Ø-315-24764-9

National Library of Canada / Bibliothèque nationale du Canada

Canadian Theses Division / Division des thèses canadiennes

Ottawa, Canada
K1A 0N4

PERMISSION TO MICROFILM — AUTORISATION DE MICROFILMER

- Please print or type — Écrire en lettres moulées ou dactylographier

Full Name of Author — Nom complet de l'auteur

GREGORY JOHN SMITH

Date of Birth — Date de naissance	Country of Birth — Lieu de naissance
AUGUST 13, 1955	CANADA

Permanent Address — Résidence fixe

820 PANORAMA DRIVE, APARTMENT 1C
PALATINE, ILLINOIS, U.S.A.
60067

Title of Thesis — Titre de la thèse

FINITE ELEMENT PRE-PROCESSOR FOR HYDROGEOLOGIC MODELLING

University — Université

UNIVERSITY OF ALBERTA

Degree for which thesis was presented — Grade pour lequel cette thèse fut présentée

MASTER OF SCIENCE

Year this degree conferred — Année d'obtention de ce grade	Name of Supervisor — Nom du directeur de thèse
1984	Dr. F.W. Schwartz

Permission is hereby granted to the NATIONAL LIBRARY OF CANADA to microfilm this thesis and to lend or sell copies of the film.

The author reserves other publication rights, and neither the thesis nor extensive extracts from it may be printed or otherwise reproduced without the author's written permission.

L'autorisation est, par la présente, accordée à la BIBLIOTHÈQUE NATIONALE DU CANADA de microfilmer cette thèse et de prêter ou de vendre des exemplaires du film.

L'auteur se réserve les autres droits de publication; ni la thèse ni de longs extraits de celle-ci ne doivent être imprimés ou autrement reproduits sans l'autorisation écrite de l'auteur.

Date	Signature
October 3, 1984	Gregory Smith

NL-91 (4/77)

THE UNIVERSITY OF ALBERTA

Finite Element Pre-processor for Hydrogeologic Modelling

by

Gregory J. Smith

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES AND RESEARCH

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE

OF Master of Science

The Department of Geology

EDMONTON, ALBERTA

Fall, 1984

THE UNIVERSITY OF ALBERTA

RELEASE FORM

NAME OF AUTHOR Gregory J. Smith

TITLE OF THESIS Finite Element Pre-processor for Hydrogeologic Modelling

DEGREE FOR WHICH THESIS WAS PRESENTED Master of Science

YEAR THIS DEGREE GRANTED Fall, 1984

Permission is hereby granted to THE UNIVERSITY OF ALBERTA LIBRARY to reproduce single copies of this thesis and to lend or sell such copies for private, scholarly or scientific research purposes only.

The author reserves other publication rights, and neither the thesis nor extensive extracts from it may be printed or otherwise reproduced without the author's written permission.

(SIGNED) Gregory J. Smith

PERMANENT ADDRESS:

520 Panorama Drive 1C

Palatine, Illinois

60067

DATED October 11 1984

THE UNIVERSITY OF ALBERTA

FACULTY OF GRADUATE STUDIES AND RESEARCH

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research, for acceptance, a thesis entitled Finite Element Pre-processor for Hydrogeologic Modelling submitted by Gregory J. Smith in partial fulfilment of the requirements for the degree of Master of Science.

Frank W. Schwartz
Supervisor

H. A. K. Charlesworth

Douglas ...

D. S. Chanasyk

Date October 9, 1984

ABSTRACT

An interactive computer package for generating a finite element mesh has been developed as a preprocessor for two and three-dimensional aquifer models. This scheme has the capability to generate triangular and quadrilateral elements in two-dimensions and tetrahedral and parallelepiped elements in three-dimensions. It is based on the Isoparametric mapping of quadrilaterals and can be used on flow domains that are described by both regular and irregular boundaries. The system is completely user interactive, utilizing a digitizing tablet and a colour graphics terminal to involve the user in all phases of the mesh generation, and also decrease the time and effort that is involved in more traditional methods of mesh construction. There are facilities to incorporate pre-positioned pumping wells and observation wells into the mesh, as well as constant head nodes, boundary fluxes and leakance elements. Output from the package is stored in a file in a format that is easily adaptable to most popular finite element codes. Some features of the code include: the ability to produce a hard copy of the mesh in any size within the limitations of the plotting hardware, and an optimization algorithm that renumbers nodes so as to increase computational efficiency. In addition to a description of the code, guidelines and hints are given for dealing with unusual shaped domains and for improving mesh design, and examples of the application of the package to a

variety of problems are presented.

ACKNOWLEDGEMENTS

The assistance provided by various people in the preparation of this thesis is gratefully acknowledged. Dr. F.W. Schwartz, the supervisor of this thesis and the person responsible for thinking up such practical research as this, provided much help throughout the writing of this work. I would like to thank the members of my examining committee, Dr. J. Toth, Dr. H.A.K. Charlesworth, Dr. D. Hackbarth and Dr. D.S. Chanasyk for their useful suggestions. Last, but not least, I would like to thank my wife, Laurie, for helping me through the past two years and supporting me to make all of this possible.

Table of Contents

LIST OF TABLES

Table	Description	Page

LIST OF FIGURES

1. INTRODUCTION

Finite element methods for solving groundwater flow problems require the discretization of the flow domain, which involves the partitioning of the system under consideration into a grid or a mesh. Typically the mesh is constructed by hand through drawing a grid over the flow domain and manually entering the mesh parameters into a data set. This procedure, especially when one encounters more complex flow systems is very tedious, time-consuming and subject to errors (Klienstreurer, 1980).

The need to automate this process is therefore apparent (Narasimhan and Witherspoon, 1980, and Klienstreurer, 1980). Techniques based on point digitizers have proven useful in reducing the time for data entry of a manually prepared grid. However, there has been relatively little work in adapting fully automatic methods used for example in structural engineering to hydrogeologic problems. These schemes have been widely used, because they provide the ability to quickly and accurately change the density of the mesh to provide an optimal solution to a given problem.

In the fields of structural and automotive engineering the problem of automatic mesh generation appears somewhat less complex than for hydrogeological applications because of regular external and internal geometries. In hydrogeologic applications, mesh generation is complicated by the irregular geometry of the flow domain. Specifically one has to deal with:

-complex internal geometries, where it is necessary to model the boundaries of lithologic units;

-complex external geometries, as for example in the case of buried channels;

-providing specific locations for observation and pumping wells that must correspond to nodal locations; and

-the necessity of compressing the grid around a pumping well to increase the mathematical accuracy in an area where the hydraulic head may be changing quite markedly.

Such constraints on the mesh to be generated are usually not considered in the schemes presently available today. Existing techniques applied to groundwater modelling have been largely problem or program specific (Klienstreurer, 1980). These schemes generally involve fitting the flow domain to an idealized grid of regularly positioned elements. This kind of constraint usually makes it difficult to accurately model an irregular flow domain.

A more general procedure for mesh generation in a pre-processor form has been developed by Sartori et al. (1982). The meshes illustrated in their work appear to be accurate and well conditioned. This scheme fits a regularly positioned grid upon an irregular geometry such that elemental boundaries do not necessarily correspond to transmissivity boundaries.

Thus, there is still a need in hydrogeological applications to extend existing procedures for mesh generation. The proven techniques in the field of structural engineering provide the logical starting point. Although many schemes exist for mesh generation, they are generally based on the following basic philosophies:

1. the mesh pattern is established by the computer from a minimal amount of information supplied in digital form; and
2. the positioning of the mesh is established by graphical-computer interaction using plotting tables or direct light pen displays (Zienkiewicz and Phillips, 1971).

The goal of my work is to extend these basic philosophies of mesh generation to hydrogeological problems. Specific objectives are: (1) to develop a mesh generator, capable of constructing a network of elements in two- or three-dimensions a level of complexity appropriate for most problems encountered in aquifer hydraulics, (2) to demonstrate the usefulness of this approach in solving practical problems, and (3) to demonstrate how grid design can improve finite element simulations.

The scheme for mesh generation used in this work is discussed in Chapter 2. It is based on an Isoparametric method developed by Zienkiewicz and Phillips (1971). The system is designed for maximum flexibility, user interaction, and is capable of generating triangular and quadrilateral shaped elements in a plane and projecting these elements in the third dimension to produce tetrahedral and parallelepiped (brick type) elements. The mesh can be layered and hence can account for the patterns of geologic layering with relative ease.

The demonstration of the mesh generation scheme provided in Chapter 2 involves the simulation of hydraulic head distributions for both a two- and three-dimensional problem. The mesh for the two-dimensional case is prepared both manually and with the mesh generation program. The results of these two simulations are compared to verify the automatic procedure and to demonstrate its utility. The three-dimensional simulation is a more complex case designed to show how the mesh generation procedure can be applied to layered geological systems.

2. DESIGN AND IMPLEMENTATION OF A FINITE ELEMENT MESH GENERATOR

2.1 EXISTING METHODS

Techniques for automatic mesh generation have been in existence in their most basic form since the late 1960's. Thwacker (1980) provides an exhaustive review of these mesh generation techniques, including an outline of the basic theory, and an analysis of the drawbacks of each scheme. Current mesh generators employ one or a combination of two of the following schemes:

1. automatic triangulation;
2. coordinate transformations;
3. smoothing procedures; or
4. blending functions (Yerry and Shephard, 1983).

The automatic triangulation technique, a scheme demonstrated in the works of Cavendish (1974), Sadeck (1980), and Suhara and Fukuda (1972), allows the greatest flexibility in modelling a very general region. Cavendish (1974) uses a random number generator to fill the computational domain with nodes to some specified density. From this population of nodes, nodes are selected to form elements in a well conditioned mesh. Nodes that are not selected are extraneous and ignored.

Sadeck (1980) takes this approach a step further. Instead of randomly generating the coordinates of the nodes, the nodes are located on the basis of the position of regularly spaced nodes around the boundary of the region in which the mesh is being constructed to form a well conditioned mesh. This procedure eliminates the need to generate the nodes first and then sort through them to produce the best conditioned element from the distribution of nodes.

The method used by Sadeck (1980) works well for regular geometric shapes, or simple forms with irregular geometry. For regions with complex geometry, which are discretized usi[illegible]drilateral elements, this method tends to form poor[illegible]nditioned elements especially where the boundary forms an acute angle. My experience with this method has shown that, although relatively simple to program it does require a substantial number of support subroutines to handle complex shapes. Yerry and Shephard (1983) state that although the triangulation algorithms have been developed that will work for general two-dimensional shapes, they have not proven popular because they occasionally produce ill-conditioned meshes and in many cases require substantial user interaction.

Coordinate transformations are used to map geometrically complex figures to a regular grid in the transformed domain. Having defined the transformation, the inverse transformation can be used to revert to the original domain. A disadvantage of this technique is that although the mesh is apparently well conditioned, depending on the coordinate transformation the mesh in reality may be very distorted. As well, another disadvantage of this scheme is that the coordinate transformation must be provided before the computation can begin. This requirement is analagous to constructing an irregular grid (Thwacker, 1980).

Isoparametric methods (Zienkiewicz and Phillips, 1971) or, smoothing procedures as Yerry and Shephard (1983) describe them, involve the smooth distribution of nodes within the computational domain. The positioning of the interior nodes is determined by the mean of the coordinates of its neighbours. This condition is equivalent to an optimal grid, which corresponds to the equilibrium condition of a system of springs connecting the nodal points (Thwacker, 1980). This scheme requires that opposite boundaries of the domain have the same number of elements. This constraint is a disadvantage because the user can not easily change the density of the elements within the domain.

2.2 DESIGN OF A FINITE ELEMENT MESH GENERATOR FOR TWO- AND THREE-DIMENSIONAL PROBLEMS

My design for an automatic mesh generator for three-dimensional groundwater flow problems is based upon the work of Zienkiewicz and Phillips (1971), which involves the application of isoparametric coordinates, or smoothing procedures as Yerry and Shephard (1983) had described them in the previous section. The work by Zienkiewicz and Phillips (1971) deals mainly with two-dimensional meshes, and indicates that the method can be easily adapted to three-dimensional problems.

The code that I have developed has the capability of generating finite element meshes in both two- and three-dimensions. It discretizes a domain using either quadrilateral or triangular elements in two-dimensions, and parallelepiped or tetrahedral elements in three-dimensions.

To facilitate both the rapid and simple entry of data into the code, a menu oriented approach has been implemented using a Summagraphics Corp. ID series tablet digitizer to input the bulk of the digital information. This approach greatly simplifies digital input of coordinate information, and provides the user with flexibility in controlling the pattern of mesh generation.

Coupled to the digitizer is an Advanc[illegible]ectronics Design AED512 colour graphics terminal th[illegible]letely supports the graphics packages necessary to [illegible]ay and manipulate the networks that are generated. The [illegible]ted

Graphics (Davis, 1977) system is a library of FORTRAN-callable subroutines for performing graphics operations. These systems are all supported by the Amdahl 5860 mainframe at the University of Alberta.

With this hardware, the amount of data that must be input manually can be greatly minimized to generate a mesh consistent with the following criteria:

1. the geometry of the boundaries, whether singly or multiply connected, must be accurately described;
2. a facility for describing the spatial variability in the hydraulic properties of the geologic material, wherever these occur;
3. a facility for grading the mesh so as to achieve different kinds of mesh compression in specific sub-areas of the domain;
4. an achievement of shapes which are approximately equidimensional and do not lead to numerical ill-conditioning;
5. a renumbering system which results in the best computational efficiency (Zienkiewicz and Phillips, 1971; Collins, 1973).

The boundary of the domain or sub-domain in any plan view can be adequately described by digitizing points around this boundary. Zones of differing transmissivities are digitized separately, with the program establishing the required continuity in the mesh geometry along boundaries. The manner in which the other three criteria are implemented in the procedure for mesh generation will become apparent in the following sections.

The mesh generating scheme is based on the isoparametric mapping of quadrilaterals. For any shaped domain the coordinates of the interior nodes can be described by the following equations (Zienkiewicz and Phillips, 1971):

$$x=N_i x_i$$
$$y=N_i y_i$$
$$z=N_i z_i \qquad (1)$$

where: N_i = the shape factor determining the placement the nodes around the domain boundary.

x_i = x-coordinate of a regularly spaced node being mapped onto the computational domain.

y_i = y-coordinate of a regularly spaced node being mapped onto the computational domain.

z_i = z-coordinate of a regularly spaced node being mapped onto the computational domain.

The shape factor N_i is described in terms of a coordinate transformation based upon the geometry of the domain boundary. In the field of hydrogeology, the geometry of the flow domain is not easily described as a simple equation.

The approach taken in this thesis involves defining the boundaries of the figure by a series of regularly spaced nodes. The spacing of these nodes is based on the desired distribution of elements within the figure. In three-dimensions the nodes are regularly spaced along the twelve edges or folds of the figure.

To obtain a regular spacing along an irregular boundary, a method based on Lagrange polynomials (Hornbeck, 1975) is incorporated into the code. Essentially this Lagrangian technique produces polynomials that are "tied down" (Hornbeck 1975) at the data points given in the digitization and interpolates positions in between. In the code, boundaries of the domain or sub-domain are represented by equations of up to a third order polynomial, which can represent most geometries that may be encountered.

The Lagrangian method interpolates evenly spaced points from an irregular distribution of sample points. For this method, consider a series of points described by a function $f(x_i)$ where the x_i are in general not evenly spaced and i can take on integer values 0 to n (meaning there are n+1 data points). To fit a polynomial interpolation between the x_i's is to fit a polynomial of degree n to these n+1 points. The mechanics of this method in this code is a moving average approach through the data points as a constantly changing polynomial. That is, utilizing four data points at a time, the code calculates evenly spaced points within a range represented by these points. As the point generation progresses, the sample points change to keep the generated points within the range represented represented by the four data points.

Because this scheme is based on the mapping of quadrilaterals, it is necessary to designate four corner points (in three-dimensions, eight corner points) of the irregular figure to obtain an equal number of regularly spaced nodes on opposite sides of the figure. With the digitized figure displayed on the graphics terminal, the user, by positioning the cursor defines the four (or eight) corners of the figure. In dealing with irregular geometries, corner points are not immediately obvious but must be designated in a manner to provide an even distribution of elements as illustrated in Figure 1.

With the boundary nodes evenly spaced around the boundary of the domain under consideration, the boundaries are completely defined but not in the equation form as in the work of Zienkiewicz and Phillips (1971). The distribution of the interior nodes becomes based on the distribution of the boundary nodes and may be described by the following set of equations:

$$N_x=(x_1-x_2)/NELM_x$$
$$N_y=(y_1-y_2)/NELM_y$$
$$N_z=(z_1-z_2)/NELM_z \qquad (2)$$

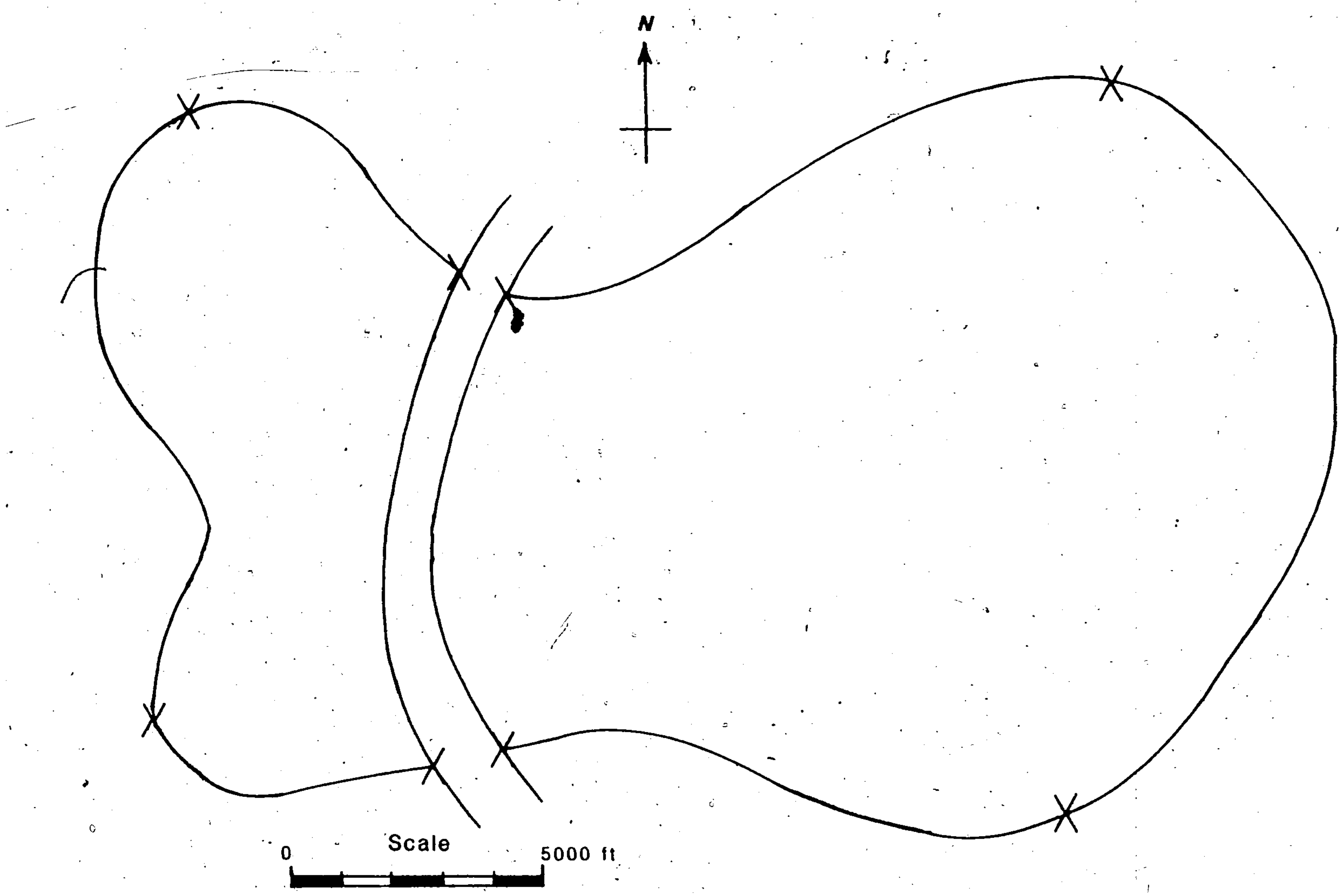

Figure 1. Flow Domain illustrating designated corner points. Corner points are shown with an X.

where: N_x = the factor determining the distribution of the nodes in the x-direction.

N_y = the factor determining the distribution of the nodes in the y-direction.

N_z = the factor determining the distribution of the nodes in the z-direction.

x_1 & x_2 = corresponding x-coordinates of nodes on opposite sides of the domain.

y_1 & x_2 = corresponding y-coordinates of nodes on opposite sides of the domain.

z_1 & z_2 = corresponding z-coordinates of nodes on opposite sides of the domain.

$NELM_x$ = the number of elements to be generated in the x-direction.

$NELM_y$ = the number of elements to be generated generated in the y-direction.

$NELM_z$ = the number of elements to be generated in the z-direction.

As the nodes are being generated, the elemental incidences are being described simultaneously.

2.3 DESCRIPTION OF THE PROGRAM

The mesh generation program consists of 13 parts: the main program and 12 subroutines: SMOOTH, DSTRB, NODE, ELMNT, SETUP, OPTNUM, NDEGEN, INTANG, ASKPLT, CNCTQ, CNCTT1, CNCTT2, as well as the Integrated Graphics Library Subroutines. The program is presently dimensioned to generate 5000 nodes and 3000 elements, which should be more than sufficient for most problems.

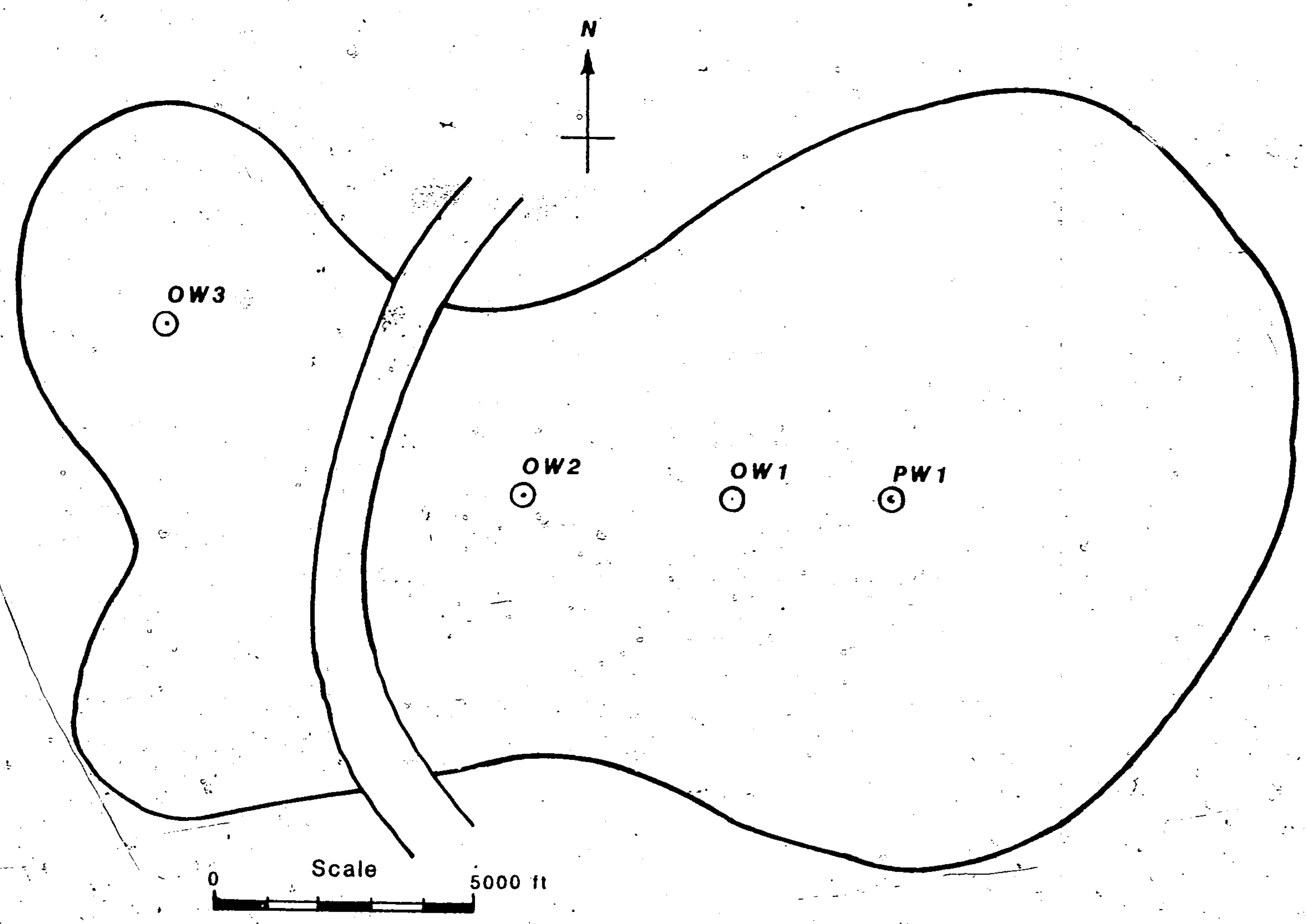

Figure 2. Example 2-D problem.

MENU

13	Generate a Mesh(after digitizing)
12	Output Data for Later Use
11	Input Boundary Fluxes
10	Future Use
9	Digitize a Transmissivity Boundary
8	Digitize an Observation Well
7	Digitize a Pumping Well
6	"Pic" Constant Head Nodes
5	"Pic" Leakance element
4	Remove Last Transmissivity Zone Generated
3	Remove Last Point Digitized
2	Stop/Begin Again
1	Re-display Menu

Figure 3. Menu illustrating options available to a user to generate a mesh.

The MAIN program initializes all the indices used in the entire program, reads in the title of the simulation and interactively sets the basic options of the program. These options determine: whether one is working in English or metric units; the scale; whether it is to be a two-dimensional or three-dimensional simulation; type of elements required; and if in three-dimensions whether a steady state or transient simulation is to be performed. The MAIN program also contains the Menu from which any of the other subroutines will be called. This is implemented as a series of 13 switches that may be accessed at any time during the construction of the mesh in a manner outlined in the flow chart, Figure 4. The 13 switches allow the user to perform a variety of tasks to synthesize a mesh.

Switch No. 1 simply re-displays the menu. This feature was incorporated into the design of the program to provide the user with an easy way to recall all of the options. Carrying most of the operating instructions with the program eliminates the need for a paper copy of the menu that could be misplaced.

Switch No. 2 activates a built-in procedure to restart or terminate the program. The instructions outlined in the user's guide in Appendix A explains how to start the program. However because the user has options throughout the entire program that may never be used, an on/off switch is provided to terminate the program.

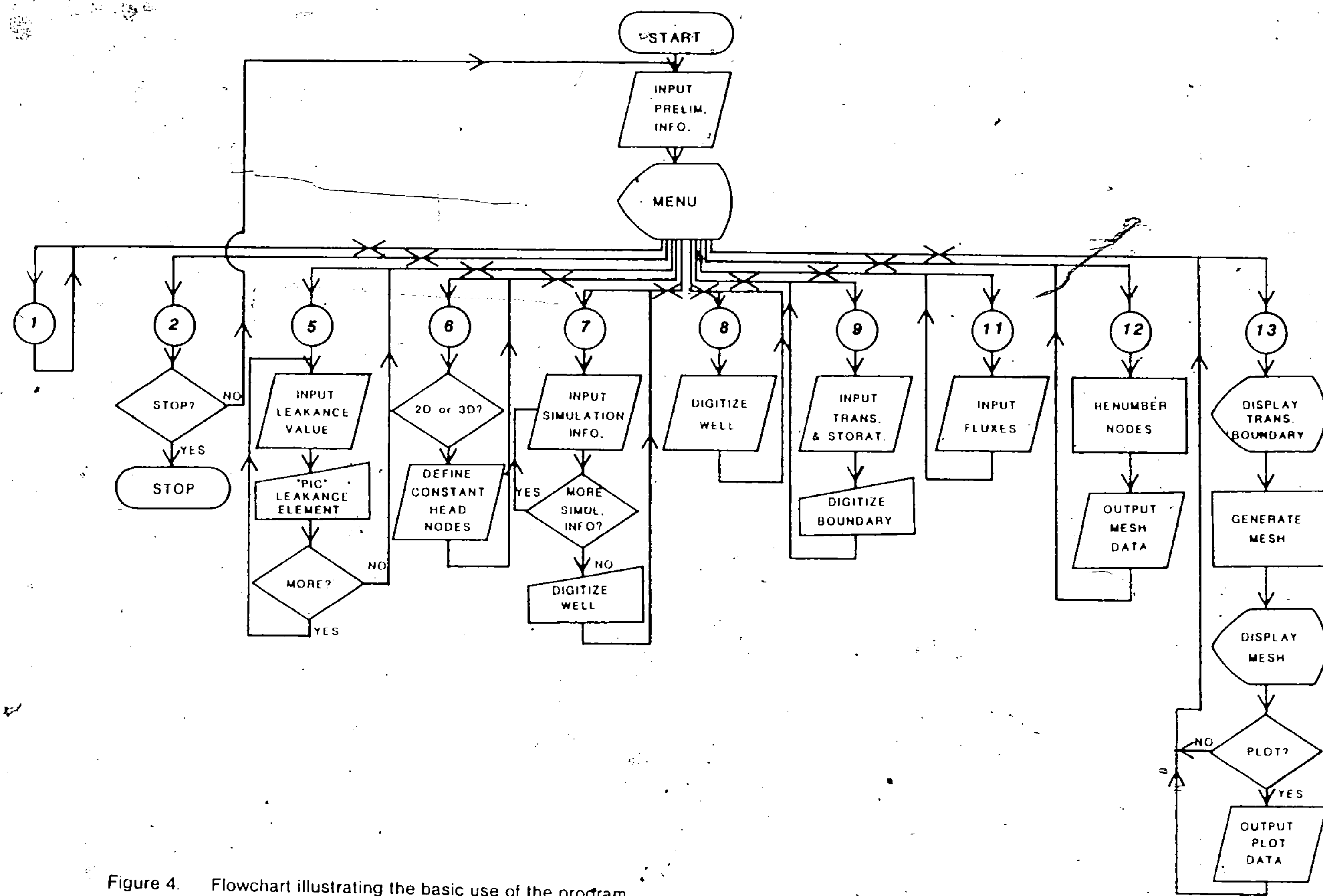

Figure 4. Flowchart illustrating the basic use of the program.

Switch No. 3 deletes the previously digitized point. This feature is incorporated so that one could take care of such hardware problems such as a faulty digitizer switch, static on the line and human error.

Switch No. 4 allows one to remove the mesh generated for the last transmissivity zone incorporated into the main mesh. This feature avoids repeating large amounts of data entry in the case where a mistake in the digitizing process goes unnoticed.

Switch No. 5 allows the user to define leakance elements, (ie: recharge associated with a river). This feature only works in two-dimensions, using a graphic routine to pick out particular elements ("Pic" routine) with the program prompting a numerical value of leakance.

Switch No. 6 defines constant head nodes in two-dimensions through a "Pic" routine and in three-dimensions constant head boundary faces. The user cannot define individual constant head nodes in three-dimensions with this program because the "Pic" routine cannot easily define the z-coordinate.

Switch No. 7 defines the location of a pumping well in the x-y plane. As well the program prompts the user to input information relating to the pump schedule. The position of the well is incorporated into the mesh as a node and in three-dimensions is continuous throughout all layers subsequent to its initial nodal definition.

Switch No. 8 defines the location of an observation well in a similiar manner to the pumping well.

Switch No. 9 allows the user to define the boundaries of a transmissivity zone with the program prompting for values of storativity and transmissivity (in three-dimensions, elemental hydraulic conductivities in x-,y- and z-directions).

Switch No. 10 is designated for future use. It was originally used to renumber the nodes to reduce bandwidth of the mesh, but this feature was subsequently incorporated into Switch No. 12.

Switch No. 11 designates boundary fluxes across the entire face of a mesh. It was felt that a boundary flux feature was needed in the mesh generation scheme, but to "Pic" out individual elements would be too costly and time consuming.

Switch No. 12 renumbers the nodes of the mesh to reduce bandwidth (Collins, 1973) and outputs the information in a format that is compatible with the Verge-Frind (Verge, 1972) two-dimensional code dealing with transient simulations for confined aquifers. In three-dimensions, a format compatible with Verge's (1975) three-dimensional groundwater flow model with minor changes in the input. The numerical information is output to device 7 and the plotting information to device 9. The workings of Collins' (1973) renumbering scheme will be discussed in Section 4.

Switch No. 13 activates the mesh generation procedure. This procedure is interactive with the user, allowing him to define the configuration of the mesh and the elemental density. If any pumping wells are present, the program will concentrate elements around the well and smooth the distribution before it is displayed on the terminal. Having displayed the mesh on the terminal, the user can rotate it to better visualize its faces and plot it on a hardcopy plotter.

The individual subroutines used in the code correspond to no particular switch discussed previously. A switch may have as many as five subroutines associated with it or it may have none. For this reason, the function of each subroutine will be discussed briefly.

Subroutine SMOOTH, used in conjunction with subroutines CNCTQ, or CNCTT1, or CNCTT2, concentrates elements around pumping wells and smooths the distribution of elements in the vicinity of the wells. Subroutine CNCTQ applies to quadrilateral and "brick type" elements around wells, whereas CNCTT1 and CNCTT2 apply to triangular and tetrahedral elements.

The concentrating subroutines CNCTQ, CNCTT1, or CNCTT2 partition the previously generated elements around a pumping well into smaller elements. The smaller elements are numbered and the incidences defined. The incidences of the elements around these partitioned elements also have their incidences redefined. A large part of these subroutines can be thought of as being largely devoted to bookeeping duties in the storage of elemental incidences.

Subroutine SMOOTH takes the coordinates of the newly generated nodes in CNCTQ, CNCTT1, or CNCTT2 and repositions them using an averaging technique based on the position of the surrounding nodes. This provides for a smooth distribution of elements in the vicinity of the pumping well.

Subroutine DSTRB re-distributes the digitized points around a perimeter of the transmissivity zone using the Lagrange interpolation routine (Hornbeck, 1975)

Subroutine NODE uses the redistributed nodes' positions to produce nodes within the interior of the transmissivity zone, with the elemental incidences being described by calls to subroutine ELMNT.

Subroutine SETUP and OPTNUM are based on algorithms developed by Collins (1973) and modified for use in this program. SETUP, as the name implies, sets up arrays that describe the interconnection relationship of each node. The purpose of these arrays is to make it easier for the renumbering routine to know which nodes are interconnected, hence the differences in numbering these nodes should be small. OPTNUM takes the results from SETUP and renumbers the nodes in the mesh based on each node's connection with its neighbours so that the bandwidth is reduced.

Subroutines INTANG and NDEGEN are used in conjunction with the subroutines for mesh compression. INTANG calculates the interior angle formed by the positioning of three specified nodes. NDEGEN uses the angle determined in INTANG and a specified radial distance defines a node in rectangular coordinates from polar coordinates.

Subroutine ASKPLT provides the user with the capability to generate a hardcopy plot. ASKPLT also allows the user to change the size of the plot. The default size is the same size that is viewed on the screen.

3. APPLICATIONS OF THE FINITE ELEMENT MESH GENERATION PROGRAM

3.1 TWO-DIMENSIONAL MESH GENERATION

To illustrate the operation and ease of use of this program, the following sample problem is included. Figure 2 depicts the two-dimensional domain of interest. The groundwater system consists of a confined aquifer enclosed on all sides by no flow boundaries, with a river providing a source for induced infiltration. There is one pumping well and three observation wells as shown on Figure 2. The transmissivity of the aquifer east of the river is 4.0×10^{-4} m^2/s, while west of the river it is 2.0×10^{-4} m^2/s. The pumping well is assumed to be pumped at a constant rate of 6.32×10^{-4} m^3/s. The leakance rate from the river is 1.0×10^{-8} m/s. This is the basic information that is required to produce a mesh.

Once execution of the program has begun using the run command in Appendix A, the first prompt will be to enter: 1) a title; 2) the units one is working in, (ie: English or metric); 3) the scale of the figure from which one will be producing the mesh; 4) whether one is working in two- or three-dimensions; 5) and whether one desires quadrilateral (brick type) or triangular (tetrahedral) type elements.

Next, the user is requested to clear the digitizer, and digitize the lower left, lower right, and upper left corners of the page of the figure that is to be digitized. The purpose of this step is to scale the figure for display on the graphics terminal.

The menu is now displayed on the screen and the user follows the steps outlined in the flow chart, Figure 4.

The first step is to digitize the boundaries of a given transmissivity zone, and fixing the transmissivity value of all the nodes in this zone. Once the transmissivity boundary is defined, the user would then re-display the menu by digitizing Switch No. 1. Before moving to the next transmissivity zone, it is necessary to define all observation and pumping wells. For the problem depicted in Figure 2, the first transmissivity zone contains two observation wells and one pumping well. By selecting Switch No. 7, the user can enter the pumping information (ie: start and finish times of the simulation, number of time steps and pumping rates) as requested, and digitize the pumping well location. Observation well locations are input using Switch No. 8. With this information, the user is able to generate a mesh for this transmissivity zone by selecting Switch No. 13. The outline of the transmissivity zone is displayed onto the screen. Because this scheme is based on the mapping of quadrilaterals, the user is prompted to indicate with the cursor the four corners of the figure (even if one desires triangular elements).

Once the four corners are indicated, the program generates boundary nodes using a method of interpolation through Lagrange polynomials (Hornbeck, 1975). The prompts request the four corners of the figure, (see Figure 1), and request the number of elements in both the x and y directions.

An important feature of this program is that observation and pumping wells are digitized prior to the mesh generation so that nodes can be placed at these locations and incorporated in the mesh.

If the transmissivity zone contains any pumping wells, the mesh is smoothed. This smoothing routine adjusts the positions of each of the interior nodes to make sure that each of the elements are well conditioned. Also, elements are concentrated around pumping wells (see Section 4) to mathematically handle the large changes in hydraulic head encountered here. The resultant mesh is then displayed on the screen with the query "PLOT?". Because we have not completed our mesh generation process the response would be "no", and the menu would then be redisplayed on the screen.

The next area of interest would be the river because the boundaries of elements must correspond to the boundaries of the river for the purposes of defining leakance elements. Selecting Switch No. 9 again, we enter the boundaries of the river as a separate transmissivity zone, and generate a mesh as before. Leakance elements are defined by selecting Switch No. 5, which the user does by indicating, with the cursor, the leakance elements and the value of leakance. This routine can be rather tedious, but does allow one to define leakance elements anywhere on the mesh.

The user could continue the mesh generation process with the area west of the river.

With the mesh completely drawn, the user can choose to have it plotted by sending the plot data into a file on output device 9 for the hard copy plotter.

By selecting Switch No. 12, the nodal points are renumbered to reduce bandwidth (Collins, 1973), and output into a file attached to device 7 in a format that is compatible with the Verge-Frind (Verge, 1972) two-dimensional, finite element program for confined aquifer simulations.

The mesh generated using this procedure is shown in Figure 5. It can be compared with the hand-drawn version illustrated in Figure 6. The hand-drawn mesh has 104 nodes, 177 elements, with a bandwidth of 23, whereas the computer generated mesh has 160 nodes, 272 elements with a bandwidth of 21. The hand drawn version required approximately 6 hours to draw, number, enter the data file and check for errors. The computer generated mesh required approximately 6 seconds of processing time on an Amdahl 5860 mainframe to produce, and approximately 20 minutes at the terminal to enter the necessary data.

The simulated hydraulic head distribution due to seven days of pumping, based on the computer generated mesh, is presented in Appendix C. By comparing these results with those simulated using the hand-generated mesh, it is possible to see that the results are very similiar. The

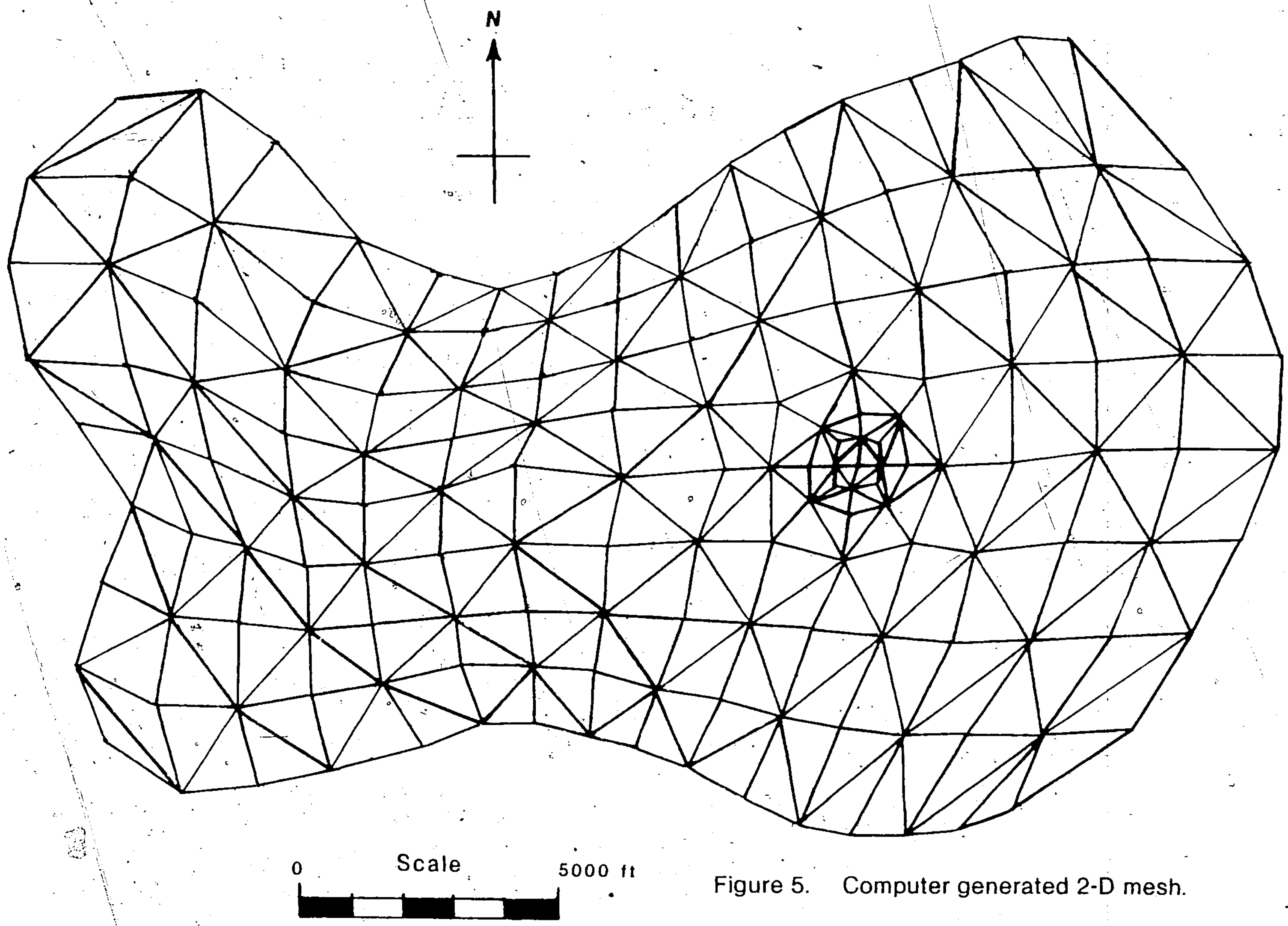

Figure 5. Computer generated 2-D mesh.

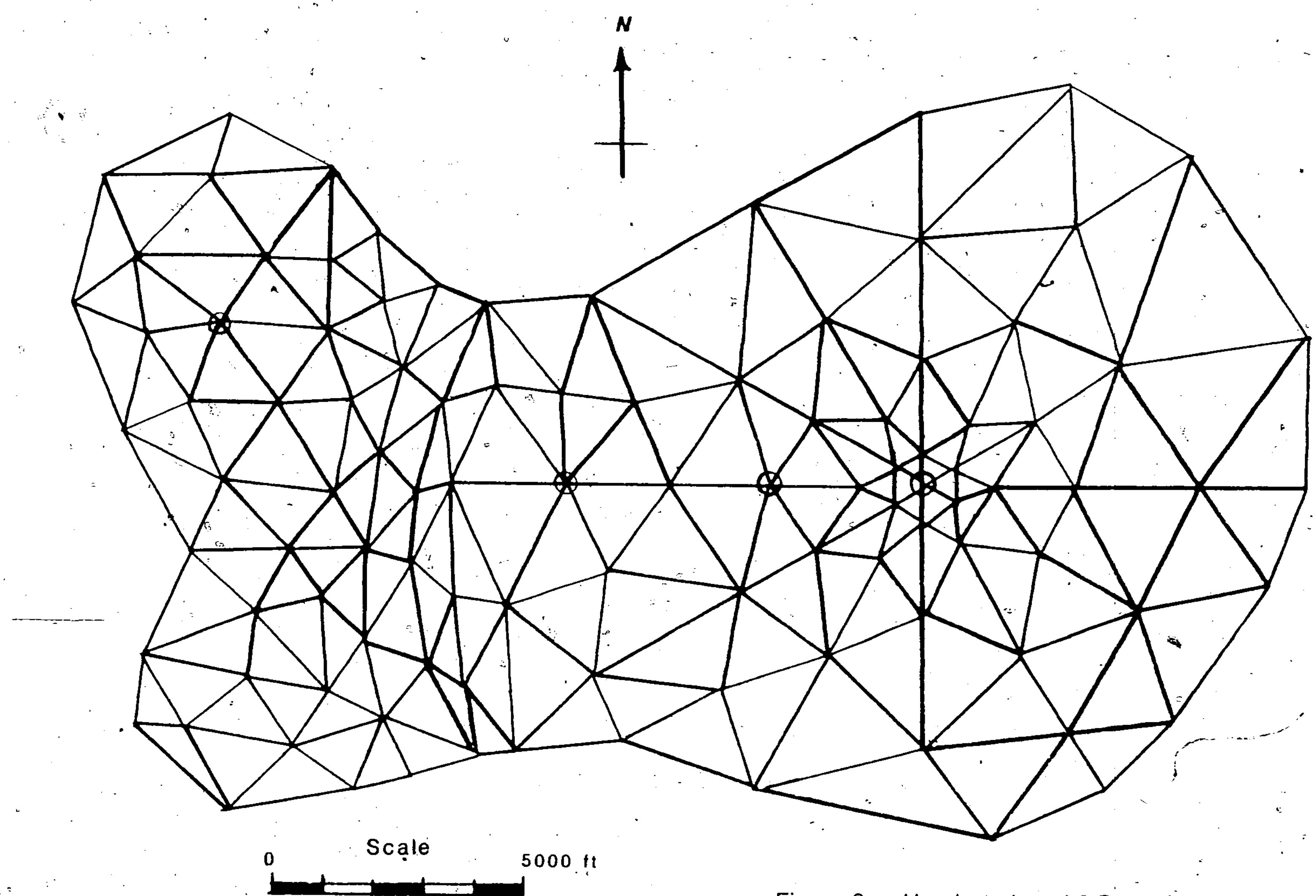

Figure 6. Hand produced 2-D mesh.

original purpose of this excercise was to determine whether it would be worthwhile to go to the expense of installing observation well number 3. According to the results of the simulations we can see that in both instances one would have to question the expense, for the drawdown indicated at this point would only be 0.03 m for the hand-drawn version, with a similiar value of 0.02 m for the computer generated version. The values of the hydraulic head differ from less than 0.01 ft in the area of observation well 3 to 3 ft. around the pumping well. The discrepancies between the two simulations are approximately in the order of 10%. The source of this discrepancy is not immediately apparent, but may be due to several sources. The sources could be accounted for in differences in mesh density, computational efficiency, and inherent inaccuracies within the simulation algorithm.

Variations in mesh density could account for a certain amount of this discrepancy. The higher density may have been better able to cover the changes in flux. This feature will be more fully discussed in the following chapter. As well, there may be inherent problems within the simulation program itself. In my experience, as much variation as is seen between these two simulations can be seen when viewing many hand drawn versions.

To further illustrate the capabilities of my algorithm, additional two-dimensional examples are included in this section. Figures 11a, 11b and 11c illustrate how easy it is,

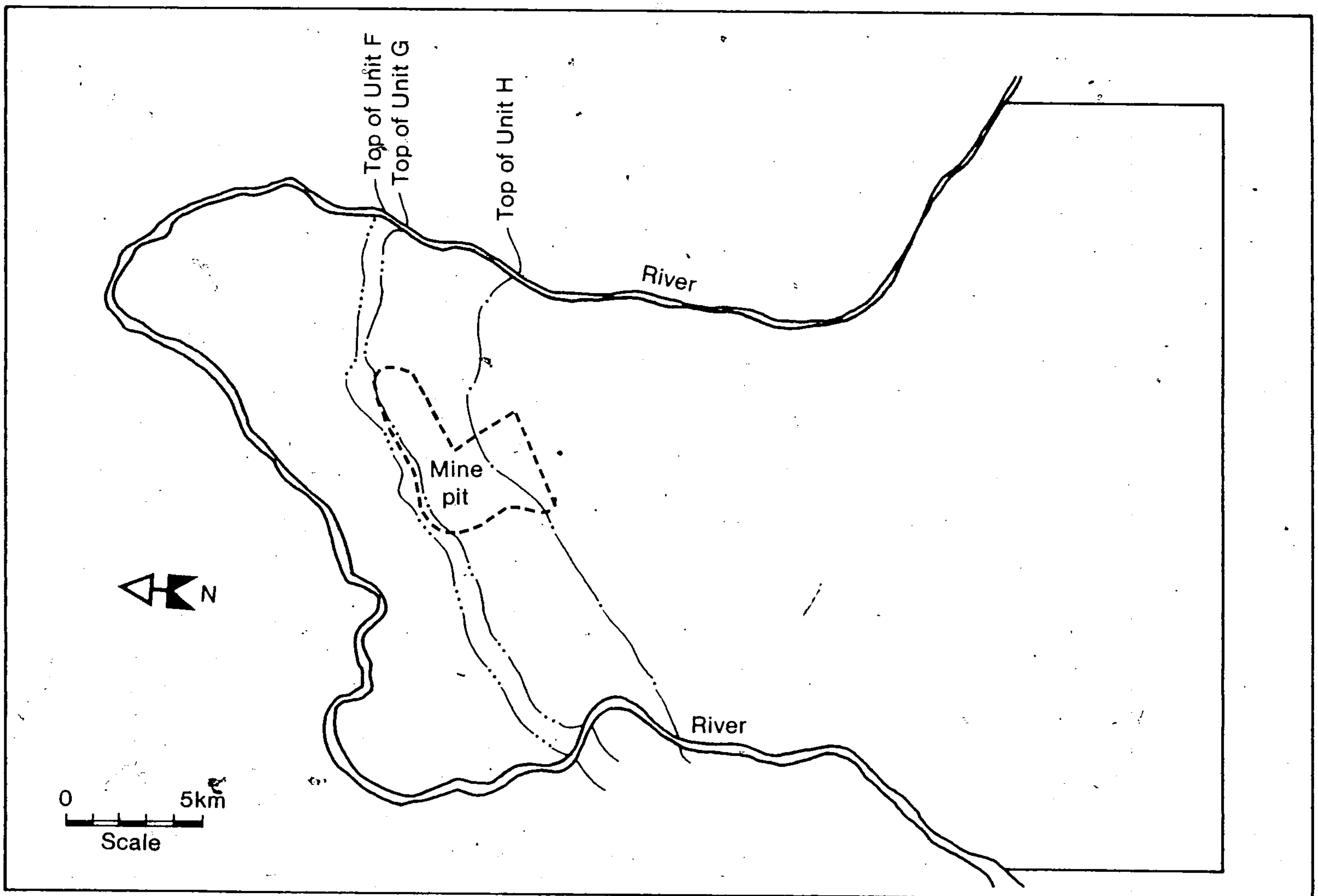

Figure 7. Map of area for 3-D simulation

with an automatic mesh generator, to increase the nodal density within a single rectangular domain. The facility to alter the density of the nodes within a domain is represented by a prompt within the mesh generation routine accessed through Switch No. 13. This prompt requests that the user indicate the number of elements desired in the x and y directions and, if working in three-dimensions, the z-direction. It is therefore the duty of the user to insure that the resultant elements are as well conditioned as possible.

Figures 8 through 14 are included to show what meshes actually look like when the domains take on a somewhat more complex shape. Figure 13 represents a shape that may be encountered if one were to model a buried channel situation. The four corners were placed as shown with an "X". Figures 13 and 14 show how important the initial placement of four corners is in controlling the character of the grid. With a triangular shaped region, it can be seen that the mesh in Figure 14 represents a smooth distribution of elements from the apex to the opposite side. In Figure 14, with the corner nodes placed as shown, there is a stretching of the elements, producing some ill-conditioned elements at the lower boundary. This last figure illustrates how it is possible to alter the mesh density along a given boundary by altering what would be an obvious placement of corner nodes to one that would alter the distribution of nodes along a given boundary. In this way any adjacent zones would not be

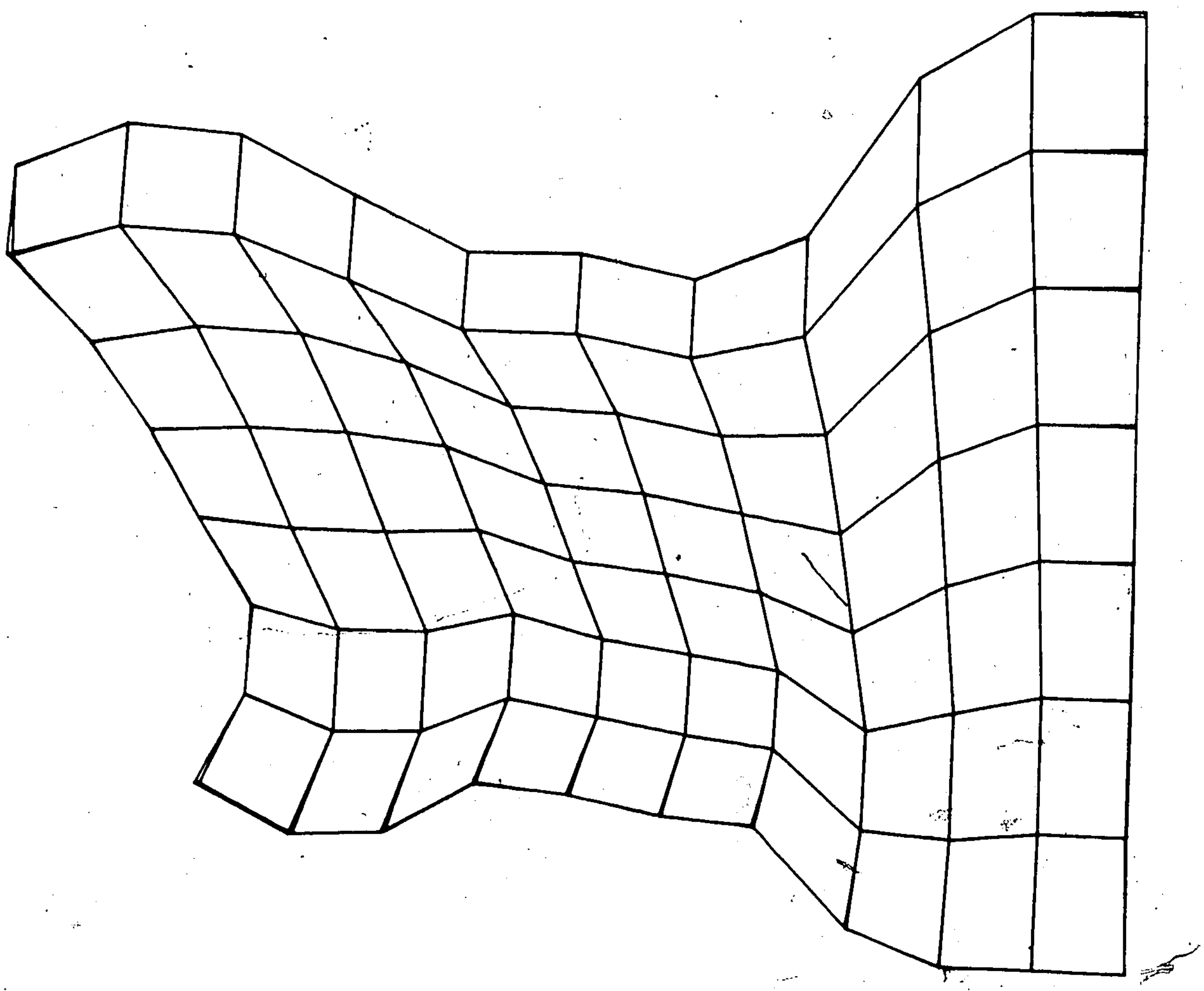

Figure 8. One layer computer generated 3-D mesh.

constrained to a set mesh density.

3.2 THREE-DIMENSIONAL MESH GENERATION

To illustrate the capabilities of the code in three-dimensional applications, consider the following hypothetical problem of assessing the influence of an open pit mine on groundwater levels. The map illustrating the layout of the simulation is shown in Figure 7. For this problem the system is discretized into eight transmissivity zones representing a layered sedimentary sequence striking approximately east-west and dipping southward. Layers E,G and H are assumed to outcrop at the surface. The hydrogeological parameters are outlined in Table 1.

TABLE 1.

SUMMARY OF BASIC DATA FOR INDIVIDUAL LAYERS IN THE MODEL

Layer No.	Predominant Lithology	Thickness (m)	Hydraulic Cond. (m/s)	Trans. (m^2/s)
H	limestone	125	$1x10^{-6}$	$1.25x10^{-4}$
G	limestone	30	$1x10^{-7}$	$3.0x10^{-6}$
F	shale	20	$1x10^{-8}$	$2.0x10^{-7}$
E	shale	15	$5x10^{-7}$	$7.5x10^{-6}$
D	ore zone	15	$1x10^{-6}$	$1.5x10^{-4}$
C	shale	20	$1x10^{-7}$	$2.0x10^{-6}$
B	shale	30	$1x10^{-9}$	$3.0x10^{-8}$
A	shale	50	$5x10^{-8}$	$2.5x10^{-6}$

The storativity is assumed to be constant throughout at a value of 0.0001.

For the purposes of defining the flow domain, the rivers bounding the three sides of the system are

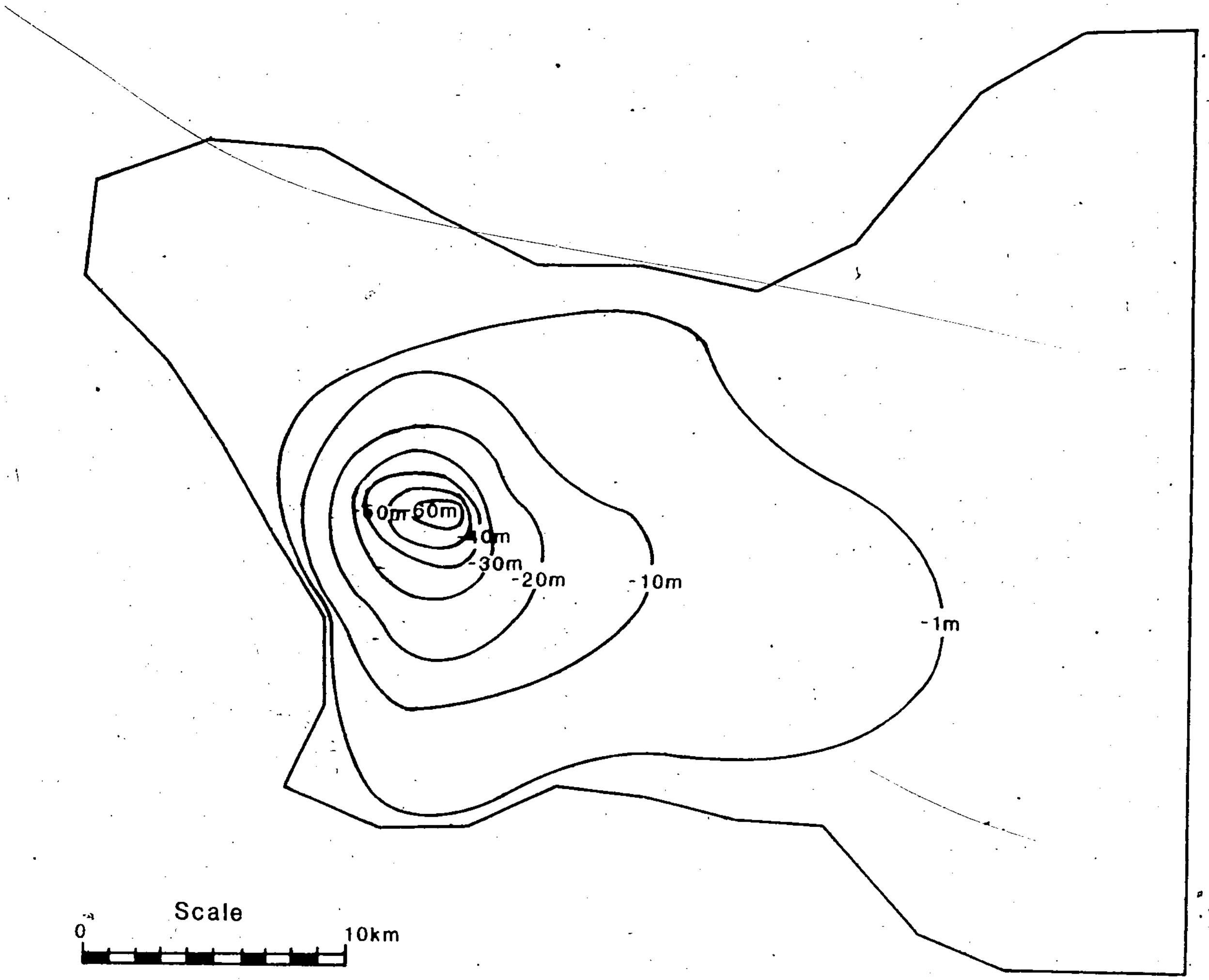

Figure 9. Drawdown cone after 15 years simulation period.

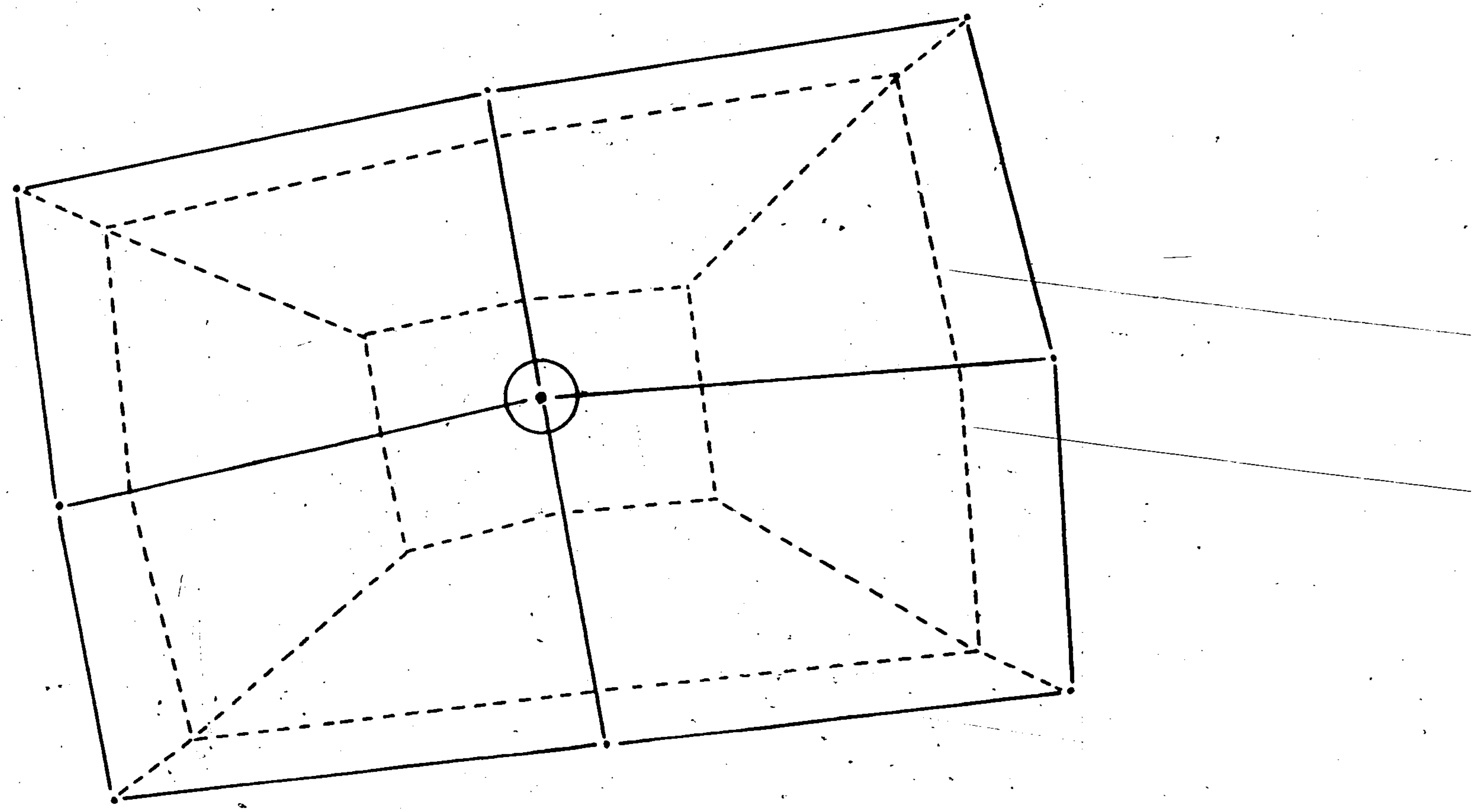

Figure 10. Figure demonstrating how the elements around a flux node are partitioned to give the impression of mesh compression. The solid lines represent the existing elements. The dashed lines represent the partitioning that is later smoothed.

represented by a series of constant head nodes at the surface. The south, bottom and top sides of the system are defined as no-flow boundaries. The simulation of dewatering the open pit mine was accomplished by defining constant head nodes that represent drawdown to the base of the mine. It is assumed that there is no recharge to the system and that the initial head distribution is consistent everywhere within the domain.

To carry out this simulation, Verge's (1975) algorithm for modelling three-dimensional transient groundwater flow was used. The simulation was carried out over a period of 15 years using a time-step of 5 years.

The mesh generated for the simulation is illustrated in Figure 8. The layer shown illustrates the general configuration of the first of eight layers of the mesh; each layer having 672 nodes, 580 elements and numbered to have a half-bandwidth of 89.

The mesh density was determined more on the basis of the memory requirements of the Verge (1975) algorithm when used on the Amdahl 5860 mainframe than on the basis of computational efficiency or detailing the solution.

The mesh required approximately 1.5 hours at the graphics terminal to input the information to construct the mesh. Approximately 32 seconds of CPU time, on the Amdahl 5860 mainframe, is necessary to construct the grid Approximately half of this computer time was involved with the renumbering of the mesh to reduce the bandwidth. In addition, approximately another hour of time is required to determine the location of constant head nodes that define the position of the river and the mine.

The drawdown cone, as seen at the surface after 15 years, is illustrated in Figure 9. This drawdown cone follows the outline of the mined area and provides a reasonable representation of what one would expect. The simulation required approximately 45 seconds of CPU time on the Amdahl 5860 mainframe.

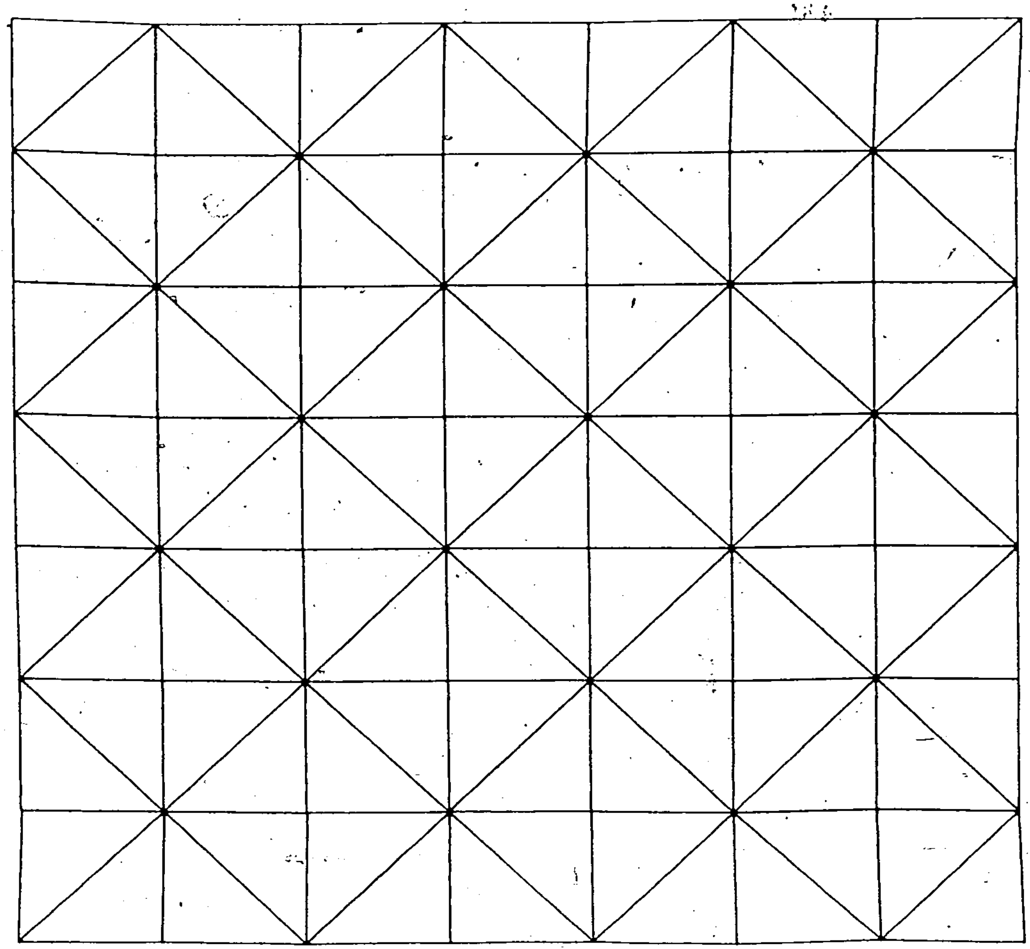

Figure 11a. Basic rectangular shaped figure showing the domain discretized into 196 triangular elements.

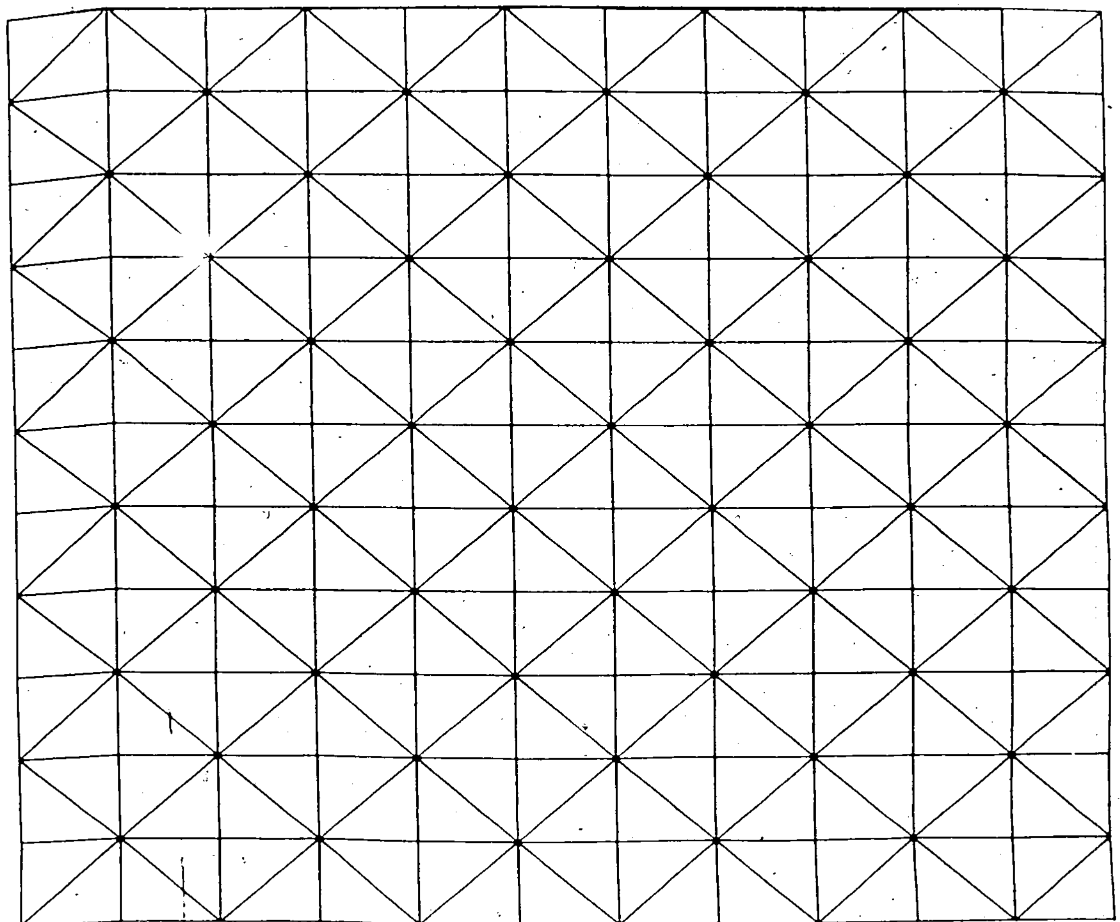

Figure 11b. Basic rectangular shaped figure showing the domain discretized into 484 triangular elements.

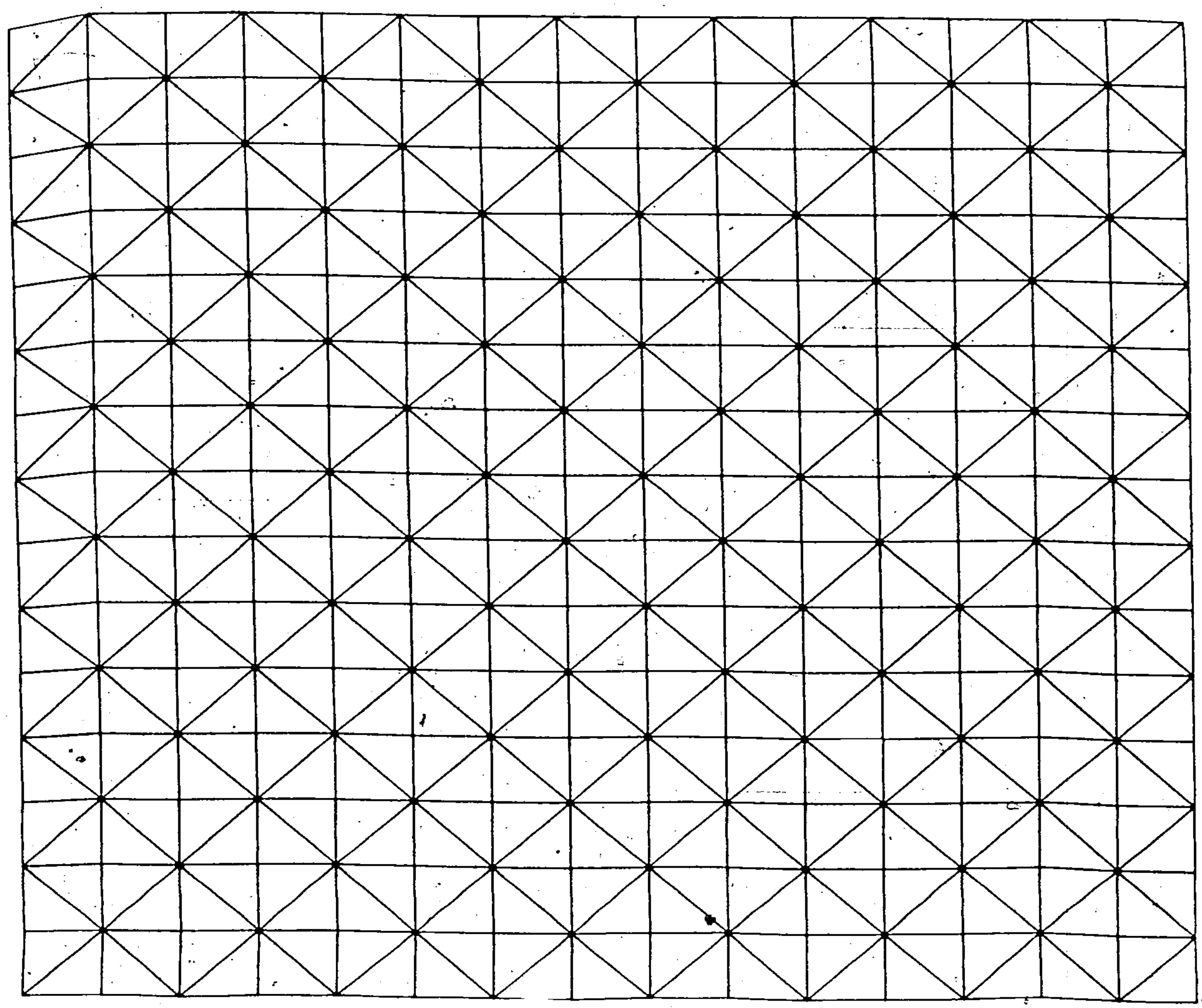

Figure 11c. Basic rectangular shaped figure showing the domain discretized into 900 triangular elements.

4. OPTIMIZATION OF THE MESH GENERATION PROCESS

With the availability of the mesh generation procedure, the problem still remains as to whether the mesh produced is optimum for the simulation in question. In the previous chapter, the results obtained from the hand-drawn mesh and the computer-produced mesh, for the same problem, were somewhat different from each other. Where it is not possible to verify numerical solutions with analytical solutions for a given problem, it is hard to know which simulation is more accurate. In the example considered, the computer generated mesh has a higher density of nodes and elements than the hand drawn version. However, the former could be considered computationally more efficient because of the smaller half-bandwidth. Arguably it could be stated that the computer produced mesh has more elements and nodes, and therefore should have provided a more exact solution. The question would then become "did we really need that many nodes?".

Bathe and Sussman (1983) state at a mesh of relatively few elements whose nodal points are placed in optimal positions can match the predictive capabilities of a coarse mesh whose nodal points are not particularily well placed. As well, the use of higher-order isoparametric elements can improve the solution if the elements were distorted in such a manner that they fit the optimal positions of the nodes. This last point will not be considered further because the generation scheme presented herein only generates linear elements.

In my code two facilities were incorporated to help optimize the mesh. There is an inherent advantage in dealing with pre-positioned flux nodes in dealing with hydrogeologic simulations. In other words, because the source of the flux is known, it is possible to compress the mesh to maximize mathematical accuracy. The manner in which this compression is performed is illustrated in Figure 10. The existing elements are represented by solid lines, while the partitioning is represented by dashed lines. It can be seen that the elements surrounding a given well (or flux node) are partitioned to form more elements. The partitioned elements are then "smoothed" using an averaging technique to reposition the nodes so that there is a smooth transition in element size away from the flux node. Time constraints did not allow me to investigate grid compression around a flux node more thoroughly. This would have involved optimizing my scheme and investigating other schemes such as that used by Bathe and Sussman (1983).

As well there is a routine to optimize the computational effort involved with the solution of the finite element mesh. This routine renumbers the nodes in order to reduce bandwidth, but unlike the algorithm provided by Collins (1973), the routine incorporated into this procedure renumbers the nodes beginning at one corner of the generated mesh. Collins' (1973) algorithm provides the ability to find the optimum node numbering scheme by successive renumberings, starting with a different node each time, and comparing the resultant bandwidths. From the solutions provided by Collins (1973), the best schemes begin numbering at a corner of the domain. It is not known whether this provides the optimal renumbering scheme in every case, but it was felt that the extra effort and cost to find the optimal condition would not result in savings in computational effort within the solution.

Logically, the efficiency of the renumbering scheme may be due in part to the geometry of the mesh (ie; there is always a long and a short axis). It is assumed that the worst case situation would be one where the mesh forms a large cube and the bandwidth would be related to the length of a diagonal through the cube. For the vast majority of cases, the renumbered mesh should have a smaller bandwidth than the original.

Workers in the field of structural engineering have come up with schemes to provide optimum meshes for finite element simulations (Bathe and Sussman, 1983; Shephard et al., 1979, 1980). These procedures involve an iterative approach begining with a crude grid that is optimized when interacted with a finite element solution. In stuctural engineering this optimization is accomplished through modifying the initial mesh until the strain energy of the solution is maximum or the potential energy is minimized (Bathe and Sussman, 1983). In hydrogeology, this approach would be analogous to modifying the mesh until the change in the potential head is minimized for each element. A fundamental approach to this problem is that the potential should be minimized not only with respect to finite element nodal displacements but also with respect to the nodal point locations (Bathe and Sussman, 1983). Defining the potential energy by Π, and the condition that the energy not fluctuate by $\partial\Pi=0$, gives the following two equations.

$$\partial\Pi/\partial u_i = 0$$
$$\partial\Pi/\partial\beta_i = 0 \qquad (3)$$

where: u_i = the nodal point displacements

β_i = all the nodal coordinates that can be changed without changing the geometry of the domain.

The effect of satisfying these equations is that the resulting mesh is concentrated around areas of greatest flux. In my code there exists a facility to concentrate flux nodes prior to the initial simulation. Therefore, satisfying these equations would involve an iterative approach once the initial solution is performed.

It is beyond the scope of this work to provide algorithms for solutions to these two equations. Indeed it is only recently that workers in this area have begun to

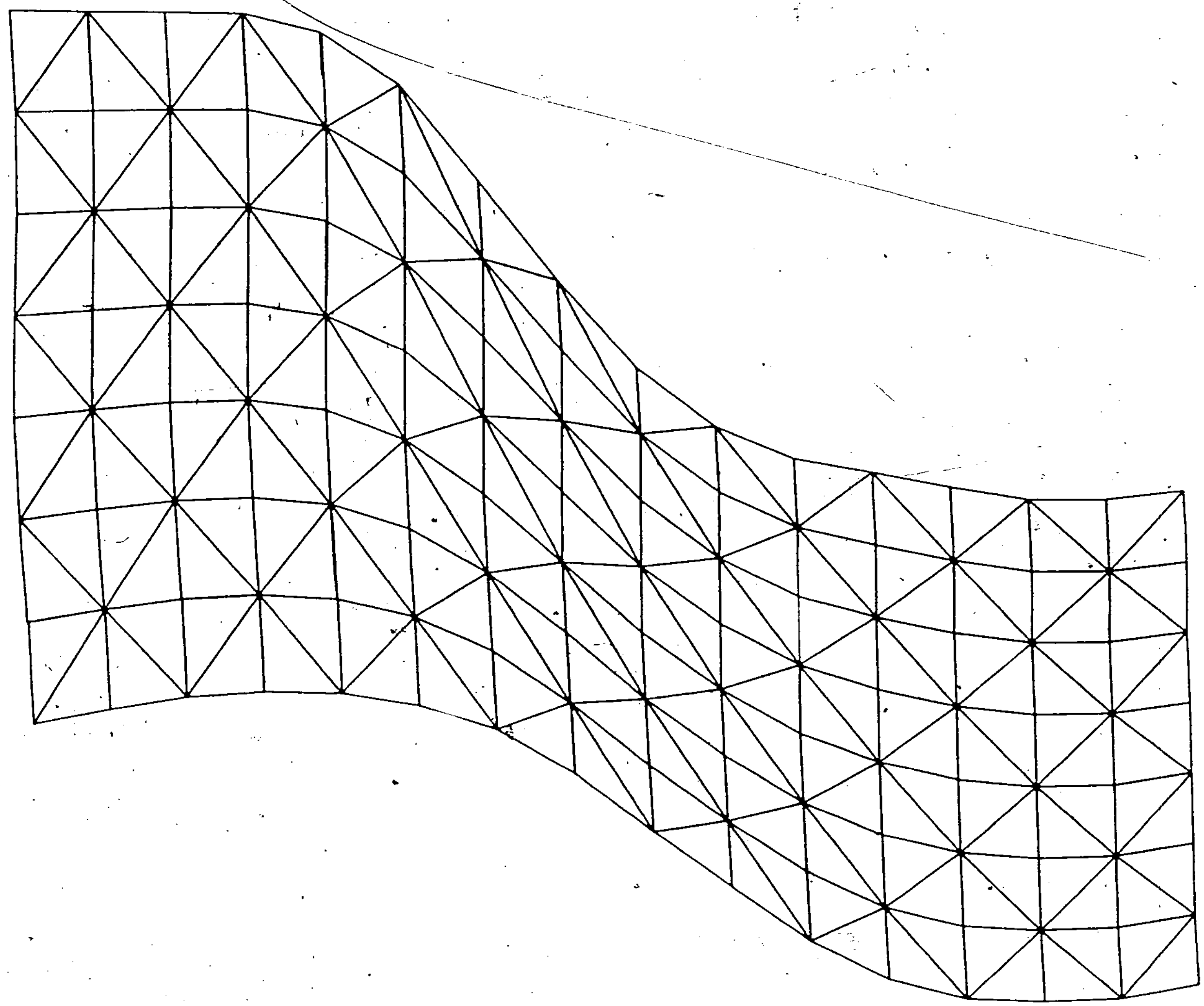

Figure 12. Irregular shaped figure of a type to represent a buried channel.

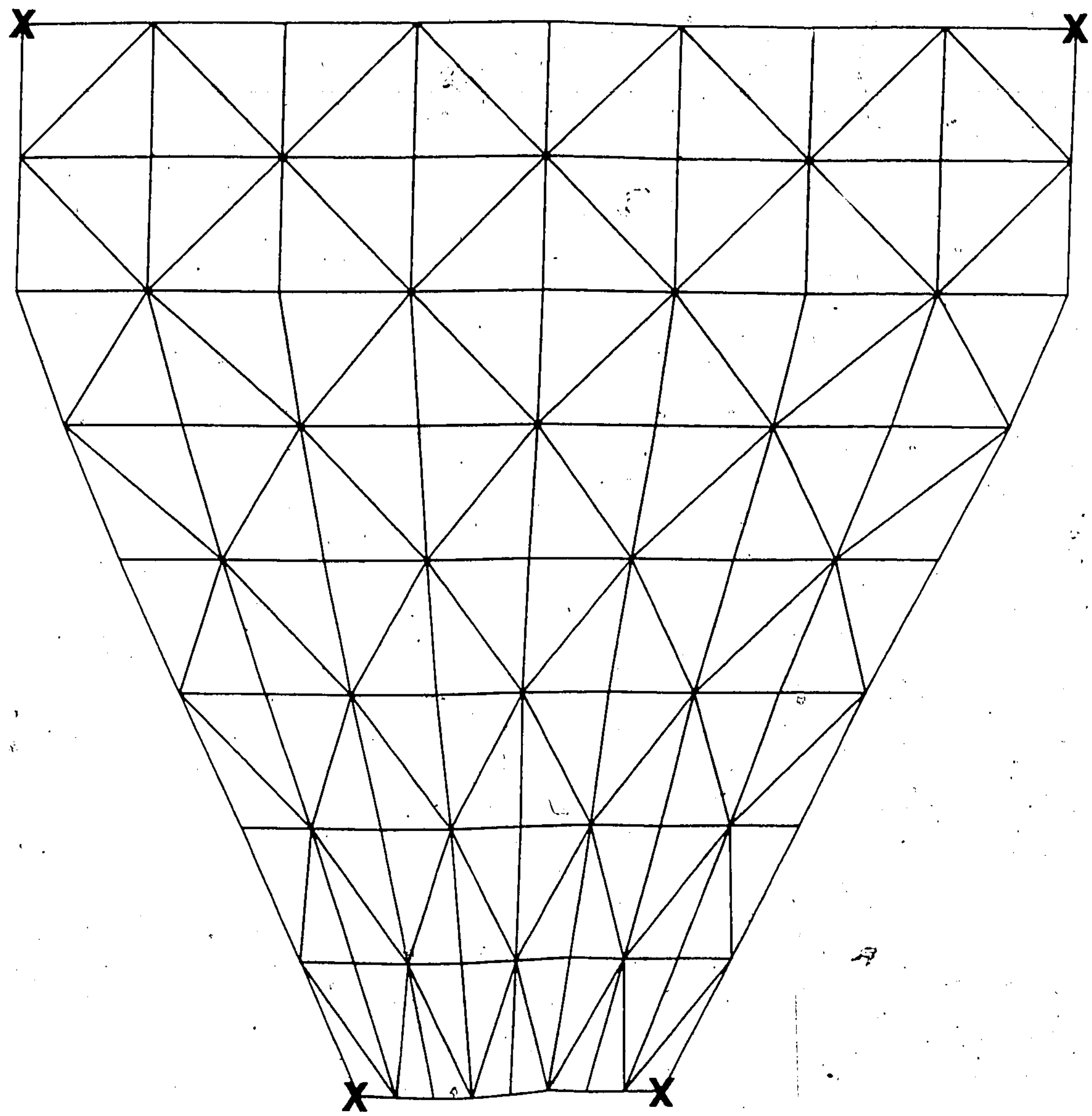

Figure 13. Triangular shaped figure discretized with a mesh of triangular elements. Corner points used to generate the mesh are designated with an "X".

deal with this problem (Bathe and Sussman, 1983; Shephard et al., 1979, 1980).

In my work there has been a modest attempt to provide a degree of built-in optimization. There exist some approximate rules and quidelines for constructing effective meshes through iterative procedures that involve the performance of analysis. The mesh generation scheme presented in this work does not easily allow one to use the approaches outlined by such authors as Shephard et al (1980) and Bathe and Sussman (1983).

It would therefore be advantageous in the synthesis of the mesh to try to bring in a degree of optimization before the simulation is attempted. Arguably this can only be attained through a great deal of experience, but two guidelines are presented with the figures to aid the uninitiated:

1. Probably the most important point is that in problems of aquifer hydraulics, the region of greatest variation in hydraulic head is known. Even though the program has a built-in facility to handle the increased change in hydraulic head, the user is advised that there should be more elements in an area where greatest changes in potential occur, and less where little change is anticipated and they are not necessary.
2. Although previously stated in the body of this work, that in dealing with isoparametric elements, it is not easy to change the density of the elements, there are a

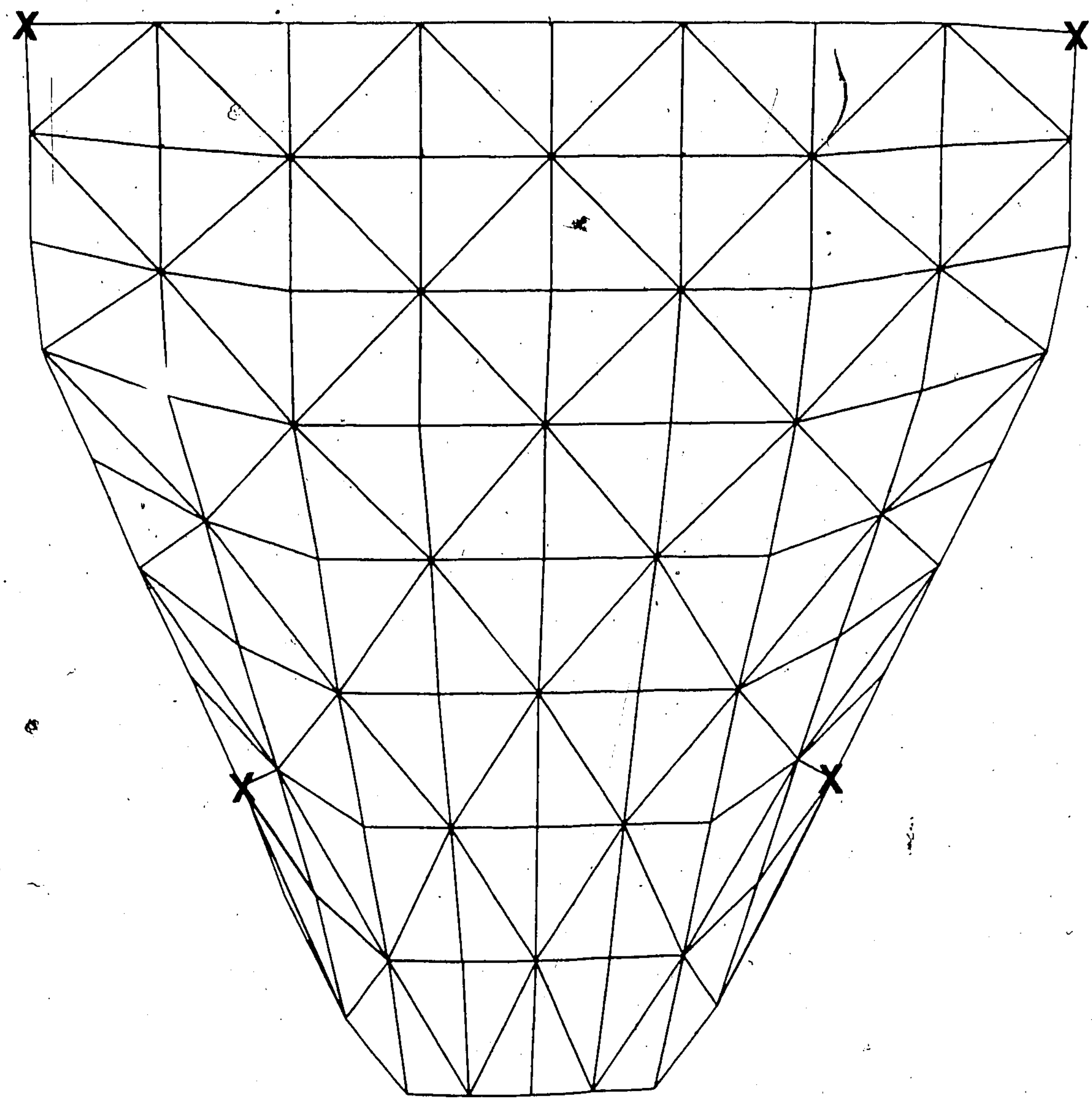

Figure 14. Triangular shaped figure showing results of designating corner points in a different manner. Corner points used to generate this mesh are designated with an "X".

few techniques involved in the placement of corner nodes that can be used to change this distribution. This is illustrated in Figures 13 and 14.

5. CONCLUSIONS AND RECOMMENDATIONS

In accordance with the objectives set out in this study, a finite element mesh generator has been successfully developed. The mesh generation scheme has proven successful and expedient in modelling finite element groundwater flow problems presented herein. It is anticipated that the generator will be successful in generating meshes where the geometry of the domain is not extremely complicated, requiring only gradual changes in mesh density. It will accept large contrasts in hydraulic conductivity, but due to its inability to produce a graded mesh, the solution may be inaccurate. As well, it is anticipated that the scheme will be successful in most cases where the geometry of the domain boundaries do not form very acute angles (ie: generally less than 10 degrees).

The amount of effort normally required to input data for a simulation has been reduced considerably through incorporating into the design of the system a digitizer and interactive graphics terminal. The digitizing tablet is used to specify both numerical information as well as options in mesh generation. The colour graphics terminal, used in conjunction with the Integrated Graphics facilities (Davis, 1977) available at the University of Alberta, allows the user to interact in the mesh generation procedure. The program is capable of generating finite element meshes in both two- and three-dimensions and has proven to be capable of handling problems of the complexity commonly associated with aquifer hydraulics.

This system operates in a mode that allows the user to be able to generate meshes without having previously absorbed volumous user manuals. The program prompts the user with options throughout the entire generation process. This "user friendly" feature should allow for the wide use of this program.

The graphics features display meshes in a series of eight colours, allowing the user to visualize where he is in the generation process. As well, hard copies can be obtained at any time during the mesh generation process.

The program, by being uniquely developed for the purposes of modelling aquifer hydraulics, has the ability to incorporate both pumping and observation wells within the mesh.

A degree of grid optimization has been incorporated into the program. This capability relates to two main areas, namely: in improving mathematical accuracy; and computational efficiency. To aid in the area of mathematical accuracy, the code compresses the mesh around flux nodes to better approximate rapid changes in hydraulic head around pumping wells. Computational efficiency is aided through the incorporation of the algorithm provided by Collins (1973) to decrease bandwidth.

Recommended areas for further study are listed below.

1. The area of grid optimization should be fully investigated. Again, as in the area of finite element

mesh generation, the field of structural engineering provides the logical starting point due to the material already existing on this problem in that field. It is suggested that the work involve using the information already supplied as input to the program such as transmissivity, storativity and pumping rates and the relationships of potential energy as provided by Bathe and Sussman(1983) as a logical starting point.

2. With the advent of the personal computer and their increasing memory capabilities and decreasing costs, it would be advantageous to investigate the feasibility of adapting this system for use with these smaller systems so as to readily allow its use by small engineering companies to aid in the speedy solution of everyday finite element simulation problems. Problems envisioned in this would be adapting the program's graphics to the capabilities of the personal computer's graphics and decreasing storage requirements for this program.

REFERENCES

Bathe, R.J. and Sussman, T.D., (1983), An Algorithm for the Construction of Optimal Finite Element Meshes in Linear Elasticity. In Computer Methods for Non-Linear Solids and Structural Mechanics. A.S.M.E. Applied Mechanics Division, vol 54, Ed. Satya N. Atluri.

Cavendish, J.C., (1974), Automatic Triangulation of Arbitrary Planar Domains for the [illegible] Element Method. Int'l J. Numerical Methods Eng., vol [illegible] pp 679-696.

Collins, R.J., (1973), Bandwidth [illegible] by Automatic Renumbering. Int'l J. Numerical Methods Eng. vol 6, pp 345 356.

Davis, A.R., (1977), Integrated Graphics User's Guide. Computing Services, University of Alberta, Edmonton, Alberta.

Hornbeck, R. W., (1975), Numerical Methods. Quantum Publishers, Inc. New York, N.Y. 310 pages.

Klienstreurer, C., (1980), Interactive Finite Element Mesh Generators for Flow Systems. Finite Elements in Water Resources, vol 3, pp7.99 - 7.109.

Narasimhan, T.N. and Witherspoon, P.A., (1980), Overview of the Finite Element Method in Groundwater Hydrology. Finite Elements in Water Resources, vol 3, pp1.29 - 1.44.

Sadeck, E.A., (1980), A Scheme for the Automatic Generation of Triangular Finite Elements. Int'l J. Numerical Methods Eng., vol 15, pp 1813-1822.

Sartori, L., Aurelio, V., Riccioni, M., and Svarca, G., (1982), Automatic Generation of Finite Element Network for Bi-Dimensional and Quasi-Three-Dimensional Equations Governing the Groundwater Flow. Finite Elements in Water Resources., vol 5, pp 8.13 - 8.23.

Shephard, M.S., Gallgher, R.H. and Abel, J.F., (1979), Experience with Interactive Computer Graphics for the Synthesis of Optimal Finite Element Meshes. In Finite Element Grid Optimization, The Third National Congress on Pressure Vessels and Piping, A.S.M.E. PVP-38, 73 pages.

S ard, M.S., Gallagher, R.H. and Abel, J.F., (1980), The Synthesis of Near-Optimum Finite Element Meshes with Interactive Computer Graphics. Int'l J. Numerical Methods Eng., vol 15, pp 1021-1039.

Suhara, J. and Fukuda, J., (1972), Automatic Mesh Generation for Finite Element Analyses. Advances in Computational Methods in Structural Mechanics and Design, (Ed. Oden, J.T., Clough, R.W. and Yamamoto, Y.) U.A.H. Press, Huntsville, Alabama.

Thwacker, W.C., (1980), A Brief Review of Techniques for Generating Irregular Computational Grids. Int'l J. Numerical Methods Eng., vol 15, No. 9, pp 1335-1341.

Verge, M.J., (1972), Linear Triangular Finite Element Model for Horizontal Aquifer Analysis with Nodal Transmissivity Values. Unpublished Paper, University of Waterloo, Department of Earth Sciences, Waterloo, Ontario, Canada.

Verge, M.J., (1975), A Three-Dimensional Saturated-Unsaturated Groundwater Flow Model for Practical Applications. Unpublished Ph.D. Thesis, Department of Earth Sciences, University of Waterloo, Waterloo, Ontario, Canada.

Yerry, M.A. and Shephard, M.S., (1983), A Modified Quadtree Approach to Finite Element Mesh Generation. Computer Graphics and Applications., pp 39 46.

Zienkiewicz, O.C. and Phillips, D.V., (1971), An Automatic Mesh Generation Scheme for Plane and Curves Surfaces by Isoparametric Coordinates. Int'l J. Numerical Methods Eng., vol 3, no. 4, pp 519-528.

APPENDIX A User's Guide

To use the automatic mesh generation program one should have at his disposal: a diagram of the figure to be modelled, with transmissivity zones, pumping wells, observation wells and areas of leakance and values of necessary constant head delineated; and hydrogeological parameters. It is also assumed that the user has the computer hardware or equivalent previously described. To run the compiled version in MTS the commands are: $RUN MESH.OBJ+*IG 5=*MSOURCE* 6=*SINK* 7=-A 9=-C

where: MESH.OBJ is the compiled version of the automatic mesh generation program.

*IG is the integrated graphics library of subroutines available at the University of Alberta.

5=*MSOURCE* specifies the input port (or device) that accepts data from the keyboard and digitizing tablet.

6=*SINK* specifies the terminal as the output port. This facilitates prompting, querying and displaying the results so that the user is involved with all aspects of the mesh generation. *SINK* displays everything on the terminal screen.

7=-A this is the port(or device) where the resultant mesh information is outputted to after the generation is completed.

9=-C this is the port to which all plots requested are sent for plotting on the CalComp Plotter.

Once the program is loaded the user will be asked to respond to a series of queries. This section is intended to aid the user as to what to expect to be able to answer as the generation routine progresses.

1. IM-a variable that determines whether one is using English or metric units. IM=0 informs the program that the user is working in English units and the program is expecting to digitize a map where "1 inch=x feet". If IM=1, the program will convert all coordinates into meters. The program is expecting to digitize a map where the scale is defined as "1:xxxx".
2. Scale-when entering the scale the format is F10.0. Therefore the user is cautioned to remember to input a period(or a comma when dealing with integer input) after the number is typed in such as 10. or 10,.
3. ID-a variable that specifies whether one is working with a two-dimensional or three-dimensional mesh. ID=0 implies that the mesh to be generated is two-dimensional and to be generated in the x-y plane. ID=1 implies the mesh to be generated is three-dimensional, generated in the x-y plane and projected in the z-direction.
4. IELEM-prompts the user as to what type of element is required. In two-dimensions, the request would be for IELEM=0, which implies triangular elements, or IELEM=1 for quadrilateral elements. In three-dimensions the choice would be between IELEM=0 for tetrahedral elements, or IELEM=1 for parallelpiped (or "brick-type")

elements. Note that this mesh generator is only capable of constructing linear elements.

5. ISS-is a prompt only used when one is working in 3-D. Where ISS=0 implies Steady State and ISS=1 implies Transient simulation. This outputs a value of 999. for the time, that triggers the steady state simulation in Verge's (1975) 3-D Groundwater model program.
6. This prompt requests the user to indicate the "LOWER LEFT" "LOWER RIGHT" and "UPPER RIGHT" corners of the map on which the flow domain is defined. These data are provided so that the computer can scale the drawing for purposes as displaying on the screen, plotting and for determining whether the map is tilted with respect to the axis of the digitizer.
7. The next prompt is the menu. The user is provided with 11 options that would control the program in the manner outlined in the flow chart in Figure 4. The following prompts will be in the order as followed by this flow chart, and a progression through the mesh synthesis process.
8. After selecting Switch No. 9, the user would be asked to enter the storativity and the transmissivity (in 3-D the hydraulic conductivities). The format for reading the values is E10.3. After this, the user would be requested to digitize the perimeter of the transmissivity zone in a counter-clockwise manner.
9. If any pumping wells are present, the user would then

select Switch No. 7. In 2-D the prompt would be to enter the "start time", "finish time", and the number of time steps for the simulation, and the pumping rate in L^3/sec. The query would then be asked whether there is any more pumping information. This question provides the opportunity to add more simulation periods as is compatible with the two-dimensional Verge-Frind (Verge, 1972) transient code for confined aquifers. In three-dimensions the user would be requested to enter the simulation period,(the time units are dependent on the time units designated in the Verge(1975) three-dimensional model), initial time step, a multiplier for the time step, number of time steps changes in delta time and the pumping rate in L^3/sec. The format for entering the start time, finish time, simulation period, initial time step, multiplier for the time step and pumping rate are all F10.0. The formats for entering the number of time steps and the number of time steps between changes in delta time are I4.

10. The next prompt would be encountered after selecting Switch No. 13 for generating a mesh. The prompt would be to designate the four (eight in three-dimensions) corner points of the figure displayed on the terminal. If one is working in three-dimensions, a prompt would request the elevation of the corner point. As well there are prompts that request that the user double check the points that he has input. The next prompt would have the

user specify the number of elements desired in the x-, y- and z-directions to obtain the desired mesh density.

11. Once the mesh is drawn, the user is prompted as to whether or not the figure displayed on the screen is to be plotted on the CalComp Plotter. From here, the user has the option of changing the size of the plot, with the default being the same size as that displayed on the screen.

12. Other Prompts include:
    Entering the leakance values in format E10.4.
    Entering the boundary fluxes in format F10.0.

Dimensioning of Program Arrays

As noted in the body of the text, the arrays are all dimensioned sufficiently large for most situations. However the following is a guide should there arise a need to change the storage requirements.

```
COMMON MEMJT(NNODE,32),IN(NELM,8),IFACE(3500,6),YN(NNODE),
       ZN(NNODE),CX(NNODE),XN(NNODE),JMEM(NNODE),NODE1(8),
       NDX(NNODE),NDY(NNODE),NDZ(NNODE),ITYPE(NNODE),
       NDWEL(20,2,25),LEAK(100),ALEAK(100),XIN(100),
       YIN(100),ISAVE(25,2),PPOSX(10),PPOSY(10),PPOSZ(10),
       OBSX(10),OBSY(10),OBSZ(10),ND(16),NND(16),COX(10),
       COY(10),COZ(10),IFE(6),NPMP(25),IPPN(10),ICRNR(10,8),
       NFACE(6,4),ICNTST(4,2),NBDND(4000)
DIMENSION TIME(10,2),NTIME(10),PMPRTE(10,10),X1(4),Y1(4),
```

```
     Z1(4),TITLE(10),BFLUX(10),A(3),B(3),NCEL(10,2),
     CHNG(10),ITCHNG(10),STOR(10)
INTEGER COL(10)
```

Where: NNODE=estimated number of nodes.

NELM =estimated number of elements.

Use of arrays in common:

MEMJT - stores the nodes connected to each node. Used in SETUP and OPTNUM.

IN - array describing incidence of each element.

IFACE - array denoting elements appearing in each of the six faces of the mesh. Used throughout the program.

XN - array storing x-coordinate of each node.

YN - array storing y-coordinate of each node.

ZN - array storing z-coordinate of each node.

CX - transmissivity of each node.

JMEM - number of nodes connected to each node. Used in SETUP and OPTNUM.

NBDND - nodes that form the perimeter of each transmissivity zone.

NDX - the adjacent node in the positive x-direction.

NDY - the adjacent node in the positive y-direction.

NDZ - the adjacent node in the positive z-direction.

ITYPE - array denoting type of each node,

ie: 0-normal

1-constant head

2-flux node.

NDWEL - defines the node numbers for observation and pumping wells in 2-D and 3-D. Dimensioned for 20 wells, observation or pumping well, and for each well to occupy 25 nodes in a vertical sense.

LEAK - defines an element that a leakance value is assigned to.

ALEAK - defines the leakance value for each element.

XIN - for inputting x-coordinates in "Pic" routines.

YIN - for inputting y-coordinates in "Pic" routines.

ISAVE - defining nodes that surround a pumping well.

PPOSX - x-coordinate of a pumping well.

PPOSY - y-coordinate of a pumping well.

PPOSZ - z-coordinate of a pumping well.

OBSX - x-coordinate of an observation well.

OBSY - y-coordinate of an observation well.

OBSZ - z-coordinate of an observation well.

ND - used in the concentration of elements around a pumping node.

NND - used in the concentration of elements around a pumping node.

COX - hydraulic conductivity in the x-direction.

COY - hydraulic conductivity in the y-direction.

COZ - hydraulic conductivity in the z-direction.

IFE - used as a counter in conjunction with IFACE.

NPMP - stores the number of nodes associated with each well.

IPPN - stores the number of simulation periods associated with each well.

ICRNR - stores the corner points associated with each transmissivity zone.

NFACE - node numbers with each face of each new transmissivity zone.

ICNTST- store common corner points.

NODE1 - book keeping array for the formation of the elements.

APPENDIX B: Program Listing

```
C*****************************************************************
C*****************************************************************
C***                                                           ***
C***   GREGORY SMITH M.Sc. THESIS PROJECT                      ***
C***                                                           ***
C*****************************************************************
C*****************************************************************
C
C A 2&3D Automatic finite element mesh generator.
C This program digitizes information from maps and
C cross-sections using a Summagraphics(Corp) ID
C series tablet digitizer.
C The system is user interactive, with the computer
C giving prompts and options to the user as to
C what to do next.
C Options:  2 or 3D mesh
C           triangular(tetrahedral) or Quadrilateral
C           (parallelpiped) elements
C           metric or English measurement system
C The elements are generated through an isoparametric
C method(Zienkiewicz and Phillips, 1971) with the
C density of the elements determined by the user.
C The elements are concentrated around pumping
C wells for numerical stability.
C After the mesh is generated and displayed and all
C manipulations are complete, the nodes are then
C renumbered to reduce bandwidth(Collins, 1973),
C then output for use by a F.E. program.
C In 3D the mesh should be generated from the
C bottom up.
C
C ..................................................
C
C Program MESH.GEN
C
C   Author
C        G.J. Smith(U. of Alberta)
C
C   Last Update
C        May 1984
C
C   Purpose
C        To set up conditions for mesh generation
C        call up the various routines to
C        generate the mesh and display and output
C        the resultant mesh.
C
C   Usage
C        Execute as a main Program
C
```

```
C   Remark
C        For complete description see:
C        Smith, G.J., Finite Element Pre-processor
C        for Hydrogeologic Modelling. Unpublished
C        M.Sc. Thesis, University of Alberta, 1984.
C
C   Subroutines Required
C        SMOOTH,DSTRB,NODE,ELMNT,NDEGEN,INTANG,
C        SETUP,OPTNUM,ASKPLT,CNCTQ,CNCTT1,CNCTT2,
C        Integrated Graphics Library Subroutines
C
C ..................................................
      COMMON MEMJT(5000,32), IN(3000,8), IFACE(3500,6), YN(5000),
     1       ZN(5000)
      COMMON CX(5000)
      COMMON XN(5000), JMEM(5000), NBDND(4000), NDX(5000), NDY(5000)
      COMMON NDZ(5000), ITYPE(5000), NDWEL(20,2,25), LEAK(100),
     1       ALEAK(100)
      COMMON XIN(100), YIN(100), ISAVE(25,2), PPOSX(10), PPOSY(10),
     1       PPOSZ(10)
      COMMON OBSX(10), OBSY(10), OBSZ(10), ND(16), NND(16)
      COMMON COX(10), COY(10), COZ(10), IFE(6), NPMP(25), IPPN(10)
      COMMON ICRNR(10,8), NFACE(6,4), ICNTST(4,2), NODE1(8)
      COMMON IP, NPTS, NPTS1, NELM, IO, ITRNS, INC, IELEM, NZ, ID, YC,
     1       XC, IPN
      COMMON DELX, DELY, SCL, IW1, IW2, IK, NBN, NPP, NXSTPS, NYSTPS,
     1       DELTZ
      DIMENSION TIME(10,2), NTIME(10), PMPRTE(10,10), X1(4), Y1(4),
     1          Z1(4)
      DIMENSION TITLE(20), BFLUX(5000), IBF(1000), A(3), B(3)
      DIMENSION NCEL(10,2), CHNG(10), ITCHNG(10), STOR(10)
      INTEGER YES /'Y'/, NO /'N'/
      INTEGER COL(10) /'RED ', 'BLUE', 'GREE', 'MAGE', 'CYAN', 'WHIT',
     1          'YELL', 'RED ', 'BLUE', 'GREE'/
      INTEGER INN(8) /5, 6, 7, 8, 1, 2, 3, 4/
      DATA BETA /0.00E+00/, DENS /0.998/, HETA /0.35/, TETA /0.35/, AL
     1     /0.0/
C Initialize all counters
   10 NLN = 0
      NBN = 0
      IW1 = 0
      IW2 = 0
      IP = 0
      NPP = 0
      IPL = 0
      NPTS1 = 1
      NPTSN1 = 1
      ICON = 0
      IPP = 0
      IPN = 0
```

```
      ITRNS = 0
      NBF = 0
      ITRNSS = 0
      IO = 0
      NELM = 0
      NELM1 = 1
      IOO = 0
      DO 20 I = 1, 6
        IFE(I) = 0
   20 CONTINUE
      DO 30 I = 1, 4999
        NDX(I) = 0
        NDY(I) = 0
        NDZ(I) = 0
        ITYPE(I) = 0
        BFLUX(I) = 0.
   30 CONTINUE
      DO 60 I = 1, 25
        DO 50 J = 1, 20
          DO 40 K = 1, 2
            NDWEL(J,K,I) = 0
   40     CONTINUE
   50   CONTINUE
   60 CONTINUE
C Begin interactive input of options
      WRITE (6,70)
   70 FORMAT ('Enter a title for this simulation')
      READ (5,1890) TITLE
      WRITE (6,80)
   80 FORMAT ('***** 2-D - 3-D FINITE ELEMENT MESH GENERATOR *****', /,
     1       'This Program will operate in two modes; 2-D and 3-D.', /,
     2       'Some Preliminary information is requires first before', /,
     3       'proceeding, Enter the frame units - IM -', /,
     4       '                    IM=0 frame units are in inches', /,
     5       '                    IM=1 frame units are in centimeters',
     6       //, ' These are the only acceptable frame units', /,
     7       ' the format is I, I1')
      READ (5,90) IM
   90 FORMAT (I1)
      WRITE (6,100)
  100 FORMAT ('Scale:?  Enter the scale of the figure to b digitized')
      IF (IM .EQ. 1) GO TO 130
      WRITE (6,110)
  110 FORMAT ('1 inch= ?ft.')
      READ (5,120) SCL
  120 FORMAT (F10.0)
      GO TO 150
  130 WRITE (6,140)
  140 FORMAT (' 1: ?')
      READ (5,120) SCL
```

```
      SCL = SCL / 100.
  150 CONTINUE
  160 FORMAT (F11.3)
      WRITE (6,170)
  170 FORMAT ('Is this Mesh Generation to be Performed in 2-D or 3-D ?',
     1       /, '         ID=0 Mesh Generation in 2-D
     2          ', /, '         ID=1 Mesh Generation in 3-D.')
      READ (5,180) ID
  180 FORMAT (I1)
      IF (ID .EQ. 1) GO TO 200
      WRITE (6,190)
  190 FORMAT ('Do you Require Triangular or Quadrilateral Elements ?', /
     1       , '               IELEM=0 Triangular Elements
     2          ', /, '          IELEM=1 Quadrilateral Elements')
      READ (5,90) IELEM
      INC = 3
      IF (IELEM .EQ. 1) INC = 4
      GO TO 240
  200 WRITE (6,210)
  210 FORMAT ('Do you require Tetrahedral or Brick Type Elements ?', /,
     1       '              IELEM=0 Tetrahedral Elements
     2       ', /, '          IELEM=1 Brick Elements')
      READ (5,90) IELEM
      INC = 3
      IF (IELEM .EQ. 1) INC = 4
      WRITE (6,220)
  220 FORMAT ('Is this simulation a steady state or transient situation?
     1', /, '            ISS=0 Steady State', /, '          ISS=1 Transien
     2t')
      READ (5,180) ISS
      WRITE (6,230)
  230 FORMAT ('In the 3-D mesh generation portion of this program', /,
     1       'a 2-D will be produced first, which will have the third ',
     2       /, 'dimension projected from it out of the page.  The progr
     3am', /, 'requires that the user begin the mesh generatiom with the
     4', /, 'the layer of the greatest transmissivity or smallest mesh '
     5       , /, 'size as a base from which subsequent meshes in the sy
     6stem', /, 'will be generated..')
  240 CONTINUE
      WRITE (6,250)
  250 FORMAT ('You are now ready to begin digitizing,
  260 ',/,'N.B.***** This program requires that the origin be located
     1',/,
     2   'at the lower left corner of the tablet.  Press the <CLEAR>
     3',/, 'switch on the digitizer',//)
C Set up scale factors
  270 WRITE (6,280)
  280 FORMAT (///, 'Enter corners of Diagram', /, 'Corners Sequence:', /
     1       , '  <LOWER LEFT>,<LOWER RIGHT>,<UPPER LEFT>')
      DO 300 I = 1, 3
```

```
        READ (5,290) IA, IB
  290   FORMAT (2X, 2I6)
        A(I) = FLOAT(IA)
        B(I) = FLOAT(IB)
  300 CONTINUE
      WRITE (6,310) IM, A(1), B(1), A(2), B(2), A(3), B(3)
  310 FORMAT ('IM= ', I1, /, 'A(1)= ', F6.0, 4X, 'B(1)= ', F6.0, /,
     1       'A(2)= ', F6.0, 4X, 'B(2)= ', F6.0, /, 'A(3)= ', F6.0, 4X,
     2       'B(3)= ', F6.0)
      DIFX = A(2) - A(1)
      DIFY = B(3) - B(1)
      SLOPEV = DIFY
      IF (A(1) .NE. A(3)) SLOPEV = DIFY / (A(3) - A(1))
      SLOPEH = (B(2) - B(1)) / DIFX
      ANGLE = ARSIN(SLOPEH)
      TUM = 1.
      IF (B(1) .EQ. B(2) .AND. A(1) .EQ. A(3)) TUM = 0.
  320 SCALE = 1000.
      IF (IM .EQ. 1) SCALE = 393.7
      ICK = 1
      N = 0
      YC = (((B(3) - B(1))/2.)*SCL) / SCALE
      XC = (((A(2) - A(1))/2.)*SCL) / SCALE
      ZC = YC * 2.
      OX = 0000.
      OY = 0000.
      GO TO 350
  330 WRITE (6,340)
  340 FORMAT ('A ZERO OR NEGATIVE VALUE IS INVALID', /, ' TRY AGAIN')
C Display of program options in menu format
  350 WRITE (6,360)
  360 FORMAT ('           NOW BEGIN DIGITIZING', //,
     1       '                MENU', /, '|----|', /,
     2       '| 13 |  Generate Mesh(after digitizing)', /, '|----|', /,
     3       '| 12 |  Output Data for Later Use', /, '|----|', /,
     4       '| 11 |  Input boundary fluxes', /, '|----|', /,
     5       '| 10 |  Future Use', /, '|----|', /, '|  9 |  Digitize a T
     6ransmissivity Boundary', /, '|----|', /, '|  8 |  Digitize an Obse
     7rvation Well', /, '|----|', /, '|  7 |  Digitize a Pumping well',
     8       /, '|----|', /, '|  6 |  "Pic" Constant Head Nodes', /,
     9       '|----|')
      WRITE (6,370)
  370 FORMAT ('|  5 |  "Pic" Leakance Elements', /, '|----|', /,
     1       '|  4 |  Remove Last Transmissivity Zone Generated', /,
     2       '|----|', /, '|  3 |  Remove Last Point Digitized', /,
     3       '|----|', /, '|  2 |  Stop/Begin Again', /, '|----|', /,
     4       '|  1 |  Re-Display Menu', /, '|----|')
  380 READ (5,390,END=2260) IX, IY
  390 FORMAT (2X, 2I6)
      IF (IX .GT. 1000) GO TO 430
```

```
      ICK = IY / 1       1
      WRITE (6,400
  400 FORMAT ('SWI    JMBER IS', 2X, I2, /)
      GO TO (500, 510, 540, 570, 910, 580, 1030, 1190, 1230, 1370,
     11380, 1590, 2000), ICK
  410 WRITE (6,420)
  420 FORMAT (/'SWITCH INVALID, 1<SWITCH<14, TRY AGAIN.')
      GO TO 380
C
C
C   TRANSFORM X AND Y CO-ORDINATES TO CORRECT ANGLE
  430 IF (TUM .EQ. O.) GO TO 440
      XX = FLOAT(IX)
      YY = FLOAT(IY)
      YTEMP = SLOPEH * (XX - A(1)) + B(1)
      YY1 = (YY - YTEMP) * COS(ANGLE)
      Y = ((YY1)/SCALE) * SCL
      IF (B(2) .EQ. B(1)) Y = ((FLOAT(IY) - B(1))/SCALE) * SCL
      XTEMP = (YY - B(1)) / SLOPEV + A(1)
      XX1 = (XX - XTEMP) * COS(ANGLE)
      X = ((XX1)/SCALE) * SCL
      GO TO 460
  440 X = ((FLOAT(IX) - A(1))/SCALE) * SCL
      Y = ((FLOAT(IY) - B(1))/SCALE) * SCL
  450 FORMAT (2F11.3)
  460 GO TO (490, 490, 490, 490, 490, 490, 1180, 1220, 1360, 490, 490,
     1490, 490), ICK
C
C LOADING ARRAY FOR FLINE SUBROUTINES
C
  470 WRITE (6,480)
  480 FORMAT ('ARRAY IS FULL')
  490 GO TO 350
C
C     SWITCH NO. 1
C
  500 GO TO 350
C
C     SWITCH NO.2
C
  510 WRITE (6,520)
  520 FORMAT ('Do you wish to terminate this mesh generation routine?')
      READ (5,1610) IANSWR
      IF (IANSWR .EQ. YES) GO TO 2260
      WRITE (6,530)
  530 FORMAT ('Restarting this program will erase all previously', /,
     1        'generated information in memory.  Do you wish to do this?'
     2       )
      READ (5,1610) IANSWR
      IF (IANSWR .EQ. YES) GO TO 10
```

```
      GO TO 350
C
C     SWITCH NO.3   Remove last point digitized
C
  540 IF (N .LT. 1) GO TO 550
      GO TO 560
  550 WRITE (6,600)
      GO TO 350
  560 N = N - 1
      ICK = ICKP
      ICKP = ICK
      GO TO 1350
C
C     SWITCH NO.4  Remove last Transmissivity zone Generated
C
  570 ITRNS = ITRNS - 1
      NPTS = NPTSP
      NPTS1 = NPTSP + 1
      GO TO 350
C
C     SWITCH NO.6   "Pic" Constant Head Nodes
C
  580 IF (N .LT. 1) GO TO 590
      GO TO 610
  590 WRITE (6,600)
  600 FORMAT ('YOU HAVE NOT YET GENERATED A MESH')
      GO TO 350
  610 IF (ID .GT. 0) GO TO 690
      WRITE (6,620)
  620 FORMAT ('MOVE THE CURSOR WITH THE JOYSTICK TO INDICATE THE', /,
     1        'CONTSTANT HEAD NODES IN THIS VIEW.  TOUCH A DIFFERENT KEY'
     2        , /, 'TO ENTER EACH DIFFERENT CONSTANT HEAD NODE.')
      ICON = 0
C ***********************************************************************
  630 CALL IGCTRL('TERMINAL', 'ERASE')
C ***********************************************************************
      CALL IGINIT
C ***********************************************************************
      CALL IGBGNS('PLOT')
C ***********************************************************************
      CALL IGMA(0.0, 0.0)
C ***********************************************************************
      DO 650 LL = 1, NELM
        JJ = IN(LL,1)
        XO = (XN(JJ) - XC) / XC
        YO = (YN(JJ) - YC) / YC
C ***********************************************************************
        CALL IGMA(XO, YO)
C ***********************************************************************
        DO 640 L = 1,, INC
```

```
          JJ = IN(LL,L)
          XO = (XN(JJ) - XC) / XC
          YO = (YN(JJ) - YC) / YC
C ***********************************************************************
          CALL IGDA(XO, YO)
C ***********************************************************************
  640   CONTINUE
        JJ = IN(LL,4)
        XO = (XN(JJ) - XC) / XC
        YO = (YN(JJ) - YC) / YC
C ***********************************************************************
        CALL IGDA(XO, YO)
C ***********************************************************************
  650 CONTINUE
C ***********************************************************************
      CALL IGENDS('PLOT')
C ***********************************************************************
      CALL IGDRON('TERMINAL')
C ***********************************************************************
      CALL IGXYIN(XIN(IC), YIN(IC))
C ***********************************************************************
      XP = XIN(IC) * XC + XC
      YP = YIN(IC) * YC + YC
      DO 670 I = 1, NPTS
        IF (XN(I) .LT. XC - 0.1*SCL .OR. XN(I) .GT. XC + 0.1*SCL)
     1     GO TO 670
        IF (YN(I) .LT. YP - 0.1*SCL .OR. YN(I) .GT. YP + 0.1*SCL)
     1     GO TO 670
        ITYPE(I) = 1
        WRITE (6,660)
  660   FORMAT ('More Constant head nodes to be defined?')
        READ (5,1610) IANSWR
        IF (IANSWR .EQ. YES) GO TO 630
C ***********************************************************************
        CALL IGCTRL('TERMINAL', 'ERASE')
C ***********************************************************************
        GO TO 350
  670 CONTINUE
      WRITE (6,680)
  680 FORMAT ('MISSED, TRY AGAIN')
      GO TO 630
  690 WRITE (6,700)
  700 FORMAT ('This routine will establish constant head nodes', /,
     1       'for each side of the generated mesh in the sequence', /,
     2       'in which they are queried.  The constant head value', /,
     3       'will be equal to the elevation of each node', //,
     4       'Top face?')
      READ (5,1610) IANSWR
      IF (IANSWR .EQ. NO) GO TO 730
      ICO = IFE(1)
```

```
      DO 720 ICO1 = 1, ICO
        DO 710 NC = 1, 4
          ITYPE(IN(IFACE(ICO1,1),NC)) = 1
710     CONTINUE
720   CONTINUE
730   WRITE (6,740)
740   FORMAT ('Bottom face?')
      READ (5,1610) IANSWR
      IF (IANSWR .EQ. NO) GO TO 770
      ICO = IFE(2)
      DO 760 ICO1 = 1, ICO
        DO 750 NC = 5, 8
          ITYPE(IN(IFACE(ICO1,2),NC)) = 1
750     CONTINUE
760   CONTINUE
770   WRITE (6,780)
780   FORMAT ('South face?')
      READ (5,1610) IANSWR
      IF (IANSWR .EQ. NO) GO TO 800
      ICO = IFE(3)
      DO 790 ICO1 = 1, ICO
        ITYPE(IN(IFACE(ICO1,3),2)) = 1
        ITYPE(IN(IFACE(ICO1,3),6)) = 1
        ITYPE(IN(IFACE(ICO1,3),7)) = 1
        ITYPE(IN(IFACE(ICO1,3),3)) = 1
790   CONTINUE
800   WRITE (6,810)
810   FORMAT ('East face?')
      READ (5,1610) IANSWR
      IF (IANSWR .EQ. NO) GO TO 830
      ICO = IFE(4)
      DO 820 ICO1 = 1, ICO
        ITYPE(IN(IFACE(ICO1,4),3)) = 1
        ITYPE(IN(IFACE(ICO1,4),7)) = 1
        ITYPE(IN(IFACE(ICO1,4),8)) = 1
        ITYPE(IN(IFACE(ICO1,4),4)) = 1
820   CONTINUE
830   WRITE(6,840)
840   FORMAT ('North face?')
      READ (5,1610) IANSWR
      IF (IANSWR .EQ. NO) GO TO 860
      ICO = IFE(5)
      DO 850 ICO1 = 1, ICO
        ITYPE(IN(IFACE(ICO1,5),[illegible])) = 1
        ITYPE(IN(IFACE(ICO1,5),8)) = 1
        ITYPE(IN(IFACE(ICO1,5),5)) = 1
        ITYPE(IN(IFACE(ICO1,5),[illegible])) = 1
850   CONTINUE
860   WRITE (6,870)
870   FORMAT ('West face?')
```

```
      READ (5,1610) IANSWR
      IF (IANSWR .EQ. NO) GO TO 890
      ICO = IFE(6)
      DO 880 ICO1 = 1, ICO
        ITYPE(IN(IFACE(ICO1,6),1)) = 1
        ITYPE(IN(IFACE(ICO1,6),5)) = 1
        ITYPE(IN(IFACE(ICO1,6),6)) = 1
        ITYPE(IN(IFACE(ICO1,6),2)) = 1
  880 CONTINUE
C Count the number of constant head nodes
  890 DO 900 JCON = 1, NPTS
        IF (ITYPE(JCON) .NE. 1) GO TO 900
        ICON = ICON + 1
  900 CONTINUE
      GO TO 350
C
C     SWITCH NO.5 DIGITIZE LEAKANCE ELEMENTS
C
  910 IF (N .LT. 1) GO TO 920
      GO TO 930
  920 WRITE (6,600)
      GO TO 350
  930 WRITE (6,940)
  940 FORMAT ('Move the cursor with the joystick to indicate the', /,
     1       'leakance elements in this view. Touch a different key to'
     2       , /, 'to enter a different leakance element')
  950 WRITE (6,960)
  960 FORMAT ('*****Enter leakance value*****')
      NLN = NLN + 1
      READ (5,970) ALEAK(NLN)
  970 FORMAT (1E10.4)
C ***************************************************************
      CALL IGCTRL('TERMINAL', 'ERASE')
C ***************************************************************
      CALL IGINIT
C ***************************************************************
      CALL IGBGNS('PLOT')
C ***************************************************************
      CALL IGMA(0.0, 0.0)
C ***************************************************************
      DO 990 LL = 1, NELM
        JJ = IN(LL,1)
        XO = (XN(JJ) - XC) / XC
        YO = (YN(JJ) - YC) / YC
C ***************************************************************
        CALL IGMA(XO, YO)
C ***************************************************************
        DO 980 L = 1, INC
          JJ = IN(LL,L)
          XO = (XN(JJ) - XC) / XC
```

```
      YO = (YN(JJ) - YC) / YC
C ***************************************************************
      CALL IGDA(XO, YO)
C ***************************************************************
  980 CONTINUE
      JJ = IN(LL,1)
      XO = (XN(JJ) - XC) / XC
      YO = (YN(JJ) - YC) / YC
C ***************************************************************
      CALL IGDA(XO, YO)
C ***************************************************************
  990 CONTINUE
C ***************************************************************
      CALL IGENDS('PLOT')
C ***************************************************************
      CALL IGDRON('TERMINAL')
C ***************************************************************
      CALL IGXYIN(XIN(IL), YIN(IL))
C ***************************************************************
      XP = XIN(IL) * XC + XC
      YP = YIN(IL) * YC + YC
      DO 1020 J = 1, NELM
        X = 0.
        Y = 0.
        DO 1000 K = 1, INC
          X = XN(IN(J,K)) + X
          Y = YN(IN(J,K)) + Y
 1000   CONTINUE
        XMEAN = X / INC
        YMEAN = Y / INC
        IF (XP .LT. XMEAN - 0.1*SCL .OR. XP .GT. XMEAN + 0.1*SCL)
     1    GO TO 1020
        IF (YP .LT. YMEAN - 0.1*SCL .OR. YP .GT. YMEAN + 0.1*SCL)
     1    GO TO 1020
        LEAK(NLN) = J
        WRITE (6,1010)
 1010   FORMAT ('More leakance elements to be defined?')
        READ (5,1610) IANSWR
        IF (IANSWR .EQ. YES) GO TO 950
C ***************************************************************
        CALL IGCTRL('TERMINAL', 'ERASE')
C ***************************************************************
      GO TO 350
 1020 CONTINUE
      WRITE (6,680)
      GO TO 950
C
C  SWITCH NO.7 DIGITIZE A PUMPING WELL IN X-Y PLANE
C
 1030 WRITE (6,1040)
```

```
 1040 FORMAT ('The location of a pumping well in the x-y plane will be',
     1       /, 'defined.  If more than one pumping well exists, select
     2switch', /, 'No. 7 again.', /, 'Enter the following pump informati
     3on', /)
      IP = IP + 1
      IF (ID .GT. 0) GO TO 1100
 1050 IPL = IPL + 1
      WRITE (6,1060)
 1060 FORMAT ('Enter the start time:')
      READ (5,120) TIME(IPL,1)
      WRITE (6,1070)
 1070 FORMAT ('Enter the finish time:')
      READ (5,120) TIME(IPL,2)
      WRITE (6,1080)
 1080 FORMAT ('Enter the number of time steps:')
      READ (5,1090) NTIME(IPL)
 1090 FORMAT (I4)
      GO TO 1150
 1100 IPL = IPL + 1
      WRITE (6,1110)
 1110 FORMAT ('Enter the simulation period:')
      READ (5,120) TIME(IPL,2)
      WRITE (6,1120)
 1120 FORMAT ('Enter the initial time step:')
      READ (5,120) TIME(IPL,1)
      WRITE (6,1130)
 1130 FORMAT ('Enter the multiplier for increasing the time step:')
      READ (5,120) CHNG(IPL)
      WRITE (6,1140)
 1140 FORMAT ('Enter the number of time steps between', /,
     1       'changes in delta time:')
      READ (5,1090) ITCHNG(IPL)
 1150 WRITE (6,1160)
 1160 FORMAT ('Enter the pumping rate in cu. ft/sec:')
      READ (5,120) PMPRTE(IPL,IP)
      WRITE (6,1170)
 1170 FORMAT ('More pumping information?')
      READ (5,1610) IANSWR
      IF (ID .GT. 0 .AND. IANSWR .EQ. YES) GO TO 1100
      IF (IANSWR .EQ. YES) GO TO 1050
      WRITE (6,1340)
      READ (5,390,END=2260) IX, IY
      IF (IX .GT. 1000) GO TO 430
 1180 IF (IP .GT. 9) GO TO 470
      PPOSX(IP) = X
      PPOSY(IP) = Y
      JCHAR = 92
      GO TO 350
C
C
```

```
C    SWITCH NO. 8 DIGITIZE OBSERVATION WELL IN X-Y PLANE
C
  1190 WRITE (6,1200)
  1200 FORMAT ('THE LOCATION OF AN OBSERVATION WELL IN THE X-Y PLANE ', /
     1
   260 'WILL BE DEFINED.  IF MORE THAN ONE PUMPING WELL EXISTS, SELECT',/
     1,
     2'SWITCH NO. 8 AGAIN.')
       IO = IO + 1
       READ (5,390,END=2260) IX, IY
       IF (IX .GT. 1000) GO TO 430
  1220 IF (IP .GT. 9) GO TO 470
       OBSX(IO) = X
       OBSY(IO) = Y
       JCHAR = 92
       GO TO 350
C
C
C  SWITCH NO. 9 DIGITIZE TRANSMISSIVITY BOUNDARY X-Y
C
  1230 WRITE (6,1240)
  1240 FORMAT ('Digitize the location of the transmissivity boundaries',
     1       /,
   260 'in the x-y plane.  It is necessary that you digitize around the',
     1 /,
   490 'entire perimeter of the transmissivity zone so that the program',
     1 /, 'will be able to generate the mesh within this boundary.  If',
     2 /, 'the user is working in 3-D, it is necessary to digitize the '
     3, /, 'perimeter of the upper surface of the zone after the lower',
     4 /,
  1250 'surface is done.  If there is more than one transmissivity zone,'
     1, /, 'select switch No.9 again after the mesh for the previous zon
     2e', /, 'has been generated', /, '*****ENTER THE STORATIVITY*****'
     3, / )
       ICKP = ICK
       NCEL(ITRNS,2) = NELM
       ITRNS = ITRNS + 1
       NCEL(ITRNS,1) = NELM1
       NPTSP = NPTS
       NPTSP1 = NPTS + 1
       N = NPTS1
       READ (5,1260) STOR(ITRNS)
  1260 FORMAT (E10.3)
       IF (ID .GT. 0) GO TO 1280
       WRITE (6,1270)
  1270 FORMAT ('Enter the transmissivity')
       READ (5,1260) COX(ITRNS)
       GO TO 1330
  1280 WRITE (6,1320)
       WRITE (6,1290)
```

```
1290 FORMAT ('Enter the hydraulic conductivity in the x-direction')
     READ (5,1260) COX(ITRNS)
     WRITE (6,1300)
1300 FORMAT ('Enter the hydraulic conductivity in the y-direction')
     READ (5,1260) COY(ITRNS)
     WRITE (6,1310)
     READ (5,1260) COZ(ITRNS)
1310 FORMAT ('Enter the hydraulic conductivity in the z-diraction')
1320 FORMAT ('Digitize the x-y coordinates of the upper surface in the'
    1       , /, 'same manner as the lower suface has been digitized fi
    2rst.')
1330 WRITE (6,1340)
1340 FORMAT ('NOW BEGIN DIGITIZING')
1350 READ (5,390,END=2260) IX, IY
     IF (IX .GT. 1000) GO TO 430
     ICK = IY / 1000 + 1
     WRITE (6,400) ICK
     IF (ICK .GT. 13) GO TO 350
     GO TO (350, 510, 540, 570, 910, 580, 1030, 1190, 1230, 1370,
    11380, 1590, 2000), ICK
1360 IF (N .GT. 4998) GO TO 470
     XN(N) = X
     YN(N) = Y
     N = N + 1
     NPTS = N - 1
     GO TO 1350
C
C  SWITCH NO. 10 Future Use
C
1370 GO TO 350
C
C  Switch No.11 Input Boundary Fluxes
C
1380 IF (ID .GT. 0) GO TO 1400
     WRITE (6,1390)
1390 FORMAT ('This switch has no signific[illegible]2-D')
     GO TO 350
1400 WRITE (6,1410)
1410 FORMAT ('This routine has the capability to give an entire', /,
    1       'face of a generated mesh a boundary flux value.  The', /,
    2       'user will be querried as to whether input a boundary', /,
    3       'flux for each face querried.', /, 'Top face?')
     IB = 0
     READ (5,1610) IANSWR
     IF (IANSWR .EQ. NO) GO TO 1440
     IB = IB + 1
     READ (5,1420) BFL
     BFL = BFL / (IFE(1)*4.)
1420 FORMAT (F10.0)
     ICO = IFE(1)
```

```
      DO 1430 ICO1 = 1, ICO
        ITYPE(IN(IFACE(ICO1,1),1)) = 2
        BFLUX(IN(IFACE(ICO1,1),1)) = BFLUX(IN(IFACE(ICO1,1),1)) + BFL
        ITYPE(IN(IFACE(ICO1,1),2)) = 2
        BFLUX(IN(IFACE(ICO1,1),2)) = BFLUX(IN(IFACE(ICO1,1),2)) + BFL
        ITYPE(IN(IFACE(ICO1,1),3)) = 2
        BFLUX(IN(IFACE(ICO1,1),3)) = BFLUX(IN(IFACE(ICO1,1),3)) + BFL
        ITYPE(IN(IFACE(ICO1,1),4)) = 2
        BFLUX(IN(IFACE(ICO1,1),4)) = BFLUX(IN(IFACE(ICO1,1),4)) + BFL
 1430 CONTINUE
 1440 WRITE (6,1450)
 1450 FORMAT ('Bottom face?')
      READ (5,1610) IANSWR
      IF (IANSWR .EQ. NO) GO TO 1470
      READ (5,1420) BFL
      BFL = BFL / (IFE(2)*4.)
      ICO = IFE(2)
      DO 1460 ICO1 = 1, ICO
        ITYPE(IN(IFACE(ICO1,2),5)) = 2
        BFLUX(IN(IFACE(ICO1,2),5)) = BFLUX(IN(IFACE(ICO1,2),5)) + BFL
        ITYPE(IN(IFACE(ICO1,2),6)) = 2
        BFLUX(IN(IFACE(ICO1,2),6)) = BFLUX(IN(IFACE(ICO1,2),6)) + BFL
        ITYPE(IN(IFACE(ICO1,2),7)) = 2
        BFLUX(IN(IFACE(ICO1,2),7)) = BFLUX(IN(IFACE(ICO1,2),7)) + BFL
        ITYPE(IN(IFACE(ICO1,2),8)) = 2
        BFLUX(IN(IFACE(ICO1,2),8)) = BFLUX(IN(IFACE(ICO1,2),8)) + BFL
 1460 CONTINUE
 1470 WRITE (6,1480)
 1480 FORMAT ('South face?')
      READ (5,1610) IANSWR
      IF (IANSWR .EQ. NO) GO TO 1500
      READ (5,1420) BFL
      BFL = BFL / (IFE(3)*4.)
      ICO = IFE(3)
      DO 1490 ICO1 = 1, ICO
        ITYPE(IN(IFACE(ICO1,3),2)) = 2
        BFLUX(IN(IFACE(ICO1,3),2)) = BFLUX(IN(IFACE(ICO1,3),2)) + BFL
        ITYPE(IN(IFACE(ICO1,3),6)) = 2
        BFLUX(IN(IFACE(ICO1,3),6)) = BFLUX(IN(IFACE(ICO1,3),6)) + BFL
        ITYPE(IN(IFACE(ICO1,3),7)) = 2
        BFLUX(IN(IFACE(ICO1,3),7)) = BFLUX(IN(IFACE(ICO1,3),7)) + BFL
        ITYPE(IN(IFACE(ICO1,3),3)) = 2
        BFLUX(IN(IFACE(ICO1,3),3)) = BFLUX(IN(IFACE(ICO1,3),3)) + BFL
 1490 CONTINUE
 1500 WRITE (6,1510)
 1510 FORMAT ('East face?')
      READ (5,1610) IANSWR
      IF (IANSWR .EQ. NO) GO TO 1530
      ICO = IFE(4)
      READ (5,1420) BFL
```

```
      BFL = BFL / (IFE(4)*4.)
      DO 1520 ICO1 = 1, ICO
        ITYPE(IN(IFACE(ICO1,4),3)) = 2
        BFLUX(IN(IFACE(ICO1,4),3)) = BFLUX(IN(IFACE(ICO1,4),3)) + BFL
        ITYPE(IN(IFACE(ICO1,4),7)) = 2
        BFLUX(IN(IFACE(ICO1,4),7)) = BFLUX(IN(IFACE(ICO1,4),7)) + BFL
        ITYPE(IN(IFACE(ICO1,4),8)) = 2
        BFLUX(IN(IFACE(ICO1,4),8)) = BFLUX(IN(IFACE(ICO1,4),8)) + BFL
        ITYPE(IN(IFACE(ICO1,4),4)) = 2
        BFLUX(IN(IFACE(ICO1,4),4)) = BFLUX(IN(IFACE(ICO1,4),4)) + BFL
 1520 CONTINUE
 1530 WRITE (6,1540)
 1540 FORMAT ('North face?')
      READ (5,1610) IANSWR
      IF (IANSWR .EQ. NO) GO TO 1560
      ICO = IFE(5)
      READ (5,1420) BFL
      BFL = BFL / (IFE(5)*4.)
      DO 1550 ICO1 = 1, ICO
        ITYPE(IN(IFACE(ICO1,5),4)) = 2
        BFLUX(IN(IFACE(ICO1,5),4)) = BFLUX(IN(IFACE(ICO1,5),4)) + BFL
        ITYPE(IN(IFACE(ICO1,5),8)) = 2
        BFLUX(IN(IFACE(ICO1,5),8)) = BFLUX(IN(IFACE(ICO1,5),8)) + BFL
        ITYPE(IN(IFACE(ICO1,5),5)) = 2
        BFLUX(IN(IFACE(ICO1,5),5)) = BFLUX(IN(IFACE(ICO1,5),5)) + BFL
        ITYPE(IN(IFACE(ICO1,5),1)) = 2
        BFLUX(IN(IFACE(ICO1,5),1)) = BFLUX(IN(IFACE(ICO1,5),1)) + BFL
 1550 CONTINUE
 1560 WRITE (6,1570)
 1570 FORMAT ('West face?')
      READ (5,1610) IANSWR
      IF (IANSWR .EQ. NO) GO TO 350
      ICO = IFE(6)
      READ (5,1420) BFL
      BFL = BFL / (IFE(6)*4.)
      DO 1580 ICO1 = 1, ICO
        ITYPE(IN(IFACE(ICO1,6),1)) = 2
        BFLUX(IN(IFACE(ICO1,6),1)) = BFLUX(IN(IFACE(ICO1,6),1)) + BFL
        ITYPE(IN(IFACE(ICO1,6),5)) = 2
        BFLUX(IN(IFACE(ICO1,6),5)) = BFLUX(IN(IFACE(ICO1,6),5)) + BFL
        ITYPE(IN(IFACE(ICO1,6),6)) = 2
        BFLUX(IN(IFACE(ICO1,6),6)) = BFLUX(IN(IFACE(ICO1,6),6)) + BFL
        ITYPE(IN(IFACE(ICO1,6),2)) = 2
        BFLUX(IN(IFACE(ICO1,6),2)) = BFLUX(IN(IFACE(ICO1,6),2)) + BFL
 1580 CONTINUE
      GO TO 350
C
C  SWITCH NO. 12  STORE INFO FOR LATER USE
C
 1590 WRITE (6,1600)
```

```
1600 FORMAT ('DO YOU WISH TO STOP AND STORE THE DATA ?')
     READ (5,1610) MQUES6
1610 FORMAT (A1)
     IF (MQUES6 .EQ. NO) GO TO 1630
     NCEL(ITRNS,2) = NELM
     WRITE (6,1620)
1620 FORMAT ('Device 7 contains digital info., Device 9 contains plot',
    */, 'data:  Store appropriate files attached to these Device
    2s.')
     GO TO 1650
1630 WRITE (6,1640)
1640 FORMAT ('THE PLOT WAS NOT ENDED, SELECT A NEW SWITCH')
     GO TO 350
C ****************************************************************
1650 CALL SETUP(IDIFF)
C ****************************************************************
     CALL OPTNUM(IDIFF, NBW)
C ****************************************************************
     IF (ID .GT. 0) GO TO 1830
     WRITE (7,1660)
1660 FORMAT ('FE')
     WRITE (7,1670) NPTS, NELM, NBW, NLN
1670 FORMAT (4I5)
     DO 1700 I = 1, NPTS
       WRITE (7,1680) I, XN(I), YN(I), ITYPE(I), CX(I)
1680   FORMAT (I5, 2F10.3, I5, F10.3)
1690   FORMAT (I5, 2F10.3, 4X, 'C', F10.4)
1700 CONTINUE
     DO 1740 J = 1, NELM
       IF (INC .EQ. 3) WRITE (7,1710) (J,(IN(J,K),K=1,3))
       IF (INC .EQ. 4) WRITE (7,1720) (J,(IN(J,K),K=1,4))
       IF (INC .EQ. 8) WRITE (7,1730) (J,(IN(J,K),K=1,8))
1710   FORMAT (4I5)
1720   FORMAT (5I5)
1730   FORMAT (9I5)
1740 CONTINUE
     DO 1760 I = 1, NLN
       WRITE (7,1750) LEAK(I), ALEAK(I)
1750   FORMAT (I5, 1E10.4)
1760 CONTINUE
     WRITE (7,1770) STORAT
1770 FORMAT (E10.3)
     DO 1810 I = 1, IPL
       WRITE (7,1780) TIME(I,1), TIME(I,2), NTIME(I), IP
1780   FORMAT (2F10.5, 2I5)
       DO 1800 I1 = 1, IP
         WRITE (7,1790) I1, I1, PMPRTE(I,I1)
1790     FORMAT ('PW', I2, I6, F10.7)
1800   CONTINUE
1810 CONTINUE
```

```
      WRITE (7,1820)
 1820 FORMAT (/, 'XX')
      GO TO 350
 1830 WRITE (7,1880)
      WRITE (7,1890) TITLE
      WRITE (7,1900) NPTS, NELM, NBW, ICON, ITRNS
      WRITE (7,1910) (I,XN(I),YN(I),ZN(I),ITYPE(I),I=1,NPTS)
      WRITE (7,1920) (J,(IN(J,INN(L)),L=1,INC),J=1,NELM)
      WRITE (7,1940) BETA, DENS
      DO 1840 IT = 1, ITRNS
        WRITE (7,1930) NCEL(IT,1), NCEL(IT,2), COX(IT), COY(IT),
     1  COZ(IT), HETA, TETA, AL
 1840 CONTINUE
      ISPL = 0
      WRITE (7,1950) ISPL
      IF (ISS .LT. 1) GO TO 1870
      DO 1860 IPU = 1, IP
        WRITE (7,1960) TIME(IPU,2), TIME(IPU,1), CHNG(IPU), ITCHNG(IPU),
     1  IPPN(IPU)
        IPNN = IPPN(IPU)
        DO 1850 IPN1 = 1, IPNN
          WRITE (7,1980) NPMP(IPN1), PMPRTE(IPN1,IPU)
 1850   CONTINUE
 1860 CONTINUE
 1870 WRITE (7,1970)
      WRITE (7,1990)
 1880 FORMAT ('ISOB                LINEAR')
 1890 FORMAT (20A4)
 1900 FORMAT (5I5)
 1910 FORMAT (I5, 3F15.3, I5)
 1920 FORMAT (9I5)
 1930 FORMAT (2I5, 7E10.3)
 1940 FORMAT (2F15.12)
 1950 FORMAT (I5)
 1960 FORMAT (3F10.2, 2I10)
 1970 FORMAT (6X, '999.')
 1980 FORMAT (I5, F10.5)
 1990 FORMAT (/, 'XXXX')
      GO TO 350
C
C     SWITCH NO.13
C
 2000 WRITE (6,2010)
 2010 FORMAT ('DO YOU WISH TO GENERATE A MESH? (Y OR N)')
      READ (5,1610) MQUES7
      IF (MQUES7 .EQ. NO) GO TO 350
      YTOP = AGSENS('TERMINAL','YSIZE')
      CALL IGTRAN('*MP*', 'SCALE', YTOP)
      IZD = 0
      INC = 4
```

```
      IF (IELEM .EQ. 0 .AND. ID .LT. 1) INC = 3
      IF (IELEM .EQ. 1 .AND. ID .EQ. 1) INC = 8
C ************************************************************
      CALL DSTRB
C ************************************************************
      IF (IP .GT. 0) CALL SMOOTH
      NCEL(ITRNS,2) = NELM
C ************************************************************
      IF (ID [illegible]O TO 2060
 2020 CALL IG[illegible]
C ************************************************************
C ************************************************************
C ************************************************************
      DO 2050 [illegible] ITRNS
        NE1 = [illegible]ITR,1)
        NE2 = NCEL(ITR,2)
        CALL IGMA(0.0, 0.0)
C ************************************************************
        DO 2040 LL = NE1, NE2
          IPIC = IGBGNS(0)
          XLIMIT = 1.0
          JJ = IN(LL,1)
          XO = (XN(JJ) - XC) / XC
          YO = ((YN(JJ) - YC)/YC)
C ************************************************************
          CALL IGMA(XO, YO)
C ************************************************************
          DO 2030 L = 1, INC
            JJ = IN(LL,L)
            XO = (XN(JJ) - XC) / XC
            YO = ((YN(JJ) - YC)/YC)
C ************************************************************
            CALL IGDA(XO, YO)
C ************************************************************
 2030     CONTINUE
          JJ = IN(LL,1)
          XO = (XN(JJ) - XC) / XC
          YO = ((YN(JJ) - YO)/YC)
C ************************************************************
          CALL IGDA(XO, YO)
C ************************************************************
 2040   CONTINUE
C ************************************************************
        CALL IGHUE(IPIC, COL(ITR))
C ************************************************************
        CALL IGENDS(IPIC)
 2050 CONTINUE
C ************************************************************
      CALL IGDRON('TERMINAL')
C ************************************************************
```

```
      CALL ASKPLT
C ************************************************************
      CALL IGCTRL('TERMINAL', 'ERASE')
C ************************************************************
      NPTS1 = NPTS + 1
      NELM1 = NELM + 1
      IF (ID .LT. 1) GO TO 350
 2060 CONTINUE
      IF (IELEM .EQ. 1) GO TO 2100
      ZO = 0.0
C ************************************************************
      CALL IGINIT
C ************************************************************
      CALL IGCTRL('TERMINAL', 'ERASE')
C ************************************************************
      CALL IGBGNS('PICT')
C ************************************************************
      DO 2160 ITR = 1, ITRNS
        NE1 = NCEL(ITR,1)
        NE2 = NCEL(ITR,2)
        CALL IGMA(0.0, 0.0, 0.0)
C ************************************************************
        DO 2090 LL = NE1, NE2
          XLIMIT = 1.0
          JJ = IN(LL,1)
          XO = (XN(JJ) - XC) / XC
          YO = ((YN(JJ) - YC)/YC)
          ZO = ZN(JJ) / ZC + 1.
C ************************************************************
          CALL IGMA(XO, YO, ZO)
C ************************************************************
          DO 2070 L = 1, INC
            JJ = IN(LL,L)
            XO = (XN(JJ) - XC) / XC
            YO = ((YN(JJ) - YC)/YC)
            ZO = ZN(JJ) / ZC + 1.
C ************************************************************
            CALL IGDA(XO, YO, ZO)
C ************************************************************
 2070     CONTINUE
          JJ = IN(LL,1)
          XO = (XN(JJ) - XC) / XC
          YO = ((YN(JJ) - YC)/YC)
          ZO = ZN(JJ) / ZC + 1.
 2080     CONTINUE
C ************************************************************
          CALL IGDA(XO, YO, ZO)
C ************************************************************
 2090   CONTINUE
        GO TO 2150
```

```
C ***********************************************************
 2100   CALL IGINIT
        ZO=0.0
C ***********************************************************
        CALL IGCTRL('TERMINAL', 'ERASE')
C ***********************************************************
        CALL IGBGNS('PICT')
C ***********************************************************
        CALL IGMA(0.0, 0.0, 0.0)
C ***********************************************************
        DO 2140 LL = NE1, NE2
          JJ = IN(LL,1)
          XO = (XN(JJ) - XC) / XC
          YO = ((YN(JJ) - YC)/YC)
          ZO = ZN(JJ)/ZC+1
C ***********************************************************
          CALL IGMA(XO, YO, ZO)
C ***********************************************************
          DO 2120 L = 1, 3
            X1(1) = (XN(IN(LL,L)) - XC) / XC
            Y1(1) = ((YN(IN(LL,L)) - YC)/YC)
            Z1(1) = (ZN(IN(LL,L)) - ZC) / ZC + 1.
            X1(2) = (XN(IN(LL,L + 4)) - XC) / XC
            Y1(2) = ((YN(IN(LL,L + 4)) - YC)/YC)
            Z1(2) = (ZN(IN(LL,L + 4)) - ZC) / ZC + 1.
            X1(3) = (XN(IN(LL,L + 5)) - XC) / XC
            Y1(3) = ((YN(IN(LL,L + 5)) - YC)/YC)
            Z1(3) = (ZN(IN(LL,L + 5)) - ZC) / ZC + 1.
            X1(4) = (XN(IN(LL,L + 1)) - XC) / XC
            Y1(4) = ((YN(IN(LL,L + 1)) - YC)/YC)
            Z1(4) = (ZN(IN(LL,L + 1)) - ZC) / ZC + 1.
C ***********************************************************
            CALL IGMA(X1(1), Y1(1), Z1(1))
C ***********************************************************
            DO 2110 L1 = 1, 4
C ***********************************************************
              CALL IGDA(X1(L1), Y1(L1), Z1(L1))
C ***********************************************************
 2110       CONTINUE
C ***********************************************************
            CALL IGDA(X1(1), Y1(1), Z1(1))
C ***********************************************************
 2120     CONTINUE
          X1(1) = (XN(IN(LL,1)) - XC) / XC
          Y1(1) = ((YN(IN(LL,1)) - YC)/YC)
          Z1(1) = (ZN(IN(LL,1)) - ZC) / ZC + 1.
          X1(2) = (XN(IN(LL,5)) - XC) / XC
          Y1(2) = ((YN(IN(LL,5)) - YC)/YC)
          Z1(2) = (ZN(IN(LL,5)) - ZC) / ZC + 1.
          X1(3) = (XN(IN(LL,8)) - XC) / XC
```

```
      Y1(3) = ((YN(IN(LL,8)) - YC)/YC)
      Z1(3) = (ZN(IN(LL,8)) - ZC) / ZC + 1.
      X1(4) = (XN(IN(LL,4)) - XC) / XC
      Y1(4) = ((YN(IN(LL,4)) - YC)/YC)
      Z1(4) = (ZN(IN(LL,4)) - ZC) / ZC + 1.
C ************************************************************
      CALL IGMA(X1(1), Y1(1), Z1(1))
C ************************************************************
      DO 2130 L1 = 1, 4
C ************************************************************
        CALL IGDA(X1(L1), Y1(L1), Z1(L1))
C ************************************************************
 2130 CONTINUE
C ************************************************************
      CALL IGDA(X1(1), Y1(1), Z1(1))
C ************************************************************
 2140 CONTINUE
 2150 CONTINUE
C ************************************************************
      CALL IGENDS('PICT')
C ************************************************************
      CALL IGHUE('PICT', COL(ITR))
 2160 CONTINUE
C ************************************************************
      CALL IGDRON('TERMINAL')
C ************************************************************
      NPTS1 = NPTS + 1
      NELM1 = NELM + 1
 2170 WRITE (6,2180)
 2180 FORMAT ('To view the results of the 3-D generation, it is', /,
     1 'necessary to ratate the picture on the screen.  The dir-',
     2 /, 'ection of rotation is dtermined by the "RIGHT HAND RULE
     3"', /, 'ie; Point the thumb of the right hand in the positive', /,
     4 'direction of the axis(Z into the screen is positive).', /,
     5 'The fingers of the right Hand will curl in the positive',
     6 /, 'direction of rotation', //, 'Enter the angles' in degree
     7s', /, 'ANGLEX ')
      READ (5,2190) ANGLEX
 2190 FORMAT (F7.2)
      ANGLEX = (ANGLEX/180) * 3.1416
      WRITE (6,2200)
 2200 FORMAT ('ANGLEY')
      READ (5,2190) ANGLEY
      ANGLEY = (ANGLEY/180.) * 3.1416
      WRITE (6,2210)
 2210 FORMAT ('ANGLEZ')
      READ (5,2190) ANGLEZ
      ANGLEZ = (ANGLEZ/180.) * 3.1416
C ************************************************************
      CALL IGTRAN('PICT', 'CURRENT', 'ROTX', ANGLEX)
```

```
C ***************************************************************
      CALL IGTRAN('PICT', 'CURRENT', 'ROTY', ANGLEY)
C ***************************************************************
      CALL IGTRAN('PICT', 'CURRENT', 'ROTZ', ANGLE
C ***************************************************************
      CALL IGDRON('TERMINAL')
C ***************************************************************
      CALL ASKPLT
C ***************************************************************
      WRITE (6,2220)
 2220 FORMAT ('DO YOU WISH TO SEE THIS VIEW IN PERSPECTIVE?')
      READ (5,1610) IANSWR
      IF (IANSWR .NE. YES) GO TO 2230
C ***************************************************************
      CALL IGTRAN('PICT', 'CURRENT', 'PROJ', 0.0, 0.5)
C ***************************************************************
      CALL IGDRON('TERMINAL')
C ***************************************************************
 2230 CALL ASKPLT
C ***************************************************************
      WRITE (6,2240)
 2240 FORMAT ('Do you wish to rotate this figure some more?')
      READ (5,1610) IANSWR
C ***************************************************************
      CALL IGCTRL('TERMINAL', 'ERASE')
C ***************************************************************
      IF (IANSWR .NE. YES) GO TO 350
      GO TO 2170
C
C
C   SWITCH NO.14
 2250 CONTINUE
      GO TO 350
C
 2260 STOP
      END
C
C
C..................................................
C
C   Subroutine SMOOTH
C
C   Author
C       G.J. Smith(U. of Alberta)
C
C   Last Update
C       May 1984
C
C   Purpose
C       To set up the concentration of elements around
```

```
C       a pumping well and smooth out the distribution
C       of elements to the pumping well.
C
C   Usage
C       CALL SMOOTH
C
C   Subroutines Required
C       CNCTQ,CNCTT1,CNCTT2
C
C
C .....................................................
      SUBROUTINE SMOOTH
      COMMON MEMJT(5000,32), IN(3000,8), IFACE(3500,6), YN(5000),
     1       ZN(5000)
      COMMON CX(5000)
      COMMON XN(5000), JMEM(5000), NBDND(4000), NDX(5000), NDY(5000)
      COMMON NDZ(5000), ITYPE(5000), NDWEL(20,2,25), LEAK(100),
     1       ALEAK(100)
      COMMON XIN(100), YIN(100), ISAVE(25,2), PPOSX(10), PPOSY(10),
     1       PPOSZ(10)
      COMMON OBSX(10), OBSY(10), OBSZ(10), ND(16), NND(16)
      COMMON COX(10), COY(10), COZ(10), IFE(6), NPMP(25), IPPN(10)
      COMMON ICRNR(10,8), NFACE(6,4), ICNTST(4,2), NODE1(8)
      COMMON IP, NPTS, NPTS1, NELM, IO, ITRNS, INC, IELEM, NZ, ID, YC,
     1       XC, IPN
      COMMON DELX, DELY, SCL, IW1, IW2, IK, NBN, NPP, NXSTPS, NYSTPS,
     1       DELTZ
C     THIS SUBROUTINE SMOOTHES OUT THE RESULTANT MESH USING
C     AN AVERAGING TECHNIQUE
   10 FORMAT (3I5)
      K = 1
      IMM = 1
      II = 1
      IFLG = 0
      IF (ITRNS .LT. 2) IP1 = 1
      I = NPTS1
      DO 30 IM = IP1, IP
        DO 20 IMM = 1, IW2
          IMM1 = IMM
          IF (ID .GT. 0) IMM1 = IMM + 1
          I2 = NDWEL(IM,2,IMM1)
          IF (I2 .EQ. 0) GO TO 30
C ************************************************************
          IF (IELEM .GT. 0) CALL CNCTQ(I2, IM, IMM1)
C ************************************************************
          IF (IELEM .LT. 1) CALL CNCTT1(I2, IM, IMM1)
C ************************************************************
   20   CONTINUE
   30 CONTINUE
      IP1 = IP
```

```
 40 CONTINUE
    DO 50 LU = 1, NBN
      IF (I .EQ. NBDND(LU)) GO TO 140
 50 CONTINUE
    DO 80 L1 = 1, 2
      DO 70 L2 = 1, IP
        DO 60 L3 = 1, IW2
          IF (I .EQ. NDWEL(L2,L1,L3)) GO TO 140
 60     CONTINUE
 70   CONTINUE
 80 CONTINUE
    DO 110 J = 1, NELM
      DO 100 L = 1, INC
        IF (I .EQ. IN(J,L)) GO TO 90
        GO TO 100
 90     ISAVE(K,1) = J
        K = K + 1
        GO TO 110
100   CONTINUE
110 CONTINUE
    X = O.
    Y = O.
    KL = O
    KK = K - 1
    DO 130 K1 = 1, KK
      DO 120 L = 1, INC
        N = IN(ISAVE(K1,1),L)
        IF (N .EQ. I) GO TO 120
        X = XN(N) + X
        Y = YN(N) + Y
        KL = KL + 1
120   CONTINUE
130 CONTINUE
    XN(I) = X / KL
    YN(I) = Y / KL
    GO TO 140
140 IF (IFLG .EQ. 1) GO TO 150
    IF (I .LT. NPTS) GO TO 160
    I = NPTS
150 IF (I .LE. NPTS1) GO TO 170
    IFLG = 1
    K = 1
    I = I - 1
    GO TO 40
160 I = I + 1
    K = 1
    GO TO 40
170 CONTINUE
    NPTS1 = NPTS + 1
    RETURN
```

```
      END
C
C
C ..........................................................
C
C   Subroutine DSTRB
C
C   Author
C       G.J. Smith(U. of Alberta)
C
C   Last Update
C       May 1984
C
C   Purpose
C       To regularuly redistribute the the digitized
C       nodes so as to set up the mesh generation
C       process.
C
C
C   Usage
C       CALL DSTRB
C
C   Subroutines Required
C       None
C
C
C ..........................................................
      SUBROUTINE DSTRB
      COMMON MEMJT(5000,32), IN(3000,8), IFACE(3500,6), YN(5000),
     1       ZN(5000)
      COMMON CX(5000)
      COMMON XN(5000), JMEM(5000), NBDND(4000), NDX(5000), NDY(5000)
      COMMON NDZ(5000), ITYPE(5000), NDWEL(20,2,25), LEAK(100),
     1       ALEAK(100)
      COMMON XIN(100), YIN(100), ISAVE(25,2), PPOSX(10), PPOSY(10),
     1       PPOSZ(10)
      COMMON OBSX(10), OBSY(10), OBSZ(10), ND(16), NND(16)
      COMMON COX(10), COY(10), COZ(10), IFE(6), NPMP(25), IPPN(10)
      COMMON ICRNR(10,8), NFACE(6,4), ICNTST(4,2), NODE1(8)
      COMMON IP, NPTS, NPTS1, NELM, IO, ITRNS, INC, IELEM, NZSTPS, ID,
     1       YC, XC, IPN
      COMMON DELX, DELY, SCL, IW1, IW2, IK, NBN, NPP, NXSTPS, NYSTPS,
     1       DELTZ
      DIMENSION XNEW(5000), YNEW(5000), ZNEW(5000), NFACE1(6,4)
      INTEGER YES /'Y'/, NO /'N'/
      IC1 = ITRNS
      J = NELM + 1
      NZSTPS = 1
      IL = 0
      IC = 0
```

```
      IFLG = 0
      LC = 0
   10 FORMAT ('IN MESH SUBROUTINE')
   20 FORMAT (A1)
   30 FORMAT (I5)
   40 FORMAT ('NPTS=', I5)
   50 FORMAT (2I5)
      YTOP = AGSENS('TERMINAL','YSIZE')
      CALL IGTRAN('*MP*', 'SCALE', YTOP)
   60 WRITE (6,70)
   70 FORMAT ('The screen will display the digitized figure.', /,
     1        'It is necessary that the user indicate the corners of', /,
     2        'the area that the mesh is to be generated.  The user,', /,
     3        ' to obtain a well conditioned mesh, must indicate the', /,
     4        ' 4 corners of the area of the tightest mesh first.  ', /,
     5        'The user will then be prompted to indicate the corners', /
     6        , ' of the adjacent portions of differring transmissivities
     7 ', /, 'Touch <RETURN> to continue.')
      READ (5,80) KEY
   80 FORMAT (I1)
C ***********************************************************
      CALL IGCTRL('TERMINAL', 'ERASE')
C ***********************************************************
      CALL IGINIT
C ***********************************************************
      CALL IGBGNS('QUAD')
C ***********************************************************
      CALL IGMA(0.0, 0.0)
C ***********************************************************
      X1 = (XN(NPTS1) - XC) / XC
      Y1 = ((YN(NPTS1) - YC)/YC)
C ***********************************************************
      CALL IGMA(X1, Y1)
C ***********************************************************
      DO 90 IA = NPTS1, NPTS
        X1 = (XN(IA) - XC) / XC
        Y1 = ((YN(IA) - YC)/YC)
C ***********************************************************
        CALL IGDA(X1, Y1)
C ***********************************************************
   90 CONTINUE
      X1 = (XN(NPTS1) - XC) / XC
      Y1 = ((YN(NPTS1) - YC)/YC)
C ***********************************************************
      CALL IGDA(X1, Y1)
C ***********************************************************
      CALL IGDRON('TERMINAL')
C ***********************************************************
      IF (ID .LT. 1) GO TO 110
      WRITE (6,100)
```

```
100 FORMAT ('Move the cursor with the joystick to indicate the 8', /,
   1       'corners of the figure in which the mesh is to be generated
   2.', /, 'Enter the lower surface first in the following order: -LOW
   3ER', /, 'LEFT-,-LOWER RIGHT-,-UPPER RIGHT-,UPPER LEFT-.  Touch a '
   4       , /, 'different key to enter each corner and then do the up
   5per         ', /, 'surface in the same manner.  Touch <RETURN> to
   6continue.')
    GO TO 130
110 WRITE (6,120)
120 FORMAT ('Move the cursor with the joystick to indicate the 4', /,
   1       ' corners of the area in which the mesh is to be generated.
   2', /, 'Enter -LOWER LEFT-, -LOWER RIGHT-, -UPPER RIGHT-, -UPPER ',
   3       /, 'LEFT-.', /, 'Touch a different key to enter each corner
   4.  Touch <RETURN>', /, ' to continue')
130 NCN = 4
    READ (5,30) IANSWR
C **********************************************************
    CALL IGCTRL('TERMINAL', 'ERASE')
C **********************************************************
    IF (ID .GT. 0) NCN = 8
140 DO 210 IC = 1, NCN
C **********************************************************
150   CALL IGXYIN(XIN(IC), YIN(IC))
C **********************************************************
      XN(NPTS + IC) = XIN(IC) * XC + XC
      YN(NPTS + IC) = (YIN(IC)*YC) + YC
      IF (ID .LT. 1) GO TO 180
      WRITE (6,160)
160   FORMAT ('Elevation?')
      READ (5,170) ZN(NPTS + IC)
170   FORMAT (F10.0)
180   WRITE (6,190)
190   FORMAT ('IS THIS CORRECT?')
      READ (5,20) IANSWR
      IF (IANSWR .EQ. NO) GO TO 150
200   FORMAT (3F13.3)
210 CONTINUE
    WRITE (6,220)
220 FORMAT ('Did all of the 4(8) corners get input ', /,
   1       'corectly?  If no, redo all 4(8).')
    READ (5,20) IANSWR
    IF (IANSWR .EQ. NO) GO TO 140
C **********************************************************
    CALL IGENDS('QUAD')
C **********************************************************
    CALL IGDELS('QUAD')
C **********************************************************
C  Generate NFACE(# of faces) array
    IFC = 1
    DO 230 M = 1, 4
```

```
          NFACE(IFC,M) = NPTS + M
          NFACE1(IFC,M) = M
          NFACE1(IFC + 1,M) = M + 4
230 CONTINUE
    IF (ID .LT. 1) GO TO 250
    DO 240 M = 1, 4
      NFACE(IFC + 1,M) = NPTS + M + 4
240 CONTINUE
    NFACE(3,1) = NPTS + 5
    NFACE(3,2) = NPTS + 1
    NFACE(3,3) = NPTS + 2
    NFACE(3,4) = NPTS + 6
    NFACE(4,1) = NPTS + 6
    NFACE(4,2) = NPTS + 2
    NFACE(4,3) = NPTS + 3
    NFACE(4,4) = NPTS + 7
    NFACE(5,1) = NPTS + 7
    NFACE(5,2) = NPTS + 3
    NFACE(5,3) = NPTS + 4
    NFACE(5,4) = NPTS + 8
    NFACE(6,1) = NPTS + 8
    NFACE(6,2) = NPTS + 4
    NFACE(6,3) = NPTS + 1
    NFACE1(6,1) = 1
    NFACE1(6,2) = 5
    NFACE1(6,3) = 6
    NFACE1(6,4) = 2
    NFACE1(3,1) = 2
    NFACE1(3,2) = 6
    NFACE1(3,3) = 7
    NFACE1(3,4) = 3
    NFACE1(4,1) = 3
    NFACE1(4,2) = 7
    NFACE1(4,3) = 8
    NFACE1(4,4) = 4
    NFACE1(5,1) = 4
    NFACE1(5,2) = 8
    NFACE1(5,3) = 5
    NFACE1(5,4) = 1
    NFACE(6,4) = NPTS + 5
250 CONTINUE
    WRITE (6,260)
260 FORMAT ('Enter the number of elements to be generated in each', /,
270 'of the X,Y, and Z directions.  Remember that common sides must',
   1 /, 'have the same number of elements along their common sides.',
   2 /, '# of X-elements= ?')
    READ (5,50) NXSTPS
    WRITE (6,280)
280 FORMAT ('# of Y-elements= ?')
    READ (5,50) NYSTPS
```

```
      IF (ID .LT. 1) GO TO 300
      WRITE (6,290)
  290 FORMAT ('# of Z-elements= ?')
      READ (5,50) NZSTPS
  300 WRITE (6,190)
      READ (5,20) IANSWR
      IF (IANSWR .EQ. NO) GO TO 250
      IFCC = 6
      IF (ID .LT. 1) IFCC = 1
      I = NPTS1
      II = NPTS1
      IC = 0
      DO 1150 IFC = 1, IFCC
        IF (ITRNS .LT. 2) GO TO 780
        IND = 0
        DO 310 M = 1, 4
          ICNTST(M,1) = 0
          ICNTST(M,2) = 0
  310   CONTINUE
        IF (ID .GT. 0) GO TO 340
        DO 330 M = 1, 4
          DO 320 N = 1, NBN
            IF (XN(NFACE(IFC,M)) .LT. (XN(NBDND(N)) - 0.5*DELX) .OR. XN(
     1         NFACE(IFC,M)) .GT. (XN(NBDND(N)) + 0.5*DELX))
     2         GO TO 320
            IF (YN(NFACE(IFC,M)) .LT. (YN(NBDND(N)) - 0.5*DELY) .OR. YN(
     1         NFACE(IFC,M)) .GT. (YN(NBDND(N)) + 0.5*DELY))
     2         GO TO 320
            NFACE(IFC,M) = NBDND(N)
            ICRNR(IC1,M) = NBDND(N)
  320     CONTINUE
  330   CONTINUE
        GO TO 780
  340   IF (IFC .GE. 2) GO TO 780
        DO 440 M = 1, 4
          NFN = IFE(IFC)
          DO 430 N = 1, NFN
            DO 350 MU = 1, 4
              MU1 = NFACE1(IFC,MU)
              IFN = IN(IFACE(N,IFC),MU1)
              IF (XN(NFACE(IFC,M)) .LT. (XN(IFN) - 0.5*DELX) .OR. XN(
     1           NFACE(IFC,M)) .GT. (XN(IFN) + 0.5*DELX)) GO TO 350
              IF (YN(NFACE(IFC,M)) .LT. (YN(IFN) - 0.5*DELY) .OR. YN(
     1           NFACE(IFC,M)) .GT. (YN(IFN) + 0.5*DELY)) GO TO 350
              IF (ID .LT. 1) GO TO 360
              IF (ZN(NFACE(IFC,M)) .LT. (ZN(IFN) - 0.5*DELZ) .OR. ZN(
     1           NFACE(IFC,M)) .GT. (ZN(IFN) + 0.5*DELZ)) GO TO 350
              GO TO 360
  350       CONTINUE
            GO TO 430
```

```
  360     IND = IND + 1
          ICNTST(M,2) = IFACE(N,IFC)
          ICNTST(IND,1) = M
          NFACE(IFC,M) = IFN
          IF (ID .LT. 1) GO TO 420
          GO TO (370, 380, 390, 400), M
  370     NFACE(3,2) = IFN
          NFACE(6,3) = IFN
          GO TO 410
  380     NFACE(3,3) = IFN
          NFACE(4,2) = IFN
          GO TO 410
  390     NFACE(4,3) = IFN
          NFACE(5,2) = IFN
          GO TO 410
  400     NFACE(5,3) = IFN
          NFACE(6,2) = IFN
  410     CONTINUE
          IF (IFC .GT. 2) GO TO 440
  420     IC = IC + 1
          ICRNR(IC1,M) = IFN
          GO TO 440
  430   CONTINUE
  440 CONTINUE
C  Redistribute the nodes along each boundary as a basis for a
C  well conditioned grid
      IF (ID .LT. 1) GO TO 780
      IF (IND .EQ. 4) IFE(IFC) = 0
      GO TO (450, 450, 690, 1150), IND
      GO TO 680
  450 DO 670 M = 1, 4
        IF (ICNTST(M,2) .EQ. 0) GO TO 460
        GO TO 670
  460   IU1 = ICNTST(1,1)
        GO TO (470, 520, 570, 620), M
  470   GO TO (670, 480, 490, 500), IU1
  480   IU = ICNTST(1,1) + NXSTPS - 1
        GO TO 510
  490   IU = ICNTST(1,1) - (NXSTP1 + 1) * (NYSTP1 - 1) - (NXSTPS - 1)
        GO TO 510
  500   IU = ICNTST(1,1) - (NXSTP1 + 1) * (NYSTPS - 1)
  510   NFACE(IFC,M) = IN(IU,2)
        ICRNR(IC1,M) = IN(IU,2)
        NFACE(3,2) = IN(IU,2)
        NFACE(6,3) = IN(IU,2)
        IC = IC + 1
        GO TO 670
  520   GO TO (530, 670, 540, 550), IU1
  530   IU = ICNTST(1,1) + NYSTPS + 1
        GO TO 560
```

```
540   IU = ICNTST(1,1) - (NXSTP1 + 1) * (NYSTPS - 1)
      GO TO 560
550   IU = ICNTST(1,1) - ((NXSTP1 + 1)*(NYSTP1 - 1) - 1)
560   NFACE(IFC,M) = IN(IU,3)
      ICRNR(IC1,M) = IN(IU,3)
      NFACE(3,3) = IN(IU,3)
      NFACE(4,2) = IN(IU,3)
      IC = IC + 1
      GO TO 670
570   GO TO (580, 590, 670, 600), IU1
580   IU = ICNTST(1,1) - ((NXSTP1 + 1)*(NYSTPS - 2) + NXSTPS - 1)
      GO TO 610
590   IU = ICNTST(1,1) - (NXSTP1 + 1) * (NYSTP1 + 1)
      GO TO 610
600   IU = ICNTST(1,1) - (NXSTPS - 1)
610   NFACE(IFC,M) = IN(IU,4)
      ICRNR(IC1,M) = IN(IU,4)
      NFACE(4,3) = IN(IU,4)
      NFACE(5,2) = IN(IU,4)
      GO TO 670
620   GO TO (630, 640, 650, 670), IU1
630   IU = ICNTST(1,1) - (NXSTP1 + 1) * (NYSTPS - 1)
      GO TO 660
640   IU = ICNTST(1,1) - ((NXSTP1 + 1 - NXSTPS) + (NXSTP1 + 1)*(
     1  NYSTPS - 1))
      GO TO 660
650   IU = ICNTST(1,1) + NXSTPS
660   NFACE(IFC,M) = IN(IU,1)
      ICRNR(IC1,M) = IN(IU,1)
      NFACE(5,3) = IN(IU,1)
      NFACE(6,2) = IN(IU,1)
      IC = IC + 1
670   CONTINUE
680   CONTINUE
      GO TO 770
690   DO 700 M = 1, 4
        IF (ICNTST(M,2) .EQ. 0) GO TO 710
700   CONTINUE
      GO TO 770
710   GO TO (720, 730, 740, 750), M
720   IU = ICNTST(2,2) - NXSTPS - 1
      NFACE(IFC,M) = IN(IU,2)
      ICRNR(IC1,M) = IN(IU,2)
      NFACE(3,2) = IN(IU,2)
      NFACE(6,3) = IN(IU,2)
      IC = IC + 1
      GO TO 760
730   IU = ICNTST(1,2) + NXSTPS - 1
      NFACE(IFC,M) = IN(IU,3)
      ICRNR(IC1,M) = IN(IU,3)
```

```
      NFACE(3,3) = IN(IU,3)
      NFACE(4,2) = IN(IU,3)
      IC = IC + 1
      GO TO 760
  740 IU = ICNTST(4,2) + NXSTPS - 1
      NFACE(IFC,M) = IN(IU,4)
      ICRNR(IC1,M) = IN(IU,4)
      NFACE(4,3) = IN(IU,4)
      NFACE(5,2) = IN(IU,4)
      IC = IC + 1
      GO TO 760
  750 IU = ICNTST(3,2) - NXSTPS + 1
      NFACE(IFC,M) = IN(IU,1)
      ICRNR(IC1,M) = IN(IU,1)
      NFACE(5,3) = IN(IU,1)
      NFACE(6,2) = IN(IU,1)
      IC = IC + 1
  760 CONTINUE
      IFE(IFC) = 0
      GO TO 1150
  770 CONTINUE
  780 DZ = 0.0
      DO 1140 L = 1, 4
  790   L1 = L
        L2 = L + 1
        IF (IFC .GT. 2 .AND. L .GT. 1) GO TO 1150
        IF (L .EQ. 4) GO TO 800
        GO TO 810
  800   L1 = 4
        L2 = 1
  810   IF (IFC .GT. 2) GO TO 820
        IF (ID .LT. 1 .AND. NDX(NFACE(IFC,L1)) .NE. 0) IC = IC + 1
        IF (L1 .LT. 3 .AND. NDX(NFACE(IFC,L1)) .NE. 0) GO TO 1140
        IF (ID .LT. 1 .AND. NDY(NFACE(IFC,L1)) .NE. 0) IC = IC + 1
        IF (NDY(NFACE(IFC,L1)) .NE. 0) GO TO 1140
  820   IF (ID .LT. 1 .AND. NDZ(NFACE(IFC,L2)) .NE. 0) IC = IC + 1
        IF (NDZ(NFACE(IFC,L2)) .NE. 0) GO TO 1140
        DX = XN(NFACE(IFC,L2)) - XN(NFACE(IFC,L1))
        DY = YN(NFACE(IFC,L2)) - YN(NFACE(IFC,L1))
        IF (ID .LT. 1) GO TO 840
        DZ = ZN(NFACE(IFC,L2)) - ZN(NFACE(IFC,L1))
        IF (IFC .GT. 2) DZ = -DZ
        IF (IFC .GT. 2) GO TO 870
  830   IF (ABS(DX) .GT. ABS(DY) .AND. ABS(DX) .GT. ABS(DZ))
     1     GO TO 850
        IF (ABS(DY) .GT. ABS(DX) .AND. ABS(DY) .GT. ABS(DZ))
     1     GO TO 860
        IF (ABS(DZ) .GT. ABS(DX) .AND. ABS(DZ) .GT. ABS(DY))
     1     GO TO 870
  840   IF (ABS(DX) .GT. ABS(DY)) GO TO 850
```

```
      GO TO 860
850   XPT = XN(NFACE(IFC,L1))
      DELX = DX / NXSTPS
      NXSTP1 = NXSTPS - 1
      YPT = YN(NFACE(IFC,L1))
      IF (ID .LT. 1) GO TO 880
      DELZ = DZ / NXSTPS
      ZPT = ZN(NFACE(IFC,L1))
      GO TO 880
860   YPT = YN(NFACE(IFC,L1))
      DELY = DY / NYSTPS
      NYSTP1 = NYSTPS - 1
      XPT = XN(NFACE(IFC,L1))
      IF (ID .LT. 1) GO TO 990
      ZPT = ZN(NFACE(IFC,L1))
      DELZ = DZ / NYSTPS
      GO TO 990
870   ZPT = ZN(NFACE(IFC,L2))
      P1 = NFACE(IFC,L1)
      P2 = NFACE(IFC,L2)
      DELZ = DZ / NZSTPS
      NZSTP1 = NZSTPS - 1
      XPT = XN(NFACE(IFC,L2))
      YPT = YN(NFACE(IFC,L2))
      LC = LC + 1
      NDZ(ICRNR(IC1,LC)) = I
      GO TO 1100
880   XNEW(I) = XPT
      ZNEW(I) = ZPT
      YNEW(I) = YPT
      IF (ID .LT. 1 .AND. DELX .LT. 0.0 .AND. IC .GT. 2)
     1   GO TO 890
      IC = IC + 1
      ICRNR(IC1,IC) = I
      NBN = NBN + 1
      NBDND(NBN) = I
      I = I + 1
890   DO 9[illegible] I[illegible] = 1, NXSTP1
        [illegible] = XP[illegible] DELX
900     [illegible]1 = II [illegible]
        II2 = II [illegible] 2
        D1 = XN([illegible])
        [illegible] = [illegible]N(II1)
        D3 = XN(II2)
        D4 = YN(II)
        D5 = YN(II1)
        D6 = YN(II2)
        IF (XN(II1) .LT. (XN(II2) - 0.01*SCL) .OR. XN(II1) .GT. (XN(
     1     II2) + 0.01*SCL)) GO TO 910
        IF (YN(II1) .LT. (YN(II2) - 0.01*SCL) .OR. YN(II1) .GT. (YN(
```

```
     1        II2) + 0.01*SCL)) GO TO 910
           II1 = II1 + 1
           II2 = II2 + 1
           D2 = XN(II1)
           D3 = XN(II2)
           D5 = YN(II1)
           D6 = YN(II2)
 910       IF (XN(II) .LT. (XN(II1) - 0.01*SCL) .OR. XN(II) .GT. (XN(
     1        II1) + 0.01*SCL)) GO TO 920
           IF (YN(II) .LT. (YN(II1) - 0.01*SCL) .OR. YN(II) .GT. (YN(
     1        II1) + 0.01*SCL)) GO TO 920
           GO TO 930
 920       IF (DELX .LT. 0.0) GO TO 940
           IF (XPT .GT. D2) GO TO 930
           GO TO 950
 930       II = II + 1
           II1 = II + 1
           II2 = II + 2
           GO TO 900
 940       IF (XPT .LT. D2) GO TO 930
           IF (INX .EQ. 1 .AND. XPT .LT. D3) GO TO 930
 950       CONTINUE
C Lagrange interpolation routine
           A = (XPT - D2) * (XPT - D3)
           B = (D1 - D2) * (D1 - D3)
           P1 = (A/B) * D4
           C = (XPT - D1) * (XPT - D3)
           D = (D2 - D1) * (D2 - D3)
           P2 = (C/D) * D5
           E = (XPT - D1) * (XPT - D2)
           F = (D3 - D1) * (D3 - D2)
           P3 = (E/F) * D6
           YNEW(I) = P1 + P2 + P3
           XNEW(I) = XPT
           CX(I) = CX(ITRNS)
           IF (ID .LT. 1) GO TO 960
           ZNEW(I) = ZNEW(I - 1) + DELZ
 960       NBN = NBN + 1
           NBDND(NBN) = I
           IF (DELX .LT. 0.00) GO TO 970
           NDX(I - 1) = I
           IF (INX .EQ. 1) NDX(ICRNR(IC1,IC)) = I
           IF (IFC .LT. 3 .AND. L .EQ. 4) NDY(I) = I - 1
           IF (IFC .LT. 3 .AND. L .EQ. 4 .AND. INX .EQ. 1)
     1     NDY(I) = ICRNR(IC1,IC)
           IF (IFC .GT. 2 .AND. L .EQ. 2) NDZ(I - 1) = I
           IF (IFC .GT. 2 .AND. L .EQ. 2 .AND. INX .EQ. 1)
     1     NDZ(ICRNR(IC1,IC)) = I
           I = I + 1
           GO TO 980
```

```
 970   NDX(I) = I - 1
       IF (INX .EQ. 1) NDX(ICRNR(IC1,IC)) = I - 1
       IF (IFC .LT. 3 .AND. L .EQ. 2) NDY(I - 1) = I
       IF (IFC .LT. 3 .AND. L .EQ. 2 .AND. INX .EQ. 1)
     1 NDY(ICRNR(IC1,IC)) = I
       IF (IFC .GT. 2 .AND. L .EQ. 4) NDZ(I) = I - 1
       IF (IFC .GT. 2 .AND. L .EQ. 4 .AND. INX .EQ. 1)
     1 NDZ(I) = ICRNR(IC1,IC)
       I = I + 1
 980  CONTINUE
      GO TO 1140
 990  YNEW(I) = YPT
      XNEW(I) = XPT
      ZNEW(I) = ZPT
      IC = IC + 1
      ICRNR(IC1,IC) = I
      NBN = NBN + 1
      NBDND(NBN) = I
      I = I + 1
      DO 1090 INY = 1, NYSTP1
        YPT = YPT + DELY
1000    II1 = II + 1
        II2 = II + 2
        D1 = YN(II)
        D2 = YN(II1)
        D3 = YN(II2)
        D4 = XN(II)
        D5 = XN(II1)
        D6 = XN(II2)
        IF (XN(II1) .LT. (XN(II2) - 0.01*SCL) .OR. XN(II1) .GT. (XN(
     1     II2) + 0.01*SCL)) GO TO 1010
        IF (YN(II1) .LT. (YN(II2) - 0.01*SCL) .OR. YN(II1) .GT. (YN(
     1     II2) + 0.01*SCL)) GO TO 1010
        II1 = II1 + 1
        II2 = II2 + 1
        D2 = YN(II1)
        D3 = YN(II2)
        D5 = XN(II1)
        D6 = XN(II2)
1010    IF (XN(II) .LT. (XN(II1) - 0.01*SCL) .OR. XN(II) .GT. (XN(
     1     II1) + 0.01*SCL)) GO TO 1020
        IF (YN(II) .LT. (YN(II1) - 0.01*SCL) .OR. YN(II) .GT. (YN(
     1     II1) + 0.01*SCL)) GO TO 1020
        GO TO 1030
1020    IF (DELY .LT. 0.0) GO TO 1040
        IF (YPT .GT. YN(II + 1)) GO TO 1030
        GO TO 1050
1030    II = II + 1
        GO TO 1000
1040    IF (YPT .LT. YN(II + 1)) GO TO 1030
```

```
 1050     CONTINUE
          IF (II .LT. NPTS - 2) GO TO 1060
          D1 = YN(NPTS - 2)
          D2 = YN(NPTS - 1)
          D3 = YN(NPTS)
          D4 = XN(NPTS - 2)
          D5 = XN(NPTS - 1)
          D6 = XN(NPTS)
C Lagrange interpolation routine
 1060     A = (YPT - D2) * (YPT - D3)
          B = (D1 - D2) * (D1 - D3)
          P1 = (A/B) * D4
          C = (YPT - D1) * (YPT - D3)
          D = (D2 - D1) * (D2 - D3)
          P2 = (C/D) * D5
          E = (YPT - D1) * (YPT - D2)
          F = (D3 - D1) * (D3 - D2)
          P3 = (E/F) * D6
          XNEW(I) = P1 + P2 + P3
          YNEW(I) = YPT
          IF (ID .LT. 1) GO TO 1070
          ZNEW(I) = ZNEW(I - 1) + DELZ
 1070     CX(I) = CX(ITRNS)
          NBN = NBN + 1
          NBDND(NBN) = I
          IF (DELY .LT. 0.00) GO TO 1080
          NDY(I - 1) = I
          IF (INY .EQ. 1) NDY(ICRNR(IC1,IC)) = I
          IF (IFC .LT. 3 .AND. L .EQ. 3) NDX(I) = I - 1
          IF (IFC .LT. 3 .AND. L .EQ. 3 .AND. INY .EQ. 1)
     1    NDY(I) = ICRNR(IC1,IC)
          IF (IFC .GT. 2 .AND. L .EQ. 1) NDZ(I - 1) = I
          IF (IFC .GT. 2 .AND. L .EQ. 1 .AND. INY .EQ. 1)
     1    NDZ(ICRNR(IC1,IC)) = I
          I = I + 1
          GO TO 1090
 1080     NDY(I) = I - 1
          IF (INY .EQ. 1) NDY(I) = ICRNR(IC1,IC)
          IF (IFC .LT. 3 .AND. L .EQ. 1) NDX(I - 1) = I
          IF (IFC .LT. 3 .AND. L .EQ. 1 .AND. INY .EQ. 1)
     1    NDX(ICRNR(IC1,IC)) = I
          IF (IFC .GT. 2 .AND. L .EQ. 3) NDZ(I) = I - 1
          IF (IFC .GT. 2 .AND. L .EQ. 3 .AND. INY .EQ. 1)
     1    NDZ(I) = ICRNR(IC1,IC)
          I = I + 1
 1090   CONTINUE
        GO TO 1140
 1100   IF (NZSTPS .LT. 2) GO TO 1130
        DO 1120 INZ = 1, NZSTP1
          XNEW(I) = XPT + ((XN(P1) - XN(P2))/NZSTPS) * INZ
```

```
            YNEW(I) = YPT + ((YN(P1) - YN(P2))/NZSTPS) * INZ
            ZNEW(I) = ZPT + DELZ * (INZ)
            CX(I) = CX(ITRNS)
            NBN = NBN + 1
            NBDND(NBN) = I
            IF (DELZ .LT. 0.00) GO TO 1110
            IF (INZ .GT. 1) NDZ(I - 1) = I
            I = I + 1
            GO TO 1120
 1110       NDZ(I) = I - 1
            IF (INZ .EQ. 1) NDZ(I) = ICRNR(IC1,LC)
            I = I + 1
 1120     CONTINUE
          NDZ(I - 1) = ICRNR(IC1,LC + 4)
          GO TO 1140
 1130     NDZ(ICRNR(IC1,LC)) = ICRNR(IC1,LC + 4)
 1140   CONTINUE
 1150 CONTINUE
 1160 FORMAT (I5, 3F13.3)
      I = I - 1
      DO 1170 I1 = NPTS1, I
        XN(I1) = XNEW(I1)
        YN(I1) = YNEW(I1)
        IF (ID .LT. 1) GO TO 1170
        ZN(I1) = ZNEW(I1)
 1170 CONTINUE
      NPTS = I
      NDX(ICRNR(IC1,4) - (NXSTPS - 1)) = ICRNR(IC1,3)
      NDX(ICRNR(IC1,4)) = ICRNR(IC1,4) - 1
      IF (IC1 .LT. 2) NDY(ICRNR(IC1,1)) = ICRNR(IC1,4) + NYSTPS - 1
      IF (ID .LT. 1) NDY(ICRNR(IC1,1)) = ICRNR(IC1,4) + NYSTPS - 1
      NDX(ICRNR(IC1,1) + NXSTPS - 1) = ICRNR(IC1,2)
      NDY(ICRNR(IC1,2) + NYSTPS - 1) = ICRNR(IC1,3)
      IF (ID .LT. 1) GO TO 1180
      NDX(ICRNR(IC1,6) - 1) = ICRNR(IC1,6)
      NDX(ICRNR(IC1,8)) = ICRNR(IC1,8) - 1
      NDY(ICRNR(IC1,5)) = ICRNR(IC1,8) + NYSTP1
      NDY(ICRNR(IC1,7) - 1) = ICRNR(IC1,7)
 1180 CONTINUE
C ***********************************************************
      CALL NODE(IC1)
C ***********************************************************
      RETURN
      END
C
C
C ..........................................................
C
C   Subroutine NODE
C
```

```
C     Author
C        G.J. Smith(U. of Alberta)
C
C     Last Update
C
C        May 1984
C
C     Purpose
C        To generate the nodes within the digitized
C        figure to form the mesh.
C
C     Usage
C        CALL NODE(IC1)
C
C     Subroutines Required
C        ELMNT
C
C
C ...............................................................
      SUBROUTINE NODE(IC1)
      COMMON MEMJT(5000,32), IN(3000,8), IFACE(3500,6), YN(5000),
     1       ZN(5000)
      COMMON CX(5000)
      COMMON XN(5000), JMEM(5000), NBDND(4000), NDX(5000), NDY(5000)
      COMMON NDZ(5000), ITYPE(5000), NDWEL(20,2,25), LEAK(100),
     1       ALEAK(100)
      COMMON XIN(100), YIN(100), ISAVE(25,2), PPOSX(10), PPOSY(10),
     1       PPOSZ(10)
      COMMON OBSX(10), OBSY(10), OBSZ(10), ND(16), NND(16)
      COMMON COX(10), COY(10), COZ(10), IFE(6), NPMP(25), IPPN(10)
      COMMON ICRNR(10,8), NFACE(6,4), ICNTST(4,2), NODE1(8)
      COMMON IP, NPTS, NPTS1, NELM, ID, ITRNS, INC, IELEM, NZSTPS, ID,
     1       YC, XC, IPN
      COMMON DELX, DELY, SCL, IW1, IW2, IK, NBN, NPP, NXSTPS, NYSTPS,
     1       DELTZ
      NDD = ICRNR(IC1,1)
      I = 1
      IFLG = 0
      J = NELM + 1
      NXSTP1 = NXSTPS - 1
      NYSTP1 = NYSTPS - 1
      IF (ID .GT. 0) NZSTP1 = NZSTPS - 1
      IL = 0
      NDDO = ICRNR(IC1,2)
   10 IL = IL + 1
      ND1 = NDD
      NDD2 = NDY(NDDO)
      IF (ID .GT. 0 .AND. IL .LT. NZSTPS) NDD = NDZ(ND1)
      ICOL = 1
      IROW = 1
```

```
      DO 350 IROW = 1, NYSTP1
        NODE1(6) = ND1
        NODE1(5) = NDY(ND1)
        NODE1(7) = NDX(ND1)
        IF (ID .GT. O) NODE1(2) = NDZ(ND1)
        NDD1 = NDY(ND1)
        ND1 = NODE1(5)
        DELTX = XN(NDD2) - XN(NDD1)
        DELTX = DELTX / NXSTPS
        IF (AMOD(FLOAT(NXSTPS),2.) .GT. 0.00) GO TC 30
        IF (IFLG .EQ. 1) GO TO 20
        IF (IFLG .EQ. O) IFLG = 1
        GO TO 30
   20   IFLG = O
   30   CONTINUE
        DO 340 ICOL = 1, NXSTP1
          IF (ICOL .EQ. 1 .AND. IROW .EQ. 1 .AND. ID .GT. O)
     1    NDD = NODE1(2)
          IF (ID .GT. O) DELTZ = ZN(ICRNR(IC1,5) + ICOL) - ZN(ICRNR(IC1,
     1    1) + ICOL)
          IF (ID .GT. O) DELTZ = DELTZ / NZSTPS
          IF (ICOL .EQ. 1) NDD1 = NDY(ND1)
          IF (ICOL .EQ. 1) NDD3 = NODE1(5)
          DELTY = (YN(ICRNR(IC1,4) - ICOL) - YN(ICOL + ICRNR(IC1,1)))
          DELTY = DELTY / NYSTPS
          IF (ID .LT. 1) GO TO 40
          ZL1 = (ZN(ICRNR(IC1,1)) - ZN(ICRNR(IC1,2))) * (NXSTPS - ICOL)
     1    / NXSTPS + ZN(ICRNR(IC1,2))
          ZL2 = (ZN(ICRNR(IC1,4)) - ZN(ICRNR(IC1,3))) * (NXSTPS - ICOL)
     1    / NXSTPS + ZN(ICRNR(IC1,3))
          ZL = (ZL1 - ZL2) * (NYSTPS - IROW) / NYSTPS + ZL2
   40     IF ((ID .GT. O .AND. IROW .GT. 1) .OR. (ID .GT. O .AND. IL .
     1    E . NZSTPS)) NODE1(3) = NDX(NODE1(2))
          NODE1(7) = NDX(NODE1(6))
          IF (NDY(NODE1(7)) .NE. O) GO TO 50
          IF (IL .GT. 1) GO TO 50
          GO TO 60
          NODE1(8) = NDY(NODE1(7))
          GO TO 100
   60     CONTINUE
          XN(NPTS + I) = XN(NODE1(5)) + DELTX
          YN(NPTS + I) = YN(NODE1(7)) + DELTY
          NODE1(8) = NPTS + I
   70     FORMAT (I5, 2F10.3)
          IF (ID .GT. O) ZN(NODE1(8)) = ZL
          IF (IL .EQ. 1 .AND. ID .GT. O) GO TO 80
          GO TO 90
   80     NBN = NBN + 1
          NBDND(NBN) = NPTS + I
          IF (IC1 .GT. 1) GO TO 90
```

```
      IFE(2) = IFE(2) + 1
      IFACE(IFE(2),2) = J
   90 CONTINUE
      IF (ID .LT. 1) CX(NPTS + I) = COX(ITRNS)
      NDX(NODE1(5)) = NPTS + I
      NDY(NODE1(7)) = NPTS + I
      I = I + 1
  100 IF (ID .LT. 1) GO TO 190
      IF (IL .EQ. NZSTPS) GO TO 120
      IF (NDZ(NODE1(5)) .NE. 0) GO TO 130
      IF (ICOL .GT. 1) GO TO 130
      XN(NPTS + I) = XN(NODE1(5))
      YN(NPTS + I) = YN(NODE1(5))
      ZN(NPTS + I) = ZN(NODE1(5)) + DELTZ
      IF (ID .LT. 1) CX(NPTS + I) = COX(ITRNS)
      NDY(NODE1(2)) = NPTS + I
      NDZ(NODE1(5)) = NPTS + I
      NODE1(1) = NPTS + I
      NBN = NBN + 1
      NBDND(NBN) = NPTS + I
      IFE(6) = IFE(6) + 1
      IFACE(IFE(6),6) = J
  110 CONTINUE
      I = I + 1
      GO TO 130
  120 NODE1(1) = NDY(NODE1(2))
      NDZ(NODE1(5)) = NODE1(1)
      NDY(NODE1(2)) = NODE1(1)
      IFE(6) = IFE(6) + 1
      IFACE(IFE(6),6) = J
  130 IF (IROW .GT. 1 .OR. IL .EQ. NZSTPS) GO TO 140
      XN(NPTS + I) = XN(NDX(NODE1(6)))
      YN(NPTS + I) = YN(NDX(NODE1(6)))
      ZN(NPTS + I) = ZN(NDX(NODE1(6))) + DELTZ
      IF (ID .LT. 1) CX(NPTS + I) = COX(ITRNS)
      NDZ(NODE1(7)) = NPTS + I
      NDX(NODE1(2)) = NPTS + I
      IF (ID .LT. 1) CX(NPTS + I) = COX(ITRNS)
      NODE1(3) = NPTS + I
      NBN = NBN + 1
      NBDND(NBN) = NPTS + I
      I = I + 1
  140 NODE1(3) = NDX(NODE1(2))
      IF (IROW .GT. 1) GO TO 150
      IFE(3) = IFE(3) + 1
      IFACE(IFE(3),3) = J
  150 IF (NDY(NODE1(3)) .EQ. 0) GO TO 160
      NODE1(4) = NDY(NODE1(3))
  160 XN(NPTS + I) = XN(NODE1(8))
      YN(NPTS + I) = YN(NODE1(8))
```

```
      ZN(NPTS + I) = ZN(NODE1(8)) + DELTZ
      NDZ(NODE1(8)) = NPTS + I
      NDY(NODE1(3)) = NPTS + I
      NDX(NODE1(1)) = NPTS + I
      NODE1(4) = NPTS + I
      IF (ID .LT. 1) CX(NPTS + I) = COX(ITRNS)
      IF (IL .EQ. NZSTPS .AND. ID .GT. 0) GO TO 170
      GO TO 180
170   NBN = NBN + 1
      NBDND(NBN) = NPTS + I
      IFE(1) = IFE(1) + 1
      IFACE(IFE(1),1) = J
180   CONTINUE
      I = I + 1
190   CONTINUE
      IF (IO .LT. 1) GO TO 250
C Look for observation wells
      DO 240 IOW = 1, IO
        IF (XN(NODE1(8)) .LE. (OBSX(IOW) - DELTX*0.5) .OR. XN(NODE1(
     1     8)) .GT. (OBSX(IOW) + DELTX*0.5)) GO TO 240
        IF (YN(NODE1(8)) .LE. (OBSY(IOW) - DELTY*0.5) .OR. YN(NODE1(
     1     8)) .GT. (OBSY(IOW) + DELTY*0.5)) GO TO 240
        IF (ID .LT. 1) GO TO 230
        DO 200 IWN = 1, IW1
          IF (NODE1(4) .EQ. NDWEL(IOW,1,IWN)) GO TO 240
200     CONTINUE
        IW1 = IW1 + 1
        XN(NODE1(4)) = OBSX(IOW)
        YN(NODE1(4)) = OBSY(IOW)
        NDWEL(IOW,1,IW1) = NODE1(4)
210     DO 220 IWN = 1, IW1
          IF (NODE1(8) .EQ. NDWEL(IOW,1,IWN)) GO TO 240
220     CONTINUE
230     IW1 = IW1 + 1
        XN(NODE1(8)) = OBSX(IOW)
        YN(NODE1(8)) = OBSY(IOW)
        NDWEL(IOW,1,IW1) = NODE1(8)
240   CONTINUE
250   CONTINUE
      IF (IP .LT. 1) GO TO 310
C Look for pumping wells
      DO 300 IPW = 1, IP
        IF (XN(NODE1(8)) .LE. (PPOSX(IPW) - DELTX*0.5) .OR. XN(
     1     NODE1(8)) .GT. (PPOSX(IPW) + DELTX*0.5)) GO TO 300
        IF (YN(NODE1(8)) .LE. (PPOSY(IPW) - DELTY*0.5) .OR. YN(
     1     NODE1(8)) .GT. (PPOSY(IPW) + DELTY*0.5)) GO TO 300
        IF (ID .LT. 1) GO TO 270
        IF (IW2 .LT. 1) GO TO 270
        DO 260 IWN = 1, IW2
          IF (NODE1(8) .EQ. NDWEL(IPW,2,IWN)) GO TO 280
```

```
  260     CONTINUE
  270     IW2 = IW2 + 1
          XN(NODE1(8)) = PPOSX(IPW)
          YN(NODE1(8)) = PPOSY(IPW)
          NDWEL(IPW,2,IW2) = NODE1(8)
          IPPN(IPW) = IW2
          ITYPE(NODE1(8)) = 2
  280     IF (ID .LT. 1) GO TO 300
          DO 290 IWN = 1, IW2
            IF (NODE1(4) .EQ. NDWEL(IPW,2,IWN)) GO TO 300
  290     CONTINUE
          XN(NODE1(4)) = PPOSX(IPW)
          YN(NODE1(4)) = PPOSY(IPW)
          NDWEL(IPW,2,IW2) = NODE1(4)
          ITYPE(NODE1(4)) = 2
          IPPN(IPW) = IW2
          IW2 = IW2 + 1
  300   CONTINUE
  310   CONTINUE
  320   FORMAT (9I5)
  330   FORMAT (4I5)
        NPTSI = NPTS + I - 1
C ***********************************************************
        CALL ELMNT(I, J, NPTSI, IROW, ICOL, ND1, NDD2, NDD3, IL, IC1,
     1        IFLG)
C ***********************************************************
  340   CONTINUE
        ICOL = ICOL + 1
C ***********************************************************
        CALL ELMNT(I, J, NPTSI, IROW, ICOL, ND1, NDD2, NDD3, IL, IC1,
     1        IFLG)
C ***********************************************************
  350 CONTINUE
      NODE1(5) = ICRNR(IC1,4)
      NODE1(6) = NODE1(5) + 1
      DO 460 IXL = 1, NXSTP1
        IF (ID .GT. 0 .OR. IELEM .EQ. 1) GO TO 380
        IF (MOD(NXSTPS,2) .GT. 0) GO TO 360
        IF (IFLG .EQ. 0) IFLG = 1
        IF (IFLG .EQ. 1) IFLG = 0
  360   IF (IFLG .EQ. 1) GO TO 370
        IN(J,1) = NODE1(5)
        IN(J,2) = NODE1(6)
        IN(J,3) = NDX(NODE1(5))
        J = J + 1
        IN(J,1) = NDX(NODE1(5))
        IN(J,2) = NODE1(6)
        IN(J,3) = NDX(NODE1(6))
        NODE1(5) = IN(J - 1,3)
        NODE1(6) = IN(J,3)
```

```
      J = J + 1
      IFLG = 1
      GO TO 460
  370 IN(J,1) = NODE1(5)
      IN(J,2) = NODE1(6)
      IN(J,3) = NDX(NODE1(6))
      J = J + 1
      IN(J,1) = NODE1(5)
      IN(J,2) = NDX(NODE1(6))
      IN(J,3) = NDX(NODE1(5))
      NODE1(5) = IN(J,3)
      NODE1(6) = IN(J,2)
      J = J + 1
      IFLG = 0
      GO TO 460
  380 NODE1(6) = ND1
      NODE1(5) = NDY(ND1)
      NODE1(7) = NDX(NODE1(6))
      NODE1(8) = NDX(NODE1(5))
      NDY(NODE1(7)) = NODE1(8)
      IF (ID .LT. 1) GO TO 410
      NODE1(1) = NDZ(NODE1(5))
      NODE1(2) = NDZ(NODE1(6))
      NODE1(3) = NDZ(NODE1(7))
      NDX(NODE1(2)) = NODE1(3)
      NDY(NODE1(2)) = NODE1(1)
      IF (IL .EQ. NZSTPS) GO TO 390
      GO TO 400
  390 NODE1(4) = NDX(NODE1(1))
      NDY(NODE1(3)) = NODE1(4)
      NDZ(NODE1(8)) = NODE1(4)
      IFE(1) = IFE(1) + 1
      IFACE(IFE(1),1) = J
      GO TO 410
  400 CONTINUE
      XN(NPTS + I) = XN(NODE1(8))
      YN(NPTS + I) = YN(NODE1(8))
      ZN(NPTS + I) = ZN(NODE1(8)) + DELTZ
      IF (ID .LT. 1) CX(NPTS + I) = COX(ITRNS)
      NODE1(4) = NPTS + I
      NBN = NBN + 1
      NBDND(NBN) = NPTS + I
      NDX(NODE1(1)) = NODE1(4)
      NDY(NODE1(3)) = NODE1(4)
      NDZ(NODE1(8)) = NODE1(4)
      I = I + 1
  410 N1 = 5
      N2 = 8
      IFE(5) = IFE(5) + 1
      IFACE(IFE(5),5) = J
```

```
        IF (IL .GT. 1 .OR. IC1 .GT. 1) GO TO 420
        IFE(2) = IFE(2) + 1
        IFACE(IFE(2),2) = J
  420   IF (IXL .GT. 1) GO TO 430
        IFE(6) = IFE(6) + 1
        IFACE(IFE(6),6) = J
  430   CONTINUE
        L = 0
        IF (ID .GT. 0) N1 = 1
        DO 440 N = N1, N2
          L = L + 1
          IN(J,L) = NODE1(N)
  440   CONTINUE
        IF (ID .LT. 1) GO TO 450
        IFE(5) = IFE(5) + 1
        IFACE(IFE(5),5) = J
  450   CONTINUE
        J = J + 1
        ND1 = NODE1(7)
  460 CONTINUE
      IF (IELEM .EQ. 1) GO TO 480
      IF (IFLG .EQ. 1) GO TO 470
      IN(J,1) = NODE1(5)
      IN(J,2) = NODE1(6)
      IN(J,3) = ICRNR(IC1,3)
      J = J + 1
      IN(J,1) = ICRNR(IC1,3)
      IN(J,2) = NODE1(6)
      IN(J,3) = NDX(NODE1(6))
      J = J + 1
      GO TO 530
  470 IN(J,1) = NODE1(5)
      IN(J,2) = NODE1(6)
      IN(J,3) = NDX(NODE1(6))
      J = J + 1
      IN(J,1) = NODE1(5)
      IN(J,2) = NDX(NODE1(6))
      IN(J,3) = ICRNR(IC1,3)
      J = J + 1
      GO TO 530
  480 NODE1(6) = ND1
      NODE1(5) = NDY(ND1)
      NODE1(7) = NDX(ND1)
      NODE1(8) = NDX(NODE1(5))
      NODE1(2) = NDZ(ND1)
      NODE1(1) = NDZ(NODE1(5))
      NODE1(3) = NDZ(NODE1(7))
      NODE1(4) = NDZ(NODE1(8))
      NDX(NODE1(1)) = NODE1(4)
      N1 = 5
```

```
      N2 = 8
      L = 0
      IF (ID .GT. 0) N1 = 1
      DO 490 N = N1, N2
        L = L + 1
        IN(J,L) = NODE1(N)
  490 CONTINUE
      IF (ID .LT. 1) GO TO 500
  500 CONTINUE
      IFE(5) = IFE(5) + 1
      IFACE(IFE(5),5) = J
      IFE(4) = IFE(4) + 1
      IFACE(IFE(4),4) = J
      IF (IL .GT. 1 .OR. IC1 .GT. 1) GO TO 510
      IFE(2) = IFE(2) + 1
      IFACE(IFE(2),2) = J
  510 IF (IL .LT. NZSTPS) GO TO 520
      IFE(1) = IFE(1) + 1
      IFACE(IFE(1),1) = J
  520 J = J + 1
      IF (ID .GT. 0) NDDO = NDZ(NDDO)
  530 IF (ID .GT. 0 .AND. IL .LT. NZSTPS) GO TO 10
      NELM = J - 1
      NPTS = NPTS + I - 1
  540 FORMAT (3I5)
      DO 580 L1 = 1, NPTS
        DO 550 L2 = 1, NPTS
          IF (L1 .EQ. NDX(L2)) GO TO 580
  550   CONTINUE
        DO 560 L3 = 1, NPTS
          IF (L1 .EQ. NDY(L3)) GO TO 580
  560   CONTINUE
        IF (ID .LT. 1) GO TO 590
        DO 570 L4 = 1, NPTS
          IF (L1 .EQ. NDZ(L4)) GO TO 580
  570   CONTINUE
        GO TO 590
  580 CONTINUE
      GO TO 600
  590 IK = L1
  600 CONTINUE
      DELX = ABS(DELX)
      DELY = ABS(DELY)
      RETURN
      END
C
C
C .............................................................
C
C   Subroutine ELMNT
```

```
C
C   Author.
C       G.J. Smith(U. of Alberta)
C
C   Last Update
C       May 1984
C
C   Purpose
C       To simultaneously form and number the elements
C       while the nodes are being generated.
C
C   Usage
C       CALL ELMNT(I,J,NPTSI,IROW,ICOL,ND1,NDD2,NDD3,IL,IC1,IFLG)
C
C   Subroutines Required
C       None
C
C...................................................
      SUBROUTINE ELMNT(I, J, NPTSI, IROW, ICOL, ND1, NDD2, NDD3, IL,
     1           IC1, IFLG)
      COMMON MEMJT(5000,32), IN(3000,8), IFACE(3500,6), YN(5000),
     1       ZN(5000)
      COMMON CX(5000)
      COMMON XN(5000), JMEM(5000), NBDND(4000), NDX(5000), NDY(5000)
      COMMON NDZ(5000), ITYPE(5000), NDWEL(20,2,25), LEAK(100),
     1       ALEAK(100)
      COMMON XIN(100), YIN(100), ISAVE(25,2), PPOSX(10), PPOSY(10),
     1       PPOSZ(10)
      COMMON OBSX(10), OBSY(10), OBSZ(10), ND(16), NND(16)
      COMMON COX(10), COY(10), COZ(10), IFE(6), NPMP(25), IPPN(10)
      COMMON ICRNR(10,8), NFACE(6,4), ICNTST(4,2), NODE1(8)
      COMMON IP, NPTS, NPTS1, NELM, IO, ITRNS, INC, IELEM, NZSTPS, ID,
     1       YC, XC, IPN
      COMMON DELX, DELY, SCL, IW1, IW2, IK, NBN, NPP, NXSTPS, NYSTPS,
     1       DELTZ
      NXSTP1 = NXSTPS - 1
      NYSTP1 = NYSTPS - 1
      IF (ID .GT. 0) NZSTP1 = NZSTPS - 1
      IF (ICOL .EQ. NXSTPS) GO TO 120
      IF (ID .GT. 0 .OR. IELEM .EQ. 1) GO TO 20
      IF (IFLG .EQ. 1) GO TO 10
C 2-D Triangular elements
      IN(J,1) = NODE1(5)
      IN(J,2) = NODE1(6)
      IN(J,3) = NODE1(8)
      J = J + 1
      IN(J,1) = NODE1(8)
      IN(J,2) = NODE1(6)
      IN(J,3) = NDX(NODE1(6))
      NDX(NODE1(5)) = IN(J,1)
```

```
      J = J + 1
      NODE1(5) = NPTSI
      NODE1(6) = IN(J -[illegible]1,3)
      IFLG = 1
      RETURN
   10 IN(J,1) = NODE1(5)
      IN(J,2) = NODE1(6)
      IN(J,3) = NDX(NODE1(6))
      J = J + 1
      IN(J,1) = NODE1(5)
      IN(J,2) = NDX(NODE1(6))
      IN(J,3) = [illegible]1(8)
      NDX(NODE[illegible]) = IN(J,3)
      J = J +[illegible]
      NODE1(5[illegible] NPT[illegible]
      NODE1(6[illegible] = IN([illegible]1,2)
      NODE1(7[illegible] = ND[illegible]DE1(6))
      IFLG = 0
      RETURN
   20 IF (IELEM .LT. 1) GO TO 100
      N1 = 5
      N2 = 8
      L = 0
      IF (ID .GT. 0) N1 = 1
      DO 30 N = N1, N2
        L = L + 1
        IN(J,L) = NODE1(N)
   30 CONTINUE
      IF (ID .LT. 1) GO TO 40
   40 CONTINUE
   50 FORMAT (9I5)
   60 FORMAT (4I5)
   70 FORMAT (I5, 3F13.3)
   80 FORMAT (2I5)
      J = J + 1
C Tetrahedral elements
      IF (ID .LT. 1) GO TO 90
      NODE1(1) = NODE1(4)
      NODE1(2) = NODE1(3)
   90 NODE1(5) = NODE1(8)
      NODE1(6) = NODE1(7)
      RETURN
  100 IF (IFLG .EQ. 1) GO TO 110
      IN(J,1) = NODE1(1)
      IN(J,2) = NODE1(2)
      IN(J,3) = NODE1(4)
      IN(J,4) = NODE1(5)
      J = J + 1
      IN(J,1) = NODE1(2)
      IN(J,2) = NODE1(7)
```

```
      IN(J,3) = NODE1(5)
      IN(J,4) = NODE1(6)
      J = J + 1
      IN(J,1) = NODE1(2)
      IN(J,2) = NODE1(3)
      IN(J,3) = NODE1(4)
      IN(J,4) = NODE1(7)
      J = J + 1
      IN(J,1) = NODE1(4)
      IN(J,2) = NODE1(5)
      IN(J,3) = NODE1(7)
      IN(J,4) = NODE1(8)
      J = J + 1
      IN(J,1) = NODE1(4)
      IN(J,2) = NODE1(2)
      IN(J,3) = NODE1(7)
      IN(J,4) = NODE1(5)
      J = J + 1
      IFLG = 1
      NODE1(1) = NODE1(4)
      NODE1(2) = NODE1(3)
      NODE1(5) = NODE1(8)
      NODE1(6) = NODE1(7)
      RETURN
  110 IN(J,1) = NODE1(1)
      IN(J,2) = NODE1(2)
      IN(J,3) = NODE1(3)
      IN(J,4) = NODE1(6)
      J = J + 1
      IN(J,1) = NODE1(3)
      IN(J,2) = NODE1(6)
      IN(J,3) = NODE1(7)
      IN(J,4) = NODE1(8)
      J = J + 1
      IN(J,1) = NODE1(3)
      IN(J,2) = NODE1(4)
      IN(J,3) = NODE1(1)
      IN(J,4) = NODE1(8)
      J = J + 1
      IN(J,1) = NODE1(1)
      IN(J,2) = NODE1(6)
      IN(J,3) = NODE1(8)
      IN(J,4) = NODE1(5)
      J = J + 1
      IN(J,1) = NODE1(3)
      IN(J,2) = NODE1(1)
      IN(J,3) = NODE1(6)
      IN(J,4) = NODE1(8)
      J = J + 1
      IFLG = 0
```

```
      NODE1(1) = NODE1(4)
      NODE1(2) = NODE1(3)
      NODE1(5) = NODE1(8)
      NODE1(6) = NODE1(7)
      RETURN
  120 CONTINUE
      IF (IELEM .GT. 0 .OR. ID .GT. 0) GO TO 140
      IF (IFLG .EQ. 1) GO TO 130
      IN(J,1) = NODE1(5)
      IN(J,2) = NODE1(6)
      IN(J,3) = ICRNR(IC1,2) + IROW
      IF (ICRNR(IC1,2) .EQ. 1) IN(J,3) = ICRNR(IC1,3) + NYSTP1 - (IROW -
     1 1)
      J = J + 1
      IN(J,1) = ICRNR(IC1,2) + IROW
      IF (ICRNR(IC1,2) .EQ. 1) IN(J,1) = ICRNR(IC1,3) + NYSTP1 - (IROW -
     1 1)
      IN(J,2) = NODE1(6)
      IN(J,3) = ICRNR(IC1,2) + IROW - 1
      IF (ICRNR(IC1,2) .EQ. 1) IN(J,3) = ICRNR(IC1,3) + NYSTP1 - (IROW -
     1 2)
      IF (IROW .EQ. 1) IN(J,3) = ICRNR(IC1,2)
      NDX(IN(J - 1,1)) = IN(J,1)
      NODE1(8) = IN(J,1)
      J = J + 1
      IFLG = 1
      GO TO 230
  130 IN(J,1) = NODE1(5)
      IN(J,2) = NODE1(6)
      IN(J,3) = ICRNR(IC1,2) + IROW - 1
      IF (ICRNR(IC1,2) .EQ. 1) IN(J,3) = ICRNR(IC1,3) + NYSTP1 - (IROW -
     1 2)
      IF (IROW .EQ. 1) IN(J,3) = ICRNR(IC1,2)
      J = J + 1
      IN(J,1) = NODE1(5)
      IN(J,2) = ICRNR(IC1,2) + IROW - 1
      IF (ICRNR(IC1,2) .EQ. 1) IN(J,2) = ICRNR(IC1,3) + NYSTP1 - (IROW -
     1 2)
      IF (IROW .EQ. 1) IN(J,2) = ICRNR(IC1,2)
      IN(J,3) = ICRNR(IC1,2) + IROW
      IF (ICRNR(IC1,2) .EQ. 1) IN(J,3) = ICRNR(IC1,3) + NYSTP1 - (IROW -
     1 1)
      NDX(IN(J,1)) = IN(J,3)
      NODE1(8) = IN(J,3)
      J = J + 1
      IFLG = 0
      GO TO 230
C Quadrilateral and parallelpiped elements
  140 NODE1(7) = NDX(NODE1(6))
      NODE1(8) = NDY(NODE1(7))
```

```
      NDX(NODE1(5)) = NODE1(8)
      NDY(NODE1(7)) = NODE1(8)
      IF (ID .LT. 1) GO TO 200
      NODE1(3) = NDZ(NODE1(7))
      NDX(NODE1(2)) = NODE1(3)
      IFE(4) = IFE(4) + 1
      IFACE(IFE(4),4) = J
      IF (IL .EQ. NZSTPS .AND. ID .GT. 0) GO TO 150
      GO TO 170
  150 NODE1(4) = NDY(NODE1(3))
      NDX(NODE1(1)) = NODE1(4)
      NDZ(NODE1(8)) = NODE1(4)
      IFE(1) = IFE(1) + 1
      IFACE(IFE(1),1) = J
      IF (IROW .GT. 1) GO TO 160
      IFE(3) = IFE(3) + 1
      IFACE(IFE(3),3) = J
  160 IF (IL .GT. 1 .OR. IC1 .GT. 1) GO TO 200
      IFE(2) = IFE(2) + 1
      IFACE(IFE(2),2) = J
      GO TO 200
  170 CONTINUE
      XN(NPTS + I) = XN(NODE1(8))
      YN(NPTS + I) = YN(NODE1(8))
      ZN(NPTS + I) = ZN(NODE1(8)) + DELTZ
      NODE1(4) = NPTS + I
      IF (ID .LT. 1) CX(NPTS + I) = COX(ITRNS)
      NDY(NODE1(3)) = NPTS + I
      NDZ(NODE1(8)) = NPTS + I
      NDX(NODE1(1)) = NODE1(4)
      NBN = NBN + 1
      NBDND(NBN) = NPTS + I
      IFE(4) = IFE(4) + 1
      IFACE(IFE(4),4) = J
      IF (IL .GT. 1 .OR. IC1 .GT. 1) GO TO 180
      IFE(2) = IFE(2) + 1
      IFACE(IFE(2),2) = J
  180 IF (IROW .GT. 1) GO TO 190
      IFE(3) = IFE(3) + 1
      IFACE(IFE(3),3) = J
  190 CONTINUE
      I = I + 1
  200 IF (IELEM .LT. 1) GO TO 250
      N1 = 5
      N2 = 8
      L = 0
      IF (ID .GT. 0) N1 = 1
      DO 210 N = N1, N2
        L = L + 1
        IN(J,L) = NODE1(N)
```

```
      210 CONTINUE
          IF (ID .LT. 1) GO TO 220
      220 CONTINUE
          J = J + 1
      230 CONTINUE
      240 FORMAT (I5)
          GO TO 270
      250 IF (IFLG .EQ. 1) GO TO 260
C Tetrahedral elements
          IN(J,1) = NODE1(1)
          IN(J,2) = NODE1(2)
          IN(J,3) = NODE1(4)
          IN(J,4) = NODE1(5)
          J = J + 1
          IN(J,1) = NODE1(2)
          IN(J,2) = NODE1(7)
          IN(J,3) = NODE1(5)
          IN(J,4) = NODE1(6)
          J = J + 1
          IN(J,1) = NODE1(2)
          IN(J,2) = NODE1(3)
          IN(J,3) = NODE1(4)
          IN(J,4) = NODE1(7)
          J = J + 1
          IN(J,1) = NODE1(4)
          IN(J,2) = NODE1(5)
          IN(J,3) = NODE1(7)
          IN(J,4) = NODE1(8)
          J = J + 1
          IN(J,1) = NODE1(4)
          IN(J,2) = NODE1(2)
          IN(J,3) = NODE1(7)
          IN(J,4) = NODE1(5)
          J = J + 1
          IFLG = 1
          NODE1(1) = NODE1(4)
          NODE1(2) = NODE1(3)
          NODE1(5) = NODE1(8)
          NODE1(6) = NODE1(7)
          GO TO 270
      260 IN(J,1) = NODE1(1)
          IN(J,2) = NODE1(2)
          IN(J,3) = NODE1(3)
          IN(J,4) = NODE1(6)
          J = J + 1
          IN(J,1) = NODE1(3)
          IN(J,2) = NODE1(6)
          IN(J,3) = NODE1(7)
          IN(J,4) = NODE1(8)
          J = J + 1
```

```
      IN(J,1) = NODE1(3)
      IN(J,2) = NODE1(4)
      IN(J,3) = NODE1(1)
      IN(J,4) = NODE1(8)
      J = J + 1
      IN(J,1) = NODE1(1)
      IN(J,2) = NODE1(6)
      IN(J,3) = NODE1(8)
      IN(J,4) = NODE1(5)
      J = J + 1
      IN(J,1) = NODE1(3)
      IN(J,2) = NODE1(1)
      IN(J,3) = NODE1(6)
      IN(J,4) = NODE1(8)
      J = J + 1
      IFLG = 0
  270 NDD2 = NDY(NODE1(8))
      ND1 = NDD3
      RETURN
      END
C
C
C ..................................................
C
C   Subroutine SETUP
C
C   Author
C       R.J. Collins 1973, modified G.J. Smith 1984
C
C   Last Update
C       May 1984
C
C   Purpose
C       To set up the interconnection of the nodes
C       in such a way as to make it easy to
C       renumber to reduce bandwidth.
C
C   Usage
C       CALL SETUP(IDIFF)
C
C   Subroutines Required
C       None
C
C
C ..................................................
      SUBROUTINE SETUP(IDIFF)
      COMMON MEMJT(5000,32), IN(3000,8), IFACE(3500,6), YN(5000),
     1       ZN(5000)
      COMMON CX(5000)
      COMMON XN(5000), JMEM(5000), NBDND(4000), NDX(5000), NDY(5000)
```

```
      COMMON NDZ(5000), ITYPE(5000), NDWEL(20,2,25), LEAK(100),
     1       ALEAK(100)
      COMMON XIN(100), YIN(100), ISAVE(25,2), PPOSX(10), PPOSY(10),
     1       PPOSZ(10)
      COMMON OBSX(10), OBSY(10), OBSZ(10), ND(16), NND(16)
      COMMON COX(10), COY(10), COZ(10), IFE(6), NPMP(25), IPPN(10)
      COMMON ICRNR(10,8), NFACE(6,4), ICNTST(4,2), NODE1(8)
      COMMON IP, NODES, NPTS1, LMENTS, IO, ITRNS, INC, IELEM, NZ, ID,
     1       YC, XC, IPN
      COMMON DELX, DELY, SCL, IW1, IW2, IK, NBN, NPP, NXSTPS, NYSTPS,
     1       DELTZ
      DIMENSION JNT(5000), NDD(3)
      IDIFF = NODES
   10 FORMAT ('NODES=', I5)
      DO 30 J = 1, NODES
        JMEM(J) = 0
        DO 20 I = 1, 32
          MEMJT(J,I) = 0
   20   CONTINUE
   30 CONTINUE
C Scan through the elements to form an array
C indicating which nodes are interconnected.
      DO 80 J = 1, LMENTS
        DO 70 I = 1, INC
          JNTI = IN(J,I)
          IF (JNTI .EQ. 0) GO TO 80
          DO 60 II = 1, INC
            IF (II .EQ. I) GO TO 60
            JJT = IN(J,II)
            IF (JNTI .EQ. JJT) GO TO 60
            MEM1 = JMEM(JNTI)
            IF (MEM1 .EQ. 0) GO TO 50
            DO 40 III = 1, MEM1
              IF (MEMJT(JNTI,III) .EQ. JJT) GO TO 60
   40       CONTINUE
   50       JMEM(JNTI) = JMEM(JNTI) + 1
            MEMJT(JNTI,JMEM(JNTI)) = JJT
            IF (IABS(JNTI - JJT) .GT. IDIFF) IDIFF = IABS(JNTI - JJT)
   60     CONTINUE
   70   CONTINUE
   80 CONTINUE
      DO 100 J = 1, NODES
        MEM1 = JMEM(J)
   90   FORMAT (9I5)
  100 CONTINUE
      IDIFF = IDIFF + 1
      RETURN
      END
C
C
```

```
C ...............................................................
C
C  Subroutine OPTNUM
C
C  Author
C      R.J. Collins 1973, modified G.J. Smith 1984
C
C  Last Update
C      May 1984
C
C  Purpose
C      To renumber the nodes of the mesh in such
C      a way as to reduce bandwidth and hence
C      increase computational efficiency.
C
C  Usage
C      CALL OPTNUM(IDIFF,MAX)
C
C  Subroutines Required
C      None
C
C
C ...............................................................
      SUBROUTINE OPTNUM(IDIFF, MAX)
      COMMON MEMJT(5000,32), IN(3000,8), IFACE(3500,6), YN(5000),
     1       ZN(5000)
      COMMON CX(5000)
      COMMON XN(5000), JMEM(5000), NBDND(4000), NDX(5000), NDY(5000)
      COMMON NDZ(5000), ITYPE(5000), NDWEL(20,2,25), LEAK(100),
     1       ALEAK(100)
      COMMON XIN(100), YIN(100), ISAVE(25,2), PPOSX(10), PPOSY(10),
     1       PPOSZ(10)
      COMMON OBSX(10), OBSY(10), OBSZ(10), ND(16), NND(16)
      COMMON COX(10), COY(10), COZ(10), IFE(6), NPMP(25), IPPN(10)
      COMMON ICRNR(10,8), NFACE(6,4), ICNTST(4,2), NODE1(8)
      COMMON IP, NODES, NPTS1, LMENTS, IO, ITRNS, INC, IELEM, NZ, ID,
     1       YC, XC, IPN
      COMMON DELX, DELY, SCL, IW1, IW2, IK, NBN, NPP, NXSTPS, NYSTPS,
     1       DELTZ
      DIMENSION NEWIN(3000,8), XNN(5000), YNN(5000), ZNN(5000),
     1          JOINT(5000)
      DIMENSION NEWCX(5000), NTYPE(5000)
      DIMENSION JNT(5000), NEWJT(5000)
      MINMAX = IDIFF
      NPN = 0
   10 FORMAT ('NPTS=', I5)
      DO 30 J = 1, NODES
        MEM1 = JMEM(J)
   20   FORMAT (9I5)
        JNT(J) = 0
```

```
   30 CONTINUE
      DO 40 J = 1, NODES
        JOINT(J) = 0
        JNT(J) = 0
        NTYPE(J) = 0
   40 NEWJT(J) = 0
      MAX = 0
      I = 1
      NEWJT(1) = IK
      JOINT(IK) = 1
      K = 1
   50 K4 = JMEM(NEWJT(I))
      IF (K4 .EQ. 0) GO TO 80
   60 FORMAT (2I5)
      DO 70 JJ = 1, K4
        K5 = MEMJT(NEWJT(I),JJ)
        IF (JOINT(K5) .GT. 0) GO TO 70
        K = K + 1
        NEWJT(K) = K5
        JOINT(K5) = K
        NDIFF = IABS(I - K)
        IF (NDIFF .GE. MINMAX) GO TO 110
        IF (NDIFF .GT. MAX) MAX = NDIFF
   70 CONTINUE
      IF (K .GE. NODES) GO TO 90
   80 I = I + 1
      GO TO 50
   90 MINMAX = MAX
      DO 100 J = 1, NODES
  100 JNT(J) = JOINT(J)
  110 CONTINUE
      IF (JNT(1) .EQ. 0) GO TO 200
      DO 140 I = 1, NODES
        ND1 = NEWJT(I)
        XNN(I) = XN(ND1)
        YNN(I) = YN(ND1)
        ZNN(I) = ZN(ND1)
        NTYPE(I) = ITYPE(ND1)
        IF (ITYPE(ND1) .NE. 2) GO TO 120
        NPN = NPN + 1
        NPMP(NPN) = I
  120   IF (ID .LT. 1) NEWCX(I) = CX(ND1)
  130   FORMAT (I5, 2F10.3)
  140 CONTINUE
      DO 160 I = 1, LMENTS
        DO 150 J = 1, INC
          JND = IN(I,J)
          NEWIN(I,J) = JOINT(JND)
  150   CONTINUE
  160 CONTINUE
```

```
      DO 170 I = 1, NODES
        XN(I) = XNN(I)
        YN(I) = YNN(I)
        ZN(I) = ZNN(I)
        ITYPE(I) = NTYPE(I)
        IF (ID .LT. 1) CX(I) = NEWCX(I)
  170 CONTINUE
      DO 190 I = 1, LMENTS
        DO 180 J = 1, INC
          IN(I,J) = NEWIN(I,J)
  180   CONTINUE
  190 CONTINUE
  200 CONTINUE
      MAX = MAX + 1
      RETURN
      END
C
C ..................................................................
C
C   Subroutine NDEGEN
C
C   Author
C       G.J. Smith
C
C   Last Update
C       May 1984
C
C   Purpose
C       This subroutine generates a node, given a
C       radial distance and an azimuth from a
C       reference node.
C
C   Usage
C       CALL NDEGEN(I,II,PHI,R,THETA,Ndd)
C
C   Subroutines Required
C
C       None
C
C
C ..................................................................
      SUBROUTINE NDEGEN(I, II, PHI, R, THETA, NDD)
      COMMON MEMJT(5000,32), IN(3000,8), IFACE(3500,6), YN(5000),
     1       ZN(5000)
      COMMON CX(5000)
      COMMON XN(5000), JMEM(5000), NBDND(4000), NDX(5000), NDY(5000)
      COMMON NDZ(5000), ITYPE(5000), NDWEL(20,2,25), LEAK(100),
     1       ALEAK(100)
      COMMON XIN(100), YIN(100), ISAVE(25,2), PPOSX(10), PPOSY(10),
     1       PPOSZ(10)
```

```
      COMMON OBSX(10), OBSY(10), OBSZ(10), ND(16), NND(16)
      COMMON COX(10), COY(10), COZ(10), IFE(6), NPMP(25), IPPN(10)
      COMMON ICRNR(10,8), NFACE(6,4), ICNTST(4,2), NODE1(8)
      COMMON IP, NPTS, NPTS1, NELM, IO, ITRNS, INC, IELEM, NZ, ID, YC,
     1       XC, IPN
      COMMON DELX, DELY, SCL, IW1, IW2, IK, NBN, NPP, NXSTPS, NYSTPS,
     1       DELTZ
   10 FORMAT (3F13.3)
      PHI = PHI + THETA
      IF (PHI .GT. 6.283) PHI = PHI - 6.283
      XA1 = COS(PHI) * R
      YA1 = SIN(PHI) * R
   20 FORMAT (2F13.3)
      YN(NPTS + II) = YN(I) + YA1
      XN(NPTS + II) = XN(I) + XA1
      IF (ID .GT. 0) ZN(NPTS + II) = ZN(I)
      NDD = NPTS + II
      RETURN
      END
C
C
C ...................................................
C
C   Subroutine INTANG
C
C   Author
C       G.J. Smith
C
C   Last Update
C       May 1984
C
C   Purpose
C       Calculates the interior angle(azimuth) from
C       the horizontal to the required node from
C       a reference node.
C
C   Usage
C       CALL INTANG(ANGLE,I,TEMP,I2,THETA)
C
C   Subroutines Required
C       None
C
C
C ...................................................
      SUBROUTINE INTANG(ANGLE, I, TEMP, I2, THETA)
      COMMON MEMJT(5000,32), IN(3000,8), IFACE(3500,6), YN(5000),
     1       ZN(5000)
      COMMON CX(5000)
      COMMON XN(5000), JMEM(5000), NBDND(4000), NDX(5000), NDY(5000)
      COMMON NDZ(5000), ITYPE(5000), NDWEL(20,2,25), LEAK(100),
```

```
     1       ALEAK(100)
      COMMON XIN(100), YIN(100), ISAVE(25,2), PPOSX(10), PPOSY(10),
     1       PPOSZ(10)
      COMMON OBSX(10), OBSY(10), OBSZ(10), ND(16), NND(16)
      COMMON COX(10), COY(10), COZ(10), IFE(6), NPMP(25), IPPN(10)
      COMMON ICRNR(10,8), NFACE(6,4), ICNTST(4,2), NODE1(8)
      COMMON IP, NPTS, NPTS1, NELM, IO, ITRNS, INC, IELEM, NZ, ID, YC,
     1       XC, IPN
      COMMON DELX, DELY, SCL, IW1, IW2, IK, NBN, NPP, NXSTPS, NYSTPS,
     1       DELTZ
      DX1 = TEMP
      X1 = ABS(DX1)
      DX2 = XN(I) - XN(I2)
      X2 = ABS(DX2)
      DY1 = 0.00
      Y1 = ABS(DY1)
      DY2 = YN(I) - YN(I2)
      Y2 = ABS(DY2)
C  Since some systems do not allow negative arguements in the tangent
C function,
C one must orientate the angle otherwise.
      IF (DX1 .GT. 0.00 .AND. DY1 .GE. 0.00) THETA1 = ATAN2(Y1,X1)
      IF (DX1 .LT. 0.00 .AND. DY1 .GT. 0.00) THETA1 = 3.1412 - ATAN2(Y1,
     1X1)
      IF (DX1 .LT. 0.00 .AND. DY1 .LT. 0.00) THETA1 = 3.1412 + ATAN2(Y1,
     1X1)
      IF (DX1 .GT. 0.00 .AND. DY1 .LT. 0.00) THETA1 = 6.282 - ATAN2(Y1,
     1X1)
      IF (DX2 .LT. 0.00 .AND. DY2 .LT. 0.00) THETA2 = ATAN2(Y2,X2)
      IF (DX2 .GT. 0.00 .AND. DY2 .LT. 0.00) THETA2 = 3.1412 - ATAN2(Y2,
     1X2)
      IF (DX2 .GT. 0.00 .AND. DY2 .GT. 0.00) THETA2 = 3.1412 + ATAN2(Y2,
     1X2)
      IF (DX2 .LT. 0.00 .AND. DY2 .GT. 0.00) THETA2 = 6.282 - ATAN2(Y2,
     1X2)
      ANGLE = (THETA2 - THETA1)
      IF (ANGLE .LT. 0.00) ANGLE = 6.282 + ANGLE
      THETA = THETA1
      IF (DX1 .GT. 0.00 .AND. X1 .GT. Y1 .AND. DY1 .LT. 0.00 .AND. DY2
     1.GT. 0.00 .AND. Y2 .GT. Y1) ANGLE = 6.282 - ANGLE
      IF (DY1 .GT. 0.00 .AND. Y1 .GT. X1 .AND. DX1 .GT. 0.00 .AND. DX2
     1.LT. 0.00 .AND. X1 .LT. X2) ANGLE = 6.282 - ANGLE
      IF (DY1 .GT. 0.00 .AND. Y1 .GT. X1 .AND. DX1 .GT. 0.00 .AND. DX2
     1.LT. 0.00 .AND. X2 .LT. X1 .AND. DY .LT. 0.00) ANGLE = 6.282 -
     2ANGLE
      RETURN
      END
C
C
C ....................................................
```

```
C
C   Subroutine ASKPLT
C
C   Author
C       R. Buttuls 1979, modified G.J. Smith
C
C   Last Update
C       May 1984
C
C   Purpose
C       Queries and generates output for use on
C       the Calcomp plotter.
C
C   Usage
C       CALL ASKPLT
C
C   Subroutines Reqired
C       Integrated Graphics Library Subroutines
C
C
C.....................................................
      SUBROUTINE ASKPLT
      INTEGER YES /'Y'/
      PRINT 10
   10 FORMAT ('&PLOT?')
      READ (5,20) IANSWR
   20 FORMAT (A1)
      IF (IANSWR .NE. YES) RETURN
      WRITE (6,30)
   30 FORMAT ('Presently the size of the plot will fit on an', /,
     1        '8.5 x 11 inch sheet of paper.  Dou you wish a ', /,
     2        'larger plot?')
      READ (5,20) IANSWR
      IF (IANSWR .NE. YES) GO TO 60
      WRITE (6,40)
   40 FORMAT ('The screen is presently equal to 7.5 inches', /,
     1        'in both the horizontal and vertical directions', /,
     2        'Enter a value "S" in inches that you wish the ', /,
     3        'dimensions of the plotted information to cover ', /,
     4        'the format is F10.0.')
      READ (5,50) S
      S = S * 2.54
   50 FORMAT (F10.0)
C *************************************************************
      CALL IGCTRL('CALCOMP', 'SIZE', S)
C *************************************************************
   60 CALL IGDRON('CALCOMP')
C *************************************************************
      PRINT 70
   70 FORMAT ('&TERMINATE PLOT?')
```

```
      READ (5,20) IANSWR
      IF (IANSWR .NE. YES) RETURN
C ***********************************************************
      CALL IGCTRL('CALCOMP', 'ENDPLOT')
C ***********************************************************
      RETURN
      END
C
C
C ..................................................
C
C   Subroutine CNCTQ
C
C   Author
C       G.J. Smith
C
C   Last Update
C       May 1984
C
C   Purpose
C       Concentrates Quadrilateral and brick type
C       elements around pumping wells.
C
C   Usage
C       CALL CNCTQ(I,IM,IMM)
C
C   Subroutines Required
C       None
C
C
C ..................................................
      SUBROUTINE CNCTQ(I, IM, IMM)
      COMMON MEMJT(5000,32), IN(3000,8), IFACE(3500,6), YN(5000),
     1       ZN(5000)
      COMMON CX(5000)
      COMMON XN(5000), JMEM(5000), NBDND(4000), NDX(5000), NDY(5000)
      COMMON NDZ(5000), ITYPE(5000), NDWEL(20,2,25), LEAK(100),
     1       ALEAK(100)
      COMMON XIN(100), YIN(100), ISAVE(25,2), PPOSX(10), PPOSY(10),
     1       PPOSZ(10)
      COMMON OBSX(10), OBSY(10), OBSZ(10), ND(16), NND(16)
      COMMON COX(10), COY(10), COZ(10), IFE(6), NPMP(25), IPPN(10)
      COMMON ICRNR(10,8), NFACE(6,4), ICNTST(4,2), NODE1(8)
      COMMON IP, NPTS, NPTS1, NELM, IO, ITRNS, INC, IELEM, NZ, ID, YC,
     1       XC, IPN
      COMMON DELX, DELY, SCL, IW1, IW2, IK, NBN, NPP, NXSTPS, NYSTPS,
     1       DELTZ
   10 FORMAT (3I5)
      K = 1
      II = 1
```

```
      K1 = 1
      JELM = NELM + 1
C Determine the interconnecting nodes and their relative position
      DO 40 J = K1, NELM
        DO 30 L = 1, INC
          IF (ID .GT. 0 .AND. K .GT. 8) GO TO 50
          IF (ID .LT. 1 .AND. K .GT. 4) GO TO 50
          IF (I .EQ. IN(J,L)) GO TO 20
          GO TO 30
   20     ISAVE(K,1) = J
          ISAVE(K,2) = L
          K = K + 1
          IF (ID .LT. 1) GO TO 40
          ISAVE(K,1) = J
          ISAVE(K,2) = L + 4
          K = K + 1
          GO TO 40
   30   CONTINUE
   40 CONTINUE
   50 KK = K - 1
      K1 = J
      TEMP = SQRT(DELX**2. + DELY**2.)
      R = 0.85 * TEMP
      R1 = 0.325 * TEMP
      K = 1
      DO 220 K1 = 1, KK
        IF (ID .LT. 1) CX(NPTS + II) = COX(ITRNS)
        IF (ID .LT. 1) CX(NPTS + II + 1) = COX(ITRNS)
        N1 = ISAVE(K1,1)
        N2 = ISAVE(K1,2)
        GO TO (60, 100, 140, 180, 60, 100, 140, 180), N2
   60   IF (N2 .GT. 4) GO TO 70
        NM1 = 1
        NM2 = 2
        NIN = 2
        IMM1 = IMM
        GO TO 80
   70   IF (IMM .GT. 2) GO TO 220
        NM1 = 9
        NM2 = 10
        NIN = 6
        IMM1 = IMM - 1
        IF (ID .LT. 1) IMM1 = IMM
C form new nodes within elements surrounding the pumping well
   80   DO 90 NM = NM1, NM2
C ***************************************************************
          CALL INTANG(ANGLE, NDWEL(IM,2,IMM), TEMP, IN(N1,NIN), THETA)
C ***************************************************************
          CALL NDEGEN(NDWEL(IM,2,IMM1), II, ANGLE, R, THETA, NND(NM))
C ***************************************************************
```

```
      II = II + 1
C ************************************************************
      CALL NDEGEN(NDWEL(IM,2,IMM1), II, ANGLE, R1, THETA, ND(NM))
C ************************************************************
      II = II + 1
      NIN = NIN + 1
 90   CONTINUE
      GO TO 220
 100  IF (N2 .GT. 4) GO TO 110
      NM1 = 3
      NM2 = 4
      NIN = 3
      IMM1 = IMM
      GO TO 120
 110  IF (IMM .GT. 2) GO TO 220
      NM1 = 11
      NM2 = 12
      NIN = 7
      IMM1 = IMM - 1
      IF (ID .LT. 1) IMM1 = IMM
 120  DO 130 NM = NM1, NM2
C ************************************************************
      CALL INTANG(ANGLE, NDWEL(IM,2,IMM), TEMP, IN(N1,NIN), THETA)
C ************************************************************
      CALL NDEGEN(NDWEL(IM,2,IMM1), II, ANGLE, R, THETA, NND(NM))
C ************************************************************
      II = II + 1
C ************************************************************
      CALL NDEGEN(NDWEL(IM,2,IMM1), II, ANGLE, R1, THETA, ND(NM))
C ************************************************************
      II = II + 1
      NIN = NIN + 1
 130  CONTINUE
      GO TO 220
 140  IF (N2 .GT. 4) GO TO 150
      NM1 = 5
      NM2 = 6
      NIN = 4
      IMM1 = IMM
      GO TO 160
 150  IF (IMM .GT. 2) GO TO 220
      NM1 = 13
      NM2 = 14
      NIN = 8
      IMM1 = IMM - 1
      IF (ID .LT. 1) IMM1 = IMM
 160  DO 170 NM = NM1, NM2
C ************************************************************
      CALL INTANG(ANGLE, NDWEL(IM,2,IMM), TEMP, IN(N1,NIN), THETA)
C ************************************************************
```

```
              CALL NDEGEN(NDWEL(IM,2,IMM1), II, ANGLE, R, THETA, NND(NM))
C ************************************************************
              II = II + 1
C ************************************************************
              CALL NDEGEN(NDWEL(IM,2,IMM1), II, ANGLE, R1, THETA, ND(NM))
C ************************************************************
              II = II + 1
              IF (N2 .GT. 4) NIN = 5
              IF (N2 .LT. 5) NIN = 1
  170     CONTINUE
          GO TO 220
  180     IF (N2 .GT. 4) GO TO 190
          NM1 = 7
          NM2 = 8
          NIN = 1
          IMM1 = IMM
          GO TO 200
  190     IF (IMM .GT. 2) GO TO 220
          NM1 = 15
          NM2 = 16
          NIN = 5
          IMM1 = IMM - 1
          IF (ID .LT. 1) IMM1 = IMM
  200     DO 210 NM = NM1, NM2
C ************************************************************
              CALL INTANG(ANGLE, NDWEL(IM,2,IMM), TEMP, IN(N1,NIN), THETA)
C ************************************************************
              CALL NDEGEN(NDWEL(IM,2,IMM1), II, ANGLE, R, THETA, NND(NM))
C ************************************************************
              II = II + 1
C ************************************************************
              CALL NDEGEN(NDWEL(IM,2,IMM1), II, ANGLE, R1, THETA, ND(NM))
C ************************************************************
              NIN = NIN + 1
              II = II + 1
  210     CONTINUE
  220 CONTINUE
        IMM1 = IMM - 1
        IF (ID .LT. 1) IMM1 = IMM
C form elements out of these newly generated nodes
        DO 250 NM = 1, 7
          IN(JELM,1) = ND(NM)
          IN(JELM,2) = NND(NM)
          IN(JELM,3) = NND(NM + 1)
          IN(JELM,4) = ND(NM + 1)
          IF (ID .LT. 1) GO TO 230
          IN(JELM,5) = ND(NM + 8)
          IN(JELM,6) = NND(NM + 8)
          IN(JELM,7) = NND(NM + 9)
          IN(JELM,8) = ND(NM + 9)
```

```
         NDZ(ND(NM + 8)) = ND(NM)
         NDZ(NND(NM + 8)) = NND(NM)
         NDZ(NND(NM + 9)) = NND(NM + 1)
         NDZ(ND(NM + 9)) = ND(NM + 1)
  230    CONTINUE
  240    FORMAT (9I5)
         JELM = JELM + 1
  250 CONTINUE
      IN(JELM,1) = ND(8)
      IN(JELM,2) = NND(8)
      IN(JELM,3) = NND(1)
      IN(JELM,4) = ND(1)
      IF (ID .LT. 1) GO TO 260
      IN(JELM,5) = ND(16)
      IN(JELM,6) = NND(16)
      IN(JELM,7) = NND(9)
      IN(JELM,8) = ND(9)
      NDZ(ND(16)) = ND(8)
      NDZ(NND(16)) = NND(8)
  260 CONTINUE
      JELM = JELM + 1.
      IN(JELM,1) = NDWEL(IM,2,IMM)
      IN(JELM,2) = ND(1)
      IN(JELM,3) = ND(2)
      IN(JELM,4) = ND(3)
      IF (ID .LT. 1) GO TO 270
      IN(JELM,5) = NDWEL(IM,2,IMM1)
      IN(JELM,6) = ND(9)
      IN(JELM,7) = ND(10)
      IN(JELM,8) = ND(11)
  270 CONTINUE
      JELM = JELM + 1
      IN(JELM,1) = ND(5)
      IN(JELM,2) = NDWEL(IM,2,IMM)
      IN(JELM,3) = ND(3)
      IN(JELM,4) = ND(4)
      IF (ID .LT. 1) GO TO 280
      IN(JELM,5) = ND(13)
      IN(JELM,6) = NDWEL(IM,2,IMM1)
      IN(JELM,7) = ND(11)
      IN(JELM,8) = ND(12)
  280 CONTINUE
      JELM = JELM + 1
      IN(JELM,1) = ND(6)
      IN(JELM,2) = ND(7)
      IN(JELM,3) = NDWEL(IM,2,IMM)
      IN(JELM,4) = ND(5)
      IF (ID .LT. 1) GO TO 290
      IN(JELM,5) = ND(14)
      IN(JELM,6) = ND(15)
```

```
      IN(JELM,7) = NDWEL(IM,2,IMM1)
      IN(JELM,8) = ND(13)
  290 CONTINUE
      JELM = JELM + 1
      IN(JELM,1) = ND(7)
      IN(JELM,2) = ND(8)
      IN(JELM,3) = ND(1)
      IN(JELM,4) = NDWEL(IM,2,IMM)
      IF (ID .LT. 1) GO TO 300
      IN(JELM,5) = ND(15)
      IN(JELM,6) = ND(16)
      IN(JELM,7) = ND(9)
      IN(JELM,8) = NDWEL(IM,2,IMM1)
  300 CONTINUE
      JELM = JELM + 1
      DO 420 K1 = 1, KK
        N1 = ISAVE(K1,1)
        N2 = ISAVE(K1,2)
        IF (N2 .GT. 4) GO TO 420
        GO TO (310, 340, 370, 400), N2
  310   IN(JELM,1) = NND(3)
        IN(JELM,2) = NND(2)
        IN(JELM,3) = IN(N1,3)
        IN(JELM,4) = IN(N1,4)
        NDX(ND(1)) = ND(2)
        NDX(NND(1)) = NND(2)
        NDY(NND(2)) = NND(3)
        NDY(ND(2)) = ND(3)
        NDY(ND(1)) = NDWEL(IM,2,IMM)
        NDY(NND(1)) = ND(1)
        NDY(IN(N1,2)) = NND(1)
        NDX(ND(2)) = NND(2)
        NDX(NND(2)) = IN(N1,3)
        IF (ID .LT. 1) GO TO 320
        IN(JELM,5) = NND(11)
        IN(JELM,6) = NND(10)
        IN(JELM,7) = IN(N1,7)
        IN(JELM,8) = IN(N1,8)
        NDX(ND(9)) = ND(10)
        NDX(NND(1)) = NND(10)
        NDY(NND(10)) = NND(11)
        NDY(ND(10)) = ND(11)
        NDY(ND(9)) = NDWEL(IM,2,IMM1)
        NDY(NND(9)) = ND(9)
        NDY(IN(N1,6)) = NND(9)
        NDX(ND(10)) = NND(10)
        NDX(NND(10)) = IN(N1,7)
  320   CONTINUE
        JELM = JELM + 1
        IN(N1,1) = NND(1)
```

```
      IN(N1,4) = NND(2)
      IF (ID .LT. 1) GO TO 330
      IN(N1,5) = NND(9)
      IN(N1,8) = NND(10)
330   CONTINUE
      GO TO 420
340   IN(JELM,1) = NND(4)
      IN(JELM,2) = NND(3)
      IN(JELM,3) = IN(N1,3)
      IN(JELM,4) = IN(N1,4)
      NDX(ND(5)) = ND(4)
      NDX(NND(5)) = ND(4)
      NDY(ND(5)) = NND(5)
      NDY(NND(5)) = IN(N1,1)
      NDY(ND(4)) = NND(4)
      NDY(NND(4)) = IN(N1,4)
      NDY(ND(3)) = ND(4)
      NDY(NND(3)) = NND(4)
      NDX(NDWEL(IM,2,IMM)) = ND(3)
      NDX(ND(3)) = NND(3)
      NDY(NDWEL(IM,2,IMM)) = ND(5)
      IF (ID .LT. 1) GO TO 350
      IN(JELM,5) = NND(12)
      IN(JELM,6) = NND(11)
      IN(JELM,7) = IN(N1,7)
      IN(JELM,8) = IN(N1,8)
      NDX(ND(13)) = ND(12)
      NDX(NND(13)) = ND(12)
      NDY(ND(13)) = NND(13)
      NDY(NND(13)) = IN(N1,5)
      NDY(ND(12)) = NND(12)
      NDY(NND(12)) = IN(N1,8)
      NDY(ND(11)) = ND(12)
      NDY(NND(11)) = NND(12)
      NDX(NDWEL(IM,2,IMM1)) = ND(11)
      NDX(ND(11)) = NND(11)
      NDY(NDWEL(IM,2,IMM1)) = ND(13)
350   CONTINUE
      JELM = JELM + 1
      IN(N1,2) = NND(5)
      IN(N1,3) = NND(4)
      IF (ID .LT. 1) GO TO 360
      IN(N1,6) = NND(13)
      IN(N1,7) = NND(12)
360   CONTINUE
      GO TO 420
370   IN(JELM,1) = IN(N1,1)
      IN(JELM,2) = NND(6)
      IN(JELM,3) = NND(5)
      IN(JELM,4) = IN(N1,4)
```

```
      NDX(ND(6)) = ND(5)
      NDX(NND(6)) = NND(5)
      NDY(ND(6)) = NND(6)
      NDY(NND(6)) = IN(N1,1)
      NDX(NND(7)) = ND(7)
      NDX(IN(N1,2)) = NND(7)
      NDY(NND(7)) = NND(6)
      NDY(ND(7)) = ND(6)
      NDX(ND(7)) = NDWEL(IM,2,IMM)
      IF (ID .LT. 1) GO TO 380
      IN(JELM,5) = IN(N1,5)
      IN(JELM,6) = NND(14)
      IN(JELM,7) = NND(13)
      IN(JELM,8) = IN(N1,8)
      NDX(ND(14)) = ND(13)
      NDX(NND(14)) = NND(13)
      NDY(ND(14)) = NND(14)
      NDY(NND(14)) = IN(N1,5)
      NDX(NND(15)) = ND(15)
      NDX(IN(N1,6)) = NND(15)
      NDY(NND(15)) = NND(14)
      NDY(ND(15)) = ND(14)
      NDX(ND(15)) = NDWEL(IM,2,IMM1)
380   CONTINUE
      JELM = JELM + 1
      IN(N1,3) = NND(7)
      IN(N1,4) = NND(6)
      IF (ID .LT. 1) GO TO 390
      IN(N1,7) = NND(15)
      IN(N1,8) = NND(14)
390   CONTINUE
      GO TO 420
400   IN(JELM,1) = IN(N1,1)
      IN(JELM,2) = IN(N1,2)
      IN(JELM,3) = NND(8)
      IN(JELM,4) = NND(7)
      NDX(ND(8)) = ND(1)
      NDX(NND(8)) = NND(1)
      NDY(NND(8)) = NND(7)
      NDY(ND(8)) = ND(7)
      NDY(IN(N1,2)) = NND(8)
      IF (ID .LT. 1) GO TO 410
      IN(JELM,5) = IN(N1,5)
      IN(JELM,6) = IN(N1,6)
      IN(JELM,7) = NND(16)
      IN(JELM,8) = NND(15)
      NDX(ND(16)) = ND(9)
      NDX(NND(16)) = NND(9)
      NDY(NND(16)) = NND(15)
      NDY(ND(16)) = ND(15)
```

```
          NDY(IN(N1,6)) = NND(16)
  410     CONTINUE
          JELM = JELM + 1
          IN(N1,1) = NND(8)
          IN(N1,4) = NND(1)
          IF (ID .LT. 1) GO TO 420
          IN(N1,5) = NND(16)
          IN(N1,8) = NND(9)
  420 CONTINUE
      IF (ID .LT. 1) GO TO 440
      DO 430 I = 1, 8
         ND(I + 8) = ND(I)
         NND(I + 8) = NND(I)
  430 CONTINUE
  440 CONTINUE
      NPTS = NPTS + II - 1
      II = 1
      NELM = JELM - 1
      RETURN
      END
C
C
C ...................................................
C
C   Subroutine CNCTT1
C
C   Author
C       G.J. Smith
C
C   Last Update
C       May 1984
C
C   Purpose
C       To concentrate triangular and tetrahedral
C       elements around a pumping well.
C
C   Usage
C       CALL CNCTT1(I,IM,IMM)
C
C   Subroutines Required
C       None
C
C
C ...................................................
      SUBROUTINE CNCTT1(I, IM, IMM)
      COMMON MEMJT(5000,32), IN(3000,8), IFACE(3500,6), YN(5000),
     1       ZN(5000)
      COMMON CX(5000)
      COMMON XN(5000), JMEM(5000), NBDND(4000), NDX(5000), NDY(5000)
      COMMON NDZ(5000), ITYPE(5000), NDWEL(20,2,25), LEAK(100),
```

```
     1        ALEAK(100)
      COMMON XIN(100), YIN(100), ISAVE(25,2), PPOSX(10), PPOSY(10),
     1        PPOSZ(10)
      COMMON OBSX(10), OBSY(10), OBSZ(10), ND(16), NND(16)
      COMMON COX(10), COY(10), COZ(10), IFE(6), NPMP(25), IPPN(10)
      COMMON ICRNR(10,8), NFACE(6,4), ICNTST(4,2), NODE1(8)
      COMMON IP, NPTS, NPTS1, NELM, IO, ITRNS, INC, IELEM, NZ, ID, YC,
     1        XC, IPN
      COMMON DELX, DELY, SCL, IW1, IW2, IK, NBN, NPP, NXSTPS, NYSTPS,
     1        DELTZ
      JELM = NELM + 1
   10 FORMAT (3I5)
      K = 1
      II = 1
   20 FORMAT (5I5)
      DO 50 J = 1, NELM
        DO 40 L = 1, INC
          IF (I .EQ. IN(J,L)) GO TO 30
          GO TO 40
   30     ISAVE(K,1) = J
          ISAVE(K,2) = L
          K = K + 1
          GO TO 50
   40   CONTINUE
   50 CONTINUE
      KK = K - 1
      TEMP = SQRT(DELX**2. + DELY**2.)
      R = 0.8 * TEMP
      R1 = 0.325 * TEMP
      K = 1
      IF (ID .LT. 1 .AND. KK .GT. 4) GO TO 510
      DO 230 K1 = 1, KK
        IF (ID .LT. 1) CX(NPTS + II) = COX(ITRNS)
        IF (ID .LT. 1) CX(NPTS + II + 1) = COX(ITRNS)
        N1 = ISAVE(K1,1)
        N2 = ISAVE(K1,2)
        IF (ID .GT. 0) GO TO 60
        GO TO (70, 110, 150), N2
   60   IF (N2 .NE. 3) GO TO 230
        IF (IN(N1,4) .EQ. I .AND. IN(N1,1) .EQ. I) GO TO 510
        IF (ABS(XN(IN(N1,3)) - XN(IN(N1,4))) .GT. ABS(YN(IN(N1,3)) - YN(
     1     IN(N1,4)))) .A ND .(XN(IN(N1,4)) .GT. XN(IN(N1,3)))) GOTO
     2     70
        IF (ABS(YN(IN(N1,3)) - YN(IN(N1,4))) .GT. ABS(XN(IN(N1,3)) - XN(
     1     IN(N1,4)))) .A ND .(YN(IN(N1,4)) .GT. YN(IN(N1,3)))) GOTO
     2     110
        IF (ABS(XN(IN(N1,3)) - XN(IN(N1,4))) .GT. ABS(YN(IN(N1,3)) - YN(
     1     IN(N1,4)))) .A ND .(XN(IN(N1,4)) .LT. XN(IN(N1,3)))) GOTO
     2     150
        IF (ABS(YN(IN(N1,3)) - YN(IN(N1,4))) .GT. ABS(XN(IN(N1,3)) - YN(
```

```
     1     IN(N1,4))).).A ND .(YN(IN(N1,4)) .LT. YN(IN(N1,3)))) GOTO
     2     190
   70  NM1 = 1
       NM2 = 2
       NIN = IN(N1,4)
       IF (ID .LT. 1) NIN = IN(N1 + 1,2)
       IMM1 = IMM
       GO TO 90
   80  IF (IMM .GT. 2) GO TO 230
       NM1 = 9
       NM2 = 10
       NIN = IN(N1,1)
       IMM1 = IMM - 1
       IF (ID .LT. 1) IMM1 = IMM
   90  DO 100 NM = NM1, NM2
C ************************************************************
         CALL INTANG(ANGLE, NDWEL(IM,2,IMM), TEMP, NIN, THETA)
C ************************************************************
         CALL NDEGEN(NDWEL(IM,2,IMM1), II, ANGLE, R, THETA, NND(NM))
C ************************************************************
         II = II + 1
C ************************************************************
         CALL NDEGEN(NDWEL(IM,2,IMM1), II, ANGLE, R1, THETA, ND(NM))
C ************************************************************
         II = II + 1
         NIN = NDX(NIN)
  100  CONTINUE
       IF (ID .GT. O .AND. NM .LT. 4) GO TO 80
       GO TO 230
  110  NM1 = 3
       NM2 = 4
       NIN = IN(N1,4)
       IF (ID .LT. 1) NIN = IN(N1,3)
       IMM1 = IMM
       GO TO 130
  120  NM1 = 11
       NM2 = 12
       NIN = IN(N1,1)
       IMM1 = IMM - 1
       IF (ID .LT. 1) IMM1 = IMM
  130  DO 140 NM = NM1, NM2
C ************************************************************
         CALL INTANG(ANGLE, NDWEL(IM,2,IMM), TEMP, NIN, THETA)
C ************************************************************
         CALL NDEGEN(NDWEL(IM,2,IMM1), II, ANGLE, R, THETA, NND(NM))
C ************************************************************
         II = II + 1
C ************************************************************
         CALL NDEGEN(NDWEL(IM,2,IMM1), II, ANGLE, R1, THETA, ND(NM))
C ************************************************************
```

```
         II = II + 1
         NIN = NDY(NIN)
  140   CONTINUE
        IF (ID .GT. 0 .AND. NM .LT. 5) GO TO 120
        GO TO 230
  150   IF (ABS(YN(IN(N1,1)) - YN(IN(N1,3))) .LT. ABS(YN(IN(N1,2)) - YN(
     1     IN(N1,3))) .AND. ID .LT. 1) GO TO 190
        NM1 = 5
        NM2 = 6
        NIN = IN(N1,4)
        IF (ID .LT. 1) NIN = IN(N1 - 1,3)
        IMM1 = IMM
        GO TO 170
  160   IF (IMM .GT. 2) GO TO 230
        NM1 = 13
        NM2 = 14
        NIN = IN(N1,1)
        IMM1 = IMM - 1
        IF (ID .LT. 1) IMM1 = IMM
  170   DO 180 NM = NM1, NM2
C ************************************************************
         CALL INTANG(ANGLE, NDWEL(IM,2,IMM), TEMP, NIN, THETA)
C ************************************************************
         CALL NDEGEN(NDWEL(IM,2,IMM1), II, ANGLE, R, THETA, NND(NM))
C ************************************************************
         II = II + 1
C ************************************************************
         CALL NDEGEN(NDWEL(IM,2,IMM1), II, ANGLE, R1, THETA, ND(NM))
C ************************************************************
         II = II + 1
         NIN = IN(N1,2)
         IF (ID .LT. 1) NIN = IN(N1 - 1,2)
         IF (NM .GT. 7) NIN = NDZ(IN(N1,2))
  180   CONTINUE
        IF (ID .GT. 0 .AND. NM .LT. 7) GO TO 160
        GO TO 230
  190   NM1 = 7
        NM2 = 8
        NIN = IN(N1,4)
        IF (ID .LT. 1) NIN = IN(N1 - 1,1)
        IMM1 = IMM
        GO TO 210
  200   IF (IMM .GT. 2) GO TO 230
        NM1 = 15
        NM2 = 16
        NIN = IN(N1,1)
        IMM1 = IMM - 1
        IF (ID .LT. 1) IMM1 = IMM
  210   DO 220 NM = NM1, NM2
C ************************************************************
```

```
                CALL INTANG(ANGLE, NDWEL(IM,2,IMM), TEMP, NIN, THETA)
C ***********************************************************
                CALL NDEGEN(NDWEL(IM,2,IMM1), II, ANGLE, R, THETA, NND(NM))
C ***********************************************************
                II = II + 1
C ***********************************************************
                CALL NDEGEN(NDWEL(IM,2,IMM1), II, ANGLE, R1, THETA, ND(NM))
C ***********************************************************
                II = II + 1
                NIN = IN(N1,2)
                IF (ID .LT. 1) NIN = IN(N1 - 1,2)
                IF (NM .GT. 9) NIN = NDZ(IN(N1,2))
              CONTINUE
              IF (ID .GT. 0 .AND. NM .LT. 9) GO TO 200
  230     CONTINUE
          IMM1 = IMM - 1
          IF (ID .LT. 1) IMM1 = IMM
          DO 280 NM = 1, 5, 2
            IN(JELM,1) = ND(NM)
            IN(JELM,2) = NND(NM)
            IN(JELM,3) = NND(NM + 1)
            IF (ID .LT. 1) GO TO 240
            IN(JELM,4) = ND(NM + 8)
  240       CONTINUE
            JELM = JELM + 1
            IN(JELM,1) = ND(NM)
            IN(JELM,2) = NND(NM + 1)
            IN(JELM,3) = ND(NM + 1)
            IF (ID .LT. 1) GO TO 250
            IN(JELM,4) = ND(NM + 9)
  250       CONTINUE
            JELM = JELM + 1
            IN(JELM,1) = ND(NM + 1)
            IN(JELM,2) = NND(NM + 1)
            IN(JELM,3) = ND(NM + 2)
            IF (ID .LT. 1) GO TO 260
            IN(JELM,4) = ND(NM + 9)
  260       CONTINUE
            JELM = JELM + 1
            IN(JELM,1) = ND(NM + 2)
            IN(JELM,2) = NND(NM + 1)
            IN(JELM,3) = NND(NM + 2)
            IF (ID .LT. 1) GO TO 270
            IN(JELM,4) = NND(NM + 10)
  270       CONTINUE
            JELM = JELM + 1
            IF (ID .LT. 1) GO TO 280
            IN(JELM,1) = NND(NM + 1)
            IN(JELM,2) = ND(NM)
            IN(JELM,3) = NND(NM + 8)
```

```
              IN(JELM,4) = ND(NM + 9)
              JELM = JELM + 1
              IN(JELM,1) = ND(NM)
              IN(JELM,2) = NND(NM + 3)
              IN(JELM,3) = ND(NM + 9)
              IN(JELM,4) = ND(NM + 8)
              JELM = JELM + 1
              IN(JELM,1) = NND(NM + 1)
              IN(JELM,2) = ND(NM + 9)
              IN(JELM,3) = NND(NM + 8)
              IN(JELM,4) = NND(NM + 9)
              JELM = JELM + 1
              IN(JELM,1) = NND(NM + 2)
              IN(JELM,2) = ND(NM + 1)
              IN(JELM,3) = NND(NM + 9)
              IN(JELM,4) = ND(NM + 10)
              JELM = JELM + 1
              IN(JELM,1) = NND(NM + 2)
              IN(JELM,2) = ND(NM + 10)
              IN(JELM,3) = NND(NM + 9)
              IN(JELM,4) = NND(NM + 10)
              JELM = JELM + 1
              IN(JELM,1) = ND(NM + 1)
              IN(JELM,2) = NND(NM + 10)
              IN(JELM,3) = ND(NM + 11)
              IN(JELM,4) = ND(NM + 10)
              JELM = JELM + 1
      280 CONTINUE
          IN(JELM,1) = ND(7)
          IN(JELM,2) = NND(7)
          IN(JELM,3) = NND(8)
          IF (ID .LT. 1) GO TO 290
          IN(JELM,4) = NN[illegible]
      290 CONTINUE
          JELM = JELM + 1
          IN(JELM,1) = ND(8)
          IN(JELM,2) = ND(7)
          IN(JELM,3) = NND(8)
          IF (ID .LT. 1) GO TO 300
          IN(JELM,4) = ND(16)
      300 CONTINUE
          JELM = JELM + 1
          IN(JELM,1) = ND(8)
          IN(JELM,2) = NND(8)
          IN(JELM,3) = ND(1)
          IF (ID .LT. 1) GO TO 310
          IN(JELM,4) = ND(16)
      310 CONTINUE
          JELM = JELM + 1
          IN(JELM,1) = ND(1)
```

```
      IN(JELM,2) = NND(8)
      IN(JELM,3) = NND(1)
      IF (ID .LT. 1) GO TO 320
      IN(JELM,4) = ND(9)
  320 CONTINUE
      JELM = JELM + 1
      IF (ID .LT. 1) GO TO 330
      IN(JELM,1) = NND(8)
      IN(JELM,2) = ND(7)
      IN(JELM,3) = NND(15)
      IN(JELM,4) = ND(16)
      JELM = JELM + 1
      IN(JELM,1) = NND(8)
      IN(JELM,2) = ND(16)
      IN(JELM,3) = NND(15)
      IN(JELM,4) = NND(16)
      JELM = JELM + 1
      IN(JELM,1) = ND(7)
      IN(JELM,2) = NND(15)
      IN(JELM,3) = ND(16)
      IN(JELM,4) = ND(15)
      JELM = JELM + 1
      IN(JELM,1) = NND(8)
      IN(JELM,2) = ND(1)
      IN(JELM,3) = ND(16)
      IN(JELM,4) = NND(9)
      JELM = JELM + 1
      IN(JELM,1) = ND(1)
      IN(JELM,2) = ND(16)
      IN(JELM,3) = NND(9)
      IN(JELM,4) = ND(9)
      JELM = JELM + 1
      IN(JELM,1) = NND(8)
      IN(JELM,2) = NND(9)
      IN(JELM,3) = ND(16)
      IN(JELM,4) = NND(16)
      JELM = JELM + 1
  330 CONTINUE
      DO 360 NM = 1, 5, 2
        IN(JELM,1) = NDWEL(IM,2,IMM)
        IN(JELM,2) = ND(NM)
        IN(JELM,3) = ND(NM + 2)
        IF (ID .LT. 1) GO TO 340
        IN(JELM,4) = NDWEL(IM,2,IMM1)
  340   CONTINUE
        JELM = JELM + 1
        IN(JELM,1) = ND(NM + 1)
        IN(JELM,2) = ND(NM + 2)
        IN(JELM,3) = ND(NM)
        IF (ID .LT. 1) GO TO 350
```

```
          IN(JELM,4) = ND(NM + 9)
  350     CONTINUE
          JELM = JELM + 1
          IF (ID .LT. 1) GO TO 360
          IN(JELM,1) = ND(NM)
          IN(JELM,2) = ND(NM + 2)
          IN(JELM,3) = NDWEL(IM,2,IMM1)
          IN(JELM,4) = NND(NM + 9)
          JELM = JELM + 1
          IN(JELM,1) = ND(NM)
          IN(JELM,2) = ND(NM + 9)
          IN(JELM,3) = NDWEL(IM,2,IMM1)
          IN(JELM,4) = ND(NM + 8)
          JELM = JELM + 1
          IN(JELM,1) = ND(NM + 2)
          IN(JELM,2) = NDWEL(IM,2,IMM1)
          IN(JELM,3) = ND(NM + 9)
          IN(JELM,4) = ND(NM + 10)
          JELM = JELM + 1
  360 CONTINUE
      IN(JELM,1) = NDWEL(IM,2,IMM)
      IN(JELM,2) = ND(7)
      IN(JELM,3) = ND(1)
      IF (ID .LT. 1) GO TO 370
      IN(JELM,4) = NDWEL(IM,2,IMM1)
  370 CONTINUE
      JELM = JELM + 1
      IN(JELM,1) = ND(8)
      IN(JELM,2) = ND(1)
      IN(JELM,3) = ND(7)
      IF (ID .LT. 1) GO TO 380
      IN(JELM,4) = ND(16)
  380 CONTINUE
      JELM = JELM + 1
      IF (ID .LT. 1) GO TO 390
      IN(JELM,1) = ND(7)
      IN(JELM,2) = ND(1)
      IN(JELM,3) = NDWEL(IM,2,IMM1)
      IN(JELM,4) = ND(16)
      JELM = JELM + 1
      IN(JELM,1) = ND(1)
      IN(JELM,2) = NDWEL(IM,2,IMM1)
      IN(JELM,3) = ND(16)
      IN(JELM,4) = ND(9)
      JELM = JELM + 1
      IN(JELM,1) = ND(7)
      IN(JELM,2) = ND(16)
      IN(JELM,3) = NDWEL(IM,2,IMM1)
      IN(JELM,4) = ND(15)
      JELM = JELM + 1
```

```
  390 DO 490 K1 = 1, KK
      N1 = ISAVE(K1,1)
      N2 = ISAVE(K1,2)
      IF (ID .GT. 0) GO TO 400
      GO TO (410, 430, 450), N2
  400 IF (N2 .NE. 3) GO TO 230
      IF (ABS(XN(IN(N1,3)) - XN(IN(N1,4))) .GT. ABS(YN(IN(N1,3)) - YN(
     1   IN(N1,4))).) .A ND .(XN(IN(N1,4)) .GT. XN(IN(N1,3)))) GOTO
     2   410
      IF (ABS(YN(IN(N1,3)) - YN(IN(N1,4))) .GT. ABS(XN(IN(N1,3)) - XN(
     1   IN(N1,4))).) .A ND .(YN(IN(N1,4)) .GT. YN(IN(N1,3)))) GOTO
     2   430
      IF (ABS(XN(IN(N1,3)) - XN(IN(N1,4))) .GT. ABS(YN(IN(N1,3)) - YN(
     1   IN(N1,4))).) .A ND .(XN(IN(N1,4)) .LT. XN(IN(N1,3)))) GOTO
     2   450
      IF (ABS(YN(IN(N1,3)) - YN(IN(N1,4))) .GT. ABS(XN(IN(N1,3)) - YN(
     1   IN(N1,4))).) .A ND .(YN(IN(N1,4)) .LT. YN(IN(N1,3)))) GOTO
     2   470
  410 IF (ID .GT. 0) GO TO 420
      IN(JELM,1) = IN(N1 + 1,3)
      IN(JELM,2) = NND(2)
      IN(JELM,3) = IN(N1,2)
      JELM = JELM + 1
      IN(JELM,1) = NND(3)
      IN(JELM,2) = NND(2)
      IN(JELM,3) = IN(N1 + 1,1)
      JELM = JELM + 1
      IN(N1,1) = NND(1)
      IN(N1,3) = NND(2)
      IN(N1 + 1,2) = NND(2)
      GO TO 490
  420 IN(JELM,1) = NND(3)
      IN(JELM,2) = NND(2)
      IN(JELM,3) = IN(N1 - 1,3)
      IN(JELM,4) = NND(11)
      JELM = JELM + 1
      IN(JELM,1) = NND(2)
      IN(JELM,2) = IN(N1,2)
      IN(JELM,3) = NND(11)
      IN(JELM,4) = NND(10)
      JELM = JELM + 1
      IN(JELM,1) = IN(N1 + 1,1)
      IN(JELM,2) = IN(N1 + 1,2)
      IN(JELM,3) = NND(2)
      IN(JELM,4) = IN(N1 + 1,4)
      JELM = JELM + 1
      IN(JELM,1) = IN(N1 + 2,1)
      IN(JELM,2) = NND(11)
      IN(JELM,3) = IN(N1 + 2,3)
      IN(JELM,4) = IN(N1 + 2,4)
```

```
      JELM = JELM + 1
      IN(JELM,1) = IN(N1 + 3,1)
      IN(JELM,2) = NND(2)
      IN(JELM,3) = IN(N1 + 3,3)
      IN(JELM,4) = NND(11)
      JELM = JELM + 1
      IN(N1 - 1,1) = NND(1)
      IN(N1 - 1,4) = NND(9)
      IN(N1,3) = NND(9)
      IN(N1 + 1,2) = NND(2)
      IN(N1 + 2,1) = NND(2)
      IN(N1 + 2,2) = NND(9)
      IN(N1 + 2,4) = NND(10)
      IN(N1 + 3,1) = NND(2)
      IN(N1 + 3,4) = NND(9)
      GO TO 490
430   IF (ID .GT. 0) GO TO 440
      IN(JELM,1) = NND(3)
      IN(JELM,2) = IN(N1,3)
      IN(JELM,3) = NND(4)
      JELM = JELM + 1
      IN(JELM,1) = NND(4)
      IN(JELM,2) = IN(N1,3)
      IN(JELM,3) = IN(N1 + 1,3)
      JELM = JELM + 1
      IN(N1 + 1,2) = NND(4)
      IN(N1,2) = NND(5)
      IN(N1,3) = NND(4)
      GO TO 490
440   IN(JELM,1) = NND(5)
      IN(JELM,2) = NND(4)
      IN(JELM,3) = IN(N1 - 1,3)
      IN(JELM,4) = NND(13)
      JELM = JELM + 1
      IN(JELM,1) = NND(4)
      IN(JELM,2) = IN(N1,2)
      IN(JELM,3) = NND(13)
      IN(JELM,4) = NND(12)
      JELM = JELM + 1
      IN(JELM,1) = IN(N1 + 1,1)
      IN(JELM,2) = IN(N1 + 1,2)
      IN(JELM,3) = NND(4)
      IN(JELM,4) = IN(N1 + 1,4)
      JELM = JELM + 1
      IN(JELM,1) = IN(N1 + 2,1)
      IN(JELM,2) = NND(13)
      IN(JELM,3) = IN(N1 + 2,3)
      IN(JELM,4) = IN(N1 + 2,4)
      JELM = JELM + 1
      IN(JELM,1) = IN(N1 + 3,1)
```

```
      IN(JELM,2) = NND(4)
      IN(JELM,3) = IN(N1 + 3,3)
      IN(JELM,4) = NND(13)
      JELM = JELM + 1
      IN(N1 - 1,1) = NND(3)
      IN(N1 - 1,4) = NND(11)
      IN(N1,3) = NND(11)
      IN(N1 + 1,2) = NND(4)
      IN(N1 + 2,1) = NND(4)
      IN(N1 + 2,4) = NND(12)
      IN(N1 + 3,1) = NND(4)
      IN(N1 + 3,4) = NND(11)
      GO TO 490
  450 IF (ID .GT. 0) GO TO 460
      IF (ABS(YN(IN(N1,1)) - YN(IN(N1,3))) .LT. ABS(YN(IN(N1,2)) - YN(
     1   IN(N1,3)))) GO TO 470
      IN(JELM,1) = IN(N1,1)
      IN(JELM,2) = NND(6)
      IN(JELM,3) = NND(5)
      JELM = JELM + 1
      IN(JELM,1) = IN(N1 - 1,1)
      IN(JELM,2) = NND(6)
      IN(JELM,3) = IN(N1,1)
      JELM = JELM + 1
      IN(N1 - 1,3) = NND(6)
      IN(N1,1) = NND(6)
      IN(N1,3) = NND(7)
      GO TO 490
  460 IN(JELM,1) = IN(N1 - 3,1)
      IN(JELM,2) = NND(6)
      IN(JELM,3) = IN(N1 - 3,3)
      IN(JELM,4) = IN(N1 - 3,4)
      JELM = JELM + 1
      IN(JELM,1) = NND(6)
      IN(JELM,2) = NND(13)
      IN(JELM,3) = IN(N1 - 2,3)
      IN(JELM,4) = NND(14)
      JELM = JELM + 1
      IN(JELM,1) = NND(5)
      IN(JELM,2) = IN(N1 - 1,2)
      IN(JELM,3) = NND(6)
      IN(JELM,4) = NND(13)
      JELM = JELM + 1
      IN(JELM,1) = IN(N1,1)
      IN(JELM,2) = IN(N1,2)
      IN(JELM,3) = NND(13)
      IN(JELM,4) = IN(N1,4)
      JELM = JELM + 1
      IN(JELM,1) = IN(N1 + 1,1)
      IN(JELM,2) = NND(6)
```

```
      IN(JELM,3) = NND(13)
      IN(JELM,4) = IN(N1 + 1,4)
      JELM = JELM + 1
      IN(N1 - 3,3) = NND(6)
      IN(N1 - 2,2) = NND(15)
      IN(N1 - 1,1) = NND(7)
      IN(N1 - 1,2) = NND(6)
      IN(N1 - 1,4) = NND(15)
      IN(N1,1) = NND(6)
      IN(N1,3) = NND(15)
      IN(N1,4) = NND(14)
      IN(N1 + 1,1) = NND(6)
      IN(N1 + 1,3) = NND(15)
      GO TO 490
470   IF (ID .GT. 0) GO TO 480
      IN(JELM,1) = IN(N1,1)
      IN(JELM,2) = IN(N1 - 1,2)
      IN(JELM,3) = NND(8)
      JELM = JELM + 1
      IN(JELM,1) = NND(8)
      IN(JELM,2) = IN(N1 - 1,2)
      IN(JELM,3) = IN(N1,2)
      JELM = JELM + 1
      IN(N1 - 1,1) = NND(1)
      IN(N1 - 1,2) = NND(8)
      IN(N1,2) = NND(8)
      IN(N1,3) = NND(7)
      GO TO 490
480   IN(JELM,1) = IN(N1 - 3,1)
      IN(JELM,2) = IN(N1 - 3,2)
      IN(JELM,3) = NND(8)
      IN(JELM,4) = IN(N1 - 3,4)
      JELM = JELM + 1
      IN(JELM,1) = IN(N1 - 2,1)
      IN(JELM,2) = NND(9)
      IN(JELM,3) = IN(N1 - 2,3)
      IN(JELM,4) = IN(N1 - 2,4)
      JELM = JELM + 1
      IN(JELM,1) = NND(1)
      IN(JELM,2) = NND(8)
      IN(JELM,3) = IN(N1 - 1,3)
      IN(JELM,4) = NND(9)
      JELM = JELM + 1
      IN(JELM,1) = NND(8)
      IN(JELM,2) = IN(N1,2)
      IN(JELM,3) = NND(9)
      IN(JELM,4) = NND(16)
      JELM = JELM + 1
      IN(JELM,1) = NND(8)
      IN(JELM,2) = IN(N1 + 1,2)
```

```
            IN(JELM,3) = NND(9)
            IN(JELM,4) = IN(N1 + 1,4)
            JELM = JELM + 1
            IN(N1 - 3,2) = NND(8)
            IN(N1 - 2,1) = NND(8)
            IN(N1 - 2,2) = NND(15)
            IN(N1 - 2,4) = NND(16)
            IN(N1 - 1,1) = NND(7)
            IN(N1 - 1,3) = NND(8)
            IN(N1 - 1,4) = NND(15)
            IN(N1,3) = NND(15)
            IN(N1 + 1,2) = NND(8)
            IN(N1 + 1,3) = NND(15)
  490 CONTINUE
      IF (ID .LT. 1) GO TO 520
      DO 500 I = 1, 8
        ND(I + 8) = ND(I)
        NND(I + 8) = NND(I)
  500 CONTINUE
      GO TO 520
C ************************************************************
  510 CALL CNCTT2(I, IM, IMM)
C ************************************************************
  520 CONTINUE
      NPTS = NPTS + II - 1
      II = 1
      NELM = JELM - 1
      RETURN
      END
C
C
C ..............................................................
C
C   Subroutine CNCTT2
C
C   Author
C       G.J. Smith(U. of Alberta)
C
C   Last Update
C       May 1984
C
C   Purpose
C       To concentrate triangular and tetrahedral
C       elements around a pumping well.
C
C   Usage
C       CALL CNCTT2(I,IM,IMM)
C
C   Subroutines Required
C       None
```

```
C
C
C ...........................................................
      SUBROUTINE CNCTT2(I, IM, IMM)
      COMMON MEMJT(5000,32), IN(3000,8), IFACE(3500,6), YN(5000),
     1       ZN(5000)
      COMMON CX(5000)
      COMMON XN(5000), JMEM(5000), NBDND(4000), NDX(5000), NDY(5000)
      COMMON NDZ(5000), ITYPE(5000), NDWEL(20,2,25), LEAK(100),
     1       ALEAK(100)
      COMMON XIN(100), YIN(100), ISAVE(25,2), PPOSX(10), PPOSY(10),
     1       PPOSZ(10)
      COMMON OBSX(10), OBSY(10), OBSZ(10), ND(16), NND(16)
      COMMON COX(10), COY(10), COZ(10), IFE(6), NPMP(25), IPPN(10)
      COMMON ICRNR(10,8), NFACE(6,4), ICNTST(4,2), NODE1(8)
      COMMON IP, NPTS, NPTS1, NELM, IO, ITRNS, INC, IELEM, NZ, ID, YC,
     1       XC, IPN
      COMMON DELX, DELY, SCL, IW1, IW2, IK, NBN, NPP, NXSTPS, NYSTPS,
     1       DELTZ
      JELM = NELM + 1
   10 FORMAT (3I5)
      K = 1
      II = 1
   20 FORMAT (5I5)
      DO 50 J = 1, NELM
        DO 40 L = 1, INC
          IF (I .EQ. IN(J,L)) GO TO 30
          GO TO 40
   30     ISAVE(K,1) = J
          ISAVE(K,2) = L
          K = K + 1
          GO TO 50
   40   CONTINUE
   50 CONTINUE
      KK = K - 1
      TEMP = SQRT(DELX**2. + DELY**2.)
      R = 0.8 * TEMP
      R1 = 0.325 * TEMP
      K = 1
      DO 240 K1 = 1, KK, 2
        IF (ID .LT. 1) CX(NPTS + II) = COX(ITRNS)
        IF (ID .LT. 1) CX(NPTS + II + 1) = COX(ITRNS)
        N1 = ISAVE(K1,1)
        N2 = ISAVE(K1,2)
        IF (ID .GT. 0) GO TO 60
        GO TO (70, 110, 150), N2
   60   IF (N2 .NE. 4) GO TO 240
        IF (IN(N1,1) .NE. NDWEL(IM,2,IMM + 1)) GO TO 240
        IF (ABS(XN(IN(N1,4)) - XN(IN(N1,3))) .GT. ABS(YN(IN(N1,4)) - YN(
     1     IN(N1,3)))) .A ND .XN(IN(N1,3)) .GT. XN(IN(N1,4))) GOTO70
```

```
      IF (ABS(YN(IN(N1,4)) - YN(IN(N1,3))) .GT. ABS(XN(IN(N1,4)) - XN(
     1   IN(N1,3)))) .A ND .YN(IN(N1,3)) .GT. YN(IN(N1,4))) GOTO110
      IF (ABS(XN(IN(N1,4)) - XN(IN(N1,3))) .GT. ABS(YN(IN(N1,4)) - YN(
     1   IN(N1,3)))) .A ND .XN(IN(N1,3)) .LT. XN(IN(N1,4))) GOTO150
      IF (ABS(YN(IN(N1,4)) - YN(IN(N1,3))) .GT. ABS(XN(IN(N1,4)) - YN(
     1   IN(N1,3)))) .A ND .YN(IN(N1,3)) .GT. YN(IN(N1,4))) GOTO190
   70 IF (ID .LT. 1 .AND. ISAVE(K1 - 1,2) .NE. 1) GO TO 240
      NM1 = 1
      NM2 = 2
      IMM1 = IMM
      NIN = IN(N1,2)
      GO TO 90
   80 IF (IMM .GT. 2) GO TO 240
      NM1 = 9
      NM2 = 10
      IMM1 = IMM - 1
      IF (ID .LT. 1) IMM1 = IMM
      NIN = NDZ(IN(N1,2))
   90 DO 100 NM = NM1, NM2
C ************************************************************
        CALL INTANG(ANGLE, NDWEL(IM,2,IMM), TEMP, NIN, THETA)
C ************************************************************
        CALL NDEGEN(NDWEL(IM,2,IMM1), II, ANGLE, R, NPTS, THETA,
     1         NND(NM))
C ************************************************************
        II = II + 1
C ************************************************************
        CALL NDEGEN(NDWEL(IM,2,IMM1), II, ANGLE, R1, THETA, ND(NM))
C ************************************************************
        II = II + 1
        NIN = NDX(NIN)
  100 CONTINUE
      IF (ID .GT. 0 .AND. NM .LT. 5) GO TO 80
      GO TO 240
  110 IF (ID .LT. 1 .AND. ISAVE(K1 - 1,2) .NE. 2) GO TO 240
      NM1 = 3
      NM2 = 4
      IMM1 = IMM
      NIN = IN(N1,2)
      GO TO 130
  120 IF (IMM .GT. 2) GO TO 240
      NM1 = 11
      NM2 = 12
      IMM1 = IMM - 1
      IF (ID .LT. 1) IMM1 = IMM
      NIN = [illegible](IN(N1,2))
  130 DO 140 NM = NM1, NM2
C ************************************************************
        CALL INTANG(ANGLE, NDWEL(IM,2,IMM), TEMP, NIN, THETA)
C ************************************************************
```

```
          CALL NDEGEN(NDWEL(IM,2,IMM1), II, ANGLE, R, THETA, NND(NM))
C ***********************************************************
          II = II + 1
C ***********************************************************
          CALL NDEGEN(NDWEL(IM,2,IMM1), II, ANGLE, R1, THETA, ND(NM))
C ***********************************************************
          II = II + 1
          NIN = NDY(NIN)
  140   CONTINUE
        IF (ID .GT. 0 .AND. NM .LT. 6) GO TO 120
        GO TO 240
  150   IF (ID .LT. 1 .AND. ISAVE(K1 + 1,2) .NE. 2) GO TO 190
        NM1 = 5
        NM2 = 6
        IMM1 = IMM
        NIN = IN(N1,2)
        GO TO 170
  160   IF (IMM .GT. 2) GO TO 240
        NM1 = 13
        NM2 = 14
        NIN = NDZ(IN(N1,2))
        IMM1 = IMM - 1
        IF (ID .LT. 1) IMM1 = IMM
  170   DO 180 NM = NM1, NM2
C ***********************************************************
          CALL INTANG(ANGLE, NDWEL(IM,2,IMM), TEMP, NIN, THETA)
C ***********************************************************
          CALL NDEGEN(NDWEL(IM,2,IMM1), II, ANGLE, R, NPTS, THETA,
     1        NND(NM), ID)
C ***********************************************************
          II = II + 1
C ***********************************************************
          CALL NDEGEN(NDWEL(IM,2,IMM1), II, ANGLE, R1, THETA, ND(NM))
C ***********************************************************
          II = II + 1
          NIN = 1
  180   CONTINUE
        GO TO 240
  190   NM1 = 7
        NM2 = 8
        IMM1 = IMM
        NIN = IN(N1,2)
        GO TO 210
  200   IF (IMM .GT. 2) GO TO 240
        NM1 = 15
        NM2 = 16
        IMM1 = IMM - 1
        IF (ID .LT. 1) IMM1 = IMM
        NIN = NDZ(IN(N1,2))
  210   DO 230 NM = NM1, NM2
```

```
C *********************************************************
          CALL INTANG(ANGLE, NDWEL(IM,2,IMM), TEMP, NIN, THETA)
C *********************************************************
          CALL NDEGEN(NDWEL(IM,2,IMM1), II, ANGLE, R, THETA, NND(NM))
C *********************************************************
          II = II + 1
C *********************************************************
          CALL NDEGEN(NDWEL(IM,2,IMM1), II, ANGLE, R1, THETA, ND(NM))
C *********************************************************
          II = II + 1
          DO 220 IU = 1, NPTS
            IF (NDY(IU) .NE. NIN) GO TO 220
            NIN = IU
            GO TO 230
  220     CONTINUE
  230   CONTINUE
        IF (ID .GT. Q .AND. NM .LT. 10) GO TO 200
  240 CONTINUE
      IMM1 = IMM
      DO 270 NM = 1, 5, 2
        IN(JELM,1) = NDWEL(IM,2,IMM)
        IN(JELM,2) = ND(NM)
        IN(JELM,3) = ND(NM + 1)
        IF (ID .LT. 1) GO TO 250
        IN(JELM,4) = NND(NM + 10)
  250   CONTINUE
        JELM = JELM + 1
        IN(JELM,1) = NDWEL(IM,2,IMM)
        IN(JELM,2) = ND(NM + 1)
        IN(JELM,3) = ND(NM + 2)
        IF (ID .LT. 1) GO TO 260
        IN(JELM,4) = ND(NM + 10)
  260   CONTINUE
        JELM = JELM + 1
        IF (ID .LT. 1) GO TO 270
        IN(JELM,1) = ND(NM + 1)
        IN(JELM,2) = ND(NM + 10)
        IN(JELM,3) = ND(NM + 8)
        IN(JELM,4) = ND(NM + 9)
        JELM = JELM + 1
        IN(JELM,1) = NDWEL(IM,2,IMM)
        IN(JELM,2) = ND(NM + 8)
        IN(JELM,3) = ND(NM + 10)
        IN(JELM,4) = NDWEL(IM,2,IMM1)
        JELM = JELM + 1
        IN(JELM,1) = NDWEL(IM,2,IMM)
        IN(JELM,2) = ND(NM + 1)
        IN(JELM,3) = ND(NM + 10)
        IN(JELM,4) = ND(NM + 8)
        JELM = JELM + 1
```

```
  270 CONTINUE
      IN(JELM,1) = ND(1)
      IN(JELM,2) = NDWEL(IM,2,IMM)
      IN(JELM,3) = ND(8)
      IF (ID .LT. 1) GO TO 280
      IN(JELM,4) = ND(9)
  280 CONTINUE
      JELM = JELM + 1
      IN(JELM,1) = ND(8)
      IN(JELM,2) = NDWEL(IM,2,IMM)
      IN(JELM,3) = ND(7)
      IF (ID .LT. 1) GO TO 290
      IN(JELM,4) = ND(16)
  290 CONTINUE
      JELM = JELM + 1
      DO 350 NM = 1, 5, 2
        IN(JELM,1) = ND(NM)
        IN(JELM,2) = NND(NM)
        IN(JELM,3) = ND(NM + 1)
        IF (ID .LT. 1) GO TO 300
        IN(JELM,4) = ND(NM + 8)
  300   CONTINUE
        JELM = JELM + 1
        IN(JELM,1) = NND(NM + 1)
        IN(JELM,2) = ND(NM + 1)
        IN(JELM,3) = NND(NM)
        IF (ID .LT. 1) GO TO 310
        IN(JELM,4) = NND(NM + 9)
  310   CONTINUE
        JELM = JELM + 1
        IF (ID .LT. 1) GO TO 320
        IN(JELM,1) = NND(NM)
        IN(JELM,2) = NND(NM + 9)
        IN(JELM,3) = ND(NM + 8)
        IN(JELM,4) = NND(NM + 8)
        JELM = JELM + 1
        IN(JELM,1) = ND(NM + 1)
        IN(JELM,2) = ND(NM + 8)
        IN(JELM,3) = NND(NM + 9)
        IN(JELM,4) = ND(NM + 9)
        JELM = JELM + 1
        IN(JELM,1) = ND(NM + 1)
        IN(JELM,2) = NND(NM)
        IN(JELM,3) = NND(NM + 9)
        IN(JELM,4) = ND(NM + 8)
        JELM = JELM + 1
  320   CONTINUE
        IN(JELM,1) = ND(NM + 2)
        IN(JELM,2) = ND(NM + 1)
        IN(JELM,3) = NND(NM + 2)
```

```
      IF (ID .LT. 1) GO TO 330
      IN(JELM,4) = ND(NM + 10)
 330  CONTINUE
      JELM = JELM + 1
      IN(JELM,1) = NND(NM + 1)
      IN(JELM,2) = NND(NM + 2)
      IN(JELM,3) = ND(NM + 2)
      IF (ID .LT. 1) GO TO 340
      IN(JELM,4) = NND(NM + 9)
 340  CONTINUE
      JELM = JELM + 1
      IF (ID .LT. 1) GO TO 350
      IN(JELM,1) = ND(NM + 1)
      IN(JELM,2) = NND(NM + 9)
      IN(JELM,3) = ND(NM + 10)
      IN(JELM,4) = ND(NM + 9)
      JELM = JELM + 1
      IN(JELM,1) = NND(NM + 2)
      IN(JELM,2) = ND(NM + 10)
      IN(JELM,3) = NND(NM + 9)
      IN(JELM,4) = NND(NM + 10)
      JELM = JELM + 1
      IN(JELM,1) = ND(NM + 1)
      IN(JELM,2) = ND(NM + 2)
      IN(JELM,3) = ND(NM + 10)
      IN(JELM,4) = NND(NM + 9)
      JELM = JELM + 1
350 CONTINUE
    IN(JELM,1) = ND(7)
    IN(JELM,2) = NND(7)
    IN(JELM,3) = ND(8)
    IF (ID .LT. 1) GO TO 360
    IN(JELM,4) = ND(15)
360 CONTINUE
    JELM = JELM + 1
    IN(JELM,1) = NND(8)
    IN(JELM,2) = ND(8)
    IN(JELM,3) = NND(7)
    IF (ID .LT. 1) GO TO 370
    IN(JELM,4) = NND(16)
370 CONTINUE
    JELM = JELM + 1
    IN(JELM,1) = ND(1)
    IN(JELM,2) = ND(8)
    IN(JELM,3) = NND(1)
    IF (ID .LT. 1) GO TO 380
    IN(JELM,4) = ND(9)
380 CONTINUE
    JELM = JELM + 1
    IN(JELM,1) = NND(8)
```

```
      IN(JELM,2) = NND(1)
      IN(JELM,3) = ND(8)
      IF (ID .LT. 1) GO TO 390
      IN(JELM,4) = NND(16)
  390 CONTINUE
      JELM = JELM + 1
      DO 490 K1 = 1, KK, 2
        N1 = ISAVE(K1,1)
        N2 = ISAVE(K1,2)
        IF (ID .LT. 1) GO TO 400
        GO TO (410, 430, 450), N2
  400   IF (N2 .NE. 4) GO TO 240
        IF (IN(N1,1) .NE. NDWEL(IM,2,IMM + 1)) GO TO 240
        IF (ABS(XN(IN(N1,4)) - XN(IN(N1,3))) .GT. ABS(YN(IN(N1,4)) - YN(
     1     IN(N1,3)))) .A ND .XN(IN(N1,3)) .GT. XN(IN(N1,4))) GOTO410
        IF (ABS(YN(IN(N1,4)) - YN(IN(N1,3))) .GT. ABS(XN(IN(N1,4)) - XN(
     1     IN(N1,3)))) .A ND .YN(IN(N1,3)) .GT. YN(IN(N1,4))) GOTO430
        IF (ABS(XN(IN(N1,4)) - XN(IN(N1,3))) .GT. ABS(YN(IN(N1,4)) - YN(
     1     IN(N1,3)))) .A ND .XN(IN(N1,3)) .LT. XN(IN(N1,4))) GOTO450
        IF (ABS(YN(IN(N1,4)) - YN(IN(N1,3))) .GT. ABS(XN(IN(N1,4)) - YN(
     1     IN(N1,3)))) .A ND .YN(IN(N1,3)) .GT. YN(IN(N1,4))) GOTO470
  410   IF (ID .GT. 0) GO TO 420
        IF (ISAVE(K1 - 1,2) .NE. 1) GO TO 490
        IN(JELM,1) = NND(1)
        IN(JELM,2) = IN(N1,3)
        IN(JELM,3) = NND(2)
        JELM = JELM + 1
        IN(JELM,1) = NND(2)
        IN(JELM,2) = IN(N1,3)
        IN(JELM,3) = NND(3)
        JELM = JELM + 1
        IN(N1,1) = NND(1)
        IN(N1 + 1,1) = NND(3)
        GO TO 490
  420   IN(JELM,1) = IN(N1 - 3,1)
        IN(JELM,2) = IN(N1 - 3,2)
        IN(JELM,3) = NND(1)
        IN(JELM,4) = IN(N1 - 3,4)
        JELM = JELM + 1
        IN(JELM,1) = IN(N1 - 2,1)
        IN(JELM,2) = NND(10)
        IN(JELM,3) = IN(N1 - 2,3)
        IN(JELM,4) = IN(N1 - 2,4)
        JELM = JELM + 1
        IN(JELM,1) = NND(2)
        IN(JELM,2) = NND(1)
        IN(JELM,3) = IN(N1 - 1,3)
        IN(JELM,4) = NND(10)
        JELM = JELM + 1
        IN(JELM,1) = NND(1)
```

```
      IN(JELM,2) = IN(N1,2)
      IN(JELM,3) = NND(10)
      IN(JELM,4) = NND(9)
      JELM = JELM + 1
      IN(JELM,1) = NND(1)
      IN(JELM,2) = IN(N1 + 1,2)
      IN(JELM,3) = NND(10)
      IN(JELM,4) = IN(N1 + 1,4)
      JELM = JELM + 1
      IN(N1 - 3,1) = NND(2)
      IN(N1 - 3,3) = NND(3)
      IN(N1 - 3,4) = NND(10)
      IN(N1 - 2,3) = NND(10)
      IN(N1 - 1,2) = NND(3)
      IN(N1,1) = NND(3)
      IN(N1,2) = NND(10)
      IN(N1,4) = NND(11)
      IN(N1 + 1,1) = NND(3)
      IN(N1 + 1,4) = NND(10)
      GO TO 490
  430 IF (ID .GT. 0) GO TO 440
      IF (ISAVE(K1 - 1,2) .NE. 2) GO TO 490
      IN(JELM,1) = IN(N1,3)
      IN(JELM,2) = NND(5)
      IN(JELM,3) = NND(4)
      JELM = JELM + 1
      IN(JELM,1) = IN(N1,3)
      IN(JELM,2) = NND(4)
      IN(JELM,3) = NND(3)
      JELM = JELM + 1
      IN(N1,2) = NND(5)
      IN(N1 + 1,2) = NND(3)
      GO TO 490
  440 IN(JELM,1) = NND(4)
      IN(JELM,2) = NND(3)
      IN(JELM,3) = IN(N1 - 1,3)
      IN(JELM,4) = NND(12)
      JELM = JELM + 1
      IN(JELM,1) = NND(3)
      IN(JELM,2) = IN(N1,2)
      IN(JELM,3) = NND(12)
      IN(JELM,4) = NND(11)
      JELM = JELM + 1
      IN(JELM,1) = IN(N1 + 1,1)
      IN(JELM,2) = IN(N1 + 1,2)
      IN(JELM,3) = NND(3)
      IN(JELM,4) = IN(N1 + 1,4)
      JELM = JELM + 1
      IN(JELM,1) = IN(N1 + 2,1)
      IN(JELM,2) = NND(12)
```

```
      IN(JELM,3) = IN(N1 + 2,3)
      IN(JELM,4) = IN(N1 + 2,4)
      JELM = JELM + 1
      IN(JELM,1) = IN(N1 + 3,1)
      IN(JELM,2) = NND(3)
      IN(JELM,3) = IN(N1 + 3,3)
      IN(JELM,4) = NND(12)
      JELM = JELM + 1
      IN(N1 - 1,2) = NND(5)
      IN(N1,1) = NND(5)
      IN(N1,2) = NND(12)
      IN(N1,4) = NND(13)
      IN(N1 + 1,1) = NND(4)
      IN(N1 + 1,3) = NND(5)
      IN(N1 + 1,4) = NND(12)
      IN(N1 + 2,3) = NND(12)
      IN(N1 + 3,2) = NND(5)
      IN(N1 + 3,3) = NND(12)
      GO TO 490
450   IF (ID .GT. 0) GO TO 460
      IF (ISAVE(K1 + 1,2) .NE. 2) GO TO 470
      IN(JELM,1) = IN(N1,1)
      IN(JELM,2) = NND(6)
      IN(JELM,3) = NND(5)
      JELM = JELM + 1
      IN(JELM,1) = IN(N1,1)
      IN(JELM,2) = NND(7)
      IN(JELM,3) = NND(6)
      JELM = JELM + 1
      IN(N1,3) = NND(7)
      IN(N1 + 1,2) = NND(5)
      GO TO 490
460   IN(JELM,1) = NND(6)
      IN(JELM,2) = NND(5)
      IN(JELM,3) = IN(N1 - 1,3)
      IN(JELM,4) = NND(14)
      JELM = JELM + 1
      IN(JELM,1) = NND(5)
      IN(JELM,2) = IN(N1,2)
      IN(JELM,3) = NND(14)
      IN(JELM,4) = NND(13)
      JELM = JELM + 1
      IN(JELM,1) = IN(N1 + 1,1)
      IN(JELM,2) = IN(N1 + 1,2)
      IN(JELM,3) = NND(5)
      IN(JELM,4) = IN(N1 + 1,4)
      JELM = JELM + 1
      IN(JELM,1) = IN(N1 + 2,1)
      IN(JELM,2) = NND(14)
      IN(JELM,3) = IN(N1 + 2,3)
```

```
      IN(JELM,4) = IN(N1 + 2,4)
      JELM = JELM + 1
      IN(JELM,1) = IN(N1 + 3,1)
      IN(JELM,2) = NND(5)
      IN(JELM,3) = IN(N1 + 3,3)
      IN(JELM,4) = NND(14)
      JELM = JELM + 1
      IN(N1 - 1,2) = NND(7)
      IN(N1,1) = NND(7)
      IN(N1,2) = NND(14)
      IN(N1,4) = NND(15)
      IN(N1 + 1,1) = NND(6)
      IN(N1 + 1,3) = NND(7)
      IN(N1 + 1,4) = NND(14)
      IN(N1 + 2,3) = NND(14)
      IN(N1 + 3,2) = NND(7)
      IN(N1 + 3,3) = NND(14)
      GO TO 490
  470 IF (ID .GT. O) GO TO 480
      IN(JELM,1) = NND(7)
      IN(JELM,2) = IN(N1,2)
      IN(JELM,3) = NND(8)
      JELM = JELM + 1
      IN(JELM,1) = NND(8)
      IN(JELM,2) = IN(N1,2)
      IN(JELM,3) = NND(15)
      JELM = JELM + 1
      IN(N1,3) = NND(7)
      IN(N1 + 1,1) = NND(1)
      GO TO 490
  480 IN(JELM,1) = IN(N1 - 3,1)
      IN(JELM,2) = IN(N1 - 3,2)
      IN(JELM,3) = NND(7)
      IN(JELM,4) = IN(N1 - 3,4)
      JELM = JELM + 1
      IN(JELM,1) = IN(N1 - 2,1)
      IN(JELM,2) = NND(16)
      IN(JELM,3) = IN(N1 - 2,3)
      IN(JELM,4) = IN(N1 - 2,4)
      JELM = JELM + 1
      IN(JELM,1) = NND(8)
      IN(JELM,2) = NND(7)
      IN(JELM,3) = IN(N1 - 1,3)
      IN(JELM,4) = NND(16)
      JELM = JELM + 1
      IN(JELM,1) = NND(7)
      IN(JELM,2) = IN(N1,2)
      IN(JELM,3) = NND(16)
      IN(JELM,4) = NND(15)
      JELM = JELM + 1
```

```
          IN(JELM,1) = NND(7)
          IN(JELM,2) = IN(N1 + 1,2)
          IN(JELM,3) = NND(16)
          IN(JELM,4) = IN(N1 + 1,4)
          JELM = JELM + 1
          IN(N1 - 3,1) = NND(8)
          IN(N1 - 3,3) = NND(1)
          IN(N1 - 3,4) = NND(16)
          IN(N1 - 2,3) = NND(16)
          IN(N1 - 1,2) = NND(1)
          IN(N1,1) = NND(1)
          IN(N1,2) = NND(16)
          IN(N1,4) = NND(9)
          IN(N1 + 1,1) = NND(1)
          IN(N1 + 1,4) = NND(16)
  490 CONTINUE
      IF (ID .LT. 1) GO TO 510
      DO 500 I = 1, 8
        ND(I + 8) = ND(I)
        NND(I + 8) = NND(I)
  500 CONTINUE
  510 CONTINUE
      NPTS = NPTS + II - 1
      II = 1
      NELM = JELM - 1
      RETURN
      END
```

APPENDIX C: Results of a Two-Dimensional Simulation

2-D MESH.GEN SIMULATION

NUMBER OF NODES 160
NUMBER OF ELEMENTS 272
ESTIMATED BANDWIDTH 21

NODE COORDINATES

NODE	X	Y	TYPE	TRANSMISSIVITY
1	7188.41	10223.19		0.0129
2	6249.98	13337.74		0.0129
3	10299.81	8418.63		0.0129
4	9358.60	11822.29		0.0129
5	7259.58	16452.30		0.0129
6	12467.23	11964.16		0.0129
7	13411.20	8858.03		0.0129
8	9894.37	15225.95		0.0129
9	12529.16	15070.29		0.0129
10	8350.11	19566.85		0.0129
11	16522.60	9659.46		0.0129
12	15575.86	12199.66		0.0129
13	10590.95	18629.61		0.0129
14	12831.79	18176.42		0.0129
15	15163.94	14739.87		0.0129
16	6181.35	22681.41		0.0129
17	18684.49	12778.97		0.0129
18	19955.87	10554.31		0.0258
19	17798.73	15003.66		0.0129
20	8939.35	22033.27		0.0129
21	11697.35	21282.56		0.0129
22	15072.63	17280.08		0.0129
23	17313.48	17228.34		0.0129
24	4244.11	25795.96		0.0129
25	21363.51	13146.84		0.0258
26	21984.61	11163.32		0.0258
27	20857.52	15391.02		0.0258
28	20545.40	17767.23		0.0258
29	9544.11	27361.77		0.0129
30	10864.82	24388.69		0.0129
31	14455.35	19820.28		0.0129
32	17213.35	19453.02		0.0129
33	3591.81	28910.52		0.0129
34	24013.36	11348.44		0.0258
35	23855.88	13325.26		0.0258
36	23757.84	15609.91		0.0258
37	23644.14	18175.10		0.0258
38	20500.91	20161.02		0.0258
39	23387.02	21267.88		0.0258

40	7227.63	28840.59	0.0258
41	10863.45	27494.82	0.0258
42	14175.17	22360.49	0.0129
43	17485.53	21677.71	0.0129
44	20867.76	22258.47	0.0258
45	4315.79	32025.07	0.0129
46	26324.20	13159.38	0.0258
47	26042.10	10876.24	0.0258
48	26616.01	15610.16	0.0258
49	26950.79	18271.25	0.0258
50	27262.17	20943.54	0.0258
51	23973.27	22829.91	0.0258
52	27331.37	23213.26	0.0258
53	8005.41	32244.25	0.0129
54	11695.02	30600.95	0.0129
55	14499.27	24900.70	0.0129
56	18135.10	23902.39	0.0129
57	21305.45	24271.19	0.0258
58	24311.75	24752.32	0.0258
59	7617.56	35139.63	0.0258
60	10889.89	35647.92	0.0129
61	28070.85	10162.00	0.0258
62	28761.99	12789.68	0.0258
63	29482.92	15515.75	0.0258
64	30267.54	18301.15	0.0258
65	31462.40	21131.85	0.0258
66	30740.48	23590.60	0.0258
67	27292.56	25374.36	0.0258
68	30249.75	26193.65	0.0258
69	14054.93	33707.08	0.0129
70	15384.64	27440.91	0.0129
71	17219.98	29981.12	0.0129
72	19074.26	26127.07	0.0129
73	21912.01	26343.88	0.0258
74	24588.02	26816.16	0.0258
75	27241.14	27646.80	0.0258
76	31167.14	12323.90	0.0258
77	30099.60	9483.86	0.0258
78	32234.43	15309.57	0.0258
79	33360.42	18481.51	0.0258
80	33318.84	21169.92	0.0258
81	33898.13	23975.45	0.0258
82	33165.87	27061.98	0.0258
83	29878.95	28746.97	0.0258
84	32503.95	29941.33	0.0258
85	20385.03	28351.76	0.0129
86	22628.35	28149.02	0.0258
87	24871.68	28712.74	0.0258
88	27115.00	29711.13	0.0258
89	29358.33	31252.10	0.0258

90	32128.34	8506.86	0.0258
91	33569.63	11796.49	0.0258
92	34890.98	15022.41	0.0258
93	35798.70	17805.59	0.0258
94	34819.30	21338.53	0.0258
95	36384.81	19533.96	0.0258
96	34969.76	19929.60	0.0258
97	36841.91	23401.88	0.0258
98	36500.46	25069.06	0.0258
99	36074.18	27889.37	0.0258
100	35305.93	22839.99	0.0258
101	35125.77	31015.34	0.0258
102	31601.66	33004.20	0.0258
103	33844.99	34307.98	0.0258
104	35952.28	11352.43	0.0258
105	34157.09	7693.09	0.0258
106	37591.07	14856.54	0.0258
107	38669.75	18484.30	0.0258
108	36023.16	22163.21	0.0258
109	35801.34	21380.29	0.0258
110	35874.18	20668.94	0.0258
111	36618.36	20502.21	0.0258
112	37319.96	20592.63	0.0258
113	37799.95	19859.51	0.0258
114	39385.82	24731.88	0.0258
115	37425.82	22065.20	0.0258
116	36802.19	22324.44	0.0258
117	38400.97	23122.17	0.0258
118	39034.49	28494.37	0.0258
119	37748.10	31879.06	0.0258
120	36088.32	35434.13	0.0258
121	36185.84	7240.91	0.0258
122	38433.77	11061.90	0.0258
123	40363.65	14705.58	0.0258
124	41779.31	18219.23	0.0258
125	38558.93	21478.49	0.0258
126	39882.98	21531.49	0.0258
127	42549.38	21749.37	0.0258
128	36688.13	21388.36	0.0258
129	37640.23	21410.09	0.0258
130	42701.86	25388.77	0.0258
131	41885.67	28927.87	0.0258
132	40371.48	32521.27	0.0258
133	38331.65	36276.93	0.0258
134	40899.41	11035.02	0.0258
135	38214.58	7146.67	0.0258
136	43277.66	14707.91	0.0258
137	44820.54	18345.17	0.0258
138	45766.28	21965.98	0.0258
139	45749.89	25584.13	0.0258

140	44911.98	29232.55	0.0258
141	43021.42	32921.56	0.0258
142	40574.97	36782.93	0.0258
143	40243.33	7362.59	0.0258
144	43720.77	11231.36	0.0258
145	46315.02	14920.78	0.0258
146	48154.61	18538.38	0.0258
147	49129.88	22143.05	0.0258
148	49269.19	25757.88	0.0258
149	48111.85	29365.49	0.0258
150	45905.13	33062.23	0.0258
151	42818.30	36966.16	0.0258
152	47059.22	11692.51	0.0258
153	42272.10	8153.72	0.0258
154	49874.09	15231.31	0.0258
155	51840.77	18770.10	0.0258
156	53092.34	22308.89	0.0258
157	53370.24	25847.68	0.0258
158	52133.61	29386.48	0.0258
159	49382.28	32925.27	0.0258
160	45061.63	36464.09	0.0258

ELEMENT DATA

ELEMENT	INCIDENCE 1	2	3	LEAKANCE
1	17	18	25	0.0
2	25	18	26	0.0
3	25	26	34	0.0
4	25	34	35	0.0
5	35	34	46	0.0
6	46	34	47	0.0
7	46	47	61	0.0
8	46	61	62	0.0
9	62	61	76	0.0
10	76	61	77	0.0
11	76	77	90	0.0
12	76	90	91	0.0
13	91	90	104	0.0
14	104	90	105	0.0
15	104	105	121	0.0
16	104	121	122	0.0
17	122	121	134	0.0
18	134	121	135	0.0
19	134	135	143	0.0
20	134	143	144	0.0
21	144	143	152	0.0
22	152	143	153	0.0
23	19	17	25	0.0

24	19	25	27	0.0
25	27	25	36	0.0
26	36	25	35	0.0
27	36	35	46	0.0
28	36	46	48	0.0
29	48	46	63	0.0
30	63	46	62	0.0
31	63	62	76	0.0
32	63	78	78	0.0
33	78	76	92	0.0
34	92	79	91	0.0
35	92	91	104	0.0
36	92	104	106	0.0
37	106	104	123	0.0
38	123	104	122	0.0
39	123	122	134	0.0
40	123	134	136	0.0
41	136	134	145	0.0
42	145	134	144	0.0
43	145	144	152	0.0
44	145	152	154	0.0
45	23	19	28	0.0
46	28	19	27	0.0
47	28	27	36	0.0
48	28	36	37	0.0
49	37	36	49	0.0
50	49	36	48	0.0
51	49	48	63	0.0
52	49	63	64	0.0
53	64	63	79	0.0
54	79	63	78	0.0
55	79	78	92	0.0
56	79	92	93	0.0
57	93	92	107	0.0
58	107	92	106	0.0
59	107	106	123	0.0
60	107	123	124	0.0
61	124	123	137	0.0
62	137	123	136	0.0
63	137	136	145	0.0
64	137	145	146	0.0
65	146	145	155	0.0
66	155	145	154	0.0
67	32	23	28	0.0
68	32	28	38	0.0
69	38	28	39	0.0
70	39	28	37	0.0
71	39	37	49	0.0
72	39	49	50	0.0
73	50	49	65	0.0

74	65	49	64	0.0
75	65	64	79	0.0
76	65	79	80	0.0
77	80	79	94	0.0
78	95	79	93	0.0
79	95	93	107	0.0
80	125	107	126	0.0
81	126	107	127	0.0
82	127	107	124	0.0
83	127	124	137	0.0
84	127	137	138	0.0
85	138	137	147	0.0
86	147	137	146	0.0
87	147	146	155	0.0
88	147	155	156	0.0
89	43	32	44	0.0
90	44	32	38	0.0
91	44	38	39	0.0
92	44	39	51	0.0
93	51	39	52	0.0
94	52	39	50	0.0
95	52	50	65	0.0
96	52	65	66	0.0
97	66	65	81	0.0
98	81	65	80	0.0
99	81	80	94	0.0
100	81	97	98	0.0
101	98	97	114	0.0
102	114	125	126	0.0
103	114	126	127	0.0
104	114	127	130	0.0
105	130	127	139	0.0
106	139	127	138	0.0
107	139	138	147	0.0
108	139	147	148	0.0
109	148	147	157	0.0
110	157	147	156	0.0
111	56	43	44	0.0
112	56	44	57	0.0
113	57	44	[illegible]	0.0
114	58	44	57	0.0
115	58	[illegible]	52	0.0
116	58	[illegible]	67	0.0
117	67	52	68	0.0
118	68	52	66	0.0
119	68	66	81	0.0
120	68	81	82	0.0
121	82	81	99	0.0
122	99	81	98	0.0
123	99	98	114	0.0

124	99	114	118	0.0
125	118	114	131	0.0
126	131	114	130	0.0
127	131	130	139	0.0
128	131	139	140	0.0
129	140	139	149	0.0
130	149	139	148	0.0
131	149	148	157	0.0
132	149	157	158	0.0
133	72	56	73	0.0
134	73	56	57	0.0
135	73	57	58	0.0
136	73	58	74	0.0
137	74	58	75	0.0
138	75	58	67	0.0
139	75	67	68	0.0
140	75	68	83	0.0
141	83	68	84	0.0
142	84	68	82	0.0
143	84	82	99	0.0
144	84	99	101	0.0
145	101	99	119	0.0
146	119	99	118	0.0
147	119	118	131	0.0
148	119	131	132	0.0
149	132	131	141	0.0
150	141	131	140	0.0
151	141	140	149	0.0
152	141	149	150	0.0
153	150	149	159	0.0
154	159	149	158	0.0
155	85	72	73	0.0
156	85	73	86	0.0
157	86	73	87	0.0
158	87	73	74	0.0
159	87	74	75	0.0
160	87	75	88	0.0
161	88	75	89	0.0
162	89	75	83	0.0
163	89	83	84	0.0
164	89	84	102	0.0
165	102	84	103	0.0
166	103	84	101	0.0
167	103	101	119	0.0
168	103	119	120	0.0
169	120	119	133	0.0
170	133	119	132	0.0
171	133	132	141	0.0
172	133	141	142	0.0
173	142	141	151	0.0

174	151	141	150	0.0
175	151	150	159	0.0
176	151	159	160	0.0
177	128	111	112	0.0
178	128	112	129	0.0
179	128	129	115	0.0
180	128	115	116	0.0
181	128	116	108	0.0
182	128	108	109	0.0
183	128	109	110	0.0
184	128	110	111	0.0
185	111	95	112	0.0
186	112	95	113	0.0
187	112	125	129	0.0
188	112	113	125	0.0
189	129	125	115	0.0
190	115	125	117	0.0
191	115	97	116	0.0
192	115	117	97	0.0
193	116	97	108	0.0
194	108	97	100	0.0
195	108	94	109	0.0
196	108	100	94	0.0
197	109	94	110	0.0
198	94	96	110	0.0
199	110	95	111	0.0
200	110	96	95	0.0
201	94	79	96	0.0
202	96	79	95	0.0
203	95	107	113	0.0
204	113	107	125	0.0
205	81	100	97	0.0
206	81	94	100	0.0
207	114	97	117	0.0
208	114	117	125	0.0
209	2	1	3	0.0
210	2	3	4	0.0
211	4	3	6	0.0
212	6	3	7	0.0
213	6	7	11	0.0
214	6	11	12	0.0
215	12	11	17	1.0E-08
216	17	11	18	1.0E-08
217	5	2	4	0.0
218	5	4	8	0.0
219	8	4	9	0.0
220	9	4	6	0.0
221	9	6	12	0.0
222	9	12	15	0.0
223	15	12	19	1.0E-08

	19	12	17	1.0E-08
	10	5	8	0.0
	10	8	13	0.0
	13	8	14	0.0
228	14	8	9	0.0
229	14	9	15	0.0
230	14	15	22	0.0
231	22	15	23	1.0E-08
232	23	15	19	1.0E-08
233	16	10	13	0.0
234	16	13	20	0.0
235	20	13	21	0.0
236	21	13	14	0.0
237	21	14	22	0.0
238	21	22	31	0.0
239	31	22	32	1.0E-08
240	32	22	23	1.0E-08
241	24	16	20	0.0
242	24	20	29	0.0
243	29	20	30	0.0
244	30	20	21	0.0
245	30	21	31	0.0
246	30	31	42	0.0
247	42	31	43	1.0E-08
248	43	31	32	1.0E-08
249	33	24	29	0.0
250	33	29	40	0.0
251	40	29	41	0.0
252	41	29	30	0.0
253	41	30	42	0.0
254	41	42	55	0.0
255	55	42	56	1.0E-08
256	56	42	43	1.0E-08
257	45	33	40	0.0
258	45	40	53	0.0
259	53	40	54	0.0
260	54	40	41	0.0
261	54	41	55	0.0
262	54	55	70	0.0
263	70	55	72	1.0E-08
264	72	55	56	1.0E-08
265	59	45	60	0.0
266	60	45	53	0.0
267	60	53	69	0.0
268	69	53	54	0.0
269	69	54	71	0.0
270	71	54	70	0.0
271	71	70	72	1.0E-08
272	71	72	85	1.0E-08

STORAGE COEFFICIENT 8.0[illegible]04

NUMBER OF CONSTANT-HEAD NODES
NUMBER OF DEGREES OF FREEDOM [illegible]

INTERVAL FROM 0.0 TO 0.0014 DAYS

WELL PW 1 NODE 1 DISCHARGE 1.3380

PIEZOMETRIC SURFACE (NODE AND DRAWDOWN)

TIME 0.0003 DAYS 30. SECONDS

1	0.00000	2	0.00000	3	-0.00000	4	-0.00000	5	0.00000	6	0.00000	7	0.00000	8	-0.00000
9	0.00000	10	0.00000	11	-0.00000	12	-0.00000	13	-0.00000	14	0.00000	15	-0.00000	16	0.00000
17	0.00000	18	0.00000	19	0.00000	20	-0.00000	21	0.00000	22	-0.00000	23	0.00000	24	0.00000
25	-0.00000	26	-0.00000	27	-0.00000	28	-0.00000	29	-0.00000	30	0.00000	31	-0.00000	32	0.00000
33	0.00000	34	0.00000	35	0.00000	36	0.00000	37	0.00000	38	-0.00000	39	0.00001	40	-0.00000
41	0.00000	42	-0.00000	43	0.00000	44	-0.00000	45	0.00000	46	-0.00001	47	0.00000	48	-0.00000
49	-0.00002	50	-0.00001	51	0.00000	52	-0.00002	53	-0.00000	54	0.00000	55	-0.00000	56	0.00000
57	-0.00000	58	0.00000	59	-0.00000	60	-0.00000	61	0.00001	62	-0.00000	63	0.00003	64	0.00004
65	0.00008	66	0.00004	67	-0.00000	68	0.00003	69	0.00000	70	0.00000	71	-0.00000	72	0.00000
73	-0.00000	74	0.00000	75	-0.00001	76	-0.00002	77	0.00001	78	0.00004	79	-0.00026	80	-0.00040
81	-0.00025	82	0.00004	83	-0.00000	84	-0.00002	85	0.00000	86	-0.00000	87	0.00000	88	0.00000
89	0.00001	90	0.00001	91	-0.00001	92	0.00009	93	-0.00032	94	0.00229	95	0.00215	96	0.00071
97	0.00211	98	-0.00030	99	0.00009	100	0.00074	101	-0.00001	102	0.00001	103	0.00001	104	-0.00002
105	0.00000	106	0.00003	107	-0.00022	108	-0.01006	109	-0.02595	110	-0.00996	111	-0.02572	112	-0.01101
113	0.00085	114	-0.00021	115	-0.00869	116	-0.02571	117	0.00087	118	0.00003	119	-0.00002	120	0.00000
121	0.00001	122	-0.00000	123	0.00003	124	0.00003	125	0.00232	126	-0.00030	127	0.00008	128	0.13083
129	-0.02822	130	0.00003	131	0.00002	132	-0.00000	133	0.00001	134	-0.00000	135	0.00000	136	-0.00000
137	-0.00002	138	-0.00001	139	-0.00002	140	-0.00000	141	-0.00000	142	0.00000	143	0.00000	144	0.00000
145	0.00000	146	0.00000	147	0.00001	148	0.00000	149	0.00000	150	0.00000	151	0.00000	152	-0.00000
153	-0.00000	154	-0.00000	155	-0.00000	156	-0.00000	157	-0.00000	158	-0.00000	159	-0.00000	160	-0.00000

TIME 0.0007 DAYS 60. SECONDS

1	0.00000	2	0.00000	3	-0.00000	4	-0.00000	5	0.00000	6	0.00000	7	0.00000	8	-0.00000
9	0.00000	10	0.00000	11	-0.00000	12	-0.00000	13	-0.00000	14	0.00000	15	-0.00000	16	0.00000
17	0.00000	18	0.00000	19	0.00000	20	-0.00000	21	0.00000	22	-0.00000	23	0.00000	24	0.00000
25	-0.00000	26	-0.00000	27	-0.00000	28	-0.00000	29	-0.00000	30	0.00000	31	-0.00000	32	0.00000
33	0.00000	34	0.00000	35	0.00000	36	0.00001	37	0.00000	38	-0.00000	39	0.00001	40	-0.00000

41	0.00000	42	-0.00000	43	0.00000	44	-0.00000	45	0.00000	46	-0.00001	47	0.00000	48	-0.00000
49	-0.00004	50	-0.00002	51	0.00001	52	-0.00004	53	-0.00000	54	0.00000	55	-0.00000	56	0.00000
57	-0.00000	58	0.00001	59	-0.00000	60	-0.00000	61	0.00001	62	-0.00001	63	0.00006	64	0.00008
65	0.00016	66	0.00008	67	-0.00000	68	0.00006	69	0.00000		0.00000	71	-0.00000	72	0.00000
73	-0.00000	74	0.00000	75	-0.00001	76	-0.00004	77	0.00001		0.00008	79	-0.00051	80	-0.00078
81	-0.00050	82	0.00008	83	-0.00001	84	-0.00004	85	0.00000		0.00000	87	0.00000	88	0.00000
89	0.00001	90	0.00003	91	-0.00003	92	0.00018	93	-0.00063	94	0.00450	95	0.00424	96	0.00146
97	0.00415	98	-0.00060	99	0.00017	100	0.00150	101	-0.00003	102	0.00001	103	0.00002	104	-0.00004
105	0.00001	106	0.00006	107	-0.00043	108	-0.02013	109	-0.05075	110	-0.01899	111	-0.05030	112	-0.02202
113	0.00171	114	-0.00041	115	-0.01742	116	-0.05039	117	0.00178	118	0.00006	119	-0.00004	120	0.00001
121	0.00001	122	-0.00000	123	0.00005	124	0.00007	125	0.00456	126	-0.00059	127	0.00015	128	0.25999
129	-0.05529	130	0.00006	131	0.00005	132	-0.00000	133	0.00001	134	-0.00001	135	0.00000	136	-0.00000
137	-0.00003	138	-0.00002	139	-0.00003	140	-0.00000	141	-0.00001	142	0.00000	143	0.00000	144	0.00000
145	0.00001	146	0.00000	147	0.00001	148	0.00000	149	0.00001	150	0.00000	151	0.00000	152	-0.00000
153	-0.00000	154	-0.00000	155	-0.00000	156	-0.00000	157	-0.00000	158	-0.00000	159	-0.00000	160	-0.00000

TIME 0.0010 DAYS 91. SECONDS

1	0.00000	2	0.00000	3	-0.00000	4	-0.00000	5	0.00000	6	0.00000	7	0.00000	8	-0.00000
9	0.00000	10	0.00000	11	-0.00000	12	-0.00000	13	-0.00000	14	0.00000	15	-0.00000	16	0.00000
17	-0.00000	18	0.00000	19	0.00000	20	-0.00000	21	0.00000	22	-0.00000	23	0.00000	24	0.00000
25	-0.00000	26	-0.00000	27	-0.00000	28	-0.00001	29	-0.00000	30	0.00000	31	-0.00000	32	0.00000
33	0.00000	34	0.00000	35	0.00000	36	0.00001	37	0.00001	38	-0.00000	39	0.00002	40	-0.00000
41	0.00000	42	-0.00000	43	0.00000	44	-0.00000	45	0.00000	46	-0.00002	47	0.00000	48	-0.00001
49	-0.00006	50	-0.00004	51	0.00001	52	0.00006	53	-0.00000	54	0.00000	55	-0.00000	56	0.00000
57	-0.00000	58	0.00001	59	-0.00000	60	0.00000	61	0.00002	62	-0.00001	63	0.00009	64	0.00011
65	0.00023	66	0.00011	67	-0.00001	68	0.00009	69	0.00000	70	0.00000	71	-0.00000	72	0.00000
73	-0.00000	74	0.00000	75	-0.00002	76	-0.00007	77	0.00002	78	0.00012	79	-0.00076	80	-0.00114
81	-0.00074	82	0.00012	83	-0.00001	84	-0.00006	85	0.00000	86	-0.00000	87	0.00000	88	0.00000
89	0.00002	90	0.00004	91	-0.00004	92	0.00028	93	-0.00092	94	0.00663	95	0.00624	96	0.00225
97	0.00613	98	-0.00087	99	0.00025	100	0.00228	101	-0.00004	102	0.00002	103	0.00004	104	-0.00006
105	0.00001	106	0.00009	107	-0.00064	108	-0.03019	109	-0.07443	110	-0.03009	111	-0.07376	112	-0.03302
113	0.00257	114	-0.00060	115	-0.02618	116	-0.07406	117	0.00272	118	0.00009	119	-0.00005	120	0.00001
121	0.00002	122	-0.00000	123	0.00007	124	0.00010	125	0.00671	126	-0.00086	127	0.00022	128	0.38752
129	-0.08125	130	0.00010	131	0.00007	132	-0.00000	133	0.00002	134	-0.00001	135	0.00000	136	-0.00001
137	-0.00005	138	-0.00003	139	-0.00004	140	-0.00001	141	-0.00001	142	0.00000	143	0.00000	144	0.00000
145	0.00001	146	0.00001	147	0.00002	148	0.00001	149	0.00001	150	0.00000	151	0.00000	152	-0.00000
153	-0.00000	154	-0.00000	155	-0.00001	156	-0.00000	157	-0.00001	158	-0.00000	159	-0.00000	160	-0.00000

TIME 0.0014 DAYS 121. SECONDS

1	0.00000	2	0.00000	3	-0.00000	4	-0.00000	5	0.00000	6	0.00000	7	0.00000	8	-0.00000
9	0.00000	10	0.00000	11	-0.00000	12	-0.00000	13	-0.00000	14	0.00000	15	-0.00000	16	0.00000
17	0.00000	18	0.00000	19	0.00000	20	-0.00000	21	0.00000	22	-0.00000	23	0.00000	24	0.00000
25	-0.00000	26	-0.00000	27	-0.00000	28	-0.00001	29	-0.00000	30	0.00000	31	-0.00000	32	0.00000
33	0.00000	34	0.00001	35	0.00000	36	0.00001	37	0.00001	38	-0.00000	39	0.00003	40	-0.00000
41	0.00000	42	-0.00000	43	0.00000	44	-0.00001	45	0.00000	46	-0.00002	47	0.00000	48	-0.00001
49	-0.00007	50	-0.00005	51	0.00001	52	-0.00008	53	-0.00000	54	0.00000	55	-0.00000	56	0.00000

57	-0.00000	58	0.00001	59	-0.00000	60	[illegible]	[illegible]	0.00003	62	-0.00001	63	0.00012	64	0.00015
65	0.00031	66	0.00015	67	-0.00001	68	0.00012	[illegible]	0.00000	70	0.00000	71	-0.00000	72	0.00000
73	-0.00000	74	0.00000	75	-0.00002	76	-0.00009	77	0.00002	78	0.00016	79	-0.00100	80	-0.00149
81	-0.00097	82	0.00016	83	-0.00001	84	[illegible]	85	0.00000	86	-0.00000	87	0.00001	88	0.00000
89	0.00003	90	0.00005	91	-0.00006	92	0.00024	93	-0.00120	94	0.00868	95	0.00818	96	0.00307
97	0.00803	98	-0.00114	99	0.00032	100	0.00306	101	-0.00005	102	0.00002	103	0.00005	104	-0.00007
105	0.00002	106	0.00012	107	-0.00084	108	-0.04025	109	-0.09703	110	-0.04024	111	-0.09616	112	-0.04400
113	0.00344	114	-0.00080	115	-0.03495	116	[illegible]	117	0.00370	118	0.00012	119	-0.00007	120	0.00002
121	0.00002	122	-0.00001	123	0.00010	124	[illegible]	125	0.00877	126	-0.00112	127	0.00029	128	0.51344
129	-0.10614	130	0.00013	131	0.00009	132	-0.00001	133	0.00002	134	-0.00002	135	0.00000	136	-0.00001
137	-0.00006	138	-0.00005	139	-0.00006	140	-0.00001	141	-0.00001	142	0.00000	143	0.00000	144	0.00000
145	0.00001	146	0.00001	147	0.00002	148	0.00001	149	0.00001	150	0.00000	151	0.00000	152	-0.00000
153	-0.00000	154	-0.00000	155	-0.00001	156	-0.00001	157	-0.00001	158	-0.00000	159	-0.00000	160	-0.00000

INTERVAL FROM 0.0014 TO 0.0070 DAYS

WELL PW 1 NODE 1 DISCHARGE 1.3380

PIEZOMETRIC SURFACE (NODE AND DRAWDOWN)

TIME 0.0028 DAYS 242. SECONDS

1	0.00000	2	0.00000	3	-0.00000	4	-0.00000	5	0.00000	6	[illegible]	7	0.00000	8	-0.00000
9	0.00000	10	0.00000	11	-0.00000	12	-0.00000	13	-0.00000	14	[illegible]	15	-0.00000	16	0.00000
17	0.00000	18	0.00000	19	0.00000	20	-0.00000	21	0.00000	22	[illegible]	[illegible]	0.00000	24	0.00000
25	-0.00000	26	-0.00000	27	-0.00000	28	-0.00001	29	-0.00000	30	0.00000	[illegible]	[illegible]	32	0.00000
33	0.00000	34	0.00001	35	0.00000	36	0.00003	37	0.00002	38	-0.00001	[illegible]	[illegible]	40	-0.00000
41	0.00000	42	-0.00000	43	0.00000	44	-0.00001	45	0.00000	46	-0.00004	[illegible]	0.00000	48	-0.00002
49	-0.00013	50	-0.00008	51	0.00002	52	-0.00015	53	-0.00000	54	0.00000	[illegible]	0.00000	56	0.00000
57	-0.00000	58	0.00003	59	-0.00000	60	-0.00000	61	0.00005	62	-0.00002	63	0.00022	64	0.00029
65	0.00055	66	0.00028	67	-0.00002	68	0.00022	69	0.00000	70	0.00000	71	-0.00000	72	0.00000
73	-0.00000	74	0.00000	75	-0.00004	76	-0.00016	77	0.00004	78	0.00031	79	-0.00188	80	-0.00261
81	-0.00181	82	0.00030	83	-0.00002	84	-0.00015	85	0.00000	86	-0.00000	87	0.00001	88	0.00000
89	0.00005	90	0.00009	91	-0.00010	92	0.00062	93	0.00212	94	0.01577	95	0.01492	96	0.00670
97	0.01469	98	-0.00203	99	0.00059	100	0.00626	101	-0.00009	102	0.00004	103	0.00009	104	-0.00014
105	0.00003	106	0.00022	107	-0.00154	108	-0.07978	109	-0.17229	110	-0.08109	111	-0.17075	112	-0.08702
113	0.00680	114	-0.00149	115	-0.06980	116	-0.17397	117	0.00790	118	0.00022	119	0.00013	120	0.00003
121	0.00004	122	-0.00001	123	0.00018	124	0.00025	125	0.01584	126	-0.00193	127	0.00052	128	0.99318
129	-0.19061	130	0.00024	131	0.00017	132	-0.00001	133	0.00004	134	-0.00003	135	0.00000	136	-0.00001
137	-0.00011	138	-0.00008	139	-0.00011	140	-0.00001	141	-0.00003	142	0.00000	143	0.00001	144	0.00000
145	0.00002	146	0.00002	147	0.00004	148	0.00002	149	0.00002	150	0.00000	151	0.00001	152	-0.00001
153	0.00000	154	-0.00000	155	-0.00002	156	-0.00001	157	-0.00002	158	-0.00000	159	-0.00001	160	-0.00000

TIME 0.0042 DAYS 363. SECONDS

1	0.00000	2	0.00000	3	-0.00000	4	-0.00000	5	0.00000	6	0.00000	7	0.00000	8	-0.00000
9	0.00000	10	0.00000	11	-0.00000	12	-0.00000	13	-0.00000	14	0.00000	15	-0.00000	16	0.00000
17	0.00000	18	0.00000	19	0.00000	20	-0.00000	21	0.00000	22	-0.00000	23	0.00000	24	0.00000
25	-0.00001	26	-0.00000	27	-0.00000	28	-0.00002	29	-0.00000	30	0.00000	31	-0.00000	32	0.00001
33	0.00000	34	0.00001	35	0.00000	36	0.00004	37	0.00002	38	-0.00001	39	0.00007	40	-0.00000
41	0.00000	42	-0.00000	43	0.00000	44	-0.00001	45	-0.00000	46	-0.00005	47	0.00001	48	-0.00003
49	-0.00019	50	-0.00012	51	0.00003	52	-0.00021	53	-0.00000	54	0.00000	55	-0.00000	56	0.00000
57	-0.00000	58	0.00004	59	-0.00000	60	-0.00000	61	-0.00007	62	-0.00003	63	0.00031	64	0.00041
65	0.00076	66	0.00040	67	-0.00002	68	0.00031	69	-0.00000	70	0.00000	71	-0.00000	72	0.00000
73	-0.00001	74	0.00000	75	-0.00005	76	-0.00023	77	0.00006	78	0.00044	79	-0.00267	80	-0.00350
81	-0.00255	82	0.00042	83	-0.00003	84	-0.00021	85	0.00000	86	-0.00000	87	0.00001	88	0.00001
89	0.00007	90	0.00013	91	-0.00014	92	0.00086	93	-0.00288	94	0.02186	95	0.02074	96	0.01052
97	0.02048	98	-0.00277	99	0.00082	100	0.00938	101	-0.00013	102	0.00006	103	0.00012	104	-0.00019
105	0.00005	106	0.00031	107	-0.00216	108	-0.11824	109	-0.23422	110	-0.12167	111	-0.23219	112	-0.12867
113	0.00991	114	-0.00212	115	-0.10401	116	-0.23920	117	0.01225	118	0.00031	119	-0.00018	120	0.00005
121	0.00006	122	-0.00002	123	0.00025	124	0.00035	125	0.02183	126	-0.00257	127	0.00071	128	1.45062
129	-0.26182	130	0.00035	131	0.00024	132	-0.00002	133	0.00005	134	-0.00004	135	0.00000	136	-0.00002
137	-0.00015	138	-0.00011	139	-0.00015	140	-0.00002	141	-0.00004	142	0.00000	143	0.00001	144	0.00000
145	0.00003	146	0.00002	147	0.00005	148	0.00002	149	0.00003	150	0.00000	151	0.00001	152	-0.00001
153	0.00000	154	-0.00001	155	-0.00002	156	-0.00001	157	-0.00002	158	-0.00001	159	-0.00001	160	-0.00000

TIME 0.0056 DAYS 484. SECONDS

1	0.00000	2	0.00000	3	-0.00000	4	-0.00000	5	0.00000	6	0.00000	7	0.00000	8	-0.00000
9	0.00000	10	0.00000	11	-0.00000	12	-0.00000	13	-0.00000	14	0.00000	15	-0.00000	16	0.00000
17	0.00000	18	0.00000	[illegible]	0.00000	20	-0.00000	21	0.00000	22	-0.00000	23	0.00000	24	0.00000
25	-0.00001	26	-0.00000	[illegible]	0.00000	28	-0.00002	29	-0.00000	30	0.00000	31	-0.00000	32	0.00001
33	0.00000	34	0.00002	[illegible]	0.00000	36	0.00004	37	0.00003	38	-0.00001	39	0.00009	40	-0.00000
41	0.00000	42	-0.00000	43	0.00000	44	-0.00002	45	0.00000	46	-0.00007	47	0.00001	48	-0.00003
49	-0.00024	50	-0.00014	51	0.00004	52	-0.00026	53	-0.00000	54	0.00000	55	-0.00000	56	0.00000
57	-0.00001	58	0.00004	59	-0.00000	60	-0.00000	61	0.00009	62	-0.00003	63	0.00040	64	0.00052
65	0.00094	66	0.00050	67	-0.00003	68	0.00039	69	0.00000	70	0.00000	71	-0.00000	72	0.00000
73	-0.00001	74	0.00000	75	-0.00007	76	-0.00028	77	0.00008	78	0.00056	79	-0.00339	80	-0.00419
81	-0.00320	82	0.00053	83	-0.00003	84	-0.00026	85	-0.00000	86	-0.00000	87	0.00002	88	0.00001
89	0.00008	90	0.00016	91	-0.00017	92	0.00107	93	-0.00350	94	0.02705	95	0.02574	96	0.01440
97	0.02547	98	-0.00337	99	0.00102	100	0.01233	101	-0.00016	102	0.00007	103	0.00015	104	-0.00023
105	0.00006	106	0.00039	107	-0.00268	108	-0.15530	109	-0.28443	110	-0.16158	111	-0.28210	112	-0.16863
113	0.01270	114	-0.00267	115	-0.13726	116	-0.29390	117	0.01661	118	0.00040	119	-0.00023	120	0.00006
121	0.00007	122	-0.00002	123	0.00031	124	0.00043	125	0.02686	126	-0.00304	127	0.00087	128	1.88724
129	-0.32137	130	0.00044	131	0.00031	132	-0.00002	133	0.00007	134	-0.00005	135	0.00000	136	-0.00003
137	-0.00019	138	-0.00013	139	-0.00019	140	-0.00002	141	-0.00005	142	0.00000	143	0.00001	144	0.00000
145	0.00003	146	0.00003	147	0.00006	148	0.00003	149	[illegible]	150	0.00000	151	0.00001	152	-0.00001
153	0.00000	154	-0.00001	155	-0.00003	156	-0.00002	157	[illegible]	158	-0.00001	159	-0.00001	160	-0.00000

TIME 0.0070 DAYS 605. SECONDS

1	0.00000	2	0.00000	3	-0.00000	4	-0.00000	5	0.00000	6	0.00000	7	0.00000	8	-0.00000
9	0.00000	10	0.00000	11	-0.00000	12	-0.00000	13	-0.00000	14	0.00000	15	-0.00000	16	0.00000
17	0.00000	18	0.00000	19	0.00001	20	-0.00000	21	0.00000	22	-0.00000	23	0.00000	24	0.00000
25	-0.00001	26	-0.00000	27	-0.00001	28	-0.00002	29	-0.00000	30	0.00000	31	-0.00000	32	0.00001
33	0.00000	34	0.00002	35	0.00000	36	0.00005	37	0.00004	38	-0.00001	39	0.00010	40	-0.00000
41	0.00000	42	-0.00000	43	0.00000	44	-0.00002	45	0.00000	46	-0.00008	47	0.00001	48	-0.00004
49	-0.00028	50	-0.00016	51	0.00005	52	-0.00030	53	-0.00000	54	0.00000	55	-0.00000	56	0.00000
57	-0.00001	58	0.00005	59	-0.00000	60	-0.00000	61	0.00010	62	-0.00004	63	0.00047	64	0.00062
65	0.00108	66	0.00059	67	-0.00003	68	0.00045	69	0.00000	70	0.00000	71	-0.00000	72	0.00000
73	-0.00001	74	0.00000	75	-0.00008	76	-0.00033	77	0.00009	78	0.00067	79	-0.00402	80	-0.00472
81	-0.00377	82	0.00062	83	-0.00004	84	-0.00030	85	0.00000	86	-0.00000	87	0.00002	88	0.00001
89	0.00010	90	0.00018	91	-0.00020	92	0.00124	93	-0.00400	94	0.03143	95	0.02999	96	0.01823
97	0.02973	98	-0.00385	99	0.00118	100	0.01505	101	-0.00018	102	0.00008	103	0.00017	104	-0.00027
105	0.00007	106	0.00045	107	-0.00312	108	-0.[illegible]072	109	-0.32432	110	-0.20047	111	-0.32188	112	-0.20660
113	0.01510	114	-0.00316	115	-0.16927	116	[illegible]	117	0.02089	118	0.00048	119	-0.00027	120	0.00007
121	0.00008	122	-0.00003	123	0.00036	124	0.00[illegible]0	125	0.03101	126	-0.00338	127	0.00099	128	2.30438
129	-0.37064	130	0.00053	131	0.00036	132	-0.00003	133	0.00008	134	-0.00006	135	0.00000	136	-0.00003
137	-0.00022	138	-0.00015	139	-0.00022	140	-0.00003	141	-0.00006	142	0.00000	143	0.00002	144	0.00000
145	0.00004	146	0.00003	147	0.00007	148	0.00003	149	0.00004	150	0.00000	151	0.00002	152	-0.00001
153	0.00000	154	-0.00001	155	-0.00003	156	-0.00002	157	-0.00003	158	-0.00001	159	-0.00001	160	0.00000

INTERVAL FROM 0.0070 TO 0.0210 DAYS

WELL PW 1 NODE 1 DISCHARGE 1.3380

PIEZOMETRIC SURFACE (NODE AND DRAWDOWN)

TIME 0.0093 DAYS 806. SECONDS

1	0.00000	2	0.00000	3	-0.00000	4	-0.00000	5	0.00000	6	0.00000	7	0.00000	8	-0.00000
9	0.00000	10	0.00000	11	-0.00000	12	-0.00000	13	-0.00000	14	0.00000	15	-0.00000	16	0.00000
17	0.00000	18	0.00000	19	0.00001	20	-0.00000	21	0.00000	22	-0.00000	23	0.00000	24	0.00000
25	-0.00001	26	-0.00000	27	-0.00001	28	-0.00003	29	-0.00000	30	0.00000	31	-0.00000	32	0.00001
33	0.00000	34	0.00002	35	0.00000	36	0.00006	37	0.00004	38	-0.00002	39	0.00012	40	-0.00000
41	0.00000	42	-0.00000	43	0.00000	44	-0.00002	45	0.00000	46	-0.00009	47	0.00001	48	-0.00005
49	-0.00032	50	-0.00018	51	0.00005	52	-0.00035	53	-0.00000	54	0.00000	55	-0.00000	56	0.00000
57	-0.00001	58	0.00006	59	-0.00000	60	-0.00000	61	0.00012	62	-0.00005	63	0.00056	64	0.00076
65	0.00125	66	0.00070	67	-0.00004	68	0.00054	69	0.00000	70	0.00000	71	-0.00000	72	0.00000
73	-0.00001	74	0.00000	75	-0.00009	76	-0.00039	77	0.00011	78	0.00082	79	-0.00485	80	-0.00521
81	-0.00448	82	0.00074	83	-0.00005	84	-0.00036	85	0.00000	86	-0.00000	87	0.00002	88	0.00001
89	0.00011	90	0.00021	91	-0.00022	92	0.00144	93	-0.00453	94	0.03681	95	0.03531	96	0.02412
97	0.03512	98	-0.00437	99	0.00138	100	0.01870	101	-0.00021	102	0.00010	103	[illegible]0.0[illegible]020	104	-0.00031
105	0.00008	106	0.00053	107	-0.00365	108	-0.24443	109	-0.36739	110	-0.26135	111	[illegible]0.3[illegible]529	112	-0.26360
113	0.01784	114	-0.00379	115	-0.21835	116	-0.39384	117	0.02745	118	0.00058	119	[illegible]031	120	0.00008

121	0.00009	122	-0.00003	123	0.00042	124	0.00059	125	0.03588	126	-0.00364	127	0.00113	128	2.95092
129	-0.42940	130	0.00064	131	0.00043	132	-0.00003	133	0.00009	134	-0.00006	135	0.00001	136	-0.00003
137	-0.00025	138	-0.00016	139	-0.00025	140	-0.00004	141	-0.00006	142	0.00000	143	0.00002	144	0.00000
145	0.06364	146	0.00004	147	0.00008	148	0.00004	149	0.00004	150	0.00000	151	0.00002	152	-0.00002
153	0.00000	154	-0.00001	155	-0.00004	156	-0.00002	157	-0.00004	158	-0.00001	159	0.00002	160	0.00000

TIME 0.0117 DAYS 1008. SECONDS

1	0.00000	2	0.00000	3	-0.00000	4	-0.00000	5	0.00000	6	0.00000	7	0.00000	8	-0.00000
9	0.00000	10	0.00000	11	-0.00000	12	-0.00000	13	-0.00000	14	0.00000	15	-0.00000	16	0.00000
17	0.00000	18	0.00000	19	0.00001	20	-0.00000	21	0.00000	22	-0.00000	23	0.00000	24	0.00000
25	-0.00001	26	-0.00000	27	-0.00001	28	-0.00003	29	-0.00000	30	0.00000	31	-0.00000	32	0.00001
33	0.00000	34	0.00003	35	0.00000	36	0.00007	37	0.00005	38	-0.00002	39	0.00013	40	-0.00000
41	0.00000	42	-0.00000	43	0.00000	44	-0.00003	45	0.00000	46	-0.00010	47	0.00001	48	-0.00006
49	-0.00036	50	-0.00019	51	0.00006	52	-0.00038	53	-0.00000	54	0.00000	55	-0.00000	56	0.00001
57	-0.00001	58	0.00007	59	-0.00000	60	-0.00000	61	0.00017	62	-0.00006	63	0.00063	64	0.00086
65	0.00135	66	0.00078	67	-0.00005	68	0.00060	69	0.00000	70	0.00000	71	-0.00000	72	0.00000
73	-0.00001	74	0.00000	75	-0.00010	76	-0.00043	77	0.00012	78	0.00093	79	-0.00549	80	-0.00539
81	-0.00499	82	0.00082	83	-0.00005	84	-0.00030	85	0.00000	86	-0.00000	87	0.00002	88	0.00001
89	0.00012	90	0.00023	91	-0.00024	92	0.01579	93	-0.00483	94	0.04056	95	0.03911	96	0.02929
97	0.03901	98	-0.00467	99	0.00151	100	0.02137	101	-0.00022	102	0.00011	103	0.00022	104	-0.00034
105	0.00009	106	0.00057	107	-0.00399	108	-0.29235	109	-0.39132	110	-0.31754	111	-0.39001	112	-0.31387
113	0.01928	114	-0.00425	115	-0.26263	116	-0.43100	117	0.03320	118	0.00065	119	-0.00035	120	0.00009
121	0.00010	122	-0.00003	123	0.00046	124	0.00064	125	0.03900	126	-0.00365	127	0.00121	128	3.55348
129	-0.36901	130	0.00073	131	0.00048	132	-0.00004	133	0.00010	134	0.00007	135	0.00001	136	-0.00004
137	-0.00027	138	-0.00017	139	-0.00028	140	-0.00004	141	-0.00007	142	0.00001	143	0.00002	144	0.00000
145	0.00005	146	0.00004	147	0.00009	148	0.00004	149	0.00005	150	0.00000	151	0.00002	152	-0.00002
153	0.00000	154	-0.00001	155	-0.00004	156	-0.00002	157	-0.00004	158	-0.00001	159	-0.00002	160	0.00000

TIME 0.0140 DAYS 1210. SECONDS

1	0.00000	2	0.00000	3	-0.00000	4	-0.00000	5	0.00000	6	0.00000	7	0.00000	8	-0.00000
9	0.00000	10	0.00000	11	-0.00000	12	-0.00000	13	-0.00000	14	0.00000	15	-0.00000	16	0.00000
17	0.00000	18	0.00000	19	0.00001	20	-0.00000	21	0.00000	22	-0.00000	23	0.00000	24	0.00000
25	-0.00001	26	-0.00000	27	-0.00001	28	-0.00003	29	-0.00000	30	0.00000	31	-0.00000	32	0.00001
33	0.00000	34	0.00003	35	0.00001	36	0.00007	37	0.00005	38	-0.00002	39	0.00013	40	-0.00000
41	0.00000	42	-0.00000	43	0.00000	44	-0.00003	45	0.00000	46	-0.00011	47	0.00001	48	-0.00006
49	-0.00038	50	-0.00020	51	0.00006	52	-0.00040	53	-0.00000	54	0.00000	55	-0.00000	56	0.00001
57	-0.00001	58	0.00007	59	-0.00000	60	-0.00000	61	0.00014	62	-0.00006	63	0.00068	64	0.00093
65	0.00140	66	0.00083	67	-0.00005	68	0.00063	69	0.00000	70	0.00000	71	-0.00000	72	0.00000
73	-0.00001	74	0.00000	75	-0.00010	76	-0.00045	77	0.00012	78	0.00100	79	-0.00594	80	-0.00533
81	-0.00531	82	0.00086	83	-0.00006	84	-0.00041	85	0.00000	86	-0.00000	87	0.00003	88	0.00001
89	0.00013	90	0.00024	91	0.00025	92	0.00164	93	-0.00494	94	0.04294	95	0.04162	96	0.03361
97	0.04162	98	-0.00478	99	0.00158	100	0.02301	101	-0.00023	102	0.00011	103	0.00022	104	-0.00035
105	0.00009	106	0.00059	107	-0.00417	108	-0.33427	109	-0.39959	110	-0.36858	111	-0.39948	112	-0.35723
113	0.01943	114	-0.00456	115	-0.30178	116	-0.45382	117	0.03803	118	0.00070	119	-0.00036	120	0.00010
121	0.00010	122	-0.00004	123	0.00047	124	0.00067	125	0.04065	126	-0.00346	127	0.00123	128	4.11621
129	-0.49286	130	0.00079	131	0.00051	132	-0.00004	133	0.00011	134	-0.00007	135	0.00001	136	-0.00004

137	-0.00027	138	-0.00017	139	-0.00029	140	-0.00004	141	-0.00008	142	0.00001	143	0.00002	144	0.00000
145	0.00005	146	0.00004	147	0.00009	148	0.00004	149	0.00005	150	0.00000	151	0.00002	152	-0.00002
153	0.00001	154	-0.00001	155	-0.00004	156	-0.00002	157	-0.00004	158	-0.00001	159	-0.00002	160	0.00000

TIME 0.0163 DAYS 1411. SECONDS

1	0.00000	2	0.00000	3	-0.00000	4	-0.00000	5	0.00000	6	0.00000	7	0.00000	8	-0.00000
9	0.00000	10	0.00000	11	-0.00000	12	-0.00000	13	-0.00000	14	0.00000	15	-0.00000	16	0.00000
17	0.00000	18	0.00000	19	0.00001	20	-0.00000	21	0.00000	22	-0.00000	23	0.00000	24	0.00000
25	-0.00001	26	-0.00000	27	-0.00001	28	-0.00003	29	-0.00000	30	0.00000	31	-0.00000	32	0.00001
33	0.00000	34	0.00003	35	0.00001	36	0.00007	37	0.00005	38	-0.00002	39	0.00013	40	-0.00000
41	0.00000	42	-0.00000	43	0.00000	44	-0.00003	45	0.00000	46	-0.00012	47	0.00001	48	-0.00007
49	-0.00038	50	-0.00019	51	0.00006	52	-0.00040	53	-0.00000	54	0.00000	55	-0.00000	56	0.00001
57	-0.00001	58	0.00007	59	-0.00000	60	-0.00000	61	0.00015	62	-0.00007	63	0.00071	64	0.00097
65	0.00140	66	0.00085	67	-0.00006	68	0.00065	69	0.00000	70	0.00000	71	-0.00000	72	0.00000
73	-0.00001	74	0.00001	75	-0.00010	76	-0.00046	77	0.00012	78	0.00105	79	-0.00622	80	-0.00508
81	-0.00547	82	0.00088	83	-0.00006	84	-0.00044	85	0.00000	86	-0.00000	87	0.00003	88	0.00001
89	0.00013	90	0.00024	91	-0.00025	92	0.00165	93	-0.00490	94	0.04415	95	0.04300	96	0.03699
97	0.04310	98	-0.00474	99	0.00160	100	0.02363	101	-0.00023	102	0.00011	103	0.00023	104	-0.00035
105	-0.00009	106	0.00059	107	-0.00420	108	-0.37012	109	-0.39504	110	-0.41421	111	-0.39643	112	-0.39369
113	0.01836	114	-0.00473	115	-0.33568	116	-0.46479	117	0.04186	118	0.00073	119	-0.00037	120	0.00010
121	0.00010	122	-0.00004	123	0.00048	124	0.00067	125	0.04104	126	-0.00313	127	0.00121	128	4.64273
129	-0.50372	130	0.00082	131	0.00053	132	-0.00004	133	0.00011	134	-0.00007	135	0.00001	136	-0.00004
137	-0.00027	138	-0.00016	139	-0.00029	140	-0.00005	141	-0.00008	142	0.00001	143	0.00002	144	0.00000
145	0.00005	146	0.00004	147	0.00009	148	0.00004	149	0.00005	150	0.00000	151	0.00002	152	-0.00002
153	0.00001	154	-0.00001	155	-0.00004	156	-0.00002	157	-0.00004	158	-0.00001	159	-0.00002	160	0.00000

TIME 0.0187 DAYS 1613. SECONDS

1	0.00000	2	0.00000	3	-0.00000	4	-0.00000	5	0.00000	6	0.00000	7	0.00000	8	-0.00000
9	0.00000	10	0.00000	11	-0.00000	12	-0.00000	13	-0.00000	14	0.00000	15	-0.00000	16	0.00000
17	0.00000	18	0.00000	19	0.00001	20	-0.00000	21	0.00000	22	-0.00000	23	0.00000	24	0.00000
25	-0.00001	26	-0.00000	27	-0.00001	28	-0.00003	29	-0.00000	30	0.00000	31	-0.00000	32	0.00001
33	0.00000	34	0.00003	35	0.00001	36	0.00007	37	0.00005	38	-0.00002	39	0.00013	40	-0.00000
41	0.00000	42	-0.00000	43	0.00000	44	[illegible]	45	0.00000	46	-0.00012	47	0.00001	48	-0.00007
49	-0.00038	50	-0.00018	51	0.00006	52	[illegible]	53	-0.00000	54	0.00000	55	-0.00000	56	0.00001
57	-0.00001	58	0.00007	59	-0.00000	60	[illegible]	61	0.00015	62	-0.00007	63	0.00073	64	0.00099
65	0.00136	66	0.00086	67	-0.00006	68	0.00065	69	0.00000	70	0.00000	71	-0.00000	72	0.00000
73	-0.00001	74	0.00001	75	-0.00010	76	-0.00046	77	0.00012	78	0.00107	79	-0.00634	80	-0.00468
81	-0.00548	82	0.00087	83	-0.00006	84	-0.00041	85	0.00000	86	-0.00000	87	0.00002	88	0.00001
89	0.00013	90	0.00023	91	-0.00024	92	0.00162	93	-0.00472	94	0.04438	95	0.04343	96	0.03941
97	0.04361	98	-0.00457	99	0.00158	100	0.02327	101	-0.00022	102	0.00011	103	0.00022	104	-0.00034
105	0.00009	106	0.00057	107	-0.00410	108	-0.39998	109	-0.37998	110	-0.45431	111	-0.38313	112	-0.42340
113	0.01618	114	-0.00478	115	-0.36430	116	-0.46591	117	[illegible]	118	0.00074	119	-0.00037	120	0.00010
121	0.00010	122	-0.00004	123	0.00046	124	0.00064	125	0.04038	126	-0.00269	127	0.00115	128	5.13627
129	-0.50384	130	0.00083	131	0.00053	132	-0.00004	133	0.00011	134	-0.00007	135	0.00001	136	-0.00004
137	-0.00026	138	-0.00015	139	-0.00029	140	-0.00005	141	-0.00008	142	0.00001	143	0.00002	144	0.00000
145	0.00005	146	0.00003	147	0.00008	148	0.00004	149	0.00005	150	0.00000	151	0.00002	152	-0.00002

81	-0.00144	82	0.00016	83	-0.00001	84	-0.00005	85	-0.00000	86	0.00000	87	0.00000	88	-0.00000
89	0.00001	90	0.00002	91	-0.00002	92	0.00026	93	-0.00045	94	0.01887	95	0.01837	96	0.0158[illegible]
97	0.01617	98	-0.00011	99	0.00020	100	-0.01190	101	-0.00001	102	0.00000	103	0.00001	104	[illegible]
105	0.00000	106	-0.00000	107	-0.00041	108	-0.35004	109	0.07623	110	-0.50111	111	0.05724	112	[illegible]
113	-0.02864	114	-0.00101	115	-0.32670	116	-0.15423	117	0.02120	118	0.00012	119	-0.00001	120	[illegible]
121	0.00000	122	-0.00001	123	0.00004	124	0.00002	125	0.00586	126	0.00197	127	-0.00208	128	[illegible]
129	-0.16695	130	0.00016	131	-0.00010	132	-0.00002	133	0.00001	134	0.00000	135	0.00000	136	[illegible]
137	0.00001	138	0.00004	139	-0.00001	140	-0.00001	141	-0.00001	142	0.00000	143	-0.00000	144	[illegible]
145	-0.00000	146	-0.00001	147	-0.00001	148	-0.00000	149	0.00000	150	-0.00000	151	0.00000	152	[illegible]
153	0.00000	154	0.00000	155	0.00000	156	0.00000	157	0.00000	158	-0.00000	159	0.00000	160	0.00000

TIME 0.0627 DAYS 5419. SECONDS

1	-0.00000	2	-0.00000	3	0.00000	4	0.00000	5	-0.00000	6	-0.00000	7	-0.00000	8	0.00000
9	-0.00000	10	-0.00000	11	0.00000	12	0.00000	13	0.00000	14	-0.00000	15	0.00000	16	-0.00000
17	-0.00000	18	-0.00000	19	-0.00000	20	0.00000	21	-0.00000	22	0.00000	23	-0.00000	24	-0.00000
25	0.00001	26	0.00000	27	0.00000	28	0.00002	29	0.00000	30	-0.00000	31	0.00000	32	-0.00001
33	-0.00000	34	-0.00001	35	-0.00000	36	-0.00003	37	-0.00002	38	0.00001	39	-0.00006	40	0.00000
41	-0.00000	42	0.00000	43	-0.00000	44	0.00001	45	-0.00000	46	0.00003	47	-0.00000	48	0.00001
49	0.00016	50	0.00009	51	-0.00003	52	0.00019	53	0.00000	54	-0.00000	55	0.00000	56	-0.00000
57	0.00000	58	-0.00003	59	0.00000	60	0.00000	61	-0.00006	62	0.00001	63	-0.00014	64	-0.00033
65	-0.00061	66	-0.00037	67	0.00002	68	-0.00024	69	-0.00000	70	-0.00000	71	0.00000	72	-0.00000
73	0.00001	74	-0.00000	75	0.00005	76	0.00019	77	-0.00006	78	-0.00033	79	0.00138	80	0.00344
81	0.00191	82	-0.00036	83	0.00002	84	0.00021	85	-0.00000	86	0.00000	87	-0.00001	88	-0.00001
89	-0.00007	90	-0.00011	91	0.00012	92	-0.00070	93	0.00245	94	-0.00387	95	-0.00579	96	-0.01896
97	-0.01362	98	0.00319	99	-0.00087	100	-0.03946	101	0.00014	102	-0.00006	103	-0.00013	104	0.00017
105	-0.00004	106	0.00031	107	0.00191	108	-0.11847	109	0.56271	110	-0.33952	111	0.53634	112	-0.09003
113	-0.05225	114	0.00253	115	-0.11115	116	0.23486	117	-0.01578	118	-0.00044	119	0.00022	120	-0.00006
121	-0.00005	122	0.00001	123	-0.00023	124	-0.00032	125	-0.02231	126	0.00350	127	-0.00078	128	10.32014
129	0.23955	130	-0.00049	131	-0.00028	132	0.00002	133	-0.00006	134	0.00004	135	-0.00000	136	0.00002
137	0.00016	138	0.00012	139	0.00019	140	0.00002	141	0.00004	142	-0.00000	143	-0.00001	144	-0.00000
145	-0.00003	146	-0.00002	147	-0.00006	148	-0.00003	149	-0.00003	150	-0.00000	151	-0.00001	152	0.00001
153	-0.00000	154	0.00001	155	0.00003	156	0.00002	157	0.00003	158	0.00001	159	0.00001	160	-0.00000

TIME 0.0836 DAYS 7221. SECONDS

1	-0.00000	2	-0.00000	3	0.00000	4	[illegible]	5	-0.00000	6	-0.00000	7	-0.00000	8	0.00000
9	-0.00000	10	-0.00000	11	0.00000	12	[illegible]	13	0.00000	14	-0.00000	15	0.00000	16	-0.00000
17	-0.00000	18	-0.00000	19	-0.00000	20	0.00000	21	-0.00000	22	0.00000	23	-0.00000	24	-0.00000
25	0.00001	26	0.00000	27	0.00000	28	0.0000[illegible]	29	0.00000	30	-0.00000	31	0.00000	32	-0.00001
33	-0.00000	34	-0.00002	35	-0.00000	36	-0.00004	37	-0.00003	38	0.00001	39	-0.00008	40	0.00000
41	-0.00000	42	0.00000	43	-0.00000	44	0.00002	45	-0.00000	46	0.00006	47	-0.00001	48	0.00004
49	0.00023	50	0.00011	51	-0.00004	52	0.00025	53	0.00000	54	-0.00000	55	0.00000	56	-0.00000
57	0.00001	58	-0.00005	59	0.00000	60	0.00000	61	-0.00009	62	0.00004	63	-0.00036	64	-0.00060
65	-0.00083	66	-0.00056	67	0.00004	68	-0.00041	69	-0.00000	70	-0.00000	71	0.00000	72	-0.00000
73	0.00001	74	-0.00000	75	0.00007	76	0.00028	77	-0.00008	78	-0.00063	79	0.00327	80	0.00360
81	0.00332	82	-0.00052	83	0.00004	84	0.00030	85	-0.00000	86	0.00000	87	-0.00002	88	-0.00001
89	-0.00009	90	-0.00014	91	0.00015	92	-0.00098	93	0.00344	94	-0.01495	95	-0.01917	96	-0.04677

97	-0.03542	98	0.00467	99	-0.00132	100	-0.04841	101	0.00019	102	-0.00008	103	-0.00018	104	0.00020
105	-0.00005	106	-0.00029	107	0.00224	108	0.18932	109	1.04276	110	-0.08670	111	1.01390	112	0.23930
113	-0.04790	114	0.00442	115	0.18330	116	0.64434	117	-0.04786	118	-0.00072	119	0.00032	120	-0.00009
121	-0.00006	122	0.00002	123	-0.00027	124	-0.00033	125	-0.03556	126	0.00256	127	-0.00087	128	11.76483
129	0.65809	130	-0.00082	131	-0.00048	132	0.00004	133	-0.00010	134	0.00004	135	-0.00001	136	0.00002
137	0.00017	138	0.00011	139	0.00025	140	0.00005	141	0.00007	142	-0.00001	143	-0.00001	144	-0.00000
145	-0.00003	146	-0.00002	147	-0.00006	148	-0.00004	149	-0.00004	150	-0.00000	151	-0.00002	152	0.00001
153	-0.00000	154	0.00001	155	0.00002	156	0.00002	157	0.00003	158	0.00001	159	0.00002	160	-0.00000

TIME 0.1044 DAYS 9023. SECONDS

1	-0.00000	2	-0.00000	3	0.00000	4	0.00000	5	-0.00000	6	-0.00000	7	-0.00000	8	0.00000
9	-0.00000	10	-0.00000	11	0.00000	12	0.00000	13	0.00000	14	-0.00000	15	0.00000	16	-0.00000
17	-0.00000	18	-0.00000	19	-0.00000	20	0.00000	21	-0.00000	22	0.00000	23	-0.00000	24	-0.00000
25	0.00001	26	0.00000	27	0.00000	28	0.00001	29	0.00000	30	-0.00000	31	0.00000	32	-0.00000
33	-0.00000	34	-0.00001	35	-0.00000	36	-0.00003	37	-0.00002	38	0.00001	39	-0.00005	40	0.00000
41	-0.00000	42	0.00000	43	-0.00000	44	0.00001	45	-0.00000	46	0.00004	47	-0.00001	48	[illegible]
49	0.00015	50	0.00006	51	-0.00002	52	0.00016	53	0.00000	54	-0.00000	55	0.00000	56	-0.00000
57	0.00000	58	-0.00003	59	0.00000	60	0.00000	61	-0.00006	62	0.00004	63	-0.00029	64	-0.00044
65	-0.00051	66	-0.00040	67	0.00004	68	-0.00032	69	-0.00000	70	-0.00000	71	0.00000	72	-0.00000
73	0.00000	74	-0.00000	75	0.00005	76	0.00018	77	-0.00005	78	-0.00048	79	0.00260	80	0.00196
81	0.00257	82	-0.00034	83	0.00002	84	0.00022	85	-0.00000	86	-0.00000	87	-0.00001	88	-0.00000
89	-0.00006	90	-0.00007	91	0.00008	92	-0.00058	93	0.00259	94	-0.01084	95	-0.01755	96	-0.06069
97	-0.04441	98	0.00430	99	-0.00109	100	-0.03590	101	0.00014	102	-0.00005	103	-0.00013	104	0.00008
105	-0.00002	106	-0.00002	107	0.00070	108	0.52670	109	1.49540	110	0.20730	111	1.46678	112	0.59263
113	-0.01612	114	0.00415	115	0.51135	116	1.04675	117	-0.06745	118	-0.00063	119	0.00026	120	-0.00007
121	-0.00002	122	0.00001	123	-0.00010	124	-0.00005	125	-0.03054	126	0.00008	127	-0.00042	128	12.88280
129	1.06276	130	-0.00076	131	-0.00044	132	0.00005	133	-0.00008	134	0.00001	135	-0.00000	136	0.00000
137	0.00006	138	0.00003	139	0.00017	140	0.00005	141	0.00006	142	-0.00001	143	-0.00000	144	0.00000
145	-0.00001	146	-0.00000	147	-0.00003	148	-0.00002	149	-0.00003	150	-0.00000	151	-0.00001	152	0.00000
153	-0.00000	154	0.00000	155	0.00001	156	0.00001	157	0.00002	158	0.00001	159	0.00001	160	-0.00000

TIME 0.1253 DAYS 10825. SECONDS

1	-0.00000	2	0.00000	3	-0.00000	4	-0.00000	5	0.00000	6	0.00000	7	-0.00000	8	-0.00000
9	0.00000	10	0.00000	11	-0.00000	12	-0.00000	13	-0.00000	14	0.00000	15	-0.00000	16	0.00000
17	0.00000	18	0.00000	19	0.00000	20	-0.00000	21	0.00000	22	-0.00000	23	0.00000	24	0.00000
25	-0.00000	26	-0.00000	27	-0.00000	28	-0.00001	29	-0.00000	30	0.00000	31	-0.00000	32	0.00000
33	0.00000	34	0.00001	35	-0.00000	36	0.00001	37	0.00001	38	-0.00000	39	0.00002	40	-0.00000
41	0.00000	42	-0.00000	43	0.00000	44	-0.00001	45	0.00000	46	-0.00002	47	-0.00000	48	0.00000
49	-0.00005	50	-0.00004	51	0.00001	52	-0.00005	53	-0.00000	54	0.00000	55	-0.00000	56	0.00000
57	-0.00000	58	0.00001	59	-0.00000	60	-0.00000	61	0.00002	62	0.00000	63	0.00007	64	0.00010
65	0.00021	66	0.00004	67	0.00001	68	0.00000	69	0.00000	70	0.00000	71	-0.00000	72	0.00000
73	-0.00000	74	-0.00000	75	-0.00001	76	-0.00008	77	-0.00002	78	0.00008	79	-0.00062	80	-0.00049
81	-0.00010	82	0.00011	83	-0.00001	84	0.00000	85	0.00000	86	-0.00000	87	0.00000	88	0.00000
89	0.00001	90	0.00007	91	-0.00006	92	0.00037	93	0.00036	94	0.00895	95	0.00001	96	-0.05872
97	-0.03900	98	0.00243	99	-0.00028	100	-0.00347	101	0.00002	102	0.00001	103	-0.00001	104	-0.00014
105	0.00004	106	0.00042	107	-0.00226	108	0.86835	109	1.91478	110	0.51393	111	1.88760	112	0.94538

113	0.03862	114	0.00176	115	0.84735	116	1.43046	117	-0.072[illegible]	118	-0.0002[illegible]	[illegible]	[illegible]00006	120	-0.00001
121	0.00005	122	-0.00002	123	0.00023	124	0.00041	125	-0.00733	126	-0.00279	[illegible]	[illegible]00039	128	13.78140
129	1.44424	130	-0.00031	131	-0.00018	132	0.00003	133	-0.00002	134	-0.00003	[illegible]	[illegible]00000	136	-0.00002
137	-0.00012	138	-0.00008	139	-0.00001	140	0.00003	141	0.00001	142	-0.00001	[illegible]3	0.00001	144	0.00000
145	0.00002	146	0.00002	147	0.00003	148	0.00000	149	0.00000	150	-0.00000	151	-0.00000	152	-0.00[illegible]1
153	0.00000	154	-0.00000	155	-0.00002	156	-0.00001	157	-0.00001	158	0.00000	159	-0.00000	160	-0.00000

TIME 0.1461 DAYS 12627. SECONDS

1	0.00000	2	-0.00000	3	-0.00000	4	-0.00000	5	0.00000	6	0.00000	7	0.00000	8	-0.00000
9	0.00000	10	0.00000	11	-0.00000	12	-0.00000	13	-0.00000	14	0.00000	15	-0.00000	16	0.00000
17	0.00000	18	0.00000	19	0.00001	20	-0.00000	21	0.00000	22	-0.00000	23	0.00000	24	0.00000
25	-0.00001	26	-0.00000	27	-0.00001	28	-0.00003	29	-0.00000	30	0.00000	31	-0.00000	32	0.00001
33	0.00000	34	0.00003	35	0.00001	36	0.00007	37	0.00005	38	-0.00001	39	0.00011	40	-0.00000
41	0.00000	42	-0.00000	43	0.00000	44	-0.00002	45	0.00000	46	-0.00011	47	0.00001	48	-0.00006
49	-0.00034	50	-0.00016	51	0.00005	52	-0.00033	53	-0.00000	54	0.00000	55	-0.00000	56	0.00000
57	-0.00001	58	0.00006	59	-0.00000	60	-0.00000	61	0.00014	62	-0.00006	63	0.00070	64	0.00092
65	0.00118	66	0.00066	67	-0.00004	68	0.00050	69	0.00000	70	0.00000	71	-0.00000	72	0.00000
73	-0.00001	74	0.00000	75	-0.00008	76	-0.00045	77	0.00012	78	0.00095	79	-0.00607	80	-0.00278
81	-0.00423	82	0.00071	83	-0.00005	84	-0.00029	85	0.00000	86	-0.00000	87	0.00002	88	0.00001
89	0.00010	90	0.00024	91	-0.00024	92	0.00167	93	-0.00261	94	0.04343	95	0.03272	96	-0.04148
97	-0.01936	98	0.00039	99	0.00095	100	0.04568	101	-0.00016	102	0.00008	103	0.00016	104	-0.00041
105	0.00011	106	0.00092	107	-0.00607	108	1.20128	109	2.30103	110	0.81778	111	2.27561	112	1.28571
113	0.11099	114	-0.00250	115	1.17749	116	1.79119	117	-0.06226	118	0.00049	119	-0.00025	120	0.00007
121	0.00012	122	-0.00006	123	0.00066	124	0.00094	125	0.03240	126	-0.00498	127	0.00136	128	14.52660
129	1.80020	130	-0.00046	131	0.00028	132	-0.00002	133	0.00007	134	-0.00009	135	0.00001	136	-0.00005
137	-0.00033	138	-0.00019	139	-0.00026	140	-0.00002	141	-0.00005	142	0.00000	143	0.00002	144	0.00000
145	0.00008	146	0.00005	147	0.00009	148	0.00004	149	0.00004	150	0.00000	151	0.00002	152	-0.00002
153	0.00001	154	-0.00001	155	-0.00004	156	-0.00002	157	-0.00004	158	-0.00001	159	-0.00002	160	0.00000

TIME 0.1670 DAYS 14429. SECONDS

1	0.00000	2	0.00000	3	-0.00000	4	-0.00000	5	0.00000	6	0.00000	7	0.00000	8	-0.00000
9	0.00000	10	0.00000	11	-0.00000	12	-0.00000	13	-0.00000	14	0.00000	15	-0.00000	16	0.00000
17	0.00000	18	0.00001	19	0.00001	20	-0.00000	21	0.00000	22	-0.00000	23	0.00001	24	0.00000
25	-0.00002	26	-0.00000	27	-0.00001	28	-0.00005	29	-0.00000	30	0.00000	31	-0.00000	32	0.00002
33	0.00000	34	0.00005	35	0.00001	36	0.00013	37	0.00008	38	-0.00003	39	0.00021	40	-0.00000
41	0.00000	42	-0.00000	43	0.00000	44	-0.0002[illegible]	45	0.00000	46	-0.00023	47	0.00002	48	-0.00015
49	-0.00067	50	-0.00029	51	0.00009	52	-0.[illegible]	53	-0.00000	54	0.00000	55	-0.00000	56	0.00001
57	-0.00001	58	0.00011	59	-0.00000	60	[illegible]	61	0.00028	62	-0.00015	63	0.00156	64	0.00191
65	0.00227	66	0.00136	67	-0.00010	68	0.00[illegible]11	69	0.00000	70	0.00000	71	-0.00000	72	0.00000
73	-0.00002	74	0.00001	75	-0.00017	76	-[illegible]87	77	0.00021	78	0.00205	79	-0.01334	80	-0.00408
81	-0.00933	82	0.00138	83	-0.00009	84	-0.00063	85	0.00000	86	-0.00000	87	0.00004	88	0.00001
89	0.00020	90	0.00043	91	-0.00044	92	0.00314	93	-0.00571	94	0.09091	95	0.07899	96	-0.01070
97	0.01338	98	-0.00357	99	0.00242	100	0.10807	101	-0.00034	102	0.00016	103	0.00035	104	-0.00069
105	0.00018	106	0.00141	107	-0.01021	108	1.51944	109	2.65651	110	1.11114	111	2.63276	112	1.60860
113	0.19617	114	-0.00824	115	1.49490	116	2.128[illegible]	117	-0.03926	118	0.00138	119	-0.00061	120	0.00017
121	0.00020	122	-0.00011	123	0.00111	124	0.00[illegible]	125	0.08637	126	-0.00561	127	0.00235	[illegible]	[illegible]46024

129	2.13129	130	0.00145	131	0.00090	132	-0.00008	133	0.00018	134	-0.00015	135	0.00003	136	-0.00008
137	-0.00053	138	-0.00029	139	-0.00054	140	-0.00008	141	-0.00013	142	0.00001	143	0.00003	144	0.00000
145	0.00009	146	0.00007	147	0.00016	148	0.00008	149	0.00009	150	0.00001	151	0.00004	152	-0.00003
153	0.00001	154	-0.00002	155	-0.00007	156	-0.00004	157	-0.00007	158	-0.00002	159	-0.00003	160	0.00001

INTERVAL FROM 0.1670 TO 1.0000 DAYS

WELL PW 1 NODE 1 DISCHARGE [illegible].3380

PIEZOMETRIC SURFACE (NODE AND DRAWDOWN)

TIME 0.2086 DAYS 18027. SECONDS

1	0.00000	2	0.00000	3	-0.00000	4	-0.00000	5	0.00000	6	[illegible]	7	0.00000	8	0.00000
9	0.00000	10	0.00000	11	-0.00000	12	-0.00000	13	-0.00000	14	0.00000	15	-0.00000	16	0.00000
17	0.00000	18	0.00001	19	0.00002	20	-0.00000	21	0.00000	22	-0.00000	23	0.00001	24	0.00000
25	-0.00004	26	-0.00000	27	-0.00002	28	-0.00009	29	-0.00000	30	0.00000	31	-0.00000	32	-0.00002
33	0.00000	34	0.00011	35	0.00003	36	0.00025	37	0.00014	38	-0.00004	39	0.00037	40	-0.00000
41	0.00000	42	-0.00000	43	0.00001	44	-0.00006	45	0.00000	46	-0.00053	47	0.00006	48	-0.00037
49	-0.00129	50	-0.00053	51	0.00014	52	-0.00113	53	-0.00000	54	0.00000	55	-0.00000	56	0.00001
57	-0.00002	58	0.00018	59	-0.00000	60	-0.00000	61	0.00053	62	-0.00037	63	0.00381	64	0.00386
65	0.00429	66	0.00254	67	-0.00024	68	0.00242	69	0.00000	70	0.00000	71	-0.00000	72	0.00000
73	-0.00003	74	0.00002	75	-0.00032	76	-0.00164	77	0.00034	78	0.00435	79	-0.03177	80	0.00105
81	-0.02026	82	0.00244	83	-0.00015	84	-0.00120	85	0.00000	86	0.00000	87	0.00006	88	0.00001
89	0.00034	90	0.00069	91	-0.00071	92	0.00575	93	-0.00870	94	0.22332	95	0.21071	96	0.08904
97	0.11876	98	-0.00776	99	0.00537	100	0.26415	101	-0.00065	102	0.00026	103	0.00067	104	-0.00107
105	0.00026	106	0.00197	107	-0.01681	108	2.08760	109	3.26719	110	1.64080	111	3.24594	112	2.18081
113	0.39194	114	-0.02272	115	2.06599	116	2.71717	117	0.04416	118	0.00340	119	-0.00134	120	0.00035
121	0.00029	122	-0.00019	123	0.00181	124	0.00193	125	0.23327	126	0.00311	127	0.00369	128	16.14816
129	2.70873	130	0.00366	131	0.00246	132	-0.00027	133	0.00040	134	-0.00021	135	0.00006	136	-0.00010
137	-0.00075	138	-0.00034	139	-0.00103	140	-0.00022	141	-0.00031	142	0.00006	143	0.00004	144	-0.00000
145	0.00011	146	0.00007	147	0.00024	148	0.00013	149	0.00018	150	0.00001	151	0.00007	152	-0.00003
153	0.00002	154	-0.00002	155	-0.00009	156	-0.00005	157	-0.00012	158	-0.00004	159	-0.00006	160	0.00002

TIME 0.2503 DAYS 21626. SECONDS

1	0.00000	2	0.00000	3	-0.00000	4	-0.00000	5	0.00000	6	0.00000	7	0.00000	8	-0.00000
9	0.00000	10	0.00000	11	-0.00000	12	-0.00000	13	-0.00000	14	0.00000	15	-0.00000	16	0.00000
17	0.00000	18	0.00001	19	0.00002	20	-0.00000	21	0.00000	22	-0.00000	23	0.00001	24	0.00000
25	-0.00005	26	0.00000	27	-0.00003	28	-0.00011	29	0.00000	30	0.00000	31	-0.00000	32	0.00002
33	-0.00000	34	0.00015	35	0.00005	36	0.00034	37	0.00016	38	-0.00004	39	0.00047	40	-0.00000
41	0.00000	42	-0.00000	43	0.00001	44	-0.00006	45	0.00000	46	-0.00083	47	0.00009	48	-0.00056
49	-0.00175	50	-0.00071	51	0.00016	52	-0.00143	53	-0.00000	54	0.00000	55	-0.00000	56	0.00001

57	-0.00002	58	0.00021	59	0.00000	60	-0.00000	61	0.00070	62	-0.00057	63	0.00629	64	0.00528
65	0.00594	66	0.00317	67	-0.00034	68	0.00352	69	0.00000	70	0.00000	71	-0.00000	72	0.00000
73	-0.00003	74	0.00003	75	-0.00042	76	-0.00216	77	0.00038	78	0.00627	79	-0.05143	80	0.01555
81	-0.02966	82	0.00296	83	-0.00017	84	-0.00155	85	0.00000	86	0.00000	87	0.00006	88	-0.00000
89	0.00040	90	0.00079	91	-0.00081	92	0.00744	93	0.00629	94	0.38123	95	0.36900	96	0.21412
97	0.25263	98	-0.00736	99	0.00770	100	0.43726	101	0.00080	102	0.00028	103	0.00086	104	-0.00118
105	0.00028	106	0.00191	107	-0.01984	108	2.59379	109	3.789[illegible]	110	2.11650	111	3.77976	112	2.68834
113	0.59811	114	-0.03786	115	2.57743	116	3.23598	117	0.15264	[illegible]	-0.00523	119	-0.00189	120	0.00047
121	0.00029	122	-0.00023	123	0.00208	124	0.00171	125	0.40525	126	0.623[illegible]	127	0.00418	128	16.95023
129	3.21781	130	0.00557	131	0.00408	132	-0.00047	133	0.00056	134	-0.00021	135	0.00008	136	-0.00008
137	-0.00074	138	-0.00028	139	-0.00133	140	-0.00036	141	-0.00047	142	0.00011	143	0.00003	144	-0.00001
145	0.00010	146	0.00004	147	0.00025	148	0.00015	149	0.00023	150	0.00001	151	0.00009	152	-0.00002
153	0.00002	154	-0.00001	155	-0.00007	156	-0.00005	157	-0.00013	158	-0.00005	159	-0.00007	160	0.00003

TIME 0.2919 DAYS 25224. SECONDS

1	0.00000	2	0.00000	3	-0.00000	4	-0.00000	5	0.00000	6	0.00000	7	0.00000	8	-0.00000
9	0.00000	10	-0.00000	11	-0.00000	12	-0.00000	13	0.00000	14	0.00000	15	-0.00000	16	-0.00000
17	0.00000	18	0.00001	19	0.00002	20	-0.00000	21	0.00000	22	-0.00000	23	0.00001	24	-0.00000
25	-0.00006	26	0.00000	27	-0.00003	28	-0.00011	29	0.00000	30	0.00000	31	-0.00000	32	0.00002
33	-0.00000	34	0.00018	35	0.00006	36	0.00038	37	0.00016	38	-0.00004	39	0.00050	40	-0.00000
41	0.00000	42	-0.00000	43	0.00000	44	-0.00005	45	0.00000	46	-0.00110	47	0.00010	48	-0.00069
49	-0.00203	50	-0.00085	51	0.00016	52	-0.00153	53	-0.00000	54	0.00000	55	-0.00000	56	0.00001
57	-0.00001	58	0.00020	59	0.00000	60	0.00000	61	0.00076	62	-0.00071	63	0.00874	64	0.00591
65	0.00722	66	0.00312	67	-0.00038	68	0.00424	69	-0.00000	70	-0.00000	71	-0.00000	72	0.00000
73	-0.00002	74	0.00003	75	-0.00045	76	-0.00236	77	0.00033	78	0.00750	79	-0.07035	80	0.03958
81	-0.03605	82	0.00284	83	-0.00012	84	-0.00163	85	0.00000	86	0.00000	87	0.00005	88	-0.00002
89	0.00036	90	0.00071	91	-0.00074	92	0.00797	93	0.00256	94	0.55532	95	0.54393	96	0.35498
97	0.40556	98	-0.00109	99	0.00909	100	0.61815	101	-0.00080	102	0.00024	103	0.00089	104	-0.00099
105	0.00022	106	0.00119	107	-0.01819	108	3.04658	109	4.26885	110	2.54512	111	4.25070	112	3.14108
113	0.80597	114	-0.05195	115	3.03671	116	3.69726	117	0.27683	118	0.00657	119	-0.00220	120	0.00051
121	0.00020	122	-0.00021	123	0.00183	124	0.00075	125	0.59197	126	0.05431	127	0.00384	128	17.62082
129	3.67128	130	0.00685	131	0.00558	132	-0.00065	133	0.00065	134	-0.00015	135	0.00008	136	0.00000
137	-0.00051	138	-0.00013	139	-0.00142	140	-0.00046	141	-0.00059	142	0.00016	143	0.00001	144	-0.00002
145	0.00005	146	-0.00001	147	0.00020	148	0.00014	149	0.00024	150	0.00001	151	0.00010	152	0.00000
153	0.00001	154	0.00000	155	-0.00004	156	-0.00003	157	-0.00012	158	-0.00005	159	-0.00007	160	0.00004

TIME 0.3336 DAYS 28823. SECONDS

1	-0.00000	2	0.00000	3	-0.00000	4	-0.00000	5	-0.00000	6	0.00000	7	0.00000	8	0.00000
9	0.00000	10	-0.00000	11	-0.00000	12	-0.00000	13	0.00000	14	-0.00000	15	-0.00000	16	-0.00000
17	0.00000	18	0.00001	19	0.00002	20	0.00000	21	-0.00000	22	-0.00000	23	0.00001	24	-0.00000
25	-0.00005	26	0.00001	27	-0.00002	28	-0.00009	29	0.00000	30	0.00000	31	-0.00000	32	0.00001
33	-0.00000	34	0.00020	35	0.00006	36	0.00038	37	0.00013	38	-0.00002	39	0.00048	40	0.00000
41	-0.00000	42	-0.00000	43	0.00000	44	-0.00003	45	-0.00000	46	-0.00133	47	0.00008	48	-0.00073
49	-0.00213	50	-0.00097	51	0.00013	52	-0.00145	53	0.00000	54	-0.00000	55	0.00000	56	0.00000
57	-0.00000	58	0.00015	59	0.00000	60	0.00000	61	0.00070	62	-0.00075	63	0.01097	64	0.00565
65	0.00828	66	0.00238	67	-0.00033	68	0.00446	69	-0.00000	70	-0.00000	71	0.00000	72	0.00000

73 -0.00001	74 0.00003	75 -0.00041	76 -0.00223	77 0.00020	78 0.00790	79 -0.08719	80 0.07244
81 -0.03854	82 0.0020[illegible]	83 -0.00001	84 -0.00145	85 -0.00000	86 0.00000	87 0.00003	88 -0.00005
89 0.00025	90 0.00047	91 -0.00051	92 0.00731	93 0.01806	94 0.73873	95 0.72819	96 0.50483
97 0.57033	98 0.01155	99 0.00946	100 0.80114	101 -0.00065	102 0.00014	103 0.00078	104 -0.00053
105 0.00009	106 -0.00016	107 -0.01133	108 3.45446	109 4.68858	110 2.93393	111 4.67132	112 3.54811
113 1.01079	114 -0.06378	115 3.45165	116 4.11147	117 0.41014	118 0.00728	119 -0.00225	120 0.00047
121 0.00003	122 -0.00011	123 0.00102	124 -0.00086	125 0.78617	126 0.09555	127 0.00287	128 18.19449
129 4.07933	130 0.00733	131 0.00683	132 -0.00079	133 0.00066	134 -0.00002	135 0.00005	136 0.00012
137 -0.00012	138 0.00005	139 -0.00131	140 -0.00049	141 -0.00067	142 0.00020	143 -0.00002	144 -0.00002
145 -0.00003	146 -0.00007	147 0.00012	148 0.00010	149 0.00022	150 -0.00000	151 0.00010	152 0.00003
153 -0.00001	154 0.00002	[illegible] [illegible]00001	156 -0.00001	157 -0.00008	158 -0.00004	159 -0.00005	160 0.00005

TIME 0.3752 DAYS [illegible] SECONDS

1 -0.00000	2 -0.00000	[illegible] [illegible]00000	4 0.00000	5 -0.00000	6 -0.00000	7 0.00000	8 0.00000
9 -0.00000	10 -0.00000	[illegible] [illegible].00000	12 -0.00000	13 0.00000	14 -0.00000	15 -0.00000	16 -0.00000
17 0.00000	18 0.00000	[illegible] 0.00001	20 0.00000	21 0.00000	22 0.00000	23 0.00000	24 -0.00000
25 -0.00004	26 0.00001	27 -0.00001	28 -0.00007	29 0.00000	30 -0.00000	31 0.00000	32 0.00000
33 -0.00000	34 0.00021	35 0.00005	36 0.00034	37 0.00009	38 -0.00001	39 0.00043	40 0.00000
41 -0.00000	42 0.00000	43 -0.00000	44 -0.00000	45 -0.00000	46 -0.00149	47 0.00004	48 -0.00066
49 -0.00211	50 -0.00108	51 0.00009	52 -0.00127	53 0.00000	54 -0.00000	55 0.00000	56 -0.00001
57 0.00001	58 0.00008	59 0.00000	60 [illegible]	61 0.00055	62 -0.00068	63 0.01283	64 0.00455
65 0.00931	66 -0.00104	67 -0.00021	68 [illegible]	69 -0.00000	70 -0.00000	71 0.00000	72 -0.00000
73 0.00001	74 0.00001	75 -0.00030	76 -0.00182	77 0.00001	78 0.00744	79 -0.10107	80 0.11304
81 -0.03665	82 0.00075	83 0.00013	84 -0.00105	85 -0.00000	86 0.00000	87 -0.00001	88 -0.00008
89 0.00009	90 0.00013	91 -0.00017	92 0.00557	93 0.03993	94 0.92656	95 0.91659	96 0.65904
97 0.74158	98 0.03046	99 0.00886	100 0.98279	101 -0.00040	102 0.00000	103 0.00055	104 0.00013
105 -0.00008	106 -0.00205	107 0.00092	108 3.82474	109 5.06786	110 3.28930	111 5.05112	112 3.91705
113 1.21009	114 -0.07260	115 3.82917	116 4.48690	117 0.54796	118 0.00730	119 -0.00205	120 0.00037
121 -0.00018	122 0.00006	123 -0.00034	124 -0.00297	125 0.98289	126 0.14546	127 0.00152	128 18.69444
129 4.44981	130 0.00696	131 0.00774	132 -0.00087	133 0.00059	134 0.00015	135 0.00000	136 0.00026
137 0.00038	138 0.00022	139 -0.00102	140 -0.00046	141 -0.00070	142 0.00023	143 -0.00006	144 -0.00003
145 -0.00012	146 -0.00012	147 0.00000	148 0.00004	149 0.00016	150 -0.00001	151 0.00009	152 0.00005
153 -0.00002	154 0.00003	155 0.00006	156 0.00002	157 -0.00003	158 -0.00002	159 -0.00003	160 0.00005

TIME 0.4169 DAYS 36020. SECONDS

1 -0.00000	2 -0.00000	3 0.00000	4 0.00000	5 -0.00000	6 [illegible]00000	7 -0.00000	8 0.00000
9 -0.00000	10 -0.00000	11 0.00000	12 0.00000	13 0.00000	14 [illegible]	15 0.00000	16 -0.00000
17 -0.00000	18 -0.00000	19 0.00000	20 0.00000	21 -0.00000	22 0.00000	23 -0.00000	24 -0.00000
25 -0.00002	26 0.00002	27 0.00000	28 -0.00005	29 0.00000	30 -0.00000	31 0.00000	32 -0.00001
33 -0.00000	34 0.00020	35 0.00003	36 0.00028	37 0.00004	38 0.00001	39 0.00037	40 0.00000
41 -0.00000	42 0.00000	43 -0.00000	44 0.00002	45 -0.00000	46 -0.00159	47 -0.00002	48 -0.00047
49 -0.00202	50 -0.00119	51 0.00005	52 -0.00102	53 0.00000	54 -0.00000	55 0.00000	56 -0.00001
57 0.00002	58 -0.00000	59 0.00000	60 0.00000	61 0.00032	62 -0.00051	63 0.01423	64 0.00270
65 0.01052	66 -0.00077	67 -0.00003	68 0.00324	69 -0.00000	70 -0.00000	71 0.00000	72 -0.00000
73 0.00002	74 -0.00001	75 -0.00013	76 -0.00117	77 -0.00020	78 0.00619	79 -0.11149	80 0.16016
81 -0.03021	82 -0.00101	83 0.00031	84 -0.00048	85 -0.00001	86 0.00000	87 -0.00004	88 -0.00011

89	-0.00011	90	-0.00029	91	0.00024	92	0.00292	93	0.06762	94	1.11539	95	1.10556	96	0.81445
97	0.91551	98	0.05527	99	0.00741	100	1.16111	101	-0.00007	102	-0.00015	103	0.00024	104	0.00094
105	-0.00028	106	-0.00437	107	0.01849	108	4.16346	109	[illegible]88	110	3.61639	111	5.39732	112	4.25404
113	1.40274	114	-0.07795	115	4.17501	116	4.83004	117	0.68732	118	0.00662	119	-0.00163	120	0.00021
121	-0.00042	122	0.00030	123	-0.00221	124	-0.00541	125	1.17884	126	0.20253	127	0.00003	128	19.13687
129	4.78889	130	0.00577	131	0.00825	132	-0.00087	133	0.00045	134	0.00035	135	-0.00007	136	0.00041
137	0.00091	138	0.00034	139	-0.00063	140	-0.00036	141	-0.00069	142	0.00024	143	-0.00009	144	-0.00002
145	-0.00020	146	-0.00016	147	-0.00012	148	-0.00003	149	0.00009	150	-0.00003	151	0.00007	152	0.00007
153	-0.00004	154	0.00004	155	-0.00011	156	0.00005	157	0.00003	158	0.00000	159	-0.00000	160	0.00005

TIME 0.4585 DAYS 39619. SECONDS

1	-0.00000	2	-0.00000	3	0.00000	4	0.00000	5	-0.00000	6	-0.00000	7	-0.00000	8	0.00000
9	-0.00000	10	-0.00000	11	0.00000	12	0.00000	13	0.00000	14	-0.00000	15	0.00000	16	-0.00000
17	-0.00000	18	-0.00001	19	-0.00001	20	0.00000	21	-0.00000	22	0.00000	23	-0.00000	24	-0.00000
25	-0.00001	26	0.00002	27	0.00001	28	-0.00002	29	0.00000	30	-0.00000	31	0.00000	32	-0.00001
33	-0.00000	34	0.00017	35	-0.00001	36	0.00021	37	-0.00000	38	0.00002	39	0.00030	40	0.00000
41	-0.00000	42	0.00000	43	-0.00001	44	0.00004	45	-0.00000	46	-0.00160	47	-0.00012	48	-0.00020
49	-0.00191	50	-0.00134	51	0.00001	52	-0.00079	53	0.00000	54	-0.00000	55	0.00000	56	-0.00001
57	0.00002	58	-0.00008	59	0.00000	60	0.00000	61	0.00004	62	-0.00023	63	0.01511	64	0.00027
65	0.01210	66	-0.00288	67	0.00020	68	0.00181	69	-0.00000	70	-0.00000	71	0.00000	72	-0.00000
73	0.00004	74	-0.00004	75	0.00008	76	-0.00034	77	-0.00041	78	0.00425	79	-0.11820	80	0.21258
81	-0.01924	82	-0.00307	83	0.00050	84	-0.00019	85	-0.00001	86	0.00000	87	-0.00008	88	-0.00013
89	-0.00032	90	-0.00074	91	0.00069	92	-0.00042	93	0.10043	94	1.30287	95	1.29270	96	0.96899
97	1.08942	98	0.08539	99	0.00531	100	1.33498	101	0.00029	102	-0.00030	103	-0.00011	104	0.00183
105	-0.00050	106	-0.00697	107	0.04113	108	4.47546	109	5.73214	110	3.91938	111	5.71542	112	4.56401
113	1.58834	114	-0.07964	115	4.49383	116	5.14596	117	0.82620	118	0.00532	119	-0.00106	120	0.00002
121	-0.00067	122	0.00058	123	-0.00454	124	-0.00797	125	1.37187	126	0.26533	127	-0.00137	128	19.53349
129	5.10141	130	0.00387	131	0.00834	132	-0.00078	133	0.00027	134	0.00058	135	-0.00016	136	0.00055
137	0.00143	138	0.00041	139	-0.00017	140	-0.00018	141	[illegible]	142	0.00023	143	-0.00012	144	-0.00001
145	-0.00027	146	-0.00019	147	-0.00023	148	-0.00010	149	0.00000	150	-0.00005	151	0.00004	152	0.00008
153	-0.00006	154	0.00004	155	0.00015	156	0.00007	157	0.00003	158	0.00003	159	0.00003	160	0.00004

TIME 0.5002 DAYS 43217. SECONDS

1	-0.00000	2	-0.00000	3	0.00000	4	0.00000	5	-0.00000	6	-0.00000	7	-0.00000	8	0.00000
9	-0.00000	10	-0.00000	11	0.00000	12	0.00000	13	0.00000	14	-0.00000	15	0.00000	16	-0.00000
17	-0.00000	18	-0.00001	19	-0.00001	20	0.00000	21	-0.00000	22	0.00000	23	-0.00001	24	-0.00000
25	0.00001	26	0.00003	27	0.00002	28	-0.00000	29	-0.00000	30	-0.00000	31	0.00000	32	-0.00002
33	0.00000	34	0.00013	35	-0.00005	36	0.00014	37	-0.00003	38	0.00003	39	0.00026	40	0.00000
41	-0.00000	42	0.00000	43	-0.00001	44	0.00006	45	-0.00000	46	-0.00155	47	-0.00022	48	0.00016
49	-0.00184	50	-0.00151	51	-0.00001	52	-0.00060	53	0.00000	54	-0.00000	55	0.00000	56	-0.00002
57	0.00003	58	-0.00015	59	-0.00000	60	0.00000	61	-0.00027	62	0.00013	63	0.01544	64	-0.00257
65	0.01420	66	-0.00516	67	0.00046	68	-0.00014	69	-0.00000	70	-0.00000	71	0.00000	72	-0.00000
73	0.00005	74	-0.00006	75	0.00033	76	0.00058	77	-0.00060	78	0.00174	79	-0.12115	80	0.26920
81	-0.00391	82	-0.00529	83	0.00067	84	0.00091	85	-0.00001	86	-0.00000	87	-0.00012	88	-0.00013
89	-0.00051	90	-0.00118	91	0.00113	92	-0.00425	93	0.13767	94	1.48741	95	1.47643	96	1.12130
97	1.26149	98	0.12018	99	0.00272	100	1.50384	101	0.00064	102	-0.00044	103	-0.00048	104	0.00275

105	-0.00071	106	-0.00974	107	0.06852	108	4.76465	109	6.02688	110	4.20158	111	6.00971	112	4.85095
113	1.76689	114	-0.07761	115	4.78941	116	5.43869	117	0.96332	118	0.00348	119	-0.00037	120	-0.00019
121	-0.00091	122	0.00090	123	-0.00724	124	-0.01048	125	1.56063	126	0.33258	127	-0.00249	128	19.89288
129	5.39118	130	0.00137	131	0.00799	132	-0.00062	133	0.00004	134	0.00082	135	-0.00025	136	0.00065
137	0.00188	138	0.00039	139	0.00030	140	0.00006	141	-0.00052	142	0.00019	143	-0.00014	144	-0.00000
145	-0.00032	146	-0.00019	147	-0.00033	148	-0.00018	149	-0.00009	150	-0.00007	151	0.00001	152	0.00008
153	-0.00007	154	0.00004	155	0.00017	156	0.00009	157	0.00014	158	0.00005	159	0.00005	160	0.00003

TIME 0.5418 DAYS 46816. SECONDS

1	-0.00000	2	-0.00000	3	0.00000	4	0.00000	5	-0.00000	6	-0.00000	7	-0.00000	8	0.00000
9	-0.00000	10	-0.00000	11	0.00000	12	0.00000	13	0.00000	14	-0.00000	15	0.00000	16	-0.00000
17	-0.00000	18	-0.00001	19	-0.00001	20	0.00000	21	-0.00000	22	0.00000	23	-0.00001	24	-0.00000
25	0.00002	26	0.00003	27	0.00003	28	0.00001	29	-0.00000	30	-0.00000	31	0.00000	32	-0.00002
33	0.00000	34	0.00007	35	-0.00010	36	0.00008	37	-0.00005	38	0.00003	39	0.00023	40	0.00000
41	-0.00000	42	0.00000	43	-0.00001	44	0.00007	45	-0.00000	46	-0.00141	47	-0.00034	48	0.00057
49	-0.00185	50	-0.00172	51	-0.00002	52	-0.00051	53	0.00000	54	-0.00000	55	0.00000	56	-0.00002
57	0.00003	58	-0.00020	59	-0.00000	60	0.00000	61	-0.00059	62	0.00056	63	0.01521	64	-0.00564
65	0.01697	66	-0.00742	67	0.00071	68	-0.00255	69	-0.00000	70	0.00000	71	0.00000	72	-0.00000
73	0.00005	74	-0.00008	75	0.00061	76	-0.00156	77	-0.00077	78	-0.00117	79	-0.12039	80	-0.32906
81	0.01554	82	-0.00753	83	0.00082	84	0.00164	85	-0.00001	86	-0.00000	87	-0.00015	88	-0.00012
89	-0.00068	90	-0.00159	91	0.00154	92	-0.00837	93	0.17865	94	1.66798	95	1.65580	96	1.27052
97	1.43049	98	0.15898	99	-0.00015	100	1.66745	101	0.00097	102	-0.00054	103	-0.00084	104	0.00364
105	-0.00090	106	-0.01254	107	0.10028	108	5.03415	109	6.30145	110	4.46566	111	6.28359	112	5.11804
113	1.93861	114	-0.07196	115	5.06484	116	5.71140	117	1.09789	118	0.00119	119	0.00039	120	-0.00039
121	-0.00112	122	0.00124	123	-0.01025	124	-0.01277	125	1.74431	126	0.40316	127	-0.00315	128	20.22152
129	5.66129	130	-0.00157	131	0.00722	132	-0.00039	133	-0.00020	134	0.00105	135	-0.00035	136	0.00071
137	0.00224	138	0.00031	139	0.00073	140	0.00035	141	-0.00037	142	0.00013	143	-0.00015	144	0.00002
145	-0.00035	146	-0.00017	147	-0.00040	148	-0.00020	149	-0.00017	150	-0.00008	151	-0.00002	152	0.00007
153	-0.00009	154	0.00003	155	0.00018	156	0.00010	157	0.00018	158	0.00007	159	0.00008	160	0.00002

TIME 0.5835 DAYS 50414. SECONDS

1	-0.00000	2	-0.00000	3	0.00000	4	0.00000	5	-0.00000	6	-0.00000	7	-0.00000	8	0.00000
9	-0.00000	10	0.00000	11	0.00000	12	0.00000	13	0.00000	14	-0.00000	15	0.00000	16	0.00000
17	-0.00001	18	-0.00001	19	-0.00001	20	0.00000	21	-0.00000	22	0.00000	23	-0.00001	24	-0.00000
25	0.00004	26	0.00004	27	0.00004	28	0.00001	29	-0.00000	30	-0.00000	31	0.00000	32	-0.00002
33	0.00000	34	0.00000	35	-0.00015	36	0.00004	37	-0.00006	38	0.00004	39	0.00025	40	0.00000
41	-0.00000	42	0.00000	43	-0.00001	44	0.00007	45	-0.00000	46	-0.00121	47	-0.00047	48	0.00101
49	-0.00197	50	-0.00197	51	-0.00002	52	-0.00055	53	0.00000	54	-0.00000	55	0.00000	56	-0.00001
57	0.00003	58	-0.00023	59	-0.00000	60	0.00000	61	-0.00089	62	0.00102	63	0.01443	64	-0.00878
65	0.02051	66	-0.00952	67	0.00094	68	-0.00536	69	-0.00000	70	0.00000	71	0.00000	72	-0.00000
73	0.00005	74	-0.00010	75	0.00090	76	0.00252	77	-0.00088	78	-0.00437	79	-0.11609	80	0.39136
81	0.03883	82	-0.00967	83	0.00092	84	0.00232	85	-0.00001	86	-0.00001	87	-0.00018	88	-0.00009
89	-0.00081	90	-0.00194	91	0.00190	92	-0.01260	93	0.22275	94	1.84396	95	1.83023	96	1.41615
97	1.59567	98	0.20118	99	-0.00315	100	1.82580	101	0.00124	102	-0.00061	103	-0.00115	104	0.00447
105	-0.00107	106	-0.01526	107	0.13600	108	5.28647	109	6.55848	110	4.71376	111	6.53973	112	5.36787
113	2.10383	114	-0.06281	115	5.32256	116	5.96664	117	1.22944	118	0.00144	119	0.00116	120	-0.00059

121	-0.00129	122	0.00158	123	-0.01349	124	-0.01467	125	1.92248	126	0.47614	127	-0.00324	128	20.52437
129	5.91421	130	-0.00481	131	0.00603	132	-0.00010	133	-0.00046	134	0.00127	135	-0.00045	136	0.00070
137	0.00248	138	0.00015	139	0.00110	140	0.00067	141	-0.00019	142	0.00005	143	-0.00016	144	0.00005
145	-0.00036	146	-0.00013	147	-0.00044	148	0.00023	149	-0.00025	150	-0.00010	151	-0.00006	152	0.00006
153	-0.00009	154	0.00002	155	0.00017	156	0.00010	157	0.00020	158	0.00008	159	0.00009	160	0.00000

TIME 0.6251 DAYS 54013. SECONDS

1	-0.00000	2	-0.00000	3	0.00000	4	0.00000	5	-0.00000	6	-0.00000	7	-0.00000	8	0.00000
9	-0.00000	10	0.00000	11	0.00000	12	0.00000	13	-0.00000	14	-0.00000	15	0.00000	16	0.00000
17	-0.00001	18	-0.00001	19	-0.00001	20	0.00000	21	-0.00000	22	0.00000	23	-0.00001	24	0.00000
25	0.00004	26	0.00004	27	0.00004	28	0.00000	29	-0.00000	30	-0.00000	31	0.00000	32	-0.00002
33	0.00000	34	-0.00008	35	-0.00019	36	0.00002	37	-0.00006	38	0.00003	39	0.00030	40	0.00000
41	-0.00000	42	0.00000	43	-0.00001	44	0.00007	45	0.00000	46	-0.00094	47	-0.00058	48	0.00146
49	-0.00223	50	-0.00225	51	0.00000	52	-0.00073	53	0.00000	54	-0.00000	55	0.00000	56	-0.00001
57	0.00003	58	-0.00023	59	-0.00000	60	-0.00000	61	-0.00116	62	0.00151	63	0.01314	64	-0.01185
65	0.02490	66	-0.01132	67	0.00112	68	-0.00850	69	0.00000	70	0.00000	71	0.00000	72	-0.00000
73	0.00005	74	-0.00011	75	0.00119	76	0.00344	77	-0.00095	78	-0.00771	79	-0.10843	80	0.45540
81	0.06563	82	-0.01158	83	0.00096	84	0.00293	85	-0.00000	86	-0.00002	87	-0.00020	88	-0.00004
89	-0.00090	90	-0.00222	91	0.00218	92	-0.01678	93	0.26941	94	2.01495	95	1.99941	96	1.55791
97	1.75654	98	0.24622	99	-0.00612	100	1.97897	101	0.00145	102	-0.00064	103	-0.00142	104	0.00519
105	-0.00119	106	-0.01779	107	0.17528	108	5.52373	109	6.80012	110	4.94768	111	6.78030	112	5.60254
113	2.26293	114	-0.05038	115	5.56471	116	6.20653	117	1.35772	118	-0.00429	119	0.00192	120	-0.00076
121	-0.00141	122	0.00190	123	-0.01689	124	-0.01606	125	2.09494	126	0.55078	127	-0.00265	128	20.80530
129	6.15198	130	-0.00821	131	0.00445	132	0.00024	133	-0.00070	134	0.00146	135	-0.00054	136	0.00062
137	0.00259	138	-0.00007	139	0.00136	140	0.00100	141	0.00002	142	-0.00004	143	-0.00015	144	0.00008
145	-0.00034	146	-0.00008	147	-0.00045	148	-0.00024	149	-0.00030	150	-0.00011	151	-0.00009	152	0.00004
153	-0.00010	154	-0.00000	155	0.00015	156	0.00009	157	0.00021	158	0.00009	159	0.00010	160	-0.00001

TIME 0.6668 DAYS 57612. SECONDS

1	0.00000	2	-0.00000	3	0.00000	4	0.00000	5	0.00000	6	-0.00000	7	-0.00000	8	-0.00000
9	-0.00000	10	0.00000	11	0.00000	12	0.00000	13	-0.00000	14	0.00000	15	0.00000	16	0.00000
17	-0.00001	18	-0.00001	19	-0.00001	20	-0.00000	21	-0.00000	22	0.00000	23	-0.00000	24	0.00000
25	0.00005	26	0.00003	27	0.00003	28	-0.00001	29	-0.00000	30	-0.00000	31	0.00000	32	-0.00001
33	0.00000	34	-0.00016	35	-0.00024	36	0.00003	37	-0.00004	38	0.00003	39	0.00039	40	0.00000
41	-0.00000	42	0.00000	43	-0.00000	44	0.00005	45	0.00000	46	-0.00061	47	-0.00069	48	0.00189
49	-0.00265	50	-0.00258	51	0.00004	52	-0.00106	53	-0.00000	54	-0.00000	55	0.00000	56	-0.00001
57	0.00002	58	-0.00021	59	-0.00000	60	-0.00000	61	-0.00138	62	0.00198	63	0.01136	64	-0.01470
65	0.03019	66	-0.01272	67	0.00124	68	-0.01192	69	0.00000	70	0.00000	71	-0.00000	72	-0.00000
73	0.00004	74	-0.00011	75	0.00147	76	0.00427	77	-0.00097	78	-0.01107	79	-0.09764	80	0.52065
81	0.09562	82	-0.01318	83	0.00093	84	0.00344	85	0.00000	86	-0.00002	87	-0.00021	88	0.00003
89	-0.00093	90	-0.00241	91	0.00238	92	-0.02078	93	0.31816	94	2.18077	95	2.16329	96	1.69568
97	1.91288	98	0.29358	99	-0.00892	100	2.12711	101	0.00159	102	-0.00063	103	-0.00162	104	0.00579
105	-0.00127	106	-0.02003	107	0.21772	108	5.74763	109	7.02809	110	5.16890	111	7.00712	112	5.82386
113	2.41627	114	-0.03487	115	5.79300	116	6.43278	117	1.48263	118	-0.00727	119	0.00263	120	-0.00090
121	-0.00148	122	0.00218	123	-0.02038	124	-0.01684	125	2.26168	126	0.62644	127	-0.00132	128	21.06733
129	6.37629	130	-0.01163	131	0.00252	132	0.00062	133	-0.00093	134	0.00163	135	-0.00060	136	0.00046

137	0.00256	138	-0.00033	139	0.00150	140	0.00134	141	0.00025	142	-0.00014	143	-0.00014	144	0.00012
145	-0.00030	146	-0.00001	147	-0.00043	148	-0.00022	149	-0.00034	150	-0.00013	151	-0.00013	152	0.00002
153	-0.00010	154	-0.00002	155	0.00012	156	0.00007	157	0.00020	158	0.00009	159	0.00011	160	-0.00003

TIME 0.7084 DAYS 61210. SECONDS

1	0.00000	2	-0.00000	3	0.00000	4	0.00000	5	0.00000	6	-0.00000	7	-0.00000	8	-0.00000
9	-0.00000	10	0.00000	11	0.00000	12	0.00000	13	-0.00000	14	0.00000	15	0.00000	16	0.00000
17	-0.00001	18	-0.00000	19	-0.00000	20	-0.00000	21	0.00000	22	0.00000	23	-0.00000	24	0.00000
25	0.00005	26	0.00003	27	0.00003	28	-0.00004	29	-0.00000	30	0.00000	31	0.00000	32	-0.00000
33	0.00000	34	-0.00024	35	-0.00027	36	0.00007	37	-0.00001	38	0.00002	39	0.00051	40	-0.00000
41	0.00000	42	0.00000	43	-0.00000	44	0.00004	45	0.00000	46	-0.00024	47	-0.00077	48	0.00227
49	-0.00323	50	-0.00294	51	0.00009	52	-0.00155	53	-0.00000	54	0.00000	55	-0.00000	56	-0.00000
57	0.00001	58	-0.00016	59	-0.00000	60	-0.00000	61	-0.00155	62	0.00244	63	0.00914	64	-0.01723
65	0.03643	66	-0.01363	67	0.00128	68	-0.01554	69	0.00000	70	0.00000	71	0.00000	72	-0.00000
73	0.00003	74	-0.00010	75	0.00174	76	0.00498	77	-0.00094	78	-0.01436	79	-0.08397	80	0.58666
81	0.12847	82	-0.01437	83	0.00082	84	0.00384	85	0.00000	86	-0.00003	87	-0.00020	88	0.00013
89	-0.00092	90	-0.00252	91	0.00249	92	-0.02448	93	0.36857	94	2.34140	95	2.32188	96	1.82943
97	2.06460	98	0.34282	99	-0.01145	100	2.27046	101	0.00164	102	-0.00058	103	-0.00176	104	0.00625
105	-0.00131	106	-0.02190	107	0.26294	108	5.95956	109	7.24384	110	5.37866	111	7.22165	112	6.03326
113	2.56424	114	-0.01654	115	6.00888	116	6.64686	117	1.60417	118	-0.01027	119	0.00326	120	-0.00100
121	-0.00149	122	0.00243	123	-0.02388	124	-0.01690	125	2.42277	126	0.70265	127	0.00081	128	21.31290
129	6.58855	130	-0.01495	131	0.00026	132	0.00101	133	-0.00113	134	0.00175	135	-0.00065	136	0.00021
137	0.00240	138	-0.00063	139	0.00151	140	0.00166	141	0.00050	142	-0.00026	143	-0.00012	144	0.00016
145	-0.00025	146	0.00007	147	-0.00038	148	-0.00019	149	-0.00035	150	-0.00013	151	-0.00017	152	-0.00001
153	-0.00009	154	-0.00005	155	0.00008	156	0.00005	157	0.00018	158	0.00009	159	0.00010	160	-0.00006

TIME 0.7501 DAYS 64809. SECONDS

1	0.00000	2	0.00000	3	0.00000	4	-0.00000	5	0.00000	6	-0.00000	7	-0.00000	8	-0.00000
9	0.00000	10	0.00000	11	0.00000	12	0.00000	13	-0.00000	14	0.00000	15	-0.00000	16	0.00000
17	-0.00000	18	-0.00000	19	0.00000	20	-0.00000	21	0.00000	22	-0.00000	23	0.00000	24	0.00000
25	0.00004	26	0.00002	27	0.00002	28	-0.00007	29	-0.00000	30	0.00000	31	-0.00000	32	0.00000
33	0.00000	34	-0.00033	35	-0.00029	36	0.00013	37	0.00003	38	0.00001	39	0.00067	40	-0.00000
41	0.00000	42	-0.00000	43	-0.00000	44	0.00002	45	0.00000	46	0.00017	47	-0.00083	48	0.00268
49	-0.00398	50	-0.00333	51	0.00015	52	-0.00221	53	-0.00000	54	0.00000	55	-0.00000	56	0.00000
57	-0.00000	58	-0.00009	59	-0.00000	60	-0.00000	61	-0.00165	62	0.00285	63	0.00653	64	-0.01935
65	0.04363	66	-0.01395	67	0.00122	68	-0.01931	69	0.00000	70	0.00000	71	-0.00000	72	0.00000
73	0.00001	74	-0.00007	75	0.00197	76	0.00557	77	-0.00086	78	-0.01748	79	-0.06766	80	0.65308
81	0.16389	82	-0.01509	83	0.00062	84	0.00412	85	0.00001	86	-0.00004	87	-0.00019	88	0.00024
89	-0.00087	90	-0.00254	91	0.00250	92	-0.02778	93	0.42029	94	2.49687	95	2.47529	96	1.95923
97	2.21170	98	0.39358	99	-0.01361	100	2.40920	101	0.00162	102	-0.00049	103	-0.00183	104	0.00655
105	-0.00129	106	-0.02335	107	0.31059	108	6.16074	109	7.44859	110	5.57806	111	7.42516	112	6.23199
113	2.70717	114	0.00439	115	6.21360	116	6.84995	117	1.72237	118	-0.01322	119	0.00379	120	-0.00107
121	-0.00144	122	0.00261	123	-0.02734	124	-0.01620	125	2.57836	126	0.77900	127	0.00377	128	21.54398
129	6.79001	130	-0.01807	131	-0.00227	132	0.00140	133	-0.00130	134	0.[illegible]0184	135	-0.00068	136	-0.00013
137	0.00211	138	-0.00095	139	0.00137	140	0.00195	141	0.00077	142	-0.00037	143	-0.00009	144	0.00020
145	-0.00018	146	0.00015	147	-0.00030	148	-0.00014	149	-0.00034	150	-0.00013	151	-0.00020	152	-0.00003

153	-0.00008	154	-0.00007	155	0.00003	156	0.00002	157	0.00014	158	0.00008	159	0.00009	160	-0.00008

TIME 0.7917 DAYS 68407. SECONDS

1	0.00000	2	0.00000	3	-0.00000	4	-0.00000	5	0.00000	6	0.00000	7	0.00000	8	-0.00000
9	0.00000	10	0.00000	11	0.00000	12	-0.00000	13	-0.00000	14	0.00000	15	-0.00000	16	0.00000
17	-0.00000	18	0.00000	19	0.00001	20	-0.00000	21	0.00000	2[illegible]	[illegible].00000	23	0.00001	24	0.00000
25	0.00003	26	0.00000	27	0.00001	[illegible]8	-0.00010	29	-0.00000	3[illegible]	[illegible].00000	31	-0.00000	32	0.00001
33	0.00000	34	-0.00041	35	-0.00030	[illegible]6	0.00023	37	0.00007	38	0.[illegible]0000	39	0.00087	40	-0.00000
41	0.00000	42	-0.00000	43	0.00000	44	0.00000	45	0.00000	46	0.[illegible]61	47	-0.00085	48	0.00284
49	-0.00491	50	-0.00374	51	0.00021	52	-0.00301	53	-0.00000	54	0.[illegible]0000	55	-0.00000	56	0.00001
57	-0.00001	58	-0.00000	59	-0.00000	60	-0.00000	61	-0.00169	62	0.00320	63	0.00357	64	-0.02098
65	0.05180	66	-0.01365	67	0.00107	68	-0.02316	69	0.00000	70	0.00000	71	-0.00000	72	0.00000
73	-0.00001	74	-0.00004	75	0.00217	76	0.00600	77	-0.00073	78	-0.02035	79	-0.04894	80	0.71962
81	0.20157	82	-0.01529	83	0.00033	84	0.00427	85	0.00001	86	-0.00004	87	-0.00016	88	0.00036
89	-0.00078	90	-0.00247	91	0.00242	92	-0.03060	93	0.47304	94	2.64729	95	2.62369	96	2.08519
97	2.35427	98	0.44551	99	-0.01531	100	2.54356	101	0.00152	102	-0.00037	103	-0.00183	104	0.00671
105	-0.00122	106	-0.02431	107	0.36034	108	6.35219	109	7.64338	110	5.76802	111	7.61871	112	6.42109
113	2.84537	114	0.02768	115	6.40819	116	7.04312	117	1.83733	118	-0.01604	119	0.00422	120	-0.00109
121	-0.00134	122	0.00273	123	-0.03070	124	-0.01469	125	2.72867	126	0.85520	127	0.00757	128	21.76221
129	6.98166	130	-0.02090	131	-0.00505	132	0.00179	133	-0.00143	134	0.00187	135	-0.00067	136	-0.00056
137	0.00170	138	-0.00128	139	0.00109	140	0.00220	141	0.00104	142	-0.00048	143	-0.00005	144	0.00025
145	-0.00009	146	0.00023	147	-0.00029	148	-0.00007	149	-0.00031	150	-0.00013	151	-0.00023	152	-0.00005
153	-0.00006	154	-0.00009	155	-0.00003	156	-0.00001	157	0.00009	158	0.00006	159	0.00008	160	-0.00010

TIME 0.8334 DAYS 72006. SECONDS

1	0.00000	2	0.00000	3	-0.00000	4	-0.00000	5	0.00000	6	0.00000	7	0.00000	8	-0.00000
9	0.00000	10	0.00000	11	-0.00000	12	-0.00000	13	-0.00000	14	0.00000	15	-0.00000	16	0.00000
17	-0.00000	18	0.00001	19	0.00002	20	-0.00000	21	0.00000	22	-0.00000	23	0.00001	24	0.00000
25	0.00001	26	-0.00001	27	-0.00000	28	-0.00015	29	-0.00000	30	0.00000	31	-0.00000	32	0.00002
33	0.00000	34	-0.00049	35	-0.00030	36	0.00035	37	0.00011	38	-0.00001	39	0.00109	40	-0.00000
41	0.00000	42	-0.00000	43	0.00000	44	-0.00002	45	0.00000	46	0.00106	47	-0.00085	48	0.00300
49	-0.00600	50	-0.00418	51	0.00028	52	-0.00396	53	-0.00000	54	0.00000	55	-0.00000	56	0.00001
57	-0.00002	58	0.00010	59	-0.00000	60	-0.00000	61	-0.00167	62	0.00348	63	0.00032	64	-0.02208
65	0.06094	66	-0.01267	67	0.00081	68	-0.02703	69	0.00000	70	0.00000	71	-0.00000	72	0.00000
73	-0.00003	74	0.00001	75	0.00233	76	0.00628	77	-0.00056	78	-0.02291	79	-0.02805	80	0.78607
81	0.24125	82	-0.01493	83	-0.00004	84	0.00429	85	0.00001	86	-0.00005	87	-0.00012	88	0.00050
89	-0.00065	90	-0.00232	91	0.00226	92	-0.03289	93	0.52655	94	2.79282	95	2.76726	96	2.20742
97	2.49243	98	0.49835	99	-0.01651	100	2.67376	101	0.00135	102	-0.00023	103	-0.00176	104	0.00670
105	-0.00111	106	-0.02475	107	0.41190	108	6.53478	109	7.82908	110	5.94933	111	7.80320	112	6.60143
113	2.97913	114	0.05311	115	6.59358	116	7.22725	117	1.94916	118	-0.01865	119	0.00453	120	-0.00107
121	-0.00119	122	0.00277	123	-0.03391	124	-0.01233	125	2.87391	126	0.93101	127	0.01222	128	21.96894
129	7.16439	130	-0.02337	131	-0.00803	132	0.00215	133	-0.00153	134	0.00186	135	-0.00064	136	-0.00108
137	0.00121	138	-0.00160	139	0.00066	140	0.00240	141	0.00131	142	-0.00059	143	-0.00001	144	0.00029
145	-0.00001	146	0.00031	147	-0.00008	148	0.00001	149	-0.00026	150	-0.00012	151	-0.00026	152	-0.00007
153	-0.00003	154	-0.00010	155	-0.00008	156	-0.00004	157	0.00004	158	0.00004	159	0.00006	160	-0.00013

TIME 0.8750 DAYS 75604. SECONDS

1	0.00000	2	0.00000	3	-0.00000	4	-0.00000	5	0.00000	6	0.00000	7	0.00000	8	-0.00000
9	0.00000	10	0.00000	11	-0.00000	12	-0.00000	13	-0.00000	14	0.00000	15	-0.00000	16	0.00000
17	0.00000	18	0.00001	19	0.00003	20	-0.00000	21	0.00000	22	-0.00000	23	0.00001	24	0.00000
25	-0.00001	26	-0.00003	27	-0.00001	28	-0.00019	29	-0.00000	30	0.00000	31	-0.00000	32	0.00003
33	0.00000	34	-0.00056	35	-0.00028	36	0.00050	37	0.00015	38	-0.00001	39	0.00133	40	-0.00000
41	0.00000	42	-0.00000	43	0.00001	44	-0.00004	45	0.00000	46	0.00153	47	-0.00081	48	0.00306
49	-0.00724	50	-0.00463	51	0.00036	52	-0.00503	53	-0.00000	54	0.00000	55	-0.00000	56	0.00002
57	-0.00003	58	0.00021	59	0.00000	60	-0.00000	61	-0.00158	62	0.00368	63	-0.00319	64	-0.02259
65	0.07103	66	-0.01098	67	0.00044	68	-0.03088	69	0.00000	70	-0.00000	71	-0.00000	72	0.00000
73	0.00005	74	0.00007	75	0.00244	76	0.00641	77	-0.00037	78	-0.02512	79	-0.00520	80	0.85225
81	0.28267	82	-0.01398	83	-0.00051	84	0.00419	85	0.00001	86	-0.00005	87	-0.00006	88	0.00064
89	-0.00050	90	-0.00210	91	0.00200	92	-0.03460	93	0.58062	94	2.93363	95	2.90623	96	2.32606
97	2.62633	98	0.55186	99	-0.01714	100	2.79999	101	0.00111	102	-0.00007	103	-0.00163	104	0.00655
105	-0.00096	106	-0.02462	107	0.46499	108	6.70925	109	8.00646	110	6.12269	111	7.97943	112	6.77381
113	3.10874	114	0.08046	115	6.77054	116	7.40309	117	2.05798	118	-0.02102	119	0.00472	120	-0.00101
121	-0.00099	122	0.00273	123	-0.03693	124	-0.00910	125	3.01429	126	1.00626	127	0.01772	128	22.16533
129	7.33895	130	-0.02542	131	-0.01118	132	0.00249	133	-0.00158	134	0.00180	135	-0.00059	136	-0.00168
137	0.00063	138	-0.00190	139	0.00009	140	0.00253	141	0.00157	142	-0.00068	143	0.00004	144	0.00034
145	0.00008	146	0.00038	147	0.00006	148	0.00009	149	-0.00018	150	-0.00010	151	-0.00028	152	-0.00008
153	-0.00000	154	-0.00011	155	-0.00013	156	-0.00008	157	-0.00003	158	0.00001	159	0.00004	160	-0.00015

TIME 0.9167 DAYS 79203. SECONDS

1	0.00000	2	0.00000	3	-0.00000	4	-0.00000	5	0.00000	6	0.00000	7	0.00000	8	-0.00000
9	0.00000	10	0.00000	11	-0.00000	12	-0.00000	13	-0.00000	14	0.00000	15	-0.00001	16	0.00000
17	0.00000	18	0.00002	19	0.00003	20	-0.00000	21	0.00000	22	-0.00001	23	0.00002	24	0.00000
25	-0.00003	26	-0.00006	27	-0.00002	28	-0.00023	29	0.00000	30	0.00000	31	-0.00000	32	0.00003
33	0.00000	34	-0.00062	35	-0.00024	36	0.00066	37	0.00018	38	-0.00002	39	0.00159	40	-0.00000
41	0.00000	42	-0.00000	43	0.00001	44	-0.00005	45	0.00000	46	0.00200	47	-0.00073	48	0.00302
49	-0.00864	50	-0.00508	51	0.00043	52	-0.00623	53	-0.00000	54	0.00000	55	-0.00000	56	0.00002
57	-0.00004	58	0.00033	59	0.00000	60	-0.00000	61	-0.00144	62	0.00380	63	-0.00689	64	-0.02250
65	0.08207	66	-0.00857	67	-0.00003	68	-0.03466	69	0.00000	70	-0.00000	71	-0.00000	72	0.00001
73	-0.00006	74	0.00014	75	0.00250	76	0.00637	77	-0.00015	78	-0.02693	79	0.01942	80	0.91803
81	0.32559	82	-0.01241	83	-0.00106	84	0.00398	85	0.00001	86	-0.00005	87	0.00000	88	0.00078
89	-0.00033	90	-0.00181	91	0.00167	92	-0.03568	93	0.63506	94	3.06990	95	3.04079	96	2.44126
97	2.75614	98	0.60585	99	-0.01717	100	2.92247	101	0.00082	102	0.00010	103	-0.00146	104	0.00625
105	-0.00077	106	-0.02392	107	0.51937	108	6.87629	109	8.17621	110	6.28874	111	8.14808	112	6.93890
113	3.23443	114	0.10953	115	6.93979	116	7.57137	117	2.16392	118	-0.02308	119	0.00479	120	-0.00092
121	-0.00074	122	0.00262	123	-0.03971	124	-0.00500	125	3.15008	126	1.08080	127	0.02406	128	22.35232
129	7.50603	130	-0.02700	131	-0.01446	132	0.00277	133	-0.00158	134	0.00169	135	-0.00050	136	-0.00235
137	-0.00001	138	-0.00216	139	-0.00061	140	0.00259	141	0.00183	142	-0.00076	143	0.00010	144	0.00038
145	0.00016	146	0.00043	147	0.00020	148	0.00018	149	-0.00009	150	-0.00007	151	-0.00030	152	-0.00009
153	0.00003	154	-0.00012	155	-0.00017	156	-0.00011	157	-0.00009	158	-0.00002	159	0.00001	160	-0.00018

TIME 0.9583 DAYS 82801. SECONDS

1	0.00000	2	0.00000	3	-0.00000	4	-0.00000	5	0.00000	6	0.00000	7	0.00000	8	-0.00000
9	0.00000	10	-0.00000	11	-0.00000	12	-0.00001	13	-0.00000	14	0.00000	15	-0.00001	16	0.00000
17	0.00000	18	0.00002	19	0.00004	20	-0.00000	21	0.00000	22	-0.00001	23	0.00002	24	0.00000
25	-0.00005	26	-0.00008	27	-0.00003	28	-0.00028	29	0.00000	30	0.00000	31	-0.00000	32	0.00004
33	-0.00000	34	-0.00066	35	-0.00019	36	0.00084	37	0.00020	38	-0.00002	39	0.00186	40	-0.00000
41	0.00000	42	-0.00000	43	0.00001	44	-0.00006	45	0.00000	46	0.00246	47	-0.00062	48	0.00286
49	-0.01017	50	-0.00554	51	0.00049	52	-0.00752	53	-0.00000	54	0.00000	55	-0.00000	56	0.00002
57	-0.00004	58	0.00045	59	0.00000	60	-0.00000	61	-0.00125	62	0.00382	63	-0.01075	64	-0.02178
65	0.09402	66	-0.00542	67	-0.00060	68	-0.03832	69	0.00000	70	-0.00000	71	-0.00000	72	0.00001
73	-0.00008	74	0.00021	75	0.00251	76	0.00618	77	0.00010	78	-0.02831	79	0.04561	80	0.98333
81	0.36982	82	-0.01022	83	-0.00168	84	0.00367	85	0.00001	86	-0.00004	87	0.00008	88	0.00092
89	-0.00015	90	-0.00146	91	0.00127	92	-0.03611	93	0.68974	94	3.20183	95	3.17115	96	2.55318
97	2.88203	98	0.66015	99	-0.01658	100	3.04137	101	0.00048	102	0.00028	103	-0.00123	104	0.00582
105	-0.00055	106	-0.02261	107	0.57481	108	7.03647	109	8.33890	110	6.44804	111	8.30976	112	7.09726
113	3.35642	114	0.14014	115	7.10195	116	7.73266	117	2.26710	118	-0.02481	119	0.00473	120	-0.00080
121	-0.00046	122	0.00243	123	-0.04222	124	-0.00004	125	3.28147	126	1.15454	127	0.03122	128	22.53076
129	7.66623	130	-0.02807	131	-0.01782	132	0.00301	133	-0.00155	134	0.00152	135	-0.00039	136	-0.00309
137	-0.00069	138	-0.00237	139	-0.00143	140	0.00257	141	0.00207	142	-0.00082	143	0.00016	144	0.00042
145	0.00023	146	0.00047	147	0.00035	148	0.00027	149	0.00002	150	0.00004	151	-0.00031	152	-0.00009
153	0.00008	154	-0.00012	155	-0.00021	156	-0.00014	157	-0.00016	158	-0.00005	159	-0.00002	160	-0.00020

TIME 1.0000 DAYS 86400. SECONDS

1	0.00000	2	0.00000	3	-0.00000	4	-0.00000	5	0.00000	6	0.00000	7	0.00000	8	-0.00000
9	0.00000	10	-0.00000	11	-0.00000	12	-0.00001	13	-0.00000	14	0.00000	15	-0.00001	16	-0.00000
17	0.00001	18	0.00002	19	0.00005	20	-0.00000	21	0.00000	22	-0.00001	23	0.00002	24	0.00000
25	-0.00008	26	-0.00011	27	-0.00004	28	-0.00032	29	0.00000	30	0.00000	31	-0.00000	32	0.00004
33	-0.00000	34	-0.00069	35	-0.00012	36	0.00104	37	0.00021	38	-0.00002	39	0.00214	40	-0.00000
41	0.00000	42	-0.00000	43	0.00001	44	-0.00007	45	0.00000	46	0.00290	47	-0.00047	48	0.00260
49	-0.01182	50	-0.00598	51	0.00055	52	-0.00889	53	-0.00000	54	0.00000	55	-0.00000	56	0.00002
57	-0.00005	58	0.00057	59	0.00000	60	-0.00000	61	-0.00101	62	0.00375	63	-0.01472	64	-0.02043
65	0.10687	66	-0.00154	67	-0.00126	68	-0.04183	69	0.00000	70	-0.00000	71	-0.00000	72	0.00001
73	-0.00009	74	0.00029	75	0.00245	76	0.00585	77	0.00035	78	-0.02925	79	0.07321	80	1.04805
81	0.41514	82	-0.00740	83	-0.00238	84	0.00327	85	0.00001	86	-0.00004	87	0.00016	88	0.00104
89	0.00003	90	-0.00107	91	0.00081	92	-0.03587	93	0.74453	94	3.32962	95	3.29754	96	2.66196
97	3.00418	98	0.71462	99	-0.01534	100	3.15689	101	0.00010	102	0.00046	103	-0.00097	104	0.00526
105	-0.00029	106	-0.02070	107	0.63111	108	7.19030	109	8.49509	110	6.60106	111	8.16500	112	7.24944
113	3.47492	114	0.17212	115	7.25757	116	7.88750	117	2.36763	118	-0.02616	119	0.00456	120	-0.00064
121	-0.00015	122	0.00215	123	-0.04443	124	0.00577	125	3.40871	126	1.22740	127	0.03918	128	22.70139
129	7.82004	130	-0.02861	131	-0.02125	132	0.00319	133	-0.00147	134	0.00131	135	-0.00026	136	-0.00388
137	-0.00138	138	-0.00252	139	-0.00236	140	0.00247	141	0.00230	142	-0.00087	143	0.00022	144	0.00045
145	0.00028	146	0.00050	147	0.00051	148	0.00036	149	0.00014	150	0.00000	151	-0.00032	152	-0.00008
153	0.00012	154	-0.00012	155	-0.00024	156	-0.00017	157	-0.00023	158	-0.00008	159	-0.00005	160	-0.00021

INTERVAL FROM 1.0000 TO 7.0000 DAYS

WELL PW 1 NODE 1 DISCHARGE 1.3380

PIEZOMETRIC SURFACE (NODE AND DRAWDOWN)

TIME 1.5000 DAYS 129600. SECONDS

1	-0.00000	2	-0.00000	3	0.00000	4	0.00000	5	-0.00000	6	-0.00000	7	0.00000	8	[illegible] 00000
9	-0.00000	10	-0.00000	11	0.00000	12	-0.00000	13	0.00000	14	-0.00000	15	0.00000	16	-[illegible]00000
17	0.00000	18	-0.00001	19	0.00002	20	0.00000	21	-0.00000	22	0.00000	23	-0.00001	24	-0.[illegible]0000
25	-0.00003	26	-0.00004	27	0.00009	28	-0.00035	29	0.00000	30	-0.00000	31	0.00000	32	-0.00001
33	-0.00000	34	0.00014	35	0.00073	36	0.00155	37	-0.00195	38	0.00026	39	0.00286	40	0.00000
41	-0.00000	42	0.00000	43	-0.00000	44	0.00029	45	-0.00000	46	0.00301	47	0.00107	48	-0.00767
49	-0.03152	50	-0.00059	51	-0.00041	52	-0.01867	53	0.00000	54	-0.00000	55	0.00000	56	-0.00000
57	0.00007	58	-0.00014	59	0.00000	60	0.00000	61	0.00086	62	-0.00359	63	-0.04765	64	0.06355
65	0.33890	66	0.12035	67	-0.01002	68	-0.04391	69	-0.00000	70	-0.00000	71	0.00000	72	-0.00000
73	0.00001	74	0.00003	75	-0.00243	76	-0.00427	77	0.00064	78	0.01706	79	0.47559	80	1.73209
81	0.98367	82	0.09279	83	-0.00902	84	0.00201	85	-0.00000	86	0.00005	87	0.00020	88	-0.00070
89	-0.00087	90	0.00185	91	-0.00125	92	0.04508	93	1.35858	94	4.48969	95	4.45429	96	3.67788
97	4.12686	98	1.32684	99	0.07532	100	4.23109	101	0.00264	102	0.00092	103	0.00073	104	-0.00198
105	0.00142	106	0.07042	107	1.30119	108	8.52587	109	9.84583	110	7.93060	111	9.81209	112	8.57767
113	4.57900	114	0.61920	115	8.60322	116	9.22893	117	3.32522	118	0.01190	119	-0.00172	120	0.00053
121	0.00190	122	-0.00315	123	-0.02114	124	0.15424	125	4.56542	126	1.97799	127	0.21010	128	24.14732
129	9.15558	130	0.03042	131	-0.04560	132	-0.00051	133	0.00038	134	-0.00287	135	0.00082	136	-0.00763
137	0.00571	138	0.01290	139	-0.01245	140	-0.00460	141	0.00137	142	0.00002	143	0.00018	144	-0.00031
145	-0.00058	146	0.00050	147	0.00080	148	-0.00023	149	0.00039	150	0.00037	151	0.00007	152	-0.00008
153	0.00011	154	-0.00010	155	0.00015	156	-0.00003	157	-0.00024	158	-0.00004	159	-0.00001	160	0.00002

TIME 2.0000 DAYS 172800. SECONDS

1	-0.00000	2	-0.00000	3	0.00000	4	0.00000	5	0.00000	6	-0.00000	7	0.00000	8	0.00000
9	0.00000	10	-0.00000	11	-0.00000	12	0.00000	13	-0.00000	14	-0.00000	15	0.00001	16	-0.00000
17	0.00001	18	0.00000	19	-0.00000	20	-0.00000	21	-0.00000	22	0.00000	23	-0.00003	24	-0.00000
25	0.00017	26	0.00016	27	0.00008	28	-0.00032	29	0.00000	30	0.00000	31	-0.00000	32	-0.00002
33	-0.00000	34	0.00060	35	0.00046	36	-0.00025	37	-0.00519	38	0.00036	39	0.00138	40	-0.00000
41	0.00000	42	-0.00000	43	0.00002	44	0.00049	45	0.00000	46	-0.00075	47	0.00041	48	-0.01717
49	-0.04151	50	0.01977	51	-0.00163	52	-0.01339	53	0.00000	54	0.00000	55	-0.00000	56	0.00002
57	0.00001	58	-0.00186	59	-0.00000	60	0.00000	61	-0.00039	62	-0.01078	63	-0.05631	64	0.19114
65	0.61183	66	0.28858	67	-0.01112	68	-0.00771	69	-0.00000	70	-0.00000	71	0.00000	72	0.00000
73	-0.00003	74	-0.00125	75	-0.00652	76	-0.01036	77	-0.00137	78	0.10546	79	0.89189	80	2.33553
81	1.52676	82	0.23552	83	-0.00655	84	0.01280	85	-0.00001	86	-0.00024	87	-0.00099	88	-0.00344
89	-0.00222	90	0.00145	91	0.00678	92	0.17949	93	1.91658	94	5.40771	95	5.37457	96	4.50062
97	5.02280	98	1.88349	99	0.21763	100	5.09646	101	0.01783	102	0.00202	103	0.00055	104	-0.00026
105	0.00092	106	0.20723	107	1.92199	108	9.55116	109	10.87991	110	8.95179	111	10.84611	112	9.60166
113	5.46906	114	1.07183	115	9.63396	116	10.[illegible]42	117	4.11195	118	0.09219	119	-0.00338	120	0.00019
121	0.00099	122	-0.00331	123	0.04390	124	0.34709	125	5.48227	126	2.62683	127	0.42508	128	25.23897

129	10.18144	130	0.13505	131	-0.04564	132	-0.00508	133	0.00081	134	-0.00378	135	0.00025	136	0.00150
137	0.03422	138	0.04747	139	-0.01228	140	-0.01097	141	-0.00169	142	0.00056	143	-0.00053	144	-0.00111
145	-0.00032	146	0.00296	147	0.00072	148	-0.00150	149	-0.00089	150	0.00005	151	0.00033	152	-0.00038
153	-0.00047	154	-0.00011	155	0.00049	156	0.00016	157	0.00001	158	0.00001	159	0.00007	160	0.00023

TIME 2.5000 DAYS 216000. SECONDS

1	-0.00000	2	0.00000	3	-0.00000	4	-0.00000	5	0.00000	6	0.00000	7	-0.00000	8	0.00000
9	0.00000	10	-0.00000	11	-0.00001	12	-0.00000	13	-0.00000	14	-0.00000	15	0.00000	16	-0.00000
17	0.00002	18	0.00004	19	0.00000	20	-0.00000	21	-0.00000	22	-0.00001	23	-0.00004	24	-0.00000
25	0.00017	26	0.00013	27	-0.00028	28	-0.00057	29	0.00000	30	0.00000	31	-0.00001	32	0.00003
33	-0.00000	34	-0.00001	35	-0.00100	36	-0.00335	37	-0.00784	38	0.00013	39	-0.00058	40	0.00000
41	0.00000	42	-0.00001	43	0.00005	44	0.00031	45	-0.00000	46	-0.00585	47	-0.00218	48	-0.02116
49	-0.03796	50	0.05563	51	-0.00133	52	0.00902	53	0.00000	54	-0.00000	55	-0.00000	56	0.00006
57	-0.00025	58	-0.00272	59	0.00000	60	0.00000	61	-0.00365	62	-0.01347	63	-0.04057	64	0.34480
65	0.90196	66	0.48270	67	-0.00080	68	0.05928	69	-0.00000	70	0.00000	71	0.00000	72	-0.00001
73	-0.00032	74	-0.00230	75	-0.00584	76	-0.00733	77	-0.00330	78	0.22229	79	1.29805	80	2.87238
81	2.02940	82	0.40343	83	0.00833	84	0.03851	85	-0.00006	86	-0.00077	87	-0.00228	88	-0.00379
89	0.00004	90	-0.00037	91	0.02727	92	0.34707	93	2.42076	94	6.16219	95	6.13264	96	5.18898
97	5.76289	98	2.38658	99	0.39163	100	5.81632	101	0.04710	102	0.00706	103	0.00081	104	0.01416
105	-0.00020	106	0.37205	107	2.48417	108	10.37547	109	11.70974	110	9.77327	111	11.67716	112	10.42710
113	6.20885	114	1.50670	115	10.46156	116	11.08363	117	4.77670	118	0.20198	119	0.00404	120	-0.00035
121	-0.00144	122	0.00547	123	0.13985	124	0.56229	125	6.23634	126	3.19497	127	0.66285	128	26.10616
129	11.00662	130	0.26872	131	-0.02184	132	-0.00653	133	-0.00037	134	0.00157	135	-0.00105	136	0.02560
137	0.08374	138	0.09968	139	0.00162	140	-0.01213	141	-0.00475	142	0.00005	143	-0.00090	144	-0.00035
145	0.00373	146	0.01001	147	0.00216	148	-0.00195	149	-0.00275	150	-0.00087	151	0.00004	152	-0.00019
153	-0.00071	154	0.00072	155	0.00068	156	0.00028	157	-0.00011	158	-0.00029	159	-0.00016	160	0.00002

TIME 3.0000 DAYS 259200. SECONDS

1	-0.00000	2	-0.00000	3	0.00000	4	-0.00000	5	0.00000	6	-0.00000	7	-0.00000	8	0.00000
9	0.00000	10	-0.00000	11	-0.00000	12	-0.00000	13	-0.00000	14	-0.00000	15	0.00000	16	0.00000
17	0.00002	18	0.00002	19	0.00001	20	-0.00000	21	-0.00000	22	-0.00001	23	-0.00005	24	-0.00000
25	-0.00019	26	-0.00030	27	-0.00097	28	-0.00114	29	0.00000	30	-0.00000	31	-0.00001	32	0.00007
33	-0.00000	34	-0.00155	35	-0.00305	36	-0.00639	37	-0.00844	38	-0.00029	39	-0.00125	40	0.00000
41	0.00000	42	-0.00001	43	0.00006	44	-0.00009	45	-0.00000	46	-0.00993	47	-0.00529	48	-0.01742
49	-0.02030	50	0.10560	51	0.00198	52	0.04751	53	0.00000	54	-0.00000	55	0.00000	56	0.00007
57	-0.00042	58	-0.00103	59	-0.00000	60	0.00000	61	-0.00665	62	-0.00931	63	-0.00374	64	0.51294
65	1.19518	66	0.68949	67	0.02183	68	0.14917	69	-0.00000	70	0.00000	71	0.00001	72	-0.00004
73	-0.00064	74	-0.00157	75	0.00207	76	0.00671	77	-0.00252	78	0.35733	79	1.68463	80	3.35435
81	2.49002	82	0.58505	83	0.03644	84	0.07910	85	-0.00015	86	-0.00099	87	-0.00200	88	0.00121
89	0.00933	90	-0.00079	91	0.06022	92	0.53341	93	2.87724	94	6.80102	95	6.77469	96	5.77981
97	6.39135	98	2.84219	99	0.58322	100	6.43079	101	0.08976	102	0.01881	103	0.00438	104	0.04180
105	0.00042	106	0.55252	107	2.99070	108	11.06203	109	12.40009	110	10.45785	111	12.36894	112	11.11550
113	6.83907	114	1.91565	115	11.15028	116	11.77138	117	5.35102	118	0.33112	119	0.02206	120	0.00093
121	-0.00304	122	0.02453	123	0.25710	124	0.78666	125	6.87477	126	3.69871	127	0.90988	128	26.82211
129	11.69406	130	0.41977	131	0.02239	132	-0.00242	133	-0.00209	134	0.01463	135	-0.00129	136	0.06404
137	0.15127	138	0.16668	139	0.02951	140	-0.00582	141	-0.00601	142	-0.00122	143	0.00052	144	0.00359

145	0.01347	146	0.02285	147	0.00679	148	-0.00022	149	-0.00396	150	-0.00194	151	-0.00078	152	0.00182
153	0.00080	154	0.00359	155	0.00130	156	0.00063	157	-0.00084	158	-0.00098	159	-0.00075	160	-0.00068

TIME 3.5000 DAYS 302400. SECONDS

1	0.00000	2	-0.00000	3	0.00000	4	0.00000	5	0.00000	6	-0.00000	7	-0.00000	8	0.00000
9	-0.00000	10	-0.00000	11	0.00002	12	0.00001	13	-0.00000	14	-0.00000	15	0.00001	16	0.00000
17	-0.00005	18	-0.00011	19	-0.00004	20	-0.00000	21	-0.00000	22	-0.00001	23	-0.00011	24	0.00000
25	-0.00090	26	-0.00106	27	-0.00179	28	-0.00183	29	0.00000	30	-0.00000	31	-0.00000	32	0.00003
33	0.00000	34	-0.00338	35	-0.00485	36	-0.00815	37	-0.00594	38	-0.00056	39	0.00078	40	0.00000
41	-0.00000	42	0.00000	43	0.00003	44	-0.00032	45	0.00000	46	-0.01117	47	-0.00727	48	-0.00532
49	0.01040	50	0.16751	51	0.00927	52	0.09982	53	0.00000	54	-0.00000	55	0.00001	56	0.00002
57	-0.00009	58	0.00441	59	-0.00000	60	-0.00000	61	-0.00715	62	0.00269	63	0.05017	64	0.68816
65	1.48353	66	0.90081	67	0.05613	68	0.25535	69	-0.00000	70	0.00001	71	0.00002	72	-0.00008
73	-0.00052	74	0.00230	75	0.01849	76	0.03189	77	0.00306	78	0.50338	79	2.04868	80	3.79069
81	2.91174	82	0.77311	83	0.07728	84	0.13323	85	-0.00020	86	-0.00013	87	0.00144	88	0.01354
89	0.02782	90	0.00276	91	0.10459	92	0.72893	93	3.29255	94	7.35457	95	7.33061	96	6.29712
97	6.93678	98	3.25687	99	0.78308	100	6.96617	101	0.14431	102	0.03908	103	0.01362	104	0.08165
105	0.00498	106	0.74051	107	3.44814	108	11.64958	109	12.99045	110	11.04399	111	12.96051	112	11.70487
113	7.38681	114	2.29707	115	11.73922	116	12.35966	117	5.85606	118	0.47223	119	0.05063	120	0.00593
121	-0.00155	122	0.05382	123	0.38825	124	1.01255	125	7.42761	126	4.15038	127	1.15792	128	27.43077
129	12.28231	130	0.58043	131	0.08300	132	0.00837	133	-0.00284	134	0.03582	135	0.00140	136	0.11504
137	0.23319	138	0.24540	139	0.07012	140	0.00869	141	-0.00418	142	-0.00240	143	0.00520	144	0.01200
145	0.02995	146	0.04203	147	0.01574	148	0.00468	149	-0.00341	150	-0.00250	151	-0.00174	152	0.00705
153	0.00548	154	0.00972	155	0.00326	156	0.00177	157	-0.00199	158	-0.00178	159	-0.00147	160	-0.00155

TIME 4.0000 DAYS 345600. SECONDS

1	0.00000	2	-0.00000	3	0.00000	4	0.00000	5	0.00000	6	-0.00001	7	0.00000	8	0.00000
9	-0.00000	10	-0.00000	11	0.00005	12	0.00003	13	-0.00000	14	-0.00000	15	0.00002	16	0.00000
17	-0.00019	18	-0.00034	19	-0.00015	20	-0.00000	21	-0.00000	22	0.00000	23	-0.00021	24	0.00000
25	-0.00173	26	-0.00189	27	-0.00244	28	-0.00230	29	-0.00000	30	-0.00000	31	0.00001	32	-0.00010
33	0.00000	34	-0.00467	35	-0.00555	36	-0.00768	37	0.00029	38	-0.00027	39	0.00637	40	-0.00000
41	-0.00000	42	0.00002	43	-0.00002	44	0.00010	45	0.00000	46	-0.00834	47	-0.00667	48	0.01491
49	0.05242	50	0.23905	51	0.02108	52	0.16343	53	-0.00000	54	-0.00000	55	0.00001	56	-0.00007
57	0.00123	58	0.01433	59	-0.00000	60	-0.00000	61	-0.00336	62	0.02271	63	0.11739	64	0.86582
65	1.76285	66	1.11176	67	0.10085	68	0.37279	69	0.00000	70	0.00001	71	0.00000	72	-0.00005
73	0.00060	74	0.01033	75	0.04380	76	0.06750	77	0.01489	78	0.65551	79	2.39014	80	4.18866
81	3.29891	82	0.96307	83	0.12973	84	0.19918	85	-0.00003	86	0.00254	87	0.00928	88	0.03428
89	0.05653	90	0.01213	91	0.15895	92	0.92750	93	3.67257	94	7.84292	95	-7.82047	96	6.75720
97	7.41840	98	3.63645	99	0.98532	100	7.44035	101	0.20906	102	0.06877	103	0.03013	104	0.13217
105	0.01514	106	0.93082	107	3.86344	108	12.16304	109	13.50608	110	11.55639	111	13.47707	112	12.21985
113	7.87066	114	2.65206	115	12.25359	116	12.87354	117	6.30655	118	0.62018	119	0.08896	120	0.01609
121	0.00484	122	0.09259	123	0.52791	124	1.23564	125	7.91493	126	4.55926	127	1.40211	128	27.95998
129	12.79625	130	0.74564	131	0.15624	132	0.02616	133	-0.00116	134	0.06498	135	0.00860	136	0.17653
137	0.32606	138	0.33302	139	0.12159	140	0.03118	141	0.00150	142	-0.00243	143	0.01436	144	0.02581
145	0.05357	146	0.06760	147	0.02962	148	0.01336	149	-0.00030	150	-0.00189	151	-0.00222	152	0.01669
153	0.01459	154	0.02009	155	0.00755	156	0.00439	157	-0.00307	158	-0.00221	159	-0.00189	160	-0.00206

TIME 4.5000 DAYS 388800. SECONDS

1	0.00000	2	-0.00000	3	0.00000	4	-0.00000	5	0.00000	6	-0.00001	7	0.00001	8	-0.00000
9	-0.00000	10	-0.00000	11	0.00009	12	0.00006	13	-0.00000	14	-0.00000	15	0.00003	16	0.00000
17	-0.00038	18	-0.00062	19	-0.00033	20	-0.00000	21	0.00000	22	0.00001	23	-0.00033	24	0.00000
25	-0.00238	26	-0.00242	27	-0.00257	28	-0.00218	29	-0.00000	30	0.00000	31	0.00002	32	-0.00029
33	0.00000	34	-0.00461	35	-0.00441	36	-0.00434	37	0.01054	38	0.00097	39	0.01601	40	-0.00000
41	-0.00000	42	0.00002	43	-0.00005	44	0.00162	45	0.00000	46	-0.00072	47	-0.00239	48	0.04262
49	0.10390	50	0.31814	51	0.03758	52	0.23600	53	-0.00000	54	0.00001	55	-0.00001	56	-0.00015
57	0.00398	58	0.02907	59	-0.00000	60	-0.00000	61	0.00591	62	0.05049	63	0.19476	64	1.04302
65	2.03117	66	1.31951	67	0.15456	68	0.49781	69	0.00001	70	-0.00001	71	-0.00006	72	0.00012
73	0.00326	74	0.02314	75	0.07781	76	0.11249	77	0.03376	78	0.81042	79	2.71014	80	4.55402
81	3.65588	82	1.15222	83	0.19245	84	0.27519	85	0.00057	86	0.00769	87	0.02236	88	0.06374
89	0.09562	90	0.02848	91	0.22185	92	1.12533	93	4.02231	94	8.27988	95	8.25824	96	7.17151
97	7.84959	98	3.98594	99	1.18635	100	7.86590	101	0.28242	102	0.10806	103	0.05476	104	0.19178
105	0.03197	106	1.12023	107	4.24281	108	12.61909	109	13.96389	110	12.01165	111	13.93552	112	12.67710
113	8.30380	114	2.98272	115	12.71020	116	13.32983	117	6.71305	118	0.77147	119	0.13598	120	0.03238
121	0.01734	122	0.13986	123	0.67234	124	1.45355	125	8.35060	126	4.93247	127	1.63975	128	28.42815
129	13.25260	130	0.91213	131	0.23894	132	0.05078	133	0.00417	134	0.10169	135	0.02143	136	0.24657
137	0.42699	138	0.42720	139	0.18199	140	0.06097	141	0.01144	142	-0.00029	143	0.02892	144	0.04561
145	0.08431	146	0.09938	147	0.04866	148	0.02611	149	0.00593	150	0.00050	151	-0.00149	152	0.03162
153	0.02902	154	0.03541	155	0.01505	156	0.00917	157	-0.00346	158	-0.00169	159	-0.00148	160	-0.00156

TIME 5.0000 DAYS 432000. SECONDS

1	-0.00000	2	0.00000	3	-0.00000	4	-0.00000	5	0.00000	6	-0.00000	7	0.00001	8	-0.00000
9	-0.00000	10	-0.00000	11	0.00011	12	0.00008	13	-0.00000	14	-0.00000	15	0.00003	16	-0.00000
17	-0.00055	18	-0.00083	19	-0.00052	20	-0.00000	21	0.00000	22	0.00001	23	-0.00042	24	0.00000
25	-0.00248	26	-0.00225	27	-0.00186	28	-0.00109	29	-0.00000	30	0.00000	31	0.00001	32	-0.00049
33	-0.00000	34	-0.00248	35	-0.00087	36	0.00224	37	0.02484	38	0.00352	39	0.02985	40	-0.00000
41	0.00000	42	0.00001	43	0.00002	44	0.00465	45	-0.00000	46	0.01205	47	0.00629	48	0.07704
49	0.16309	50	0.40298	51	0.05872	52	0.31553	53	-0.00000	54	0.00003	55	-0.00007	56	-0.00012
57	0.00856	58	0.04870	59	-0.00000	60	-0.00000	61	0.02134	62	0.08552	63	0.27974	64	1.21796
65	2.28783	66	1.52248	67	0.21587	68	0.62776	69	0.00001	70	-0.00006	71	-0.00018	72	0.00055
73	0.00794	74	0.04107	75	0.12004	76	0.16573	77	0.05995	78	0.96594	79	3.01035	80	4.89143
81	3.98656	82	1.33892	83	0.26409	84	0.35966	85	0.00182	86	0.01579	87	0.04109	88	-0.10177
89	0.14468	90	0.05237	91	0.29199	92	1.32016	93	4.34591	94	8.67531	95	8.65395	96	7.54834
97	8.23996	98	4.30948	99	1.38410	100	8.25193	101	0.36298	102	0.15666	103	0.08778	104	0.25905
105	0.05603	106	1.30681	107	4.59150	108	13.02944	109	14.37571	110	12.42137	111	14.34773	112	13.08834
113	8.69579	114	3.29139	115	13.12086	116	13.74027	117	7.08335	118	0.92378	119	0.19059	120	0.05530
121	0.03667	122	0.19466	123	0.81896	124	1.66507	125	8.74454	126	5.27556	127	1.86948	128	28.84806
129	13.66306	130	1.07778	131	0.32854	132	0.08183	133	0.01402	134	0.14540	135	0.04061	136	0.32348
137	0.53365	138	0.52604	139	0.24952	140	0.09723	141	0.02579	142	0.00486	143	0.04948	144	0.07167
145	0.12191	146	0.13701	147	0.07285	148	0.04303	149	0.01556	150	0.00518	151	0.00113	152	0.05239
153	0.04939	154	0.05610	155	0.02644	156	0.01669	157	-0.00249	158	0.00034	159	0.00027	160	0.00060

TIME 5.5000 DAYS 475200. SECONDS

1	-0.00000	2	0.00000	3	-0.00000	4	-0.00000	5	0.00000	6	0.00001	7	0.00002	8	-0.00000
9	0.00000	10	-0.00000	11	0.00010	12	0.00008	13	-0.00000	14	0.00000	15	0.00002	16	-0.00000
17	-0.00062	18	-0.00084	19	-0.00067	20	0.00000	21	-0.00000	22	-0.00001	23	-0.00042	24	-0.00000
25	-0.00170	26	-0.00104	27	-0.00000	28	0.00126	29	0.00000	30	0.00000	31	-0.00003	32	-0.00060
33	-0.00000	34	0.00222	35	0.00547	36	0.01226	37	0.04311	38	0.00763	39	0.04785	40	-0.00000
41	0.00001	42	-0.00005	43	0.00026	44	0.00948	45	-0.00000	46	0.03005	47	0.01977	48	0.11733
49	0.22847	50	0.49209	51	0.08431	52	0.40040	53	-0.00000	54	0.00006	55	-0.00018	56	0.00011
57	0.01524	58	0.07313	59	-0.00000	60	0.00000	61	0.04317	62	0.12720	63	0.37037	64	1.38959
65	2.53285	66	1.71989	67	0.28356	68	0.76075	69	0.00003	70	-0.00014	71	-0.00039	72	0.00136
73	0.01499	74	0.06417	75	0.16979	76	0.22616	77	0.09340	78	1.12065	79	3.29248	80	5.20463
81	4.29429	82	1.52228	83	0.34336	84	0.45119	85	0.00393	86	0.02717	87	0.06563	88	0.14786
89	0.20298	90	0.08390	91	0.36823	92	1.51072	93	4.64684	94	9.03648	95	9.01501	96	7.89396
97	8.59666	98	4.61051	99	1.57743	100	8.60521	101	0.44956	102	0.21398	103	0.12905	104	0.33275
105	0.08746	106	1.48949	107	4.91385	108	13.40254	109	14.75004	110	12.79394	111	14.72227	112	13.46206
113	9.05378	114	3.58032	115	13.49408	116	14.11334	117	7.42337	118	1.07559	119	0.25178	120	0.08498
121	0.06314	122	0.25605	123	0.96606	124	1.86966	125	9.10410	126	5.59292	127	2.09071	128	29.22887
129	14.03613	130	1.24125	131	0.42305	132	0.11881	133	0.02894	134	0.19556	135	0.06649	136	0.40587
137	0.64426	138	0.62808	139	0.32263	140	0.13907	141	0.04456	142	0.01368	143	0.07636	144	0.10404
145	0.16593	146	0.18006	147	0.10201	148	0.06408	149	0.02875	150	0.01256	151	0.00628	152	0.07927
153	0.07600	154	0.08235	155	0.04218	156	0.02742	157	0.00042	158	0.00438	159	0.00387	160	0.00501

TIME 6.0000 DAYS 518400. SECONDS

1	-0.00000	2	0.00000	3	-0.00000	4	-0.00000	5	-0.00000	6	0.00002	7	0.00003	8	0.00000
9	0.00000	10	-0.00000	11	0.00004	12	0.00006	13	-0.00000	14	-0.00000	15	-0.00002	16	-0.00000
17	-0.00050	18	-0.00050	19	-0.00069	20	0.00000	21	-0.00001	22	-0.00003	23	-0.00025	24	-0.00000
25	0.00027	26	0.00155	27	0.00323	28	0.00511	29	0.00000	30	-0.00000	31	-0.00010	32	-0.00054
33	-0.00000	34	0.00989	35	0.01487	36	0.02577	37	0.06514	38	0.01351	39	0.06983	40	-0.00001
41	0.00001	42	-0.00016	43	0.00077	44	0.01632	45	-0.00000	46	0.05320	47	0.03822	48	0.16275
49	0.29873	50	0.58431	51	0.11406	52	0.48929	53	-0.00000	54	0.00010	55	-0.00038	56	0.00066
57	0.02421	58	0.10213	59	0.00000	60	0.00000	61	0.07139	62	0.17489	63	0.46513	64	1.55731
65	2.76664	66	1.91140	67	0.35655	68	0.89546	69	0.00004	70	-0.00028	71	-0.00070	72	0.00266
73	0.02467	74	0.09234	75	0.22631	76	0.29278	77	0.13381	78	1.27366	79	3.55822	80	5.49675
81	4.58194	82	1.70183	83	0.42910	84	0.54857	85	0.00706	86	0.04200	87	0.09586	88	0.20138
89	0.26961	90	0.12288	91	0.44960	92	1.69634	93	4.92798	94	9.36889	95	9.34705	96	8.21315
97	8.92509	98	4.89192	99	1.76579	100	8.93093	101	0.54113	102	0.27925	103	0.17817	104	0.41186
105	0.12613	106	1.66772	107	5.21344	108	13.74470	109	15.09323	110	13.13564	111	15.06555	112	13.80463
113	9.38323	114	3.85158	115	13.83622	116	14.45542	117	7.73770	118	1.22594	119	0.31865	120	0.12133
121	0.09675	122	0.32323	123	1.11253	124	2.06720	125	9.43483	126	5.88805	127	2.30336	128	29.57736
129	14.37813	130	1.40173	131	0.52097	132	0.16118	133	0.04925	134	0.25160	135	0.09914	136	0.49267
137	0.75745	138	0.73220	139	0.40003	140	0.18570	141	0.06764	142	0.02662	143	0.10963	144	0.14261
145	0.21589	146	0.22811	147	0.13587	148	0.08912	149	0.04551	150	0.02294	151	0.01445	152	0.11230
153	0.10894	154	0.11415	155	0.06255	156	0.04171	157	0.00581	158	0.01085	159	0.00971	160	0.01216

TIME 6.5000 DAYS 561600. SECONDS

1	-0.00000	2	-0.00000	3	-0.00000	4	0.00000	5	-0.00000	6	0.00003	7	0.00002	8	0.00000
9	0.00001	10	-0.00000	11	-0.00008	12	-0.00000	13	-0.00000	14	-0.00001	15	-0.00007	16	-0.00000
17	-0.00009	18	0.00033	19	-0.00049	20	-0.00000	21	-0.00003	22	-0.00006	23	0.00016	24	-0.00000
25	0.00368	26	0.00576	27	0.00804	28	0.01064	29	0.00000	30	-0.00001	31	-0.00021	32	-0.00022
33	-0.00001	34	0.02074	35	0.02747	36	0.04271	37	0.09068	38	0.02127	39	0.09555	40	-0.00001
41	0.00001	42	-0.00034	43	0.00161	44	0.02530	45	-0.00000	46	0.08128	47	0.06159	48	0.21261
49	0.37282	50	0.67870	51	0.14765	52	0.58118	53	0.00000	54	0.00015	55	-0.00066	56	0.00160
57	0.03558	58	0.13542	59	0.00000	60	0.00000	61	0.10575	62	0.22795	63	0.56287	64	1.72083
65	2.98983	66	2.09699	67	0.43392	68	1.03093	69	0.00006	70	-0.00046	71	-0.00111	72	0.00451
73	0.03710	74	0.12535	75	0.28885	76	0.36472	77	0.18075	78	1.42442	79	3.80914	80	5.77038
81	4.85192	82	1.87738	83	0.52028	84	0.65076	85	0.01134	86	0.06032	87	0.13153	88	0.26157
89	0.34364	90	0.16892	91	0.53528	92	1.87679	93	5.19174	94	9.67681	95	9.65444	96	8.50970
97	9.22947	98	5.15611	99	1.94900	100	9.23314	101	0.63687	102	0.35167	103	0.23458	104	0.49554
105	0.17172	106	1.84129	107	5.49323	108	14.06075	109	15.41016	110	13.45125	111	15.38244	112	14.12091
113	9.68840	114	4.10704	115	14.15214	116	14.77132	117	8.02995	118	1.37421	119	0.39041	120	0.16408
121	0.13732	122	0.39549	123	1.25766	124	2.25783	125	9.74105	126	6.16384	127	2.50761	128	29.89868
129	14.69392	130	1.55871	131	0.62114	132	0.20845	133	0.07504	134	0.31301	135	0.13845	136	0.58299
137	0.87222	138	0.83756	139	0.48067	140	0.23640	141	0.09489	142	0.04395	143	0.14923	144	0.18716
145	0.27129	146	0.28071	147	0.17413	148	0.11800	149	0.06580	150	0.03654	151	0.02603	152	0.15134
153	0.14813	154	0.15139	155	0.08766	156	0.05976	157	0.01406	158	0.02006	159	0.01813	160	0.02244

TIME 7.0000 DAYS 604800. SECONDS

1	-0.00000	2	-0.00000	3	-0.00000	4	0.00000	5	-0.00000	6	0.00005	7	0.00001	8	0.00000
9	0.00000	10	-0.00000	11	-0.00027	12	-0.00011	13	-0.00000	14	-0.00002	15	-0.00015	16	-0.00000
17	0.00068	18	0.00177	19	0.00000	20	-0.00000	21	-0.00007	22	-0.00010	23	0.00088	24	0.00000
25	0.00873	26	0.01178	27	0.01454	28	0.01797	29	0.00000	30	-0.00005	31	-0.00035	32	0.00043
33	-0.00001	34	0.03487	35	0.04328	36	0.06298	37	0.11947	38	0.03098	39	0.12472	40	-0.00001
41	0.00000	42	-0.00059	43	0.00284	44	0.03648	45	-0.00000	46	0.11399	47	0.08973	48	0.26629
49	0.44987	50	0.77455	51	0.18475	52	0.67526	53	0.00001	54	0.00021	55	-0.00104	56	0.00302
57	0.04940	58	0.17267	59	0.00000	60	0.00001	61	0.14590	62	0.28580	63	0.66270	64	1.88005
65	3.20312	66	2.27678	67	0.51490	68	1.16651	69	0.00007	70	-0.00071	71	-0.00164	72	0.00700
73	0.05235	74	0.16289	75	0.35666	76	0.44123	77	0.23368	78	1.57265	79	4.04669	80	6.02769
81	5.10631	82	2.04891	83	0.61600	84	0.75690	85	0.01684	86	0.08208	87	0.17230	88	0.32770
89	0.42413	90	0.22153	91	0.62461	92	2.05206	93	5.44015	94	9.96366	95	9.94065	96	8.78664
97	9.51317	98	5.40510	99	2.12708	100	9.51509	101	0.73607	102	0.43040	103	0.29766	104	0.58310
105	0.22382	106	2.01021	107	5.75566	108	14.35448	109	15.70463	110	13.74456	111	15.67680	112	14.41471
113	9.97266	114	4.34833	115	14.44563	116	15.06485	117	8.30307	118	1.52007	119	0.46641	120	0.21286
121	0.18450	122	0.47224	123	1.40106	124	2.44180	125	10.02621	126	6.42266	127	2.70381	128	30.19682
129	14.98729	130	1.71196	131	0.72271	132	0.26015	133	0.10625	134	0.37932	135	0.18416	136	0.67615
137	0.98784	138	0.94355	139	0.56371	140	0.29059	141	0.12614	142	0.06581	143	0.19494	144	0.23737
145	0.33163	146	0.33745	147	0.21648	148	0.15050	149	0.08953	150	0.05348	151	0.04128	152	0.19618
153	0.19335	154	0.19383	155	0.11750	156	0.08169	157	0.02548	158	0.03226	159	0.02938	160	0.03612

FINAL BANDWIDTH 21