	0.454
0.454	0.525	0.597	0.597	0.632	0.650	0.650	0.687
0.709	0.709	0.117	0.117	0.209	0.302	0.302	0.378
0.454	0.454	0.525	0.597	0.597	0.632	0.650	0.650
0.687	0.709	0.709	0.117	0.117	0.209	0.302	0.302
.378	.454	.454	.525	.597	.597	.632	.650

.650	.687	.709	.709	.117	.117	.209	.302
.302	.378	.454	.454	.525	.597	.597	.632
.650	.650	.687	.709	.709			

1	2	0	0	6
2	3	0	1	7
3	4	0	2	8
4	5	0	3	9
5	0	0	4	10
6	7	1	0	11
7	8	2	6	12
8	9	3	7	13
9	10	4	8	14
10	0	5	9	15
11	12	6	0	16
12	13	7	11	17
13	14	8	12	18
14	15	9	13	19
15	0	10	14	20
16	17	11	0	21
17	18	12	16	22
18	19	13	17	23
19	20	14	18	24
20	0	15	19	25
21	22	16	0	26
22	23	17	21	27
23	24	18	22	28
24	25	19	23	29
25	0	20	24	30
26	27	21	0	31
27	28	22	26	32
28	29	23	27	33
29	30	24	28	34
30	0	25	29	35
31	32	26	0	36
32	33	27	31	37
33	34	28	32	38
34	35	29	33	39
35	0	30	34	40
36	37	31	0	41
37	38	32	36	42
38	39	33	37	43
39	40	34	38	44
40	0	35	39	45
41	42	36	0	46
42	43	37	41	47
43	44	38	42	48
44	45	39	43	49
45	0	40	44	50
46	47	41	0	0
47	48	42	46	0
48	49	43	47	0
49	50	44	48	0
50	0	45	49	0

5 13
3
3
3
3
5

1	3	3	1	2	3	4	5	6	7	8
2	3	3	10	11	12	13	3	2	1	9
3	3	3	15	16	17	18	12	11	10	14
4	3	3	20	21	22	23	17	16	15	19
5	5	3	25	26	27	28	22	21	20	24
6	3	3	32	33	1	8	7	29	30	31
7	3	3	35	36	10	9	1	33	32	34
8	3	3	38	39	15	14	10	36	35	37
9	3	3	41	42	20	19	15	39	38	40
10	5	3	44	45	25	24	20	42	41	43
11	3	3	49	50	32	31	30	46	47	48
12	3	3	52	53	35	34	32	50	49	51
13	3	3	55	56	38	37	35	53	52	54
14	3	3	58	59	41	40	38	56	55	57
15	5	3	61	62	44	43	41	59	58	60
16	3	3	66	67	49	48	47	63	64	65
17	3	3	69	70	52	51	49	67	66	68
18	3	3	72	73	55	54	52	70	69	71
19	3	3	75	76	58	57	55	73	72	74
20	5	3	78	79	61	60	58	76	75	77
21	3	3	83	84	66	65	64	80	81	82
22	3	3	86	87	69	68	66	84	83	85
23	3	3	89	90	72	71	69	87	86	88
24	3	3	92	93	75	74	72	90	89	91
25	5	3	95	96	78	77	75	93	92	94
26	3	3	100	101	83	82	81	97	98	99
27	3	3	103	104	86	85	83	101	100	102
28	3	3	106	107	89	88	86	104	103	105
29	3	3	109	110	92	91	89	107	106	108
30	5	3	112	113	95	94	92	110	109	111
31	3	3	117	118	100	99	98	114	115	116
32	3	3	120	121	103	102	100	118	117	119
33	3	3	123	124	106	105	103	121	120	122
34	3	3	126	127	109	108	106	124	123	125
35	5	3	129	130	112	111	109	127	126	128
36	3	3	134	135	117	116	115	131	132	133
37	3	3	137	138	120	119	117	135	134	136
38	3	3	140	141	123	122	120	138	137	139
39	3	3	143	144	126	125	123	141	140	142
40	5	3	146	147	129	128	126	144	143	145
41	3	3	151	152	134	133	132	148	149	150
42	3	3	154	155	137	136	134	152	151	153
43	3	3	157	158	140	139	137	155	154	156
44	3	3	160	161	143	142	140	158	157	159
45	5	3	163	164	146	145	143	161	160	162
46	3	3	168	169	151	150	149	165	166	167
47	3	3	171	172	154	153	151	169	168	170
48	3	3	174	175	157	156	154	172	171	173
49	3	3	177	178	160	159	157	175	174	176
50	5	3	180	181	163	162	160	178	177	179

SECTION B

PROPERTIES data file for region shown in figure 4.

Description of input data set for PROPERTIES is given in appendix 2, section B.

```
EXAMPLE PUMP TEST
  100       10.0    1
 .00000100    .27000
 480  273  13   5    240
 262
 264
 266
 268
 270
 2
   150.     30.0      10 .0000005
   2
   7
  12
  17
  22
  27
  32
  37
  42
  47
     1.0     0.1      20 .000002
   1
   6
  11
  16
  21
  26
  31
  36
  41
  46
   3
   8
  13
  18
  23
  28
  33
  38
  43
  48
-2.78    0.0       .001      .958      1.2       1000
```

SECTION C

OB.WELL data file used to generate figure 5.

Description of input data set for OB.WELL is given in appendix 2, section C.

```
 1
  600   682.0
 181
```

SECTION D

Observed field-data file used to generate figure 5.

Description of input data set for actual pump test data files is given in appendix 2, section D.

```
  26 600.  138.7
1.        139.1
2.        139.33
3.        139.4
4.        139.5
5.        139.8
6.        140.
9.        140.25
12.       140.5
18.       141.1
24.       141.5
30.       141.8
45.       142.25
70.       142.9
100.      143.33
140.      143.75
220.      144.5
335.      145.3
455.      145.9
605.      146.5
695.      146.66
845.      147.
965.      147.5
1055.     147.7
1175.     147.75
1295.     147.8
1385.     148.
```

APPENDIX 5

Example of 'OUTPUT' file from MESH 'P' mode for example shown in figure 3.

CONNECTIVITY DATA

SUBREGION	SHARED BOUNDARY 1	2	3	4
1	2	0	0	3
2	0	0	1	4
3	4	1	0	5
4	0	2	3	6
5	6	3	0	0
6	0	4	5	0

1

**** ARRANGEMENT OF GLOBAL NODE AND FINITE ELEMENT NODE NUMBER, IN SUBREGION 1 ****

NUMBER OF INCREMENTS =

3 ROWS AND 3 COLUMNS

GLOBAL NODE NUMBERS 1 2 3 4 5 6 7 8

FINITE-ELEMENT NODE NUMBERS

8	9	1
10	11	2
12	13	3

ELEMENT NUMBERS, FINITE-ELEMENT NODE NUMBERS, AND NODE COORDINATES

NEL	NODE NUMBERS			X(1)	Y(1)	X(2)	Y(2)	X(3)	Y(3)
1	10	11	9	0.4000	0.0500	0.5000	0.0500	0.5000	0.0000
2	10	9	8	0.4000	0.0500	0.5000	0.0000	0.4000	0.0000
3	11	2	1	0.5000	0.0500	0.6000	0.0500	0.6000	0.0000
4	11	1	9	0.5000	0.0500	0.6000	0.0000	0.5000	0.0000
5	12	13	11	0.4000	0.1000	0.5000	0.1000	0.5000	0.0500
6	12	11	10	0.4000	0.1000	0.5000	0.0500	0.4000	0.0500
7	13	3	2	0.5000	0.1000	0.6000	0.1000	0.6000	0.0500
8	13	2	11	0.5000	0.1000	0.6000	0.0500	0.5000	0.0500

*NOTE: COORDINATES ARE MULTIPLIED BY .001

1

**** ARRANGEMENT OF GLOBAL NODE AND FINITE ELEMENT NODE NUMBER, IN SUBREGION 2 ****

NUMBER OF INCREMENTS =

5 ROWS AND 3 COLUMNS

GLOBAL NODE NUMBERS 10 11 12 13 3 2 1 9

FINITE-ELEMENT NODE NUMBERS

12	13	3
14	15	4
16	17	5
18	19	6
20	21	7

ELEMENT NUMBERS, FINITE-ELEMENT NODE NUMBERS, AND NODE COORDINATES

NEL	NODE NUMBERS			X(1)	Y(1)	X(2)	Y(2)	X(3)	Y(3)
9	14	15	13	0.4000	0.1250	0.5000	0.1250	0.5000	0.1000
10	14	13	12	0.4000	0.1250	0.5000	0.1000	0.4000	0.1000
11	15	4	3	0.5000	0.1250	0.6000	0.1250	0.6000	0.1000
12	15	3	13	0.5000	0.1250	0.6000	0.1000	0.5000	0.1000
13	16	17	15	0.4000	0.1500	0.5000	0.1500	0.5000	0.1250

14	16	15	14	0.4000	0.1500	0.5000	0.1250	0.4000	0.1250
15	17	5	4	0.5000	0.1500	0.6000	0.1500	0.6000	0.1250
16	17	4	15	0.5000	0.1500	0.6000	0.1250	0.5000	0.1250
17	18	19	17	0.4000	0.1750	0.5000	0.1750	0.5000	0.1500
18	18	17	16	0.4000	0.1750	0.5000	0.1500	0.4000	0.1500
19	19	6	5	0.5000	0.1750	0.6000	0.1750	0.6000	0.1500
20	19	5	17	0.5000	0.1750	0.6000	0.1500	0.5000	0.1500
21	20	21	19	0.4000	0.2000	0.5000	0.2000	0.5000	0.1750
22	20	19	18	0.4000	0.2000	0.5000	0.1750	0.4000	0.1750
23	21	7	6	0.5000	0.2000	0.6000	0.2000	0.6000	0.1750
24	21	6	19	0.5000	0.2000	0.6000	0.1750	0.5000	0.1750

*NOTE: COORDINATES ARE MULTIPLIED BY .001

1

**** ARRANGEMENT OF GLOBAL NODE AND FINITE ELEMENT NODE NUMBER, IN SUBREGION 3 ****

NUMBER OF INCREMENTS =

3 ROWS AND 3 COLUMNS

GLOBAL NODE NUMBERS 17 18 1 8 7 14 15 16

FINITE-ELEMENT NODE NUMBERS

22	23	8
24	25	10
26	27	12

ELEMENT NUMBERS, FINITE-ELEMENT NODE NUMBERS, AND NODE COORDINATES

NEL	NODE NUMBERS			X(1)	Y(1)	X(2)	Y(2)	X(3)	Y(3)
25	24	25	23	0.2000	0.0500	0.3000	0.0500	0.3000	0.0000
26	24	23	22	0.2000	0.0500	0.3000	0.0000	0.2000	0.0000
27	25	10	8	0.3000	0.0500	0.4000	0.0500	0.4000	0.0000
28	25	8	23	0.3000	0.0500	0.4000	0.0000	0.3000	0.0000
29	26	27	25	0.2000	0.1000	0.3000	0.1000	0.3000	0.0500
30	26	25	24	0.2000	0.1000	0.3000	0.0500	0.2000	0.0500
31	27	12	10	0.3000	0.1000	0.4000	0.1000	0.4000	0.0500
32	27	10	25	0.3000	0.1000	0.4000	0.0500	0.3000	0.0500

*NOTE: COORDINATES ARE MULTIPLIED BY .001

1

**** ARRANGEMENT OF GLOBAL NODE AND FINITE ELEMENT NODE NUMBER, IN SUBREGION 4 ****

NUMBER OF INCREMENTS =

5 ROWS AND 3 COLUMNS

GLOBAL NODE NUMBERS 20 21 10 9 1 18 17 19

FINITE-ELEMENT NODE NUMBERS

26	27	12
28	29	14
30	31	16
32	33	18
34	35	20

ELEMENT NUMBERS, FINITE-ELEMENT NODE NUMBERS, AND NODE COORDINATES

NEL	NODE NUMBERS			X(1)	Y(1)	X(2)	Y(2)	X(3)	Y(3)
33	28	29	27	0.2000	0.1250	0.3000	0.1250	0.3000	0.1000
34	28	27	26	0.2000	0.1250	0.3000	0.1000	0.2000	0.1000
35	29	14	12	0.3000	0.1250	0.4000	0.1250	0.4000	0.1000
36	29	12	27	0.3000	0.1250	0.4000	0.1000	0.3000	0.1000
37	30	31	29	0.2000	0.1500	0.3000	0.1500	0.3000	0.1250
38	30	29	28	0.2000	0.1500	0.3000	0.1250	0.2000	0.1250
39	31	16	14	0.3000	0.1500	0.4000	0.1500	0.4000	0.1250

40	31	14	29	0.3000	0.1500	0.4000	0.1250	0.3000	0.1250
41	32	33	31	0.2000	0.1750	0.3000	0.1750	0.3000	0.1500
42	32	31	30	0.2000	0.1750	0.3000	0.1500	0.2000	0.1500
43	33	18	16	0.3000	0.1750	0.4000	0.1750	0.4000	0.1500
44	33	16	31	0.3000	0.1750	0.4000	0.1500	0.3000	0.1500
45	34	35	33	0.2000	0.2000	0.3000	0.2000	0.3000	0.1750
46	34	33	32	0.2000	0.2000	0.3000	0.1750	0.2000	0.1750
47	35	20	18	0.3000	0.2000	0.4000	0.2000	0.4000	0.1750
48	35	18	33	0.3000	0.2000	0.4000	0.1750	0.3000	0.1750

*NOTE: COORDINATES ARE MULTIPLIED BY .001

1

**** ARRANGEMENT OF GLOBAL NODE AND FINITE ELEMENT NODE NUMBER, IN SUBREGION 5 ****

NUMBER OF INCREMENTS =

3 ROWS AND 5 COLUMNS

GLOBAL NODE NUMBERS 25 26 17 16 15 22 23 24

FINITE-ELEMENT NODE NUMBERS

36	37	38	39	22
40	41	42	43	24
44	45	46	47	26

ELEMENT NUMBERS, FINITE-ELEMENT NODE NUMBERS, AND NODE COORDINATES

NEL	NODE NUMBERS			X(1)	Y(1)	X(2)	Y(2)	X(3)	Y(3)
49	40	41	37	0.0000	0.0500	0.0500	0.0500	0.0500	0.0000
50	40	37	36	0.0000	0.0500	0.0500	0.0000	0.0000	0.0000
51	41	42	38	0.0500	0.0500	0.1000	0.0500	0.1000	0.0000
52	41	38	37	0.0500	0.0500	0.1000	0.0000	0.0500	0.0000
53	42	43	39	0.1000	0.0500	0.1500	0.0500	0.1500	0.0000
54	42	39	38	0.1000	0.0500	0.1500	0.0000	0.1000	0.0000
55	43	24	22	0.1500	0.0500	0.2000	0.0500	0.2000	0.0000
56	43	22	39	0.1500	0.0500	0.2000	0.0000	0.1500	0.0000
57	44	45	41	0.0000	0.1000	0.0500	0.1000	0.0500	0.0500
58	44	41	40	0.0000	0.1000	0.0500	0.0500	0.0000	0.0500
59	45	46	42	0.0500	0.1000	0.1000	0.1000	0.1000	0.0500
60	45	42	41	0.0500	0.1000	0.1000	0.0500	0.0500	0.0500
61	46	47	43	0.1000	0.1000	0.1500	0.1000	0.1500	0.0500
62	46	43	42	0.1000	0.1000	0.1500	0.0500	0.1000	0.0500
63	47	26	24	0.1500	0.1000	0.2000	0.1000	0.2000	0.0500
64	47	24	43	0.1500	0.1000	0.2000	0.0500	0.1500	0.0500

*NOTE: COORDINATES ARE MULTIPLIED BY .001

1

**** ARRANGEMENT OF GLOBAL NODE AND FINITE ELEMENT NODE NUMBER, IN SUBREGION 6 ****

NUMBER OF INCREMENTS =

5 ROWS AND 5 COLUMNS

GLOBAL NODE NUMBERS 28 29 20 19 17 26 25 27

FINITE-ELEMENT NODE NUMBERS

44	45	46	47	26
48	49	50	51	28
52	53	54	55	30
56	57	58	59	32
60	61	62	63	34

ELEMENT NUMBERS, FINITE-ELEMENT NODE NUMBERS, AND NODE COORDINATES

NEL	NODE NUMBERS			X(1)	Y(1)	X(2)	Y(2)	X(3)	Y(3)
65	48	49	45	0.0000	0.1250	0.0500	0.1250	0.0500	0.1000

66	48	45	44	0.0000	0.1250	0.0500	0.1000	0.0000	0.1000
67	49	50	46	0.0500	0.1250	0.1000	0.1250	0.1000	0.1000
68	49	46	45	0.0500	0.1250	0.1000	0.1000	0.0500	0.1000
69	50	51	47	0.1000	0.1250	0.1500	0.1250	0.1500	0.1000
70	50	47	46	0.1000	0.1250	0.1500	0.1000	0.1000	0.1000
71	51	28	26	0.1500	0.1250	0.2000	0.1250	0.2000	0.1000
72	51	26	47	0.1500	0.1250	0.2000	0.1000	0.1500	0.1000
73	52	53	49	0.0000	0.1500	0.0500	0.1500	0.0500	0.1250
74	52	49	48	0.0000	0.1500	0.0500	0.1250	0.0000	0.1250
75	53	54	50	0.0500	0.1500	0.1000	0.1500	0.1000	0.1250
76	53	50	49	0.0500	0.1500	0.1000	0.1250	0.0500	0.1250
77	54	55	51	0.1000	0.1500	0.1500	0.1500	0.1500	0.1250
78	54	51	50	0.1000	0.1500	0.1500	0.1250	0.1000	0.1250
79	55	30	28	0.1500	0.1500	0.2000	0.1500	0.2000	0.1250
80	55	28	51	0.1500	0.1500	0.2000	0.1250	0.1500	0.1250
81	56	57	53	0.0000	0.1750	0.0500	0.1750	0.0500	0.1500
82	56	53	52	0.0000	0.1750	0.0500	0.1500	0.0000	0.1500
83	57	58	54	0.0500	0.1750	0.1000	0.1750	0.1000	0.1500
84	57	54	53	0.0500	0.1750	0.1000	0.1500	0.0500	0.1500
85	58	59	55	0.1000	0.1750	0.1500	0.1750	0.1500	0.1500
86	58	55	54	0.1000	0.1750	0.1500	0.1500	0.1000	0.1500
87	59	32	30	0.1500	0.1750	0.2000	0.1750	0.2000	0.1500
88	59	30	55	0.1500	0.1750	0.2000	0.1500	0.1500	0.1500
89	60	61	57	0.0000	0.2000	0.0500	0.2000	0.0500	0.1750
90	60	57	56	0.0000	0.2000	0.0500	0.1750	0.0000	0.1750
91	61	62	58	0.0500	0.2000	0.1000	0.2000	0.1000	0.1750
92	61	58	57	0.0500	0.2000	0.1000	0.1750	0.0500	0.1750
93	62	63	59	0.1000	0.2000	0.1500	0.2000	0.1500	0.1750
94	62	59	58	0.1000	0.2000	0.1500	0.1750	0.1000	0.1750
95	63	34	32	0.1500	0.2000	0.2000	0.2000	0.2000	0.1750
96	63	32	59	0.1500	0.2000	0.2000	0.1750	0.1500	0.1750

BANDWIDTH QUANTITY IS 32 CALCULATED IN ELEMENT 95

```
C                             APPENDIX 6
C
C                           RADFLOW-S Code
C
C********************************************************************** MAN  70
C                                                                       MAN  80
C     A GALERKIN FINITE-ELEMENT FLOW MODEL FOR THE TRANSIENT            MAN  90
C     RESPONSE OF A RADIALLY-SYMMETRIC AQUIFER (Reilly, 1984)           MAN 100
C                                                                       MAN 110
C       PR, PZ = FT/DAY  ; ZE, RE=FT  ; Q=CFS ; TIME =DAYS              MAN 120
C     QRCH=FT/DAY (+ MEANS RECHARGE ; - MEANS DISCHARGE)                MAN 130
C                                                                       MAN 140
C     CONSTANT DRAWDOWN NODES MUST BE NUMBERED FIRST                    MAN 150
C           (I.E.   1,2,3,.........)                                    MAN 160
C     PPR=PRIMARY RADIAL HYDRAULIC CONDUCTIVITY                         MAN 170
C     PPZ=PRIMARY VERTICAL HYDRAULIC CONDUCTIVITY                       MAN 180
C     NDIF=# OF ELEMENTS WITH PROPERTIES DIFFERERING FROM THE
C      PRIMARY PROPERTIES
C     NE=# OF ELEMENTS; NN=# OF NODES;NCH=# OF CONSTANT DRAWDOWN NODES  MAN 200
C     NQ=# OF NODES DISCHARGING;NND=NODES DISCH.                        MAN 210
C     IF NODE HAS AN 'UNCONFINED' BOUNDARY PUT A '1' IN COL. 35         MAN 220
C     IF NODE HAS A RECHARGE TOP BOUNDARY PUT A '1' IN COL. 40          MAN 230
C     IF NODE HAS A CONSTANT ZERO DRAWDOWN PUT A '1' IN COL. 45         MAN 240
C                                                                       MAN 250
C********************************************************************** MAN 260
C
C*******************************************************************************
C                             GLOSSARY
C
C*  NNE = # OF ELEMENTS WITH PROPERTIES DIFFERING FROM THE PRIMARY PROPERTIES
C*  XBAR,YBAR = ARRAYS OF COORDINATES FOR THE CENTERS OF OBSERVATION WELL
C*               SCREEN LOCATIONS, UP TO FIVE OBSERVATION WELLS
C*  N1,N2,N3 = LINEAR TRIANGULAR SHAPE FUNCTIONS
C*  PHI = AN ARRAY OF HEAD(S) FOR EACH OBSERVATION WELL FOR EACH ITERATION,
C*        UP TO FIVE OBSERVATION WELLS.
C*  POB,TOB = CUMULATIVE ARRAY OF CALCULATED HEAD(S) AT SIMULATION TIMES, FOR
C*             EACH OBSERVATION WELL(S), UP TO FIVE OBSERVATION WELLS AND UP TO
C*             100 ITERATIONS AS PRESENTLY DIMENSIONED.
C*  EPSI = A GAUSSIAN NORM STATISTIC COMPARING THE PREDICTED AND OBSERVED
C*          DRAWDOWNS  IN EACH OBSERVATION WELL(S), UP TO FIVE WELLS
C*  SS = AN ARRAY CONTAINING THE SPECIFIC STORAGE VALUES FOR EACH TRIANGULAR
C*        ELEMENT
C*******************************************************************************
C
      DIMENSION NG(480,3), PR(480), PZ(480), RE(273), ZE(273), IFLUX(273MAN  20
     1), IQ(273), IRCH(273), NSCON(273), F(273), S1(273), S3(260), S5(26MAN  30
     20), F2(260), WG1(260,59), WTRA1(260,59), R(3), Z(3), ZI(3), RI(3),MAN  40
     3 L(3), W(3,3), WR(3,3), WZ(3,3), WW(3,3), WST(3,3), NND(50), TITLEMAN  50
     4(20),SS(480)                                                          MAN
C*******************************************************************************
C   ARRAYS THAT ARE INTRODUCED IN THE RADFLOW-S VERSION
C*******************************************************************************
      DIMENSION NNE(5)
      REAL N1,N2,N3,XBAR(5),YBAR(5),PHI(5)
```

```
      REAL POB(5,100),TOB(5,100),EPSI(5)
      COMMON /PT/ POB,TOB
C     NNE= OBSERVATION WELL ELEMENT NUMBERS
      COMMON /ELEM/NG,PR,PZ
      COMMON /NODES/RE,ZE,IFLUX,IQ,IRCH,NSCON,F,S1
      COMMON /GLOBAL/WG1,WTRA1
      COMMON /EPS/EPSI
C*********************************************************************
C
C     A GALERKIN FINITE-ELEMENT FLOW MODEL FOR THE TRANSIENT
C     RESPONSE OF A RADIALLY SYMMETRIC AQUIFER
C
C       PR, PZ = FT/DAY  ; ZE, RE=FT  ; Q=CFS ; TIME =DAYS
C     QRCH=FT/DAY (+ MEANS RECHARGE ; - MEANS DISCHARGE)
C
C     CONSTANT DRAWDOWN NODES MUST BE NUMBERED FIRST
C           (I.E.   1,2,3,.........)
C     PPR=PRIMARY RADIAL HYDRAULIC CONDUCTIVITY
C     PPZ=PRIMARY VERTICAL HYDRAULIC CONDUCTIVITY
C     NDIF=# OF DIFFERING HYDRAULIC CONDUCTIVITY ELEMENTS
C     NE=# OF ELEMENTS; NN=# OF NODES;NCH=# OF CONSTANT DRAWDOWN NODES
C     NQ=# OF NODES DISCHARGING;NND=NODES DISCH.
C     IF NODE HAS AN 'UNCONFINED' BOUNDARY PUT A '1' IN COL. 35
C     IF NODE HAS A RECHARGE TOP BOUNDARY PUT A '1' IN COL. 40
C     IF NODE HAS A CONSTANT ZERO DRAWDOWN PUT A '1' IN COL. 45
C
C*********************************************************************
C
C
C     INPUT-OUTPUT
C
C******************************************************************************
C   FILES USED FOR I/O INTRODUCED IN RADFLOW-S
C******************************************************************************
      OPEN(UNIT=55,FILE='RADFLOW.INPUT',STATUS='OLD')
      OPEN(UNIT=56,FILE='RAD.OUTPUT',STATUS='NEW')
      OPEN(UNIT=57,FILE='OB.WELL',STATUS='OLD')
C
C     CALCULATE CONSTANTS
C
      TPI=3.1416*2.
      CONV1=1440.*60.
C
      READ (55,11) TITLE
   11 FORMAT (20A4)
      WRITE (56,12) TITLE
   12 FORMAT (1H1,10X,20A4)
      READ (55,13) PPR,PPZ,IPP
   13 FORMAT (2F10.0,I10)
      READ (55,14) S,SY
   14 FORMAT (2F10.0)
      WRITE (56,15) S
   15 FORMAT (1H0,36H COEFFICIENT OF   SPECIFIC  STORAGE=,E12.5,5H 1/FT)
      WRITE (56,16) SY
```

```
   16 FORMAT (1H0,17H SPECIFIC YIELD =,F10.4)                            MAN 440
      WRITE (56,17) PPR,PPZ                                             MAN 450
   17 FORMAT (1H0,38H THE PRIMARY HYDRAULIC CONDUCTIVITY = ,F10.3,12H  RMAN 460
     1ADIALLY &,F10.3,21H  VERTICALLY (FT/DAY))                         MAN 470
      READ (55,18) NE,NN,NCH,NQ,NDIF                                    MAN 480
   18 FORMAT (5I5)                                                      MAN 490
      WRITE (56,19) NE,NN,NCH,NQ,NDIF                                   MAN 500
   19 FORMAT (1H0,I5,13H   ELEMENTS   ,I5,9H  NODES   ,I5,22H  CONSTANT VAMAN 510
     1LUE NODES,I5,18H NODES DISCHARGING,I5,23H ELEMENTS OF DIFF. H.C.) MAN 520
      IF (NQ.EQ.0) GO TO 24                                             MAN 530
      WRITE (56,20)                                                     MAN 540
   20 FORMAT (1H0,17HDISCHARGING NODES)                                 MAN 550
      DO 23 I=1,NQ                                                      MAN 560
      READ (55,21) NND(I)                                               MAN 570
   21 FORMAT (I5)                                                       MAN 580
      WRITE (56,22) NND(I)                                              MAN 590
   22 FORMAT (1X,I9)                                                    MAN 600
   23 CONTINUE                                                          MAN 610
   24 DO 25 I=1,NE                                                      MAN 620
      PR(I)=PPR                                                         MAN 630
      PZ(I)=PPZ                                                         MAN 640
C*******************************************************************************
C   INITIALIZE SPECIFIC STORAGE IN EACH TRIANGULAR ELEMENT
C*******************************************************************************
      SS(I)=S
   25 CONTINUE                                                          MAN 650
C                                                                       MAN 660
C*******************************************************************************
C     DEFINE ELEMENTS OF DIFFERENT CONDUCTIVITIES                       MAN 670
C*******************************************************************************
C                                                                       MAN 680
      IF (NDIF.EQ.0) GO TO 31                                           MAN 690
C*******************************************************************************
C    SOME SECTION OF THE ORIGINAL RADFLOW SOURCE CODE HAVE BEEN COMMENTED
C    OUT WITH AN 'CS' IN RADFLOW-S BECAUSE OF REDUNDANCIES WITH THE MESH
C    PRE-PROCESSOR.  THEY REMAIN AS COMMENTED STATEMENTS TO HELP THE
C    PROGRAM REVIEWER
C*******************************************************************************
C      WRITE (56,26)
CS26   FORMAT (1H0,1X,39H, ELEMENTS WITH DIFFERENT CONDUCTIVITIES)
CS     WRITE (56,27)                                                    MAN 720
CS  27 FORMAT (1H0,1X,7HELEMENT,5X,11HRADIAL H.C.,5X,13HVERTICAL H.C.)  MAN 730
      DO 30 K=1,NDIF                                                    MAN 740
      READ (55,28) IE,PR(IE),PZ(IE),SS(IE)                              MAN 750
   28 FORMAT (I10,2F10.0,5X,F10.0)                                      MAN 760
CS     WRITE (56,29) IE,PR(IE),PZ(IE)                                   MAN 770
CS  29 FORMAT (1X,I5,10X,F10.3,10X,F10.3)                               MAN 780
   30 CONTINUE                                                          MAN 790
C                                                                       MAN 800
C*******************************************************************************
C     READ NODAL ORDER OF EACH ELEMENT                                  MAN 810
C*******************************************************************************
C                                                                       MAN 820
CS  31 WRITE (56,32)                                                    MAN 830
```

```
CS  32 FORMAT (1H0,7HELEMENT,10X,11HNODAL ORDER)                        MAN 840
31     DO 35 I=1,NE                                                      MAN 850
       READ (55,33) IE,NG(IE,1),NG(IE,2),NG(IE,3)                        MAN 860
   33 FORMAT (4I5)                                                       MAN 870
CS      WRITE (56,34) IE,NG(IE,1),NG(IE,2),NG(IE,3)                      MAN 880
CS  34 FORMAT (5X,I5,5X,3(2X,I5))                                        MAN 890
   35 CONTINUE                                                           MAN 900
C********************************************************************************
C      READ # OF OBSERVATION WELLS
C      MAXIMUM OF 5 OBSERVATION WELLS
C********************************************************************************
       READ(57,221) NNO
221    FORMAT(5(1X,I4))
C********************************************************************************
C      READ LOCATION FOR OBSERVATION WELLS IN EACH ELEMENT
C********************************************************************************
       DO 223 I=1,NNO
        READ(57,222) XBAR(I),YBAR(I)
222     FORMAT (2(1X,F6.0))
223    CONTINUE
C********************************************************************************
C      READ ELEMENT NUMBERS FOR OB. WELLS
C********************************************************************************
       DO 224 I=1,NNO
        READ(57,225) NNE(I)
225     FORMAT(5(1X,I4))
224    CONTINUE
C********************************************************************************
C                                                                        MAN 910
C      READ NODAL COORDINATES FOR EACH NODE                              MAN 920
C                                                                        MAN 930
C********************************************************************************
CS       WRITE (56,36)                                                    MAN 940
CS   36 FORMAT (1H0,17HNODAL INFORMATION)                                 MAN 950
CS       WRITE (56,37)                                                    MAN 960
CS   37 FORMAT (1X,4HNODE,10X,1HR,10X,1HZ,10X,10HUNCONFINED,10X,16HSURFACEMAN 970
CS     1 RECHARGE,10X,13HCONSTANT HEAD)                                   MAN 980
       DO 40 J=1,NN                                                      MAN 990
       IQ(J)=0                                                           MAN1000
       READ (55,38) IND,RE(IND),ZE(IND),IFLUX(IND),IRCH(IND),NSCON(IND)  MAN1010
   38 FORMAT (I10,2F10.0,3I5)                                            MAN1020
CS       WRITE (56,39) IND,RE(IND),ZE(IND),IFLUX(IND),IRCH(IND),NSCON(IND) MAN1030
CS   39 FORMAT (1X,I5,2F10.2,3(15X,I5))                                   MAN1040
   40 CONTINUE                                                           MAN1050
C                                                                        MAN1060
C********************************************************************************
C      CHECK INPUT DATA FOR CONSISTENCY                                  MAN1070
C********************************************************************************
C                                                                        MAN1080
       CALL CHECK(NE,NN,NCH,NG,RE,ZE,NSCON,IERR)                         MAN1090
       IF (IERR.EQ.1) GO TO 91                                           MAN1100
C                                                                        MAN1110
C********************************************************************************
C      DEFINE DISCHARGING NODES                                          MAN1120
```

```
C*****************************************************************************
C                                                                      MAN1130
      IF (NQ.EQ.0) GO TO 46                                            MAN1140
      DO 41 K=1,NQ                                                     MAN1150
      IQ(NND(K))=1                                                     MAN1160
   41 CONTINUE                                                         MAN1170
C                                                                      MAN1180
C*****************************************************************************
C     DEFINE TOTAL TRANSMISSIVITY OF SCREEN LENGTH                     MAN1190
C*****************************************************************************
C                                                                      MAN1200
      TRTOT=0.                                                         MAN1210
      DO 45 J=1,NE                                                     MAN1220
      DO 44 I=1,3                                                      MAN1230
      IF (IQ(NG(J,I)).NE.1) GO TO 44                                   MAN1240
      IF (I.EQ.1) GO TO 42                                             MAN1250
      IF (IQ(NG(J,I-1)).NE.1) GO TO 44                                 MAN1260
      TI=PR(J)*ABS(ZE(NG(J,I))-ZE(NG(J,I-1)))                          MAN1270
      GO TO 43                                                         MAN1280
   42 IF (IQ(NG(J,3)).NE.1) GO TO 44                                   MAN1290
      TI=PR(J)*ABS(ZE(NG(J,1))-ZE(NG(J,3)))                            MAN1300
   43 TRTOT=TRTOT+TI                                                   MAN1310
   44 CONTINUE                                                         MAN1320
   45 CONTINUE                                                         MAN1330
C                                                                      MAN1340
C*****************************************************************************
C     DETERMINE HALF BAND WIDTH AND NEEDED MATRIX WIDTH                MAN1350
C*****************************************************************************
C                                                                      MAN1360
   46 IHBW1=0                                                          MAN1370
      DO 50 I=1,NE                                                     MAN1380
      DO 49 J=1,3                                                      MAN1390
      IF (J.EQ.1) GO TO 47                                             MAN1400
      IF (NSCON(NG(I,J-1)).EQ.1) GO TO 49                              MAN1410
      IF (NSCON(NG(I,J)).EQ.1) GO TO 49                                MAN1420
      IHBW=IABS(NG(I,J)-NG(I,J-1))                                     MAN1430
      GO TO 48                                                         MAN1440
   47 IF (NSCON(NG(I,1)).EQ.1) GO TO 49                                MAN1450
      IF (NSCON(NG(I,3)).EQ.1) GO TO 49                                MAN1460
      IHBW=IABS(NG(I,1)-NG(I,3))                                       MAN1470
   48 IF (IHBW.LT.IHBW1) GO TO 49                                      MAN1480
      IHBW1=IHBW                                                       MAN1490
   49 CONTINUE                                                         MAN1500
   50 CONTINUE                                                         MAN1510
      M=2*IHBW1+1                                                      MAN1520
      WRITE (56,51) IHBW1,M                                            MAN1530
   51 FORMAT (1H0,22HTHE HALF BAND WIDTH IS,I5,32H  AND THE WIDTH OF THEMAN1540
     1 MATRIX IS,I5)                                                   MAN1550
      DO 52 II=1,NN                                                    MAN1560
      S1(II)=0.0                                                       MAN1570
   52 CONTINUE                                                         MAN1580
      MM=NN-NCH                                                        MAN1590
C                                                                      MAN1600
C     SET UP PUMPING PERIOD                                            MAN1610
```

```
C
      TITO=0.0
      DO 90 NPP=1,IPP
      DO 53 II=1,NN
      F(II)=0.0
   53 CONTINUE
      READ (55,54) Q,QRCH,DELT,TMAXF,TSM,NTS
   54 FORMAT (5F10.0,I10)
      DT=DELT
      TM=0.0
      DO 55 I=1,NTS
      TM=TM+DT
      IF (TM.GE.TMAXF) GO TO 56
      DT=TSM*DT
   55 CONTINUE
      GO TO 57
   56 DELT=TMAXF/TM*DELT
      NTS=I
   57 WRITE (56,58) NPP,DELT,NTS
   58 FORMAT (1H1,14HPUMPING PERIOD,I10,/,20H INITIAL TIME STEP =,F10.5,
     15H DAYS,/,46H NUMBER OF TIME STEPS IN THIS PUMPING PERIOD =,I10)
      WRITE (56,59) Q
   59 FORMAT (1H0,11HDISCHARGE= ,E12.5,6H   CFS)
      WRITE (56,60) QRCH
   60 FORMAT (1H0,10HRECHARGE= ,F7.2,8H  FT/DAY)
      Q=Q*CONV1
C
C     VALUE OF SINK MATRIX
C
      QTOT=0.0
      DO 65 K=1,NE
      DO 61 J=1,3
      L(J)=NG(K,J)
      R(J)=RE(L(J))
      Z(J)=ZE(L(J))
   61 CONTINUE
      DO 64 I=1,3
      DO 63 J=1,3
      IF (IQ(L(J)).NE.1) GO TO 62
      IF (IQ(L(I)).NE.1) GO TO 62
      BLEN=SQRT((ABS(R(I)-R(J)))**2+(ABS(Z(I)-Z(J)))**2)
      F(L(I))=Q*PR(K)*BLEN/(2.*TRTOT)+F(L(I))
      QTOT=Q*PR(K)*BLEN/(2.*TRTOT)+QTOT
   62 IF (IRCH(L(J)).NE.1) GO TO 63
      IF (IRCH(L(I)).NE.1) GO TO 63
      AREA=3.1416*ABS(R(J)**2-R(I)**2)
      F(L(I))=QRCH*AREA/2.+F(L(I))
   63 CONTINUE
   64 CONTINUE
   65 CONTINUE
      DO 66 I=1,MM
      F2(I)=F(I+NCH)
   66 CONTINUE
C
```

```
C     CHECK DISCHARGE
C
      QTOT=QTOT/CONV1
      WRITE (56,67) QTOT
   67 FORMAT (1H0,28H CALCULATED WELL DISCHARGE =,F10.2,4H CFS)
C
C     START TIME LOOP
C*********************************************************************
      TIME=0.0
C*****************************************************************************
C     NIT= IS COUNTER FOR PRINT ROUTINE
C*****************************************************************************
      NIT=0
      DO 89 IT=1,NTS
C*****************************************************************************
C     INITIALIZE VARIABLES
C*****************************************************************************
C
      DO 69 I=1,MM
      DO 68 J=1,M
      WTRA1(I,J)=0.0
      WG1(I,J)=0.0
   68 CONTINUE
   69 CONTINUE
      DO 79 K=1,NE
C*****************************************************************************
C   ASSIGN SPECIFIC STORAGE FOR EACH ELEMENT DURING CONDUCTIVITY
C   MATRIX ASSEMBLY
C*****************************************************************************
      S=SS(K)
      DO 70 J=1,3
      L(J)=NG(K,J)
      R(J)=RE(L(J))
      Z(J)=ZE(L(J))
   70 CONTINUE
C
C     AVERAGE DISTANCE R OF ELEMENT
C
      RBAR=(R(1)+R(2)+R(3))/3.
C
      DO 72 I=1,3
      DO 71 J=1,3
      W(I,J)=0.0
   71 CONTINUE
   72 CONTINUE
      ZI(1)=Z(2)-Z(3)
      ZI(2)=Z(3)-Z(1)
      ZI(3)=Z(1)-Z(2)
      RI(1)=R(3)-R(2)
      RI(2)=R(1)-R(3)
      RI(3)=R(2)-R(1)
      DEL=(ZI(1)*R(1)+ZI(2)*R(2)+ZI(3)*R(3))/2.
C
C     CALCULATE ELEMENT MATRICIES
```

```
C
      DO 78 I=1,3
      DO 77 J=1,3
      WR(I,J)=TPI*RBAR*PR(K)*ZI(I)*ZI(J)/(4.*DEL)
      WZ(I,J)=TPI*RBAR*PZ(K)*RI(I)*RI(J)/(4.*DEL)
      IF (I.EQ.J) GO TO 73
      WST(I,J)=S*TPI*R(I)*DEL/(DELT*12.)
      GO TO 74
   73 WST(I,J)=S*TPI*R(I)*DEL/(DELT*6.)
   74 WW(I,J)=WR(I,J)+WZ(I,J)
      IF (I.NE.J) GO TO 76
      IF (IFLUX(L(I)).NE.1) GO TO 76
      AREA=0.0
      DO 75 KK=1,3
      IF (KK.EQ.I) GO TO 75
      IF (IFLUX(L(KK)).NE.1) GO TO 75
      AREA=3.1416*ABS(R(I)*R(I)-R(KK)*R(KK))
   75 CONTINUE
      W(I,J)=-SY*AREA/(2.*DELT)
C
C       ASSEMBLE ELEMENT MATRICIES INTO GLOBAL
C
   76 IF (NSCON(L(J)).EQ.1) GO TO 77
      IF (NSCON(L(I)).EQ.1) GO TO 77
      II=NG(K,I)-NCH
      JJ=NG(K,J)-NCH
      MTRAN=JJ-II+(M+1)/2
C
C       GLOBAL TRANSIENT MATRIX
C
      WTRA1(II,MTRAN)=WST(I,J)+W(I,J)+WTRA1(II,MTRAN)
C
C       GLOBAL MATRIX
C
      WG1(II,MTRAN)=WW(I,J)+WG1(II,MTRAN)
   77 CONTINUE
   78 CONTINUE
   79 CONTINUE
C
C
C       SUM OF GLOBAL AND TRANSIENT COEFFICIENT MATRICIES
C                ON LEFT HAND SIDE
C
      DO 81 I=1,MM
      DO 80 J=1,M
      WG1(I,J)=WTRA1(I,J)+WG1(I,J)
   80 CONTINUE
   81 CONTINUE
C
C       MULT. OF MATRICIES ON RIGHT HAND SIDE
C
      DO 82 I=1,MM
      S3(I)=S1(I+NCH)
      S5(I)=0.0
```

```
   82 CONTINUE
      CALL MLTBM(WTRA1,S3,S5,MM,M)
      DO 83 I=1,MM
      S5(I)=S5(I)+F2(I)
   83 CONTINUE
C
C      SOLUTION
C
      CALL SOLVE(1,WG1,S5,MM,IHBW1,MM,M)
      CALL SOLVE(2,WG1,S5,MM,IHBW1,MM,M)
      NNN=NCH+1
      DO 84 I=NNN,NN
      S1(I)=S5(I-NCH)
   84 CONTINUE
C
      TIME=TIME+DELT
      TIMM=TIME*1440.
      ACTT=TIME+TITO
C*******************************************************************************
C     WRITE OUT OBSERVATION WELL RESULTS AFTER EACH ITERATION
C*******************************************************************************
      DO 801 KK=1,NNO
      LN=NNE(KK)
       XBARE=XBAR(KK)
       YBARE=YBAR(KK)
      DO 802 JJ=1,3
       L(JJ)=NG(LN,JJ)
       R(JJ)=RE(L(JJ))
802    Z(JJ)=ZE(L(JJ))
      A1=R(2)*Z(3)-R(3)*Z(2)
      A2=R(3)*Z(1)-R(1)*Z(3)
      A3=R(1)*Z(2)-R(2)*Z(1)
      B1=Z(2)-Z(3)
      B2=Z(3)-Z(1)
      B3=Z(1)-Z(2)
      C1=R(3)-R(2)
      C2=R(1)-R(3)
      C3=R(2)-R(1)
      DELT2=(B1*R(1)+B2*R(2)+B3*R(3))
      N1=(A1+B1*XBARE+C1*YBARE)/DELT2
      N2=(A2+B2*XBARE+C2*YBARE)/DELT2
      N3=(A3+B3*XBARE+C3*YBARE)/(2*DELT2)
      PHI(KK)=N1*S1(L(1))+N2*S1(L(2))+N3*S1(L(3))
801   CONTINUE
C*******************************************************************************
C     PRINT OUT OBSERVATION WELL DATA FOR EACH ITERATION
C*******************************************************************************
      WRITE(56,850) ACTT
850   FORMAT(1X,'THE OBSERVATION WELL DATA AT TIME =',F10.3,' DAYS')
      DO 860 KK=1,NNO
       WRITE (56,851) KK,PHI(KK)
851    FORMAT(1X,'AT OBS WELL #',I2,2X,'THE DRAWDOWN IS ',F6.3,1X,
     1'FEET')
      POB(KK,IT)=PHI(KK)
```

```
      TOB(KK,IT)=ACTT*1440
860   CONTINUE
C*******************************************************************************
C   AN ARITHMETIC LOGICAL TEST FOR PRINTING OUTPUT FOR ENTIRE FLOW FIELD
C   EVERY TENTH ITERATION OR LAST ITERATION
C*******************************************************************************
      IF (IT .EQ. NTS) GO TO 998
      NIT=NIT+1
      NITT=NIT/10
      NIP=NITT*10
      IF(NIT .NE. NIP) GO TO 999
C                                                                       MAN3310
C       OUTPUT                                                          MAN3320
C                                                                       MAN3330
998   WRITE (56,85) ACTT                                                MAN3340
   85 FORMAT (1H1,30HTOTAL TIME IN THE SIMULATION =,F10.3,5H DAYS)      MAN3350
      WRITE (56,86) TIME,TIMM                                           MAN3360
   86 FORMAT (1H0,18HTHE DRAWDOWN AFTER,E10.3,9H  DAYS OR,E10.3,26HMIN. MAN3370
     1IN THE PUMPING PERIOD)                                            MAN3380
      WRITE (56,87)                                                     MAN3390
   87 FORMAT (1H0,10X,12H NODE NUMBER,5X,8HDRAWDOWN,10X,12H NODE NUMBER,MAN3400
     15X,8HDRAWDOWN,10X,12H NODE NUMBER,5X,8HDRAWDOWN)                  MAN3410
      WRITE (56,88) (I,S1(I),I=1,NN)                                    MAN3420
   88 FORMAT (10X,I5,10X,F10.3,10X,I5,10X,F10.3,10X,I5,10X,F10.3)       MAN3430
999   DELT=DELT*TSM                                                     MAN3440
   89 CONTINUE                                                          MAN3450
      TITO=TIME+TITO                                                    MAN3460
   90 CONTINUE                                                          MAN3470
      GO TO 93                                                          MAN3480
   91 WRITE (56,92)                                                     MAN3490
   92 FORMAT (1H1,49H**TERMINATION OF PROGRAM DUE TO INPUT DATA ERRORS) MAN3500
      GO TO 935
C*******************************************************************************
C  CALL DRAWDOWN PLOTTING ROUTINE FOR SIMULATION AND FOR TEST
C  DATA AT THE OBSERVATION WELL
C*******************************************************************************
93    DO 931 NPLT=1,NNO
931   CALL PTPLOT(NPLT,NTS)
      WRITE(56,993) (EPSI(I),I=1,NNO)
993   FORMAT(1X,5(/,F10.5))
935   STOP
      END                                                               MAN3520-
      SUBROUTINE CHECK(NE,NN,NCH,NG,RE,ZE,NSCON,IERR)                   CHK 010
      DIMENSION NG(NE,3),RE(NN),ZE(NN),NSCON(NN)                        CHK 020
C                                                                       CHK 030
C     THIS SUBROUTINE CHECKS THE ELEMENT INPUT DATA FOR CONSISTANCY     CHK 040
C                                                                       CHK 050
      IERR=0                                                            CHK 060
      NCK=0                                                             CHK 070
C                                                                       CHK 080
C     CHECK NUMBER AND ORDER OF CONSTANT HEAD NODES                     CHK 090
C                                                                       CHK 100
C     FIRST CHECK IF CONSTANT HEAD NODES ARE THE FIRST NODES NUMBERED   CHK 110
C                                                                       CHK 120
```

```
      DO 10 I=1,NCH
      NCK=NSCON(I)+NCK
   10 CONTINUE
      IF(NCK.EQ.NCH) GO TO 30
      IERR=1
      WRITE(56,20) NCH,NCK
   20 FORMAT(1H1,20H*** PROGRAM EXPECTED,I5,39H CONSTANT HEAD NODES BUT
     1ONLY THE FIRST,I5,28H  NODES WERE FLAGGED AS SUCH)
C
C     THEN CHECK THE TOTAL NUMBER OF CONSTANT HEAD FLAGS
C
   30 NCK=0
      DO 40 I=1,NN
      NCK=NSCON(I)+NCK
   40 CONTINUE
      IF(NCK.EQ.NCH) GO TO 60
      IERR=1
      WRITE(56,50)
   50 FORMAT(1H0,71H***TOTAL NUMBER OF CONSTANT HEAD FLAGS DOES NOT AGRE
     1E WITH NCH(# CODED))
C
C     NEXT CHECK IS TO INSURE THAT ALL ELEMENTS ARE NUMBERED
C     COUNTERCLOCKWISE
C
   60 DO 100 K=1,NE
      L=NG(K,1)
      M=NG(K,2)
      N=NG(K,3)
      A=(RE(L)-RE(N))*ZE(M)+(RE(M)-RE(L))*ZE(N)+(RE(N)-RE(M))*ZE(L)
      IF(A.LT.0.) GO TO 100
      IERR=1
      WRITE(56,70) K
   70 FORMAT(1H0,17H***EITHER ELEMENT,I5,23H  IS NUMBERED CLOCKWISE,/,11
     1X,92HOR THE VERTICAL COORDINATES ARE NOT POSITIVE DOWNWARD STARTIN
     2G WITH ZERO AT THE TOP BOUNDARY)
  100 CONTINUE
      RETURN
      END
      SUBROUTINE MLTBM(A,B,R,MM,M)
      DIMENSION A(MM,M),B(MM),R(MM)
C
C     MULT. OF A BANDED MATRIX(ORIGINALLY MM*MM)
C     WITH A VECTOR MATRIX
C     ( COMPACTED BANDED MATRIX OF M*MM AND VECTOR OF MM*1 )
C
      DO 1 I=1,MM
      R(I)=0.
      DO 2 J=1,M
      K=J+I-(M+1)/2
      IF(K.LT.1) GO TO 2
      IF(K.GT.MM) GO TO 2
      R(I)=A(I,J)*B(K)+R(I)
    2 CONTINUE
    1 CONTINUE
```

```
      RETURN
      END
      SUBROUTINE SOLVE(KKK,B,R,NEQ,IHALFB,NDIM,MDIM)
C     ********************************************
C
C    ASYMMETRIC BAND MATRIX EQUATION SOLVER
C     ORIGINALLY PROGRAMED BY JAMES O. DUGUID
C
C    KKK=1 TRIANGULARIZES THE BAND MATRIX B
C    KKK=2 SOLVES FOR RIGHT SIDE R, SOLUTION RETURNS IN R
C
      DIMENSION B(NDIM,MDIM), R(NDIM)
      NRS=NEQ-1
      IHBP=IHALFB+1
      IF (KKK.EQ.2) GO TO 30
C
C   TRIANGULARIZE MATRIX A USING DOOLITTLE METHOD
C
      DO 20 K=1,NRS
      PIVOT=B(K,IHBP)
      KK=K+1
      KC=IHBP
      DO 10 I=KK,NEQ
      KC=KC-1
      IF (KC.LE.0) GO TO 20
      C=-B(I,KC)/PIVOT
      B(I,KC)=C
      KI=KC+1
      LIM=KC+IHALFB
      DO 10 J=KI,LIM
      JC=IHBP+J-KC
   10 B(I,J)=B(I,J)+C*B(K,JC)
   20 CONTINUE
      GO TO 100
C
C   MODIFY LOAD VECTOR R
C
   30 NN=NEQ+1
      IBAND=2*IHALFB+1
      DO 70 I=2,NEQ
      JC=IHBP-I+1
      JI=1
      IF (JC.LE.0) GO TO 40
      GO TO 50
   40 JC=1
      JI=I-IHBP+1
   50 SUM=0.0
      DO 60 J=JC,IHALFB
      SUM=SUM+B(I,J)*R(JI)
   60 JI=JI+1
   70 R(I)=R(I)+SUM
C
C    BACK SOLUTION
C
```

```
      R(NEQ)=R(NEQ)/B(NEQ,IHBP)
      DO 90 IBACK=2,NEQ
      I=NN-IBACK
      JP=I
      KR=IHBP+1
      MR=MIN0(IBAND,IHALFB+IBACK)
      SUM=0.0
      DO 80 J=KR,MR
      JP=JP+1
   80 SUM=SUM+B(I,J)*R(JP)
   90 R(I)=(R(I)-SUM)/B(I,IHBP)
  100 RETURN
      END
C
C***********************************************************************
C  SUBROUTINE PTPLOT   WRITTEN BY A. PUCCI AND D. POPE 1/2/87
C                      USGS  WEST TRENTON ,NEW JERSEY
C                      (609) 771-3900
C***********************************************************************
      SUBROUTINE PTPLOT(NPLT,NTS)
C***********************************************************************
C PROGRAM TO PLOT PUMPING TEST DATA. FROM RADFLOW MODEL OUTPUT AT EACH
C OBSERVATION WELL, AND FROM THE OBSERVATION WELL DATA AT EACH SITE.
C***********************************************************************
C                            GLOSSARY
C
C   GXMAX,GYMAX,GYMIN,GXMIN = USER SUPPLIED LIMITS TO LOG-LOG GRAPH OF     ****
C                            WATER LEVEL CHANGES AGAINST REDUCED TIME     ****
C***********************************************************************
      CHARACTER INPUT*10,OUTPUT*10,TITLE1*50,TITLE2*50,ANS*1
      REAL GXMAX,GXMIN,GYMAX,GYMIN,XAXIS,YAXIS
      REAL POB(5,100),TOB(5,100),EPSI(5)
      INTEGER CHOICE
C GDIU          INITIALIZE VARIABLES FOR GDIU DEVICE CALLS
       CHARACTER*1   OPT          /*  DEVICE OPTION CODE
       CHARACTER*50  DES          /*  DEVICE DESCRIPTION
       CHARACTER*32  PALET        /*  DEVICE PALET
       LOGICAL       FLAGS(3)     /*  DEVICE OPTION FLAGS
C GDIU                            /*    1-PLOT FILE REQUIRED
C GDIU                            /*    2-DEVICE IS NOT A GRAPHIC TERMINAL
C GDIU                            /*    3-DEVICE HAS GRAPHIC INPUT
       REAL          SIZE(2)      /*  OPTIONAL MAX WIDTH AND HEIGHT OF PLOT
       CHARACTER*128 PLTFIL       /*  NAME OF PLOT FILE, IF NEEDED
       INTEGER       LUPLOT       /*  LOGICAL UNIT TO OPEN PLOT FILE ON
       INTEGER       NDEV         /*  DEVICE NUMBERS
       CHARACTER*8   COLOR        /*  PLOT COLOR
      COMMON /PT/ POB,TOB
      COMMON /EPS/ EPSI
C**********************************************************************
C**     EACH OUTPUTFILE WILL BE SEPARATE FOR AN OBSERVATION WELL
C**     WITH THE TEST DATA
C**********************************************************************
C
C*************************************************
```

```
C    TERMINAL PROMPTS TO USER FOR OUTPUT CONTROL
C*******************************************************
C**    OUTPUT FILE NAMES FOR EACH OBSERVATION WELL PLOT MUST BE UNIQUE
C
      PRINT *,'BEGIN PLOTTING SECTION'
      PRINT *,'     '
      PRINT *,'ENTER TITLE FIRST LINE'
      PRINT *,'FORMAT  ''title words$'''
      READ(1,*) TITLE1
      PRINT *,'ENTER TITLE SECOND LINE'
      PRINT *,'FORMAT '' second title$'''
      READ(1,*) TITLE2
      WRITE(1,*)' THE NEXT FOUR INPUTS: XMAX,XMIN,YMAX,YMIN, REFER'
      WRITE(1,*)' TO THE LOG CYCLES THAT ARE CHOSEN FOR PLOTTING '
      WRITE(1,*)' DATA AND CALCULATED DRAWDOWNS AT THE OBSERVATION'
      WRITE(1,*)' WELLS AS TYPE CURVES, USING LOG-LOG PLOTS.'
      WRITE(1,*)' USE 5 LOG CYCLES FOR X-AXIS, 3 FOR Y AXIS.'
      WRITE(1,*)' I.E. MAX X =0., MIN X=-5., MAX Y= 1., MIN Y=-2.'
      PRINT *,'MAX-X VALUE ='
      READ(1,*)GXMAX
      PRINT *,'MIN -X VALUE ='
      READ(1,*) GXMIN
      PRINT *,'MAX Y VALUE ='
      READ(1,*)GYMAX
      PRINT *,'MIN Y VALUE='
      READ(1,*) GYMIN
      PRINT *,'NAME THE INPUT FILE'
      READ'(A)', INPUT
C*******************************************************************************
C      OPEN OBSERVATION-WELL DATA FILE
C*******************************************************************************
      OPEN (61,FILE=INPUT,STATUS='OLD')
      PRINT *,'TYPE OF DRAWDOWN DATA?(1=DRAWDOWN,2=WATER LEVEL)'
      READ (1,*) CHOICE2
C*******************************************************************************
C     SET CONTROL COUNTER FOR EPSILON SUBROUTINE ,SO THAT THE EPSILON
C      STATISTIC IS CALCULATED ONLY ONCE.
C*******************************************************************************
      NEP=0
7     NEP=NEP+1
8     NDEV = 0
      LUPLT = 13
C**  SUBROUTINE TO QUERY USER FOR DEVICE
      CALL G_QRDV(NDEV)
C
C**  SUBROUTINE RETURNS ATTRIBUTES FOR SELECTED DEVICE
      CALL G_DVDF(OPT,DES,PALET,FLAGS,SIZE,-NDEV,IRTN)
      IF (IRTN .NE. NDEV) GO TO 8
C
C        SET FLAG SPECIFYING THAT A ZETA PLOT HAS BEEN PRODUCED
      IF(FLAGS(1)) THEN
        NOZETA = 1
        PRINT *,'TYPE IN THE NAME OF THE OUTPUT FILE'
        READ '(A)',OUTPUT
```

```
         OPEN(LUPLT,FILE=OUTPUT,STATUS='NEW')
        ELSE
         NOZETA = 0
       ENDIF
C** SUBROUTINE TO INITIALIZE SELECTED DEVICE   **
      CALL G_INIT(NDEV,1,1,LUPLT,IERR)
      IF(IERR.NE.0) STOP 'ERROR TRYING TO INITIALIZE GRAPHIC DEVICE'
C** SUBROUTINE TO  PUT TERMINAL IN GRAPHIC MODE   **
      CALL G_GRMO(NDEV,1,1,LUPLT,IERR)
      IF(IERR .NE. 0)STOP 'ERROR TRYING TO GO TO GRAPHIC MODE'
C
C      BASIC DISSPLA GRAPH SETTUP
      CALL PAGE(11.5,9.5)
      CALL AREA2D(9.0625,5.5)
      CALL HEADIN( TITLE1,100,1.4,2 )
      CALL HEADIN( TITLE2,100,1.2,2 )
      CALL XNAME('LOG T/R**2$',100)
      CALL YNAME('LOG S$',100)
10    CALL GRAF(GXMIN,1.,GXMAX,GYMIN,1.,GYMAX)
C********************************************************************************
C**     CALL DATA-PLOTTING SUBROUTINE
C********************************************************************************
      CALL PUMPLT(61,NPLT,NTS,CHOICE2,N)
      CALL ENDPL(0)
C**  RETURN SCREEN TO TEXT MODE
      CALL G_TXMO(NDEV,1,1,LUPLT,IERR)
C**  GDIU CALL WHICH IS THE EQUIVALENT OF THE DISSPLAY DONEPL CALL
      CALL G_DNPL(NDEV,1,1,LUPLT,IERR)
C********************************************************************************
C        CALL SUBROUTINE EPSILON IF THIS IS THE FIRST PLOT OF THIS DATA
C********************************************************************************
      IF (NEP .EQ. 1) THEN
       CALL EPSILON(NPLT,N,NTS)
       WRITE(1,*) 'EPSI= ',EPSI(NPLT)
      ENDIF
C********************************************************************************
C      GIVES OPTION TO MAKE A ZETA PLOT IF A PLOT TO THE TERMINAL WAS
C          PRODUCED FIRST
C********************************************************************************
      IF (NOZETA .EQ. 1) GO TO 40
      WRITE(1,*) 'DO YOU WANT ANOTHER PLOT? (Y OR N)'
      READ(1,20) ANS
20    FORMAT(A1)
      IF (ANS .EQ. 'N') GO TO 40
       CLOSE (13)
       REWIND (61)
       GO TO 7
40    CLOSE (61)
      RETURN
      END
C
C
C***************************************************************************
      SUBROUTINE PUMPLT(F,NPLT,NTS,CHOICE2,N)
```

```
C***************************************************************************
C
C***************************************************************************
C**                          GLOSSARY
C**
C**    XT,YT = LOGARITHMS OF OBSERVED REDUCED TIME AND RESIDUAL WATER
C**            LEVELS FOR EACH OBSERVATION WELL
C**   XPT,YPT = LOGARITHMS OF SIMULATED REDUCED TIME AND RESIDUAL WATER
C**             LEVELS FOR EACH OBSERVATION WELL
C**   N = NUMBER OF WATER LEVEL OVSERVATIONS
C**   R = DISTANCE OF OBSERVATION WELLS FROM PUMPING WELL
C**   RE = REFERENCE ELEVATION FOR WATER LEVEL ELEMENTS (0 IS DUMMY
C**        VALUE FOR DATA WITH RESIDUAL WATER LEVELS
C***************************************************************************
      INTEGER CHOICE2,N,F
      REAL S(100),T(100),XT(100),YT(100),R,RE
      REAL XPT(100),YPT(100)
      REAL POB(5,100),TOB(5,100)
      COMMON /PT/POB,TOB
      COMMON /TS/ T,S
C***************************************************************************
C      READ ACTUAL TEST DATA FROM DATA FILE
C***************************************************************************
      READ(F,10) N,R,RE
10    FORMAT(1X,I3,1X,F5.0,1X,F5.1)
      WRITE(1,10) N,R,RE
      DO 20 I=1,N
       READ(F,15) T(I),S(I)
C      WRITE(1,15)T(I),S(I)
15     FORMAT(F10.0,F10.3)
20     CONTINUE
C***************************************************************************
C     WHAT KIND OF DRAWDOWN?
C***************************************************************************
      TP=0.0
      IF (CHOICE2.EQ.1) GO TO 50
      TP=RE
50    DO 105 J=1,N
       S(J)=S(J)-TP
       XT(J)=LOG10(T(J)/(R*R))
105    YT(J)=LOG10(S(J))
C***************************************************************************
C     PREPARE MODEL DATA FOR PLOTTING
C***************************************************************************
      DO 106 J=1,NTS
       XPT(J)=LOG10(TOB(NPLT,J)/(R*R))
106    YPT(J)=LOG10(POB(NPLT,J))
C***************************************************************************
C     PLOT OBSERVED TEST DATA
C***************************************************************************
      CALL MARKER(16)
      CALL CURVE(XT,YT,N,-1)
C***************************************************************************
C     PLOT MODEL OUTPUT DATA
```

```
C***************************************************************************
      CALL MARKER(2)
      CALL CURVE(XPT,YPT,NTS,+1)
      RETURN
      END
C
C
C***************************************************************************
      SUBROUTINE EPSILON(NPLT,N,NTS)
C
      REAL POB(5,100),TOB(5,100),T(100),S(100),EPSI(5)
      COMMON /PT/POB,TOB
      COMMON /TS/T,S
      COMMON /EPS/ EPSI
C***************************************************************************
C 'NPLT' IS THE OBSERVATION WELL BEING PLOTTED
C 'N' IS THE NUMBER OF DATA POINTS AT EACH OBSERVATION WELL
C 'NTS' IS THE NUMBER OF CALCULATION POINTS FROM RADFLOW
C THE FIRST DATA POINT SHOULD BE AFTER THE FIRST CALCULATED POINT
C***************************************************************************
      DO 10 K=1,N
       DO 15 J=1,NTS
           TOBJ=TOB(NPLT,J)
           TK=T(K)
           TEST=TOBJ-TK
           IF(TEST.GE.0.) THEN
           GO TO 11
           END IF
15    CONTINUE
      IF (TOB(NPLT,J) .LT. T(K)) RETURN
C***************************************************************************
C    INTERPOLATE SIMULATED RESIDUAL WATER LEVELS TO SAME TIME AS OBSERVED
C     WATER LEVELS FOR COMPARISON
C***************************************************************************
11    IF (J .EQ. 1) GO TO 10
           T1=TK-TOB(NPLT,J-1)
      T2=TOBJ-TOB(NPLT,J-1)
           TT=T1/T2
           HIP=POB(NPLT,J)-POB(NPLT,J-1)
           HIP=HIP*TT + POB(NPLT,J-1)
           EPSI(NPLT)=EPSI(NPLT)+(HIP - S(K))**2/S(K)**2
10         CONTINUE
      RETURN
      END
```


☆ U.S. GOVERNMENT PRINTING OFFICE: 1988-0-512-443