4-1972

Digital Computer Modeling of Limestone Groundwater Systems

Digital Object Identifier: https://doi.org/10.13023/kwrri.rr.50

John Thrailkill
University of Kentucky

David P. Beiter
University of Kentucky

Right click to open a feedback form in a new tab to let us know how this document benefits you.

Follow this and additional works at: https://uknowledge.uky.edu/kwrri_reports

Part of the Computer Sciences Commons, Earth Sciences Commons, and the Environmental Sciences Commons

Repository Citation
https://uknowledge.uky.edu/kwrri_reports/145

This Report is brought to you for free and open access by the Kentucky Water Resources Research Institute at UKnowledge. It has been accepted for inclusion in KWRRI Research Reports by an authorized administrator of UKnowledge. For more information, please contact UKnowledge@lsv.uky.edu.
DIGITAL COMPUTER MODELING OF LIMESTONE GROUNDWATER SYSTEMS

Dr. John Thrailkill
Principal Investigator

Graduate Student Assistant: David P. Beiter

Project Number A-030-KY (Completion Report)
Agreement Number 14-31-0001-3217 (FY 1971)
Period of Project - July, 1970 - August, 1971

University of Kentucky Water Resources Institute
Lexington, Kentucky

The work on which this report is based was supported in part by funds provided by the Office of Water Resources Research, United States Department of the Interior, as authorized under the Water Resources Research Act of 1964.

April, 1972
ABSTRACT

Because limestone groundwater flows mainly in discrete openings, limestone aquifers are fundamentally different from aquifers in granular rocks. A digital computer program which simulates flow in a limestone aquifer as a pipe network was written and compared with the Sinkhole Plain aquifer of west-central Kentucky.

A reasonably good fit between observed parameters of the aquifer and those calculated were obtained under assumed conditions of both laminar and turbulent flow in the aquifer. The indicated gross permeability of the aquifer is 5600 meaners with an assumed aquifer thickness of 100 feet. The location and discharge of springs along the streams bounding the aquifer are predicted.

With further refinements to the computational routines, additional features of the aquifer can be modelled, and more refined predictions can be made of water budget parameters, location of flow paths, and development of the aquifer.

KEY WORDS: limestone aquifer/ground-water basin/digital computer models/flownets.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>CHAPTERS</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ABSTRACT</strong></td>
<td></td>
</tr>
<tr>
<td><strong>ACKNOWLEDGMENTS</strong></td>
<td></td>
</tr>
<tr>
<td><strong>LIST OF TABLES</strong></td>
<td></td>
</tr>
<tr>
<td><strong>LIST OF FIGURES</strong></td>
<td></td>
</tr>
<tr>
<td><strong>CHAPTERS</strong></td>
<td></td>
</tr>
<tr>
<td>I  INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>Significance of Research</td>
<td>1</td>
</tr>
<tr>
<td>Project Objectives</td>
<td>2</td>
</tr>
<tr>
<td>II DESCRIPTION OF PROGRAM KLAM</td>
<td>3</td>
</tr>
<tr>
<td>(Kentucky Limestone Aquifer Model)</td>
<td></td>
</tr>
<tr>
<td>Geometry</td>
<td>4</td>
</tr>
<tr>
<td>General Computational Procedure</td>
<td>4</td>
</tr>
<tr>
<td>Summary of Program</td>
<td>5</td>
</tr>
<tr>
<td>III MODEL 1 - SINKHOLE PLAIN AQUIFER-LAMINAR FLOW</td>
<td>6</td>
</tr>
<tr>
<td>General</td>
<td>6</td>
</tr>
<tr>
<td>Parameters</td>
<td>8</td>
</tr>
<tr>
<td>Results</td>
<td>8</td>
</tr>
<tr>
<td>IV MODEL 2 - SINKHOLE PLAIN AQUIFER-TURBULENT FLOW</td>
<td>16</td>
</tr>
<tr>
<td>General and Parameters</td>
<td>16</td>
</tr>
<tr>
<td>Results</td>
<td>16</td>
</tr>
<tr>
<td>V  MODEL 3 - NORTH-CENTRAL SINKHOLE PLAIN AQUIFER-LAMINAR FLOW</td>
<td>20</td>
</tr>
<tr>
<td>General and Parameters</td>
<td>20</td>
</tr>
<tr>
<td>Results</td>
<td>23</td>
</tr>
<tr>
<td>VI CONCLUDING SUMMARY</td>
<td>25</td>
</tr>
<tr>
<td><strong>PUBLICATIONS AND TRAINING ACCOMPLISHED</strong></td>
<td>26</td>
</tr>
<tr>
<td><strong>REFERENCES</strong></td>
<td>27</td>
</tr>
</tbody>
</table>

APPENDIX - KENTUCKY LIMESTONE AQUIFER MODEL

iv
## LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Comparison of Observed and Calculated Heads for Model 1 (feet)</td>
<td>13</td>
</tr>
<tr>
<td>2</td>
<td>Comparison of Observed Flows in the Aquifer and Model 1 Calculated Branch Flows (ft³/min)</td>
<td>15</td>
</tr>
<tr>
<td>3</td>
<td>Comparison of Observed and Model 1 Calculated Discharges on the Aquifer Boundary (ft³/min)</td>
<td>15</td>
</tr>
<tr>
<td>4</td>
<td>Comparison of Observed and Calculated Heads for Model 2 (feet)</td>
<td>18</td>
</tr>
<tr>
<td>5</td>
<td>Comparison of Observed Flows in the Aquifer and Model 2 Calculated Branch Flows (ft³/min)</td>
<td>19</td>
</tr>
<tr>
<td>6</td>
<td>Comparison of Observed and Model 2 Calculated Discharges on the Aquifer Boundary (ft³/min)</td>
<td>19</td>
</tr>
<tr>
<td>7</td>
<td>Comparison of Observed and Model 3 Calculated Flows and Discharges (ft³/min)</td>
<td>23</td>
</tr>
<tr>
<td>Figure</td>
<td>Illustration</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>------------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>1</td>
<td>Indexing Scheme for Network Parameters</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>Sinkhole Plain Aquifer - Location Map</td>
<td>7</td>
</tr>
<tr>
<td>3</td>
<td>Sinkhole Plain Aquifer - Network and Values of Specified Parameters</td>
<td>9</td>
</tr>
<tr>
<td>4</td>
<td>Sinkhole Plain Aquifer - Model 1 Values of Flow in Branches</td>
<td>10</td>
</tr>
<tr>
<td>5</td>
<td>Sinkhole Plain Aquifer - Model 1 Values of Recharge</td>
<td>11</td>
</tr>
<tr>
<td>6</td>
<td>Sinkhole Plain Aquifer - Model 1 Values of Head</td>
<td>12</td>
</tr>
<tr>
<td>7</td>
<td>Sinkhole Plain Aquifer - Model 2 Values of Head</td>
<td>17</td>
</tr>
<tr>
<td>8</td>
<td>North-Central Sinkhole Plain Aquifer - Geologic Map</td>
<td>21</td>
</tr>
<tr>
<td>9</td>
<td>North-Central Sinkhole Plain Aquifer - Model 3 Recharge</td>
<td>22</td>
</tr>
<tr>
<td>10</td>
<td>North-Central Sinkhole Plain Aquifer - Model 3 Values of Head</td>
<td>24</td>
</tr>
</tbody>
</table>
CHAPTER I
INTRODUCTION

Significance of Research

In many areas, water found in openings in the rock (groundwater) is a major water supply. Where the underlying rock is granular, such as the alluvium along major streams, the study, evaluation, and production of this groundwater is facilitated by the existence of a well-developed theory of the granular groundwater body.

Much of Kentucky, as well as large portions of other states, is underlain by limestone. Groundwater in limestone (and similar rocks) behaves quite differently than that in granular rocks. The permeability is localized in discrete openings, rather than in the intergranular pores of granular rocks, and the flow is often turbulent. The distribution of openings is irregular, and the openings themselves have been and are being enlarged by the flow of water within them.

Although flow assumptions borrowed from granular rock theory (e.g., linear relationship between head loss and flow) are widely applied (with varying degrees of success) to limestones, it is evident that concepts and methods which take into account the nature of the limestone aquifer need to be developed.

The importance of limestone groundwater to Kentucky may be judged by the following summary derived from data in Mull, et al. (1971).

Twenty-nine of Kentucky’s 120 counties obtain some of their public and industrial water from limestone groundwater. Of these, 7 derive between 10 and 50% of their supply from this source, and six obtain more than half their supply from limestone. In this latter group, Hart, Russell, and Scott counties get more than 80% of their public and industrial water supply from this source, and Allen county receives all of its supply from limestone groundwater.
Currently, the probabilities of obtaining a yield sufficient for public or industrial water supply from a well drilled in a limestone area are quite low. An inspection of the data in Mull, et al (1971) shows only about one-fifth of the wells listed have a yield of 100 gallons/minute or greater. Large flows of water do exist, however, as evidenced by several springs discharging more than 1000 gallons/minute. It is evident that the development of a theory of limestone groundwater which would allow the prediction of the location of these large flows would be of considerable economic benefit.

A better understanding of groundwater in limestone would have benefits other than water supply. The prediction of reservoir leakage and of pollution paths is currently a difficult matter in the absence of a well-developed theory of limestone groundwater.

Project Objectives

The original objectives of the project (as described in the original proposal) were:

To develop digital computer model techniques to describe limestone groundwater basins in two and three dimensions. Models will be developed for actual limestone aquifer systems with boundary conditions determined by geologic and geochemical data now available or to be acquired as part of the project. Numerical techniques will be used to obtain solutions to the models. Results will be made available on specific systems to guide groundwater surface water development and aid in pollution control. Training of one or two graduate students in the techniques developed will be accomplished.

Midway through the first year of the planned two-year period the project was revised by shortening it to 14 months and reducing the funds to 42% of the original amount. This necessitated a drastic limitation of the project objectives, as discussed in a later section of this report.
CHAPTER II

DESCRIPTION OF PROGRAM KLAM
(KENTUCKY LIMESTONE AQUIFER MODEL)

The aquifer is approximated as a 2-dimensional quadrilateral network of pipes. Each branch (pipe segment) has assigned or calculated values of flow volume, flow resistance (permeability), and length. Each node (junction of pipe segments) has assigned or calculated values of head and flow volume introduced from outside the system (recharge flow). The network is numbered as being in the first quadrant. The indexing scheme for nodes and branches is shown in Figure 1, together with the variables and the sign convention for flows.

\[ H, FX, FY, FR \]  
\[ KX, KY \]  
\[ LX, LY \]  

INDEXING SCHEME FOR NETWORK PARAMETERS

\textbf{FIGURE 1}
Geometry

1. The node flow error \( E \) at a node is calculated as
\[
E_{i,j} = FX_{i,j} + FY_{i,j} - FR_{i,j} - FX_{i-1,j} - FY_{i,j-1}
\]
where the symbols are as shown in Figure 1.

2. The branch flow error \( E_- \) in the four branches associated with the node is calculated as
\[
E_{X_{i,j}} = \frac{KX_{i,j} \cdot (H_{i,j} - H_{i+1,j})}{LX_{i,j}} - FX_{i,j}
\]
for the branch to the right of the node and similarly for the remaining branches. In this expression \( n \) is 1 for the laminar flow models and 2 for the turbulent flow models.

3. The flow in each branch (including the recharge flow) about the node is corrected by an amount \( C_- \) calculated as
\[
C_{X_{i,j}} = \frac{-R \cdot (1 - W) \cdot E_{i,j}}{M_{i,j}} + W \cdot E_{X_{i,j}}
\]
for the branch to the right of the node and similarly for the remaining branches. \( R \) is a relaxation coefficient, \( W \) a parameter to allow the relative influence of the two errors to be changed, and \( M \) is the number of branches around the node.

4. The head at the node is calculated from the heads at adjacent nodes and the new branch flows by averaging expressions similar to
\[
H_{i,j} = \frac{LX_{i,j} \cdot (FX_{i,j})^n}{KX_{i,j}} + H_{i+1,j}
\]

5. After the above computations are performed for each node, the entire procedure is iterated until convergence occurs.
Summary of Program

The program is written in Fortran IV and executed on an IBM 360/65 computer. When dimensioned to accommodate a 36 x 27 node network, 118K bytes of storage were required and 100 iterations required about 8 minutes of central processing unit time.

The program consists of a main program and 14 sub-routines.

1. MAIN - reads program parameter card, writes out parameters, and performs iterative computations described above.
2. IN1 - reads data for each node on separate card.
3. IN2 - reads data for all nodes on first card and changes for individual nodes on later cards.
4. IN3 - reads data for each node on cards punched by OUT4.
5. CAL1 - calculates KX and KY if undefined.
6. CAL2 - counts entered value and writes messages.
7. CAL3 - assigns starting values of unspecified flows.
8. CAL4 - assigns starting values of unspecified heads.
9. OUT1 - lists values after specified number of iterations.
10. OUT2 - lists final values in table form.
11. OUT3 - fills arrays for map output of final values.
12. OUT3A - lists final values in map form.
13. OUT3B - writes heading for map output.
14. OUT3C - writes error table for map output.
15. OUT4 - punches cards with final values.

In the initial stages, the program was titled LASP but this was changed to KLAM (Kentucky Limestone Aquifer Model) during the project. A complete listing of the program (slightly expanded for readability) will be found in the Appendix.
CHAPTER III

MODEL 1 - SINKHOLE PLAIN AQUIFER - LAMINAR FLOW

General

The Sinkhole Plain Aquifer was considered to be a continuous aquifer underlying the area shown on Figure 2. Its boundaries were taken to be the streams indicated, which are perennial and of sufficient size to be represented as double lines on 1/24,000 scale maps.

All of the rocks of the area are nearly flat-lying, with a general regional dip to the north of about 10 meters per kilometer. Pennsylvanian rocks crop out near the Green and Barren rivers in the northwest part of the area, otherwise only rocks of Mississippian age outcrop. The Pennsylvanian and upper Mississippian rocks are interbedded sandstones and shales with thin limestones in the Mississippian. The lower Mississippian rocks are almost entirely limestones and dolomites.

North of the south-facing Chester Escarpment, which roughly follows a line connecting the towns of Munfordville, Horse Cave, Cave City, Park City, and Bowling Green, most of the area is underlain by upper Mississippian and Pennsylvanian rocks, with the lower Mississippian limestones and dolomites cropping out only in the bottom of sinkholes and valleys. The portion of the area south of the escarpment, known as the Sinkhole Plain, is underlain by lower Mississippian rocks and is a typical karst, with few surface streams and, in many areas, a very high density of sinkholes. The average altitude is about 220 meters (750 feet) with an average local relief of about 20 meters (65 feet). The relief is fine-textured; a characteristic sinkhole diameter being 100 meters (300 feet).

The eastern part of the plateau north of the Chester escarpment is also a karst, but of a significantly different form. The average altitude of the
plateau tops is about 260 meters (850 feet) and that of the intervening sinkholes is 200 meters (650 feet). The local relief is thus three times that of the Sinkhole Plain. The texture is also much coarser with an average sinkhole diameter of about 1 km.

Parameters

A grid with a standard node interval of 12,500 feet was established (Figure 3). Nodes on the boundary were moved to fall on the bounding streams and the lengths of the branches connecting these nodes to the grid were shortened where necessary. Note that the orientation of the branch does not affect the computations. These boundary nodes were assigned head values taken from 1/24,000 scale topographic maps, and the recharge for these nodes was left unspecified (Figure 3). This model represents boundary geometry and heads prior to the construction and filling of a reservoir on the Barren River in the southernmost part of the area.

There is very little surface drainage in the area, and a recharge for the interior nodes was calculated assuming that all non-evapotranspired precipitation (i.e., runoff plus infiltration) enters the aquifer as recharge. In a study of a geologically and climatologically similar area about 120 km to the west (Walker, 1956), the runoff varied from 3.4 inches to 70.7 inches per year over a ten year period. A figure of 30 in/yr was adopted for this model. This is equivalent to a recharge flow of 740 ft³/min at each node.

During execution, the permeability (inverse flow resistance) of the branches was adjusted to obtain a fit to observed heads at interior points in the area (Figure 3).

Results

A solution judged to be satisfactory was obtained after 560 iterations. The trial and error method used to determine a permeability is not very efficient, and convergence could have been obtained in only 100 or so iterations if the permeability had been held constant. Final values for the branch flow, recharge flow, and head are shown on Figures 4, 5, and 6 respectively.
The final permeability used was 650,000 ft$^3$/min (the appropriate units are those of discharge), which was judged to provide a satisfactory fit to the observed head at the interior points as shown in Table 1.

**TABLE 1**

**COMPARISON OF OBSERVED AND CALCULATED HEADS FOR MODEL 1 (feet)**

<table>
<thead>
<tr>
<th>Point</th>
<th>Observed Head (Figure 3)</th>
<th>Head Interpolated from Figure 6</th>
<th>Difference (Model 1 - observed)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>475</td>
<td>465</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>480</td>
<td>500</td>
<td>+ 20</td>
</tr>
<tr>
<td>3</td>
<td>575</td>
<td>550</td>
<td>- 25</td>
</tr>
<tr>
<td>4</td>
<td>435</td>
<td>470</td>
<td>+ 35</td>
</tr>
<tr>
<td>5</td>
<td>585</td>
<td>565</td>
<td>- 20</td>
</tr>
<tr>
<td>6</td>
<td>435</td>
<td>545</td>
<td>+ 110</td>
</tr>
<tr>
<td>7</td>
<td>525</td>
<td>565</td>
<td>+ 40</td>
</tr>
<tr>
<td>8</td>
<td>595</td>
<td>600</td>
<td>+ 5</td>
</tr>
<tr>
<td>9</td>
<td>595</td>
<td>595</td>
<td>-</td>
</tr>
<tr>
<td>10</td>
<td>515</td>
<td>505</td>
<td>-</td>
</tr>
<tr>
<td>11</td>
<td>545</td>
<td>555</td>
<td>+ 10</td>
</tr>
<tr>
<td>12</td>
<td>655</td>
<td>665</td>
<td>+ 10</td>
</tr>
<tr>
<td>13</td>
<td>585</td>
<td>580</td>
<td>-</td>
</tr>
<tr>
<td>14</td>
<td>625</td>
<td>630</td>
<td>+ 5</td>
</tr>
</tbody>
</table>

Nine of the fourteen points were sinkholes which apparently did not reach the water table (no water was shown in them on the topographic map) and thus suggest only an upper limit for the head (points 1, 2, 8 through 14 in Table 1).
Assuming active flow occurs in the aquifer to a depth of 100 feet and with the pipe spacing of 12,500 ft, each branch corresponds to a cross-sectional area of the aquifer of $1.25 \times 10^6$ ft$^2$. The pipe flow permeability of $6.5 \times 10^5$ ft$^3$/min therefore corresponds to a diffuse flow permeability of .52 ft/min, or 5600 meinzers (1 meinzer is equivalent to $9.3 \times 10^{-5}$ ft/min). This is a relatively high permeability, about that of a very well sorted, medium-grained sand (Davis and DeWiest, 1968, p. 164).

Data with which to evaluate the calculated branch flows and recharge (discharge) flows are limited. Flow estimates in streams appearing in four deep sinks or caves have been made, and are compared with the corresponding branch flows in Table 2. The location of these points is shown on Figure 4. Similarly, the calculated discharge is compared with the observed discharge in some large springs on the Green River (Figure 5 and Table 3).

The order-of-magnitude accordance between most observed branch flows and discharges and those calculated as Model 1 (Tables 2 and 3) suggests the model has some validity. The fact that the values calculated depended on the node spacing may indicate that a texture of about that used is present in the aquifer.

A water table map of part of the area (Cushman, 1968) based on well data is shown on Figure 6. Although large differences exist between the observed head and the calculated head, it is interesting to note that the groundwater divide between the Green and Barren Rivers suggested by the observed water table contours nearly coincides with the divide calculated for Model 1 (flow-line A on Figure 6).

Most of the observed discrepancies between Model 1 and observations of the aquifer can probably be explained by the general nature of the Model 1 parameters, such as recharge assumed constant both in time and space, uniform permeability throughout the aquifer, and uniform spacing of flow conduits. The very bad fit between water table contours and calculated heads near the center of the area, for example, can probably best be explained.
by much higher permeabilities in this area (which could be used in a refine-
ment of Model 1). It appears, however, that Model 1 provides a reasonably
good approximation of the aquifer and that the calculated parameters have
some predictive value.

TABLE 2

COMPARISON OF OBSERVED FLOWS IN THE AQUIFER AND MODEL 1
CALCULATED BRANCH FLOWS (ft$^3$/min)

<table>
<thead>
<tr>
<th>Point (Figure 4)</th>
<th>Apparent direction</th>
<th>Observed flow and reference</th>
<th>Corresponding branch between nodes indicated</th>
<th>Model 1 flow</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>north</td>
<td>3300 (Brown, 1966)</td>
<td>12, 9 and 12, 10</td>
<td>2700 north</td>
</tr>
<tr>
<td>2</td>
<td>north</td>
<td>2700 (Brown, 1966)</td>
<td>12, 8 and p2, 9</td>
<td>1600 north</td>
</tr>
<tr>
<td>3</td>
<td>north</td>
<td>4700 (Brown, 1966)</td>
<td>18, 9 and 18, 10</td>
<td>2200 north</td>
</tr>
<tr>
<td>4</td>
<td>west</td>
<td>340 (Thrailkill, 1970)</td>
<td>10, 6 and 11, 6</td>
<td>1000 west</td>
</tr>
</tbody>
</table>

TABLE 3

COMPARISON OF OBSERVED AND MODEL 1 CALCULATED DISCHARGES
ON THE AQUIFER BOUNDARY (ft$^3$/min)

<table>
<thead>
<tr>
<th>Point (Figure 5)</th>
<th>Observed discharge and reference</th>
<th>Corresponding node</th>
<th>Model 1 discharge</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>8000 (Brown, 1966)</td>
<td>12, 11</td>
<td>4900</td>
</tr>
<tr>
<td>2</td>
<td>95 (Brown, 1966)</td>
<td>13, 11</td>
<td>4700</td>
</tr>
<tr>
<td>3</td>
<td>2000 (Brown, 1966)</td>
<td>14, 12</td>
<td>3300</td>
</tr>
</tbody>
</table>
CHAPTER IV

MODEL 2 - SINKHOLE PLAIN AQUIFER TURBULENT FLOW

General and Parameters

The boundaries and parameters of Model 2 are the same as Model 1 except that turbulent flow in the wholly rough region was specified by assigning the value of 2.0 to the parameter $n$ in steps 2 and 4 of the computational procedure discussed earlier. As with Model 1, the permeability was adjusted during execution to fit observed interior heads.

Results

A solution was obtained after 100 iterations starting with Model 1 values of variables. Head contours (water table elevation) are shown on Figure 7, and comparisons of Model 2, Model 1, and observed head, branch flows, and discharges are given in Tables 4, 5, and 6. The permeability arrived at by the same adjustment procedure used for Model 1 was $2 \times 10^9 \text{ ft}^6/\text{min}^2$, or $(45,000 \text{ ft}^3/\text{min})^2$.

It is apparent that Model 2 is so similar to Model 1 that observed aquifer flows and heads are insufficient to test their relative validity.
<table>
<thead>
<tr>
<th>Point</th>
<th>Observed head (Figure 3)</th>
<th>Model 2 head interpolated from Figure 7</th>
<th>Difference (Model 2 - observed)</th>
<th>Model 1 difference (from Table 1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>475</td>
<td>445</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>480</td>
<td>470</td>
<td>-</td>
<td>+ 20</td>
</tr>
<tr>
<td>3</td>
<td>575</td>
<td>520</td>
<td>- 55</td>
<td>- 25</td>
</tr>
<tr>
<td>4</td>
<td>435</td>
<td>480</td>
<td>+ 45</td>
<td>+ 35</td>
</tr>
<tr>
<td>5</td>
<td>585</td>
<td>530</td>
<td>- 55</td>
<td>- 20</td>
</tr>
<tr>
<td>6</td>
<td>435</td>
<td>530</td>
<td>+ 95</td>
<td>+ 90</td>
</tr>
<tr>
<td>7</td>
<td>525</td>
<td>545</td>
<td>+ 20</td>
<td>+ 40</td>
</tr>
<tr>
<td>8</td>
<td>595</td>
<td>585</td>
<td>-</td>
<td>+ 5</td>
</tr>
<tr>
<td>9</td>
<td>595</td>
<td>590</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>10</td>
<td>515</td>
<td>505</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>11</td>
<td>545</td>
<td>545</td>
<td>-</td>
<td>+ 10</td>
</tr>
<tr>
<td>12</td>
<td>655</td>
<td>650</td>
<td>-</td>
<td>+ 10</td>
</tr>
<tr>
<td>13</td>
<td>585</td>
<td>565</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>14</td>
<td>625</td>
<td>610</td>
<td>-</td>
<td>+ 5</td>
</tr>
</tbody>
</table>
### Table 5

Comparison of observed flows in the aquifer and model 2 calculated branch flows (ft³/min). See Table 2 for reference for observed flow and branch coordinates.

<table>
<thead>
<tr>
<th>Point (Figure 4)</th>
<th>Apparent direction</th>
<th>Observed flow</th>
<th>Model 2 flow</th>
<th>Model 1 flow (from Table 2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>north</td>
<td>3300</td>
<td>2500</td>
<td>north 2700</td>
</tr>
<tr>
<td>2</td>
<td>north</td>
<td>2700</td>
<td>1600</td>
<td>north 1600</td>
</tr>
<tr>
<td>3</td>
<td>north</td>
<td>4700</td>
<td>2400</td>
<td>north 2200</td>
</tr>
<tr>
<td>4</td>
<td>west</td>
<td>340</td>
<td>1700</td>
<td>west 1000</td>
</tr>
</tbody>
</table>

### Table 6

Comparison of observed and model 2 calculated discharges on the aquifer boundary (ft³/min). See Table 3 for reference for observed flow and node coordinates.

<table>
<thead>
<tr>
<th>Point (Figure 5)</th>
<th>Observed discharge</th>
<th>Model 2 discharge</th>
<th>Model 1 discharge (from Table 1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>8000</td>
<td>4200</td>
<td>4900</td>
</tr>
<tr>
<td>2</td>
<td>95</td>
<td>5600</td>
<td>4700</td>
</tr>
<tr>
<td>3</td>
<td>2000</td>
<td>3400</td>
<td>3300</td>
</tr>
</tbody>
</table>
CHAPTER V

MODEL 3 - NORTH-CENTRAL SINKHOLE PLAIN AQUIFER - LAMINAR FLOW

General and Parameters

In order to investigate flow conditions using a finer grid and with more realistic (and complex) conditions, Model 3 was constructed of the north-central part of the Sinkhole Plain aquifer (Figure 2). Its boundaries were the Green River, the 600 ft. head contour from Model 1, and two Model 1 flow-lines (B and C on Figure 6). A grid spacing of 2500 feet was used.

Other than the finer grid spacing, the principal difference between this model and the previous ones is that it embodies an attempt to represent spatially varying recharge. As outlined earlier, the principal stratigraphic break divides the sedimentary rocks into a thick sequence of limestones and dolomites, of which the Girkin Limestone is the uppermost, overlain by a sequence of thin sandstones, shales, and limestones with the Big Clifty Sandstone at the base directly overlying the Girkin Limestone. The distribution of these two sequences (Gildersleeve, 1963, 1965; Haynes, 1962, 1964a, b, 1966; Klemic, 1963; Richards, 1964) is shown on Figure 8. To at least a good first approximation, no recharge enters the aquifer in areas underlain by the upper sequence (Big Clifty Sandstone and above) due to impermeable shales at the base of the Big Clifty Sandstone. In Model 3, the recharge of 29.6 ft³/min (equivalent to 30 in/year) assigned to each node was introduced at nodes in the lower sequence (open circles on Figure 9) but diverted to the nearest lower sequence node (dotted circles on Figure 9) from nodes located in the outcrop area of the upper sequence (solid circles on Figure 9). Where two or more lower sequence nodes were equi-distant from an upper sequence node, the recharge was divided equally.
Aquifer flow entering the model from the southeast was taken to be that calculated in Model 1.

**Results**

After 1300 iterations, a satisfactorily convergent solution was still not obtained for Model 3 for reasons which will be discussed below. An approximate solution yielded the results shown on Figure 10 and listed in Table 7. A permeability of 130,000 ft$^3$/min was used; the same as that used for Model 1.

**TABLE 7**

**COMPARISON OF OBSERVED AND MODEL 3 CALCULATED FLOWS AND DISCHARGES (ft$^3$/min)**

<table>
<thead>
<tr>
<th>Point (Figure 10)</th>
<th>Observed flow or discharge (See Figures 4 and 5 for reference)</th>
<th>Corresponding node or branch</th>
<th>Model 3 flow or discharge</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 + 2</td>
<td>discharge: 8000</td>
<td>7,20 + 8,20</td>
<td>- 400</td>
</tr>
<tr>
<td>3 + 4</td>
<td>discharge: 95</td>
<td>13,21 + 13,22</td>
<td>2400</td>
</tr>
<tr>
<td>5</td>
<td>discharge: 2000</td>
<td>21,23</td>
<td>7300</td>
</tr>
<tr>
<td>6</td>
<td>flow north: 3300</td>
<td>7,17 - 7,18</td>
<td>650 south</td>
</tr>
<tr>
<td>7</td>
<td>flow north: 2700</td>
<td>9,10 - 9,11</td>
<td>1400 south</td>
</tr>
<tr>
<td>8</td>
<td>flow north: 4700</td>
<td>36,13 - 36,14</td>
<td>9100 north</td>
</tr>
</tbody>
</table>
CHAPTER VI

CONCLUDING SUMMARY

The results presented in this report are believed to demonstrate the validity of the basic approach investigated. Models 1 and 2 of the entire Sinkhole Plain Aquifer show close correspondence with observed aquifer parameters. It has been shown that Model 1, calculated for laminar flow, and Model 2, calculated for turbulent (wholly rough) flow are sufficiently similar to require detailed observations of the aquifer to discriminate the flow regime in the aquifer, which confirms an earlier conclusion by the writer (Thrailkill, 1968). The existence of large discharges from the aquifer (probably in the form of underwater springs) predicted by the models suggests areas favorable for groundwater exploration and indicates flow paths of groundwater pollution in the area.

As stated in the introduction, the original objectives of the project were significantly reduced in number and scale as a result of time and funding reductions imposed on the project after its initiation. The revised objectives were to (1) write a digital computer program to model the flow in a limestone aquifer, (2) use the program to construct models of a real aquifer of interest, and (3) study the changes in an aquifer with time based on its response to currently available geochemical parameters. Objectives (1) and (2) were accomplished, although there are many routines in the program which require revision to improve their efficiency. The improvements needed in the program and the accomplishment of Objective (3) were prevented by the exhaustion of computer funds, even though a significant amount of non-project funds were also used. It is felt that the major objectives of the project were accomplished and that further work in this direction will provide considerable insight into the nature of the limestone aquifer.
PUBLICATIONS AND TRAINING ACCOMPLISHED

Publications

None as yet (other than progress report)

Training accomplished

One graduate student, David P. Beiter, was supported by and contributed to the project for the academic year 1970-71. Mr. Beiter's research project for the Ph.D. dissertation involves a study of the kinetic factors in the solutional enlargement of flow conduits in limestone.

Some of the results of this project are being utilized in a graduate course in hydrogeology now being offered.
REFERENCES


Haynes, Donald D., 1964b, Geology of the Mammoth Cave Quadrangle, Kentucky. U.S. G. S. Map GQ-351.


APPENDIX

KENTUCKY LIMESTONE AQUIFER MODEL
**KENTUCKY LIMESTONE AQUIFER MODEL**

**LLOWEST STATEMENT NUMBER IS 510**
**HIGHEST STATEMENT NUMBER IS 586**
**OTHER NUMBERS AVAILABLE ARE 585**

**CARD COLUMNS**

1 THRU 5  NO. OF NODES PER ROW (IMAX) - INTEGER
6 THRU 10 NO. OF ROWS OF NODES (JMAX) - INTEGER
11 THRU 15 MAXIMUM ACCEPTABLE ERROR IN FLOW AT A NODE (FERROR)
16 THRU 20 MAXIMUM ACCEPTABLE ERROR IN HEAD IN A BRANCH (HERROR)
21 THRU 25 TURBULENT FLOW EXPONENT (XP)
26 THRU 30 COEFFICIENT TO OVERRELAX (IF GREATER THAN 1) (ORLX)
31 THRU 35 MAXIMUM NO. OF ITERATIONS PERMITTED (IMAXIT) - INTEGER

A 1 IN THIS COLUMN READS DATA WITH SUBROUTINE IN1
A 1 IN THIS COLUMN READS DATA WITH SUBROUTINE IN2
A 1 IN THIS COLUMN READS INITIAL VALUES WITH SUBROUTINE IN3

A 1 IN THIS COLUMN CAUSES OUTPUT WITH SUBROUTINE OUT1
A 1 IN THIS COLUMN CAUSES OUTPUT WITH SUBROUTINE OUT2
A 1 IN THIS COLUMN CAUSES OUTPUT WITH SUBROUTINE OUT3
44 A 1 IN THIS COLUMN PUNCHES FINAL VALUES WITH SUBROUTINE OUT4
45 A DIGIT (N) IN THIS COLUMN CAUSES OUTPUT WITH OUT1 EVERY 2**N AND (2**N)+1 ITERATIONS
46 MULTIPLIER (LEAVE BLANK FOR I) OF HEAD ASSIGNED TO RECHARGE NODES IN SUBROUTINE CAL3 (MFAC)
47 A 1 IN THIS COLUMN CONTINUES EXECUTION IF NUMBER OF ENTERED VARIABLES IS NOT EQUAL TO REQUIRED NUMBER
48 A DIGIT (QI) IN THIS COLUMN IS AMOUNT IN TENTHS OF HEAD ERROR USED FOR FLOW ADJUSTMENT
49 A 1 IN THIS COLUMN MAKES QI AMOUNT IN HUNDREDTHS, A 2 IN THOUSANDTHS, AND A 3 IN TEN-THOUSANDTHS
50 A 1 IN THESE COLUMNS WILL CAUSE DELETION OF MAPS WRITTEN UNDER SUBROUTINE OUT3 AS FOLLOWS:
51 MAP 1 (ENTERED HEAD - BH)
52 MAP 2 (FINAL HEAD - H)
53 MAP 3 (ENTERED RECHARGE - BFR)
54 MAP 4 (FINAL RECHARGE - FR)
55 MAP 5 (ENTERED BRANCH FLOW - BFX AND BFY)
56 MAP 6 (FINAL BRANCH FLOW - FX AND FY)
57 MAP 7 (ENTERED PERMEABILITY - BXX AND BKY)
58 MAP 8 (FINAL PERMEABILITY - KX AND KY)
59 MAP 9 (BRANCH LENGTH - BLX AND BLY)
60 DECADE COMMON VARIABLES
61 COMMON H(38,27), BH(38,27), FX(38,27), BFX(38,27), FY(38,27), BFY(38,27)
62 FR(38,27), BFR(38,27), XX(38,27), BXX(38,27), KY(38,27), BKY(38,27), BLX(38,27), BLY(38,27), IA(38,27),
63 I,J,IMAX,JMAX,ITER,CTER,ERINF,IMAX1,JMAX1,XP
64 REAL KK, KY
DECLARE MAIN VARIABLES

INTEGER HEADIV, FREBR(5), QI

READ PARAMETER CARD

READ (5,531) IMAX, JMAX, FERROR, HERROR, XP, ORLX, MAXIT, IO1, IO4, IO6,
2 IO2, IO3, IO5, IO7, MM1, MFAC, MGO, QI, JQ, ID1, IO2, IO3, IO4,
3 IO5, IO6, IO7, IO8, IO9

531 FORMAT (215, 4F5.0, 15, 311, 2X, 411, 6X, 511, 5X, 911)

IF (MFAC.EQ.0) MFAC=1

WRITE OUT PARAMETRERS

WRITE (6,523)
523 FORMAT ('IPARAMETERS USED IN EXECUTION ARE:',')
WRITE (6,524) IMAX
524 FORMAT ('O MAXIMUM VALUE OF I (IMAX) IS ',IO5)
WRITE (6,525) JMAX
525 FORMAT ('O MAXIMUM VALUE OF J (JMAX) IS ',IO5)
WRITE (6,526) FERROR
526 FORMAT ('O MAXIMUM ERROR IN FLOW AT ANY NODE (FERROR) IS ',G10
2.3)
WRITE (6,561) HERROR
561 FORMAT ('O MAXIMUM ERROR IN HEAD IN ANY BRANCH (HERROR) IS ',G
210.3)
WRITE (6,585) QI
585 FORMAT ('O DISTRIBUTION FACTOR FOR HEAD ADJUSTMENT (QI) IS ',IO1
21)

IF (JQ.NE.0) WRITE (6,586) JQ
586 FORMAT ('O DISTRIBUTION FACTOR (QI) MULTIPLIED BY 10**-11)
WRITE (6,559) XP
559 FORMAT ('O FLOW EXPONENT (XP) IS ',G10.3)
WRITE (6,527) ORLX
527 FORMAT ('O RELAXATION COEFFICIENT (ORLX) IS ',G10.3)
WRITE (6,528) MAXIT
528 FORMAT ('O MAXIMUM NUMBER OF ITERATIONS PERMITTED (MAXIT) IS '
2.15)

IF (IO1.EQ.1) WRITE (6,571)
571 FORMAT ('O DATA READ WITH SUBROUTINE IN1')
IF (104.EQ.1) WRITE (6,574)
574 FORMAT ("0 DATA READ WITH SUBROUTINE IN2")
IF (104.EQ.1) WRITE (6,553) MFAC
553 FORMAT ("0 MULTIPLIER OF ",11, CC" USED WITH CAL3")
IF (106.EQ.1) WRITE (6,540)
540 FORMAT ("0 INITIAL VALUES ENTERED WITH SUBROUTINE IN3")
IF (102.EQ.1) WRITE (6,572)
572 FORMAT ("0 OUTPUT WITH SUBROUTINE OUT1")
IF (102.EQ.1) WRITE (6,575) MM1,MM1
572 FORMAT ("0 OUTPUT EVERY 2**11,4 AND (2**11,4)+1 ITERA-
2TIONS")
IF (103.EQ.1) WRITE (6,573)
573 FORMAT ("0 OUTPUT WITH SUBROUTINE OUT2")
IF (105.EQ.1) WRITE (6,557)
557 FORMAT ("0 OUTPUT WITH SUBROUTINE OUT3. THE FOLLOWING MAPS HAV
2E BEEN DELETED:")
IF (101.EQ.1) WRITE (6,556)
556 FORMAT ("0 MAP 1 (ENTERED HEAD - BH)")
IF (102.EQ.1) WRITE (6,555)
555 FORMAT ("0 MAP 2 (FINAL HEAD - H)")
IF (103.EQ.1) WRITE (6,554)
554 FORMAT ("0 MAP 3 (ENTERED RECHARGE - BFR)")
IF (104.EQ.1) WRITE (6,522)
522 FORMAT ("0 MAP 4 (FINAL RECHARGE - FR)")
IF (105.EQ.1) WRITE (6,521)
521 FORMAT ("0 MAP 5 (ENTERED BRANCH FLOW - BFX AND BFY)")
IF (106.EQ.1) WRITE (6,517)
517 FORMAT ("0 MAP 6 (FINAL BRANCH FLOW - FX AND FY)")
IF (107.EQ.1) WRITE (6,513)
513 FORMAT ("0 MAP 7 (ENTERED PERMEABILITY - BKX AND BKY)")
IF (108.EQ.1) WRITE (6,576)
576 FORMAT ("0 MAP 8 (FINAL PERMEABILITY - KX AND KY)")
IF (109.EQ.1) WRITE (6,577)
577 FORMAT ("0 MAP 9 (BRANCH LENGTH - BLX AND BLY)")
IF (101.NE.1.AND.102.NE.1.AND.103.NE.1.AND.104.NE.1.AND.105.NE.1.
2.AND.106.NE.1.AND.107.NE.1.AND.108.NE.1.AND.109.NE.1) WRITE
3 (6,578)
578 FORMAT ("0 NO MAPS DELETED")
IF (107.EQ.1) WRITE (6,539)
539 FORMAT ("0 FINAL VALUES PUNCHED WITH SUBROUTINE OUT4")
C INITIALIZE ALL ARRAYS
C
IMAXM1=IMAX-1
JMAXM1=JMAX-1
DO 529 I=1,IMAX
DO 530 J=1,JMAX
HI(I,J)=0.
BH(I,J)=0.
FX(I,J)=0.
BFX(I,J)=0.
FY(I,J)=0.
BFY(I,J)=0.
FR(I,J)=0.
BFR(I,J)=0.
KX(I,J)=0.
BKX(I,J)=0.
KY(I,J)=0.
BKY(I,J)=0.
BLX(I,J)=0.
BLY(I,J)=0.
530 CONTINUE
529 CONTINUE
IF (I01.EQ.1) CALL IN1
IF (I04.EQ.1) CALL IN2
C
C TEST FOR CORRECT NUMBER OF ENTERED VALUES
C
CALL CAL2 (MGO)
C
C ASSIGN INITIAL VALUES
C
IF (I06.EQ.1) CALL IN3
IF (I06.NE.1) CALL CAL3 (MFAC,KCHEK)
C
************* BEGIN ITERATIONS *********************
ifdef (KCHEK.EQ.0) CALL CALL1
C
C ************** BEGIN HEAD CALCULATIONS **********************
C
MFX1=1
MFY1=1
MFX2=1
MFY2=1
MFX1=ABS(FX(I-1,J))
MFY1=ABS(FY(I,J-1))
MFX2=ABS(FX(I,J))
MFY2=ABS(FY(I,J))
C
C CHECK FOR ENTERED VALUE AND SKIP CALCULATIONS
C
514 IF (BH(I,J).NE.0.) GO TO 515
C
C CALCULATE H IF I NE 1, J NE 1, I NE IMAX, J NE JMAX, I-1 KX NE 0, J-1
C KX NE 0, KX NE 0, OR KY NE 0
C
IF (I.EQ.1.OR.J.EQ.1.OR.I.EQ.IMAX.OR.J.EQ.JMAX) GO TO 516
IF (KX(I-1,J).LT.0.0001.OR.KY(I,J).LT.0.0001.OR.KX(I-1,J).LT.0.0001
? OR.KY(I-1,J).LT.0.0001) GO TO 516
4   (FY(I,J).LT.1.OE-70.AND.FY(I,J).GT.-1.OE-70)) GOTO 516
H(I,J)=((-BLX(I-1,J)*MFX1*(AFX1**XP)/KX(I-1,J)))+HI(I-1,J)
2   +((-BLY(I,J-1)*MFY1*(AFY1**XP)/KY(I,J-1))+HI(I,J-1))
3   +((-BLX(I,J)*MFX2*(AFX2**XP)/KX(I,J)))HI(I+1,J)
4   +((-BLY(I,J)*MFY2*(AFY2**XP)/KY(I,J)))HI(J+1,I))/4.
C SET BRANCH TERMS EQ 0 IF ALL FOUR BRANCHES NOT PRESENT
C
516 IF (I.EQ.1) GO TO 518
518 AHEAD=0.0
GO TO 520
519 AHEAD=((-BLX(I-1,J)*MFX1*(AFX1**XP)/KX(I-1,J)))+HI(I-1,J)
520 IF (J.EQ.1) GO TO 552
521 IF (KY(I,J-1).LT.0.0001) GO TO 552
522 IF (FY(I,J-1).LT.1.OE-70.AND.FY(I,J-1).GT.-1.OE-70)) GO TO 552
523 BHEAD=0.0
GO TO 564
524 IF (I.EQ.IMAX.OR.KX(I,J).LT.0.0001) GO TO 565
525 IF (FX(I,J).LT.1.OE-70.AND.FX(I,J).GT.-1.OE-70)) GO TO 565
526 CHEAD=0.0
GO TO 567
527 CHEAD=((-BLY(I,J-1)*MFY1*(AFY1**XP)/KY(I,J-1)))+HI(J-1)
528 IF (J.EQ.JMAX.OR.KY(I,J).LT.0.0001) GO TO 568
529 IF (FY(I,J).LT.1.OE-70.AND.FY(I,J).GT.-1.OE-70)) GO TO 568
530 DHEAD=0.0
GO TO 570
531 DHEAD=((-BLX(I,J)*MFX2*(AFX2**XP)/KX(I,J)))+HI(I+1,J)
C COUNT NUMBER OF BRANCHES
C
570 HEADIV=4
IF (AHEAD.EQ.0.0) HEADIV=HEADIV-1
IF (BHEAD.EQ.0.0) HEADIV=HEADIV-1
IF (CHEAD.EQ.0.0) HEADIV=HEADIV-1
IF (DHEAD.EQ.0.0) HEADIV=HEADIV-1
IF (HEADIV.EQ.0.0) GO TO 515

C SUM BRANCH TERMS AND DIVIDE BY NUMBER OF BRANCHES
C
H(I,J)=(AHEAD+BHEAD+CHEAD+DHEAD)/HEADIV
C
C *********************** CALCULATE ERROR IN FLOW AT NODE ***********************
C
C CALCULATE FLOW ERROR AND COMPARE WITH FERROR
C
515 CONTINUE
IF (I.EQ.1) GO TO 534
IF (J.EQ.1) GO TO 535
ERINF=FX(I,J)+FY(I,J)-FR(I,J)-FX(I-1,J)-FY(I,J-1)
GO TO 536
534 IF (J.EQ.1) GO TO 537
ERINF=FX(I,J)+FY(I,J)-FR(I,J)-FY(I,J-1)
GO TO 536
535 ERINF=FX(I,J)+FY(I,J)-FR(I,J)
GO TO 536
536 ABER=ABS(ERINF)
IF (ABER.GT.FERROR) CNTER=CNTER+1

C CALCULATE HEAD ERRORS IN BRANCHES
C
MFLR=1
MFLU=1
MFLL=1
MFLD=1
AFLR=0.
AFLU=0.
AFLL=0.
AFLD=0.
CALFLR=0.
CALFLU=0.
CALFLL=0.
CALFLD=0.
IF (I.EQ.IMAX) GO TO 581
IF (((KX(I,J)*HI(I,J)&#8722;HI(I+1,J))/BLX(I,J)).LT.0.) MFLR=-1
AFLR=ABS((KX(I,J)*HI(I,J)&#8722;HI(I+1,J))/BLX(I,J))
581 IF (J.EQ.JMAX) GO TO 582
IF (((KY(I,J)*HI(I,J)&#8722;HI(I,J+1))/BLY(I,J)).LT.0.) MFLU=-1
AFLU=ABS((KY(I,J)*HI(I,J)&#8722;HI(I,J+1))/BLY(I,J))
582 IF (I.EQ.1) GO TO 583
IF (((KX(I-1,J)*HI(I,J)-HI(I,J-1))/BLX(I-1,J)).GT.0.) MFLL=-1
AFLL=ABS((KX(I-1,J)*HI(I,J)-HI(I,J-1))/BLX(I-1,J))
583 IF (J.EQ.1) GO TO 584
IF (((KY(I-1,J)*HI(I,J)-HI(I,J-1))/BLY(I-1,J)).GT.0.) MFLD=-1
AFLD=ABS((KY(I-1,J)*HI(I,J)-HI(I,J-1))/BLY(I-1,J))
584 IF (AFLR.NE.0.) CALFLR=(AFLR**(1./XP))*MFLR
IF (AFLU.NE.0.) CALFLU=(AFLU**(1./XP))*MFLU
IF (AFLL.NE.0.) CALFLL=(AFLL**(1./XP))*MFLL
IF (AFLD.NE.0.) CALFLD=(AFLD**(1./XP))*MFLD
C
C DISTRIBUTE FLOW ERROR
C
DO 538 L=1,5
538 FREQBR(I)=1
IF (I.EQ.IMAX.OR.BFX(I,J).NE.0..OR.BKX(I,J).EQ.12345.) FREQBR(I)=0
IF (J.EQ.JMAX.OR.BFY(I,J).NE.0..OR.BKY(I,J).EQ.12345.) FREQBR(I)=0
IF (I.EQ.1) GO TO 580
IF (BFX(I-1,J).NE.0..OR.BKX(I-1,J).EQ.12345.) FREQBR(I)=0
580 IF (J.EQ.1) GO TO 544
IF (BFY(I,J-1).NE.0..OR.BKY(I,J-1).EQ.12345.) FREQBR(I)=0
544 IF (FREB(I,J).NE.0.) FREQBR(5)=0
IF (I.EQ.1) FREQBR(3)=0
IF (J.EQ.1) FREQBR(4)=0
TOTFRE=0.
DO 545 L=1,5
545 TOTFRE=TOTFRE+FREQBR(I)
IF (TOTFRE.EQ.0.) GO TO 562
IF (FREQBR(I).EQ.1) FX(I,J)=FX(I,J)&#8722;QRLX*((QF*ERINF/TOTFRE)
+QH*(CALFLR-FX(I,J)))
IF (FREQBR(2).EQ.1) FY(I,J)=FY(I,J)&#8722;QRLX*((QF*ERINF/TOTFRE)
+QH*(CALFLU-FY(I,J)))
IF (FREQBR(3).EQ.1) FX(I-1,J)=FX(I-1,J)&#8722;QRLX*((QF*ERINF/TOTFRE)
+QH*(CALFLL-FX(I-1,J)))
562 IF (FREQBR(4).EQ.1) FY(I,J)=FY(I,J)&#8722;QRLX*((QF*ERINF/TOTFRE)
+QH*(CALFLD-FY(I,J)))
IF (FREQBR(5).EQ.1) FX(I,J)=FX(I,J)&#8722;QRLX*((QF*ERINF/TOTFRE)
+QH*(CALFLF-FX(I,J)))
IF (FREBR(4).EQ.1) FY(I,J-1)=FY(I,J-1)+ORLX*(QF*ERINF/TOTFRE)
2 +QH*(CALFLD-FY(I,J-1))
IF (FREBR(5).EQ.1) FR(I,J)=FR(I,J)+ORLX*(ERINF/TOTFRE)
C
C CALCULATE HEAD ERRORS AND COMPARE WITH HERROR
C
HEDERX=0.
HEDERY=0.
IF (I.EQ.IMAX) GO TO 547
IF (K(I,J).LT.0.0001) GO TO 547
HEDERX=((BLX(I,J)*MFX2*(AFX2**XP))/K(I,JJ)-H(I,J)+H(I+1,J)
547 IF (J.EQ.JMAX) GO TO 546
IF (K(Y(I,J).LT.0.0001) GO TO 546
HEDERY=((BLY(I,J)*MFY2*(AFY2**XP))/K(I,J)+H(I,J)+H(I,J+1)
546 CONTINUE
ABERX=ABS(HEDERX)
ABERY=ABS(HEDERY)
IF (ABERX.GT.HERROR) KNTR=KNTR+1.
IF (ABERY.GT.HERROR) KNTR=KNTR+1.
IF ((KX.EQ.1) CALL OUT3C (KNTR,HEDERX,HEDERY)
562 IF (IO2.EQ.1) CALL OUT1 (MM1,HEDERX,HEDERY,KNTR)
551 CONTINUE
512 CONTINUE
511 CONTINUE
C
C ****************** TEST FOR NEXT ITERATION **********************
C
IF (IO3.EQ.1.AND.(ITER.EQ.MAXIT-1).OR.(CINTER.EQ.0.AND.KNTR.EQ.0))
2 CALL OUT2 (K02)
IF (IO7.EQ.1.AND.(ITER.EQ.MAXIT).OR.(CINTER.EQ.0.AND.KNTR.EQ.0))
2 CALL OUT4
IF (IO5.EQ.1) GO TO 558
IF (ITER.GE.MAXIT) GO TO 548
IF (CINTER.EQ.0.AND.KNTR.EQ.0) GO TO 549
GO TO 550
548 WRITE (6,532) ITER,MAXIT
532 FORMAT ('EXECUTION TERMINATED - NUMBER OF ITERATIONS (",IS",")
2ALS MAXIMUM SPECIFIED (",IS")')
RETURN
549 WRITE (6,533) FERROR,HERROR,ITER
533 FORMAT ('EXECUTION COMPLETED - ALL FLOW ERRORS BELOW MAXIMUM (",G
540 MAIN 3690
541 MAIN 3700
542 MAIN 3710
543 MAIN 3720
544 MAIN 3730
545 MAIN 3740
546 MAIN 3750
547 MAIN 3760
548 MAIN 3770
549 MAIN 3780
550 MAIN 3790
551 MAIN 3800
552 MAIN 3810
553 MAIN 3820
554 MAIN 3830
555 MAIN 3840
556 MAIN 3850
557 MAIN 3860
558 MAIN 3870
559 MAIN 3880
560 MAIN 3890
561 MAIN 3900
562 MAIN 3910
563 MAIN 3920
564 MAIN 3930
565 MAIN 3940
566 MAIN 3950
567 MAIN 3960
568 MAIN 3970
569 MAIN 3980
570 MAIN 3990
571 MAIN 4000
572 MAIN 4010
573 MAIN 4020
574 MAIN 4030
575 MAIN 4040
576 MAIN 4050
577 MAIN 4060
578 MAIN 4070
579 MAIN 4080
580 MAIN 4090
AND ALL HEAD ERRORS BELOW MAXIMUM (',G12.5,') - ',15, ITE 3R."
RETURN
558 CONTINUE
IF (INTER.EQ.0.AND.KNTR.EQ.0) GO TO 510
IF (ITER.EQ.MAXIT-1) GO TO 510
IF (ITER.EQ.MAXIT) GO TO 579
GO TO 550
510 CALL OUT3B (KK,FERROR,HERROR,MAXIT,KNTR)
IF (KK.EQ.1) GO TO 550
579 CONTINUE
IF (IO5.EQ.1) CALL OUT3 (ID1,ID2,ID3,ID4,ID5,ID6,ID7,ID8,ID9)
RETURN
END
SUBROUTINE IN1
C ********************************************
C ****************** SUBROUTINE IN1 ********************
C ********************************************
C STATEMENT NUMBERS USED ARE 401,405,407,408,409,500,509,510
C
C DATA CARDS ARE NODE CARDS
C
C CAN BE IN ANY ORDER BUT MUST BE ONE FOR EACH I AND J IN RECTANGULAR
C ARRAY. ANY BLANK OR ZERO PUNCHED FIELDS WILL BE CONSIDERED
C UNKNOWNS. ENTERED ZERO'S MUST BE PUNCHED 12345 IN LAST 5 COLUMNS
C OR 12345. IN ANY PART OF FIELD. TOTAL NUMBER OF SPECIFIED HEAD,
C FLOW, AND PERMEABILITY VALUES MUST = (3*IMAX*JMAX)-IMAX-JMAX.
C BLANK OR ZERO IN LENGTH FIELDS WILL BE CONSIDERED UNIT LENGTH,
C OTHERWISE PUNCH NUMBER. ZERO (12345) LENGTHS NOT PERMITTED.
C
C CARD COLUMNS
C
C 1 THRU 5 NODE NUMBER IN ROW (I) - INTEGER
C 6 THRU 10 ROW NUMBER (J) - INTEGER
C 11 THRU 20  HEAD AT NODE (BH(I,J))
C 21 THRU 30  FLOW IN +X DIRECTION FROM NODE (BFX(I,J))
C 31 THRU 40  FLOW IN +Y DIRECTION FROM NODE (BFY(I,J))
C 41 THRU 50  RECHARGE FLOW INTO NODE (BFR(I,J))
C 51 THRU 60  PERMEABILITY IN +X DIRECTION FROM NODE (BXX(I,J))
C 61 THRU 70  PERMEABILITY IN +Y DIRECTION FROM NODE (BKY(I,J))
C 71 THRU 75  LENGTH IN +X DIRECTION FROM NODE (BLX(I,J))
C 76 THRU 80  LENGTH IN +Y DIRECTION FROM NODE (BLY(I,J))
C DECLARE COMMON VARIABLES
C COMMON  H(38,27), BH(38,27), FX(38,27), BFX(38,27), FY(38,27),
B 2   BFY(38,27), FR(38,27), BFR(38,27), KX(38,27), BKX(38,27),
3   KY(38,27), BKY(38,27), BLX(38,27), BLY(38,27), IA(38,27),
4   I, J, IMAX, JMAX, ITER, CNTER, ERINF, I1, MAX1, JMAX1, XP
   REAL KX, KY
   INTEGER CNTER
C READ NODE CARDS
C
NODMAX=IMAX*JMAX
DO 500 L=1,NODMAX
  500 READ (5,401) I,J,BH(I,J),BFX(I,J),BFY(I,J),BFR(I,J),BKX(I,J),
   2   BKY(I,J),BLX(I,J),BLY(I,J)
401 FORMAT (15,6F10.0,2F5.0)
C CORRECT LENGTH VALUES
C
  DO 405 I=1,IMAX
  DO 409 J=1,JMAX
  IF (BLX(I,J).EQ.0.) BLX(I,J)=1.
  IF (BLY(I,J).EQ.0.) BLY(I,J)=1.
409 CONTINUE
405 CONTINUE
C WRITE OUT ENTERED DATA
C
WRITE(6,408)
408 FORMAT('I J BH BFX BFY BFR BXX BKY BLX BLY'//)
DO 509 I=1,IMAX

DO 510 J=1,JMAX
   WRITE(6,407) I,J,BH(I,J),BFX(I,J),BFY(I,J),BFR(I,J),BKX(I,J),
   \ \ \ \ \ \ \ \ \ \ BKY(I,J),BLX(I,J),BLY(I,J)
407 FORMAT (',12X,15,5X,15,3X,8F10.3)
510 CONTINUE
509 CONTINUE
C TEST FOR CORRECT NUMBER OF ENTERED VALUES
C CALL CAL2
C ASSIGN INITIAL VALUES
C CALL CAL3 (MFAC,KCHEK)
RETURN
END

SUBROUTINE IN2
C ******************** DATA FORMAT ********************
C FIRST DATA CARD (AFTER PARAMETER CARD) IS GENERAL TO LOAD ALL ARRAYS.
C ENTER VALUES IN THREE OF FIRST SIX FOR CORRECT NUMBER OF ENTERED
C VALUES. BLANK OR ZERO CONSIDERED UNKNOWN, 12345.0 FOR ZERO, DECIMAL
C POINT TO RIGHT OF LAST POSITION UNLESS PUNCHED, ZERO BLX AND BLY
C NOT PERMITTED.
C CARD COLUMNS
C 1 THRU 10  HEAD (BH)
C 11  20  FLOW IN BRANCH TO RIGHT, POSITIVE IF TO RIGHT (BFX)
C 21  30  FLOW IN BRANCH ABOVE, POSITIVE IF UP (BFY)
C 31  40  RECHARGE FLOW, POSITIVE IF INTO SYSTEM (BFR)
C 41  50  PERMEABILITY IN BRANCH TO RIGHT (BKX)
C 51  60  PERMEABILITY IN BRANCH ABOVE (BKY)
C 61  70  LENGTH OF BRANCH TO RIGHT (BLX)

IN1  0690
IN1  0700
IN1  0710
IN1  0720
IN1  0730
IN1  0740
IN1  0750
IN1  0760
IN1  0770
IN1  0780
IN1  0790
IN1  0800
IN1  0810
IN1  0820
IN1  0830
IN1  0840
IN2  0010
IN2  0020
IN2  0030
IN2  0040
IN2  0050
IN2  0060
IN2  0070
IN2  0080
IN2  0090
IN2  0100
IN2  0110
IN2  0120
IN2  0130
IN2  0140
IN2  0150
IN2  0160
IN2  0170
IN2  0180
IN2  0190
IN2  0200
IN2  0210
IN2  0220
IN2  0230
IN2  0240
IN2  0250
C 71 80 LENGTH OF BRANCH ABOVE (BLY)
C C=C
C REMAINING DATA CARDS ARE CHANGE CARDS WITH FOUR NODES PER CARD
C CARD COLUMNS
C
1  DIGIT INDICATING VARIABLE:
  21  1 FOR BH  5 FOR BKK
  41  2 FOR BFX  6 FOR BKY
  61  3 FOR BFY  7 FOR BLX
  4 FOR BFR  8 FOR BLY
0 TO SIGNAL END OF DATA
C 2 THRU 5 NODE NUMBER IN ROW (I) - INTEGER
C 22  25
C 42  45
C 62  65
C 6,26,46,66 BLANK
C 7 THRU 10 ROW NUMBER (J) - INTEGER
C 27  30
C 47  50
C 67  70
C C=C
C 11 THRU 20 VALUE OF VARIABLE
C 31  40
C 51  60
C 71  80
C DECLARE COMMON VARIABLES
C COMMON H(38,27), BH(38,27), FX(38,27), BFX(38,27), FY(38,27),
  2 BFY(38,27), FR(38,27), BFR(38,27), KX(38,27), BKX(38,27),
  3 KY(38,27), BKY(38,27), BLX(38,27), BLY(38,27), IA(38,27),
REAL KX,KY
INTEGER CNTER

DECLARE IN2 VARIABLES
REAL IG1,IG2,IG3,IG4,IG5,IG6,IG7,IG8

READ GENERAL DATA CARD
READ (5,108) IG1,IG2,IG3,IG4,IG5,IG6,IG7,IG8
108 FORMAT (8F10.0)
DO 109 I=1,IMAX
DO 110 J=1,JMAX
BHll,Jl=IG1
BRll,Jl=IG4
110 CONTINUE
DO 111 I=1,IMAXM1
DO 112 J=1,JMAXM1
BFxl,Jl=IG2
BKX(J,J)=IG5
BLX(J,J)=IG7
112 CONTINUE
DO 113 I=1,IMAX
DO 114 J=1,JMAX
BFYl,Jl=IG3
BKY(J,J)=IG6
RLY(J,J)=IG8
114 CONTINUE
DO 115 I=1,IMAX
DO 116 J=1,JMAX
IF (IV1.EQ.0) GO TO 115
IF (IV1.EQ.1) BHl,J)=VL
115 CONTINUE

C READ CHANGE DATA CARDS
C
116 READ (5,100) IV1,IV2,IV3,IV4,IV5,IV6,IV7,IV8
100 FORMAT (4(l1,1X,14,F10.0))
1=11
J=Jl
IF (IV1.EQ.0) GO TO 115
IF (IV1.EQ.1) BHll,J)=VL
IF (IV1.EQ.2)  BFX(I,J)=V1
IF (IV1.EQ.3)  BFY(I,J)=V1
IF (IV1.EQ.4)  BFR(I,J)=V1
IF (IV1.EQ.5)  BKX(I,J)=V1
IF (IV1.EQ.6)  BKY(I,J)=V1
IF (IV1.EQ.7)  BLX(I,J)=V1
IF (IV1.EQ.8)  BLY(I,J)=V1
I=12
J=J2
IF (IV2.EQ.0)  GO TO 115
IF (IV2.EQ.1)  BHI(I,J)=V2
IF (IV2.EQ.2)  BFX(I,J)=V2
IF (IV2.EQ.3)  BFY(I,J)=V2
IF (IV2.EQ.4)  BFR(I,J)=V2
IF (IV2.EQ.5)  BKX(I,J)=V2
IF (IV2.EQ.6)  BKY(I,J)=V2
IF (IV2.EQ.7)  BLX(I,J)=V2
IF (IV2.EQ.8)  BLY(I,J)=V2
I=13
J=J3
IF (IV3.EQ.0)  GO TO 115
IF (IV3.EQ.1)  BHI(I,J)=V3
IF (IV3.EQ.2)  BFX(I,J)=V3
IF (IV3.EQ.3)  BFY(I,J)=V3
IF (IV3.EQ.4)  BFR(I,J)=V3
IF (IV3.EQ.5)  BKX(I,J)=V3
IF (IV3.EQ.6)  BKY(I,J)=V3
IF (IV3.EQ.7)  BLX(I,J)=V3
IF (IV3.EQ.8)  BLY(I,J)=V3
I=14
J=J4
IF (IV4.EQ.0)  GO TO 115
IF (IV4.EQ.1)  BHI(I,J)=V4
IF (IV4.EQ.2)  BFX(I,J)=V4
IF (IV4.EQ.3)  BFY(I,J)=V4
IF (IV4.EQ.4)  BFR(I,J)=V4
IF (IV4.EQ.5)  BKX(I,J)=V4
IF (IV4.EQ.6)  BKY(I,J)=V4
IF (IV4.EQ.7)  BLX(I,J)=V4
IF (IV4.EQ.8)  BLY(I,J)=V4
GO TO 116
C WRITE OUT ENTERED DATA

115 WRITE (6,101)
101 FORMAT ('DATA ENTERED USING SUBROUTINE IN2')
   WRITE (6,102)
102 FORMAT ('0',14X,'I',9X,'J',9X,'BH',8X,'BFY',7X,'BFR',8X,'BFX',8X,'BFY',8X,'BFR',8X,'BFX',8X,'BFY',8X,'BFR',8X,'BFX')
   DO 103 I=1,IMAX
      WRITE (6,104)
      WRITE (6,107) I,J,BH(I,J),BFY(I,J),BFR(I,J),BFX(I,J),BKY(I,J),BLX(I,J),BLY(I,J)
   103 CONTINUE
104 FORMAT ('*')
105 CONTINUE
106 CONTINUE
   RETURN
SUBROUTINE IN3

C HIGHEST STATEMENT NUMBER IS 104
C DECLARE COMMON VARIABLES
C
   COMMON H(38,27), BH(38,27), FX(38,27), BFX(38,27), FY(38,27),
   2 BFY(38,27), FR(38,27), BFR(38,27), KX(38,27), BKX(38,27),
   3 KY(38,27), BKY(38,27), BLX(38,27), BLY(38,27), IA(38,27),
   4 I,J,IMAX,JMAX,ITER,CNTER,ERINF,IMAXM1,JMAXM1,XP
   REAL KX,KY
   INTEGER CNTER
C
C READ INITIAL VALUES FROM CARDS
C
   READ (5,100) NCARD
100 FORMAT (110)
   DO 101 N=1,NCARD
READ (5,102) I,J,H(I,J),FX(I,J),FY(I,J),FR(I,J),KX(I,J),KY(I,J)
102 FORMAT (215,6E10.3)
101 CONTINUE
C ASSIGN ENTERED VALUES IF DEFINED (TO PERMIT CHANGES)

DO 103 I=1,IMAX
DO 104 J=1,JMAX
IF (BH(I,J).NE.0.) H(I,J)=BH(I,J)
IF (BH(I,J).EQ.12345.) H(I,J)=0.
IF (BFX(I,J).NE.0.) FX(I,J)=BFX(I,J)
IF (BFX(I,J).EQ.12345.) FX(I,J)=0.
IF (BFY(I,J).NE.0.) FY(I,J)=BFY(I,J)
IF (BFY(I,J).EQ.12345.) FY(I,J)=0.
IF (BFY(I,J).NE.0.) FR(I,J)=BFY(I,J)
IF (BFY(I,J).EQ.12345.) FR(I,J)=0.
IF (BKX(I,J).NE.0.) KX(I,J)=BKX(I,J)
IF (BKX(I,J).EQ.12345.) KX(I,J)=0.
IF (BKX(I,J).NE.0.) KY(I,J)=BKX(I,J)
IF (BKX(I,J).EQ.12345.) KY(I,J)=0.
104 CONTINUE
103 CONTINUE
RETURN
END

SUBROUTINE CALI

C ************************** SUBROUTINE CALL ****************************
C
C HIGHEST STATEMENT NUMBER IS 101
C
C DECLARE COMMON VARIABLES

COMMON H(38,27), BH(38,27), FX(38,27), BFX(38,27), FY(38,27),
2 BFY(38,27), FR(38,27), BFR(38,27), KX(38,27), BKX(38,27),
3 KY(38,27), BKY(38,27), BLX(38,27), BLY(38,27), IA(38,27),
4 I,J,IMAX,JMAX,ITER,CNTER,ERINF,IMAXM1,JMAXM1,XP
REAL KX,KY
INTEGER CNTER
IF (I.EQ.1MAX .OR. BKX(I,J).NE.0.) GO TO 100
HDIF = H(I,J) - H(I+1,J)
IF (HDIF .LT. 0.00001 .AND. HDIF .GT. -0.00001) HDIF = 1.
IF (FX(I,J) .LT. 1.0E-70 .AND. FX(I,J) .GT. -1.0E-70) GO TO 100
AFX2 = ABS(FX(I,J))
KX(I,J) = ABS((BLX(I,J) * (AFX2 ** XP)) / HDIF)
100 IF (J .EQ. JMAX .OR. BKY(I,J) .NE. 0) GO TO 101
HDIF = H(I,J) - H(I,J+1)
IF (HDIF .LT. 0.00001 .AND. HDIF .GT. -0.00001) HDIF = 1.
IF (FY(I,J) .LT. 1.0E-70 .AND. FY(I,J) .GT. -1.0E-70) GO TO 101
AFY2 = ABS(FY(I,J))
KY(I,J) = ABS((BLY(I,J) * (AFY2 ** XP)) / HDIF)
101 RETURN
END
SUBROUTINE CAL2 (MGO)
C
C "************************************************** SUBROUTINE CAL2 ******************************************
C ************************************************** COMMON VARIABLES **************************************************
C COMMON H(38,27), BH(38,27), FX(38,27), BFX(38,27), FY(38,27),
2 BFY(38,27), FR(38,27), BFR(38,27), KX(38,27), BKX(38,27),
3 KY(38,27), BKY(38,27), BLX(38,27), BLY(38,27), IA(38,27),
4 I,J,IMAX,JMAX,ITER,CNTER,ERINF,IMAXM1,JMAXM1,XP
REAL KX,KY
INTEGER CNTER
C
C "************************************************** COUNT ENTERED VALUES **************************************************
C KOUNT = 0
DO 100 I = 1,IMAX
DO 101 J = 1,JMAX
IF (BH(I,J) .NE. 0) KOUNT = KOUNT + 1
IF (BFR(I,J) .NE. 0) KOUNT = KOUNT + 1
101 CONTINUE
100 CONTINUE
DO 102 I = 1,IMAXM1
DO 103 J = 1, JMAX
  IF (BFX(I, J).NE.0.) KOUNT = KOUNT + 1
  IF (BKX(I, J).NE.0.) KOUNT = KOUNT + 1
103 CONTINUE
102 CONTINUE
  DO 104 I = 1, IMAX
    DO 105 J = 1, JMAXM1
    IF (BFY(I, J).NE.0.) KOUNT = KOUNT + 1
    IF (BKY(I, J).NE.0.) KOUNT = KOUNT + 1
105 CONTINUE
104 CONTINUE
C MAKE TESTS
C
  KBV = (3*IMAX*JMAX)-IMAX-JMAX
  KDIF = IABS(KBV-KOUNT)
  KK1 = 0
  IF (KBV.GT.KOUNT) KK1 = 1
  IF (KBV.LT.KOUNT) KK1 = 2
C WRITE MESSAGES
C
  WRITE (6, 107)
  107 FORMAT ('DATA CHECKED USING SUBROUTINE CAL2')
  IF (KK1.EQ.0) GO TO 108
  IF (KK1.EQ.1) GO TO 109
  GO TO 110
108 WRITE (6, 111) KOUNT, KBV
111 FORMAT ('NUMBER OF ENTERED VALUES (', 15, ') EQUALS REQUIRED NUMBER (', 15, ')')
  RETURN
109 CONTINUE
  IF (MGZ.EQ.1) GO TO 114
  WRITE (6, 112) KOUNT, KBV, KDIF
  112 FORMAT ('EXECUTION TERMINATED - NUMBER OF ENTERED VALUES (', 215, ') IS LESS THAN REQUIRED NUMBER (', 15, ') BY ', 15)
  CALL EXIT
110 CONTINUE
  IF (MGZ.EQ.1) GO TO 114
  WRITE (6, 113) KOUNT, KBV, KDIF
  113 FORMAT ('EXECUTION TERMINATED - NUMBER OF ENTERED VALUES (', 15, ')')
215, ') EXCEEDS REQUIRED NUMBER (', 15, ') BY ', 15
CALL EXIT
114 CONTINUE
WRITE (6, 115) KOUNT, KBV
115 FORMAT ('0 NUMBER OF ENTERED VALUES (', 15, ') NOT EQUAL TO REQUIRED NUMBER (', 15, ') - EXECUTION CONTINUES')
RETURN
END
SUBROUTINE CAL3 (MFAC, KCHEK)
C
C ********************************************************************
C SUBROUTINE CAL3 ********************************************************
C
C HIGHEST STATEMENT NUMBER IS 137
C
C DECLARE COMMON VARIABLES
C
COMMON H(38, 27), BH(38, 27), FX(38, 27), BFX(38, 27), FY(38, 27),
2 BFY(38, 27), FR(38, 27), BFR(38, 27), KX(38, 27), BKX(38, 27),
3 KY(38, 27), BKY(38, 27), BLX(38, 27), BLY(38, 27), IA(38, 27),
4 J, I, JMAX, JMAX, ITER, CNTER, ERINF, IMAXM1, JMAXM1, XP
REAL KX, KY
INTEGER CNTER
C
C SUM AND COUNT SPECIFIED NON-ZERO BH
C
SUMH=0.
NUMH=0
DO 100 I=1, IMAX
DO 101 J=1, JMAX
IF (BH(I, J), EQ, 12345), OR, BH(I, J), EQ, 0.), GO TO 103
SUMH=SUMH+BH(I, J)
NUMH=NUMH+1
103 CONTINUE
101 CONTINUE
100 CONTINUE
C
C SUM AND COUNT SPECIFIED K AND SUM L
C
SUMK=0.
NUMK=0
SUML=0.
NUML=(2*IMAX*JMAX)-IMAX-JMAX
DO 104 I=1,IMAX+1
DO 105 J=1,JMAX
IF (BKX(I,J).EQ.12345 OR BKX(I,J).EQ.0) GO TO 106
SUMK=SUMK+BKX(I,J)
NUMK=NUMK+1
106 SUML=SUML+BLX(I,J)
105 CONTINUE
104 CONTINUE
DO 107 I=1,IMAX
DO 108 J=1,JMAX+1
IF ( BKY(I,J).EQ.12345 OR BKY(I,J).EQ.0 ) GO TO 109
SUMK=SUMK+BKY(I,J)
NUMK=NUMK+1
109 SUML=SUML+BLY(I,J)
108 CONTINUE
107 CONTINUE
C CALCULATE BFR BALANCE
C
SUMFR=0.
DO 110 I=1,IMAX
DO 112 J=1,JMAX
IF ( BFR(I,J).NE.12345 AND BFR(I,J).NE.0 ) SUMFR=SUMFR+BFR(I,J)
110 CONTINUE
112 CONTINUE
C COUNT UNSPECIFIED BFR EXCLUDING THOSE WITH INAPPROPRIATE HEAD
C
IF ( NUMH.EQ.0 ) NUMH=1
HEADAV=SUMH/NUMH
NUMFR=0
DO 113 I=1,IMAX
DO 114 J=1,JMAX
IF ( SUMFR.LE.0. ) GO TO 113
IF ( BFR(I,J).NE.0 ) GO TO 114
IF ( BH(I,J).EQ.12345 AND HEADAV.GE.0 ) NUMFR=NUMFR+1
2 NUMFR=NUMFR+1
GO TO 114
113 IF (BFR(I,J).NE.0) GO TO 114
    IF (BH(I,J).EQ.12345..AND..HEADAV.LE.0.) NUMFR=NUMFR+1
    IF (BH(I,J).NE.0..AND..BH(I,J).NE.12345..AND..BH(I,J).GE..HEADAV)
      2 NUMFR=NUMFR+1
114 CONTINUE
112 CONTINUE
111 CONTINUE
C DISTRIBUTE FR TO BALANCE
C
    IF (NUMFR.EQ.0) NUMFR=1
    FRBAL=SUMFR/NUMFR
    DO 115 I=1,IMAX
      DO 116 J=1,JMAX
        IF (BFR(I,J).EQ.0) GO TO 137
        FR(I,J)=BFR(I,J)
        IF (BFR(I,J).EQ.12345) FR(I,J)=0.
        GO TO 118
117 IF (SUMFR.LE.0.) GO TO 117
116 CONTINUE
115 CONTINUE
C ASSIGN INITIAL VALUES OF H
C
    IF (NUMK.EQ.0) NUMK=1
    KAVER=SUMK/NUMK
    LENGAV=SUML/NUML
    IF (KAVER.EQ.0.) KAVER=1.
    RATIO=LENGAV/KAVER
    IF (RATIO.EQ.0.) RATIO=1.
    DO 119 I=1,IMAX
      DO 120 J=1,JMAX

IF (BH(I,J) .NE. 0.) H(I,J) = BH(I,J)
IF (BH(I,J) .EQ. 0. .AND. BFR(I,J) .NE. 0. .AND. BFR(I,J) .NE. 12345.)
  2  H(I,J) = HEADAV + (MFAC * RATIO * BFR(I,J))
IF (BH(I,J) .EQ. 0. .AND. FR(I,J) .NE. 0.)
  2  H(I,J) = HEADAV + (MFAC * RATIO * FR(I,J))
120 CONTINUE
119 CONTINUE
  CALL CAL4
  DO 121 I = 1, IMAX
  DO 122 J = 1, JMAX
  IF (H(I,J) .EQ. 12345.) H(I,J) = 0.
122 CONTINUE
121 CONTINUE

C ASSIGN INITIAL VALUES OF FX, FY, KX, AND KY
C
  KCHEK = 0
  DO 129 I = 1, IMAXM1
  DO 130 J = 1, JMAX
  FX(I,J) = BFX(I,J)
  IF (BFX(I,J) .EQ. 12345.) FX(I,J) = 0.
  IF (BKX(I,J) .NE. 0.) GO TO 135
  KX(I,J) = KAVE
  KCHEK = 1
  GO TO 131
135 KX(I,J) = BKX(I,J)
  IF (BKX(I,J) .EQ. 12345.) KX(I,J) = 0.
131 CONTINUE
130 CONTINUE
129 CONTINUE
  DO 132 I = 1, IMAX
  DO 133 J = 1, JMAXM1
  FY(I,J) = BFY(I,J)
  IF (BFY(I,J) .EQ. 12345.) FY(I,J) = 0.
  IF (BKY(I,J) .NE. 0.) GO TO 136
  KY(I,J) = KAVE
  KCHEK = 1
  GO TO 134
136 KY(I,J) = BKY(I,J)
  IF (BKY(I,J) .EQ. 12345.) KY(I,J) = 0.
134 CONTINUE
C WRITE OUT ASSIGNED VALUES
C
WRITE (6,123) MFAC
123 FORMAT ('INITIAL VALUES CALCULATED USING SUBROUTINE CAL3 WITH HEA
2D MULTIPLIER OF ',11)
WRITE (6,124)
124 FORMAT ('0',14X,'I',9X,'J',10X,'H',12X,'FX',12X,'FY',12X,'FR',12X,
2 'KK',12X,'KY')
DO 125 1=1,IMAX
WRITE (6,126)
126 FORMAT (' ',14X,'I',9X,'J',10X,'H(I,J),FX(I,J),FY(I,J),FR(I,J),KK(I,J),KY(I,J)
128 FORMAT (' ',12X,15,5X,15,3X,6(G10.1,4X)',
127 CONTINUE
125 CONTINUE
RETURN
END

SUBROUTINE CAL4
C
HIGHEST STATEMENT NUMBER IS 115
C
DECLARE COMMON VARIABLES
C
COMMON H(38,27), BH(38,27), FX(38,27), BFX(38,27), FY(38,27),
2 BFY(38,27), FR(38,27), BFR(38,27), KX(38,27), BXX(38,27),
3 KY(38,27), KXY(38,27), BLX(38,27), BLY(38,27), IA(38,27),
4 I,J,IMAX,JMAX,ITER,CNTER,ERINF,IMAX1,JMAX1,XP
REAL KK,KY
INTEGER CNTER
C
SET IA EQ 1 FOR NODES WITH ASSIGNED VALUES OF H
DO 100 1=1,IMAX
DO 101 J=1,JMAX
IA(I,J)=0
IF (H(I,J).EQ.0.) GO TO 102
IA(I,J)=1
102 CONTINUE
101 CONTINUE
100 CONTINUE
NLIM=INAX+JMAX-2
DO 103 I=1,IMAX
DO 104 J=1,JMAX
IF (IA(I,J).NE.1) GO TO 105
DO 106 N=1,NLIM
DO 107 M=1,N
C SET THIRD QUADRANT
C
IF (I-M.LT.1) GO TO 108
IF (J-N+M.LT.1) GO TO 108
IF (IA(I-M,J-N+M).EQ.0) GO TO 109
IF (IA(I-M,J-N+M).LE.N+1) GO TO 108
IA(I-M,J-N+M)=N+1
108 CONTINUE
C
C SET FOURTH QUADRANT
C
IF (I-N+M.LT.1) GO TO 110
IF (J+M.GT.JMAX) GO TO 110
IF (IA(I-N+M,J+M).EQ.0) GO TO 111
IF (IA(I-N+M,J+M).LE.N+1) GO TO 110
111 H(I-N+M,J+M)=H(I,J)
IA(I-N+M,J+M)=N+1
110 CONTINUE
C
C SET FIRST QUADRANT
C
IF (I+M.GT.IMAX) GO TO 112
IF (J+N-M.GT.JMAX) GO TO 112
IF (IA(I+M,J+N-M).EQ.0) GO TO 113
IF (IA(I+M,J+N-M).LE.N+1) GO TO 112
113 H(I+M,J+N-M)=H(I,J)
IAll+M,J+N-Ml=N+l
112 CONTINUE

C SET SECOND QUADRANT
C
IF (I+M-N-M.GT.IMAX) GO TO 114
IF (J-M.LT.1) GO TO 114
IF (IA(I+M-N-M,J-M).EQ.0) GO TO 115
IF (IA(I+M-N-M,J-M).LE.N+1) GO TO 114
115 H(I+M-N-M,J-M)=H(I,J)
IA(I+M-N-M,J-M)=N+1
114 CONTINUE
107 CONTINUE
106 CONTINUE
105 CONTINUE
104 CONTINUE
103 CONTINUE
RETURN
END

SUBROUTINE OUT1 (IM1,HEDERX,HEDERY,KNTR)

******************************************************************************
******************************************************************************
******************************************************************************
******************************************************************************

HIGHEST STATEMENT NUMBER IS 106

DECLARE COMMON VARIABLES

COMMON H(38,27), BH(38,27), FX(38,27), BFX(38,27), FY(38,27),
2 BFY(38,27), FR(38,27), BFR(38,27), KX(38,27), BKX(38,27),
3 KY(38,27), BKY(38,27), BLX(38,27), BLY(38,27), IA(38,27),
4 I,J,IMAX,JMAX,ITER,CNTER,ERINF,IMAXM1,JMAXM1,XP

REAL KX,KY
INTEGER CNTER

WRITE HEADING FOR PRELIMINARY VALUES

IF (ITER.EQ.1.AND.I.EQ.1.AND.J.EQ.1) GO TO 100
GO TO 104
100 WRITE (6,106)
106 FORMAT (*10OUTPUT WITH SUBROUTINE OUT1*)
WRITE (6,102)
102 FORMAT(*1ITERATION I J H FX FY
2 FR KX KY ERINF FEC HEDERX HEDERY,
3 HEC**1)
C CHECK FOR WRITE
C
104 IF (ITER.EQ.1.AND.I.EQ.1.AND.J.EQ.11 MK1=0
 IF (ITER.EQ.1.AND.I.EQ.1.AND.J.EQ.1) MP1=2**MM1
 IF (ITER.EQ.1.OR.ITER.EQ.2) GO TO 101
 IF (ITER.EQ.MK1.OR.ITER.EQ.MK1+1) GO TO 101
 RETURN
C WRITE VALUES AT NODE
C
101 WRITE (6,103) ITER,I,J,H(I,J),FX(I,J),FY(I,J),FR(I,J),KX(I,J),
2 KY(I,J),ERINF,CNTER,HEDERX,HEDERY,KNTR
103 FORMAT (* 10,2X,15,5X,15,5X,15,3X,7Gl0.3,1X,13,1X,2Gl0.3,1X,131
 IF (I.NE.IMAX.OR.J.NE.JMAX) GO TO 105
 IF (ITER.EQ.MK1+1) MK1=MK1+MPI
105 RETURN
END
SUBROUTINE OUT2 (KO2)
C
C******************** SUBROUTINE OUT2 ********************
C
C HIGHEST STATEMENT NUMBER IS 122
C
C DECLARE COMMON VARIABLES
C
COMMON H(38,27), BH(38,27), FX(38,27), BFX(38,27), FY(38,27),
2 BFY(38,27), FR(38,27), BFR(38,27), KX(38,27), BKX(38,27),
3 KY(38,27), BKY(38,27), BLX(38,27), BLY(38,27), IA(38,27),
4 I,J,IMAX,JMAX,ITER,CNTER,ERINF,IMAX1,JMAX1,XP
REAL KX,KY
INTEGER CNTER
IF (KO2.EQ.1) GO TO 121
KO2=KO2+1
WRITE (6,120)
120 FORMAT ('OUTPUT WITH SUBROUTINE OUT2')
WRITE (6,100)
100 FORMAT ('FINAL VALUES LISTED BELOW. ENTERED VALUE OF 12345 MEANS ZERO. ENTERED VALUE OF ZERO MEANS NO VALUE (VALUE TO BE SOLVED 3D FOR)')
WRITE (6,122)
122 FORMAT ('IF TERMINATION DUE TO ITERATION COUNT, VALUES LISTED ARE FOR NEXT TO LAST ITERATION')
WRITE (6,101)
101 FORMAT ('J ENTERED HEAD FINAL HEAD ENTERED RECHARGE FINAL RECHARGE')
WRITE (6,106)
106 FORMAT ('FLOW INTO NODE IS POSITIVE')
DO 102 J=1,JMAX
WRITE (6,103)
103 FORMAT (')
DO 104 J=1,JMAX
WRITE (6,105) I,J,BH(I,J),H(I,J),BFR(I,J),FR(I,J)
105 FORMAT (')
104 CONTINUE
102 CONTINUE
WRITE (6,107)
107 FORMAT ('J ENTERED FLOW IN FINAL FLOW IN ENTERED FLOW IN FINAL FLOW IN')
WRITE (6,108)
108 FORMAT ('BRANCH TO RIGHT BRANCH TO RIGHT BRANCH ABOVE BRANCH ABOVE')
WRITE (6,109)
109 FORMAT ('FLOW TO RIGHT IS POSITIVE')
DO 110 I=1,IMAX
WRITE (6,103)
110 FORMAT (')
DO 111 J=1,JMAX
WRITE (6,105) I,J,BF(I,J),F(I,J),BF(I,J),F(I,J)
111 CONTINUE
110 CONTINUE
WRITE (6,112)
112 FORMAT ('J ENTERED PERM. IN FINAL PERM. IN ENTERED PERM. IN FINAL PERM. IN')
WRITE (6,108)
DO 113 I=1,IMAX
WRITE (6,103)
DO 114 J=1,JMAX
WRITE (6,105) I,J,BKX(I,J),KX(I,J),BKY(I,J),KY(I,J)
114 CONTINUE
113 CONTINUE
WRITE (6,115)
115 FORMAT (I,J) ENTERED LENGTH OF ENTERED LENGTH OF
WRITE (6,116)
116 FORMAT (10X,'BRANCH TO RIGHT BRANCH ABOVE')
DO 117 I=1,IMAX
WRITE (6,103)
DO 118 J=1,JMAX
WRITE (6,119) I,J,BLX(I,J),BLY(I,J)
119 FORMAT (215,2G20.8)
118 CONTINUE
117 CONTINUE
121 RETURN
END

SUBROUTINE OUT3 (ID1,ID2,ID3,ID4,ID5,ID6,ID7,ID8,ID9)

C *** SUBROUTINE OUT3 ************
C
C HIGHEST STATEMENT NUMBER IS 134
C
C DECLARE COMMON VARIABLES
C
COMMON H(38,27), BH(38,27), FX(38,27), BFX(38,27), FY(38,27),
2 BFY(38,27), FR(38,27), BFR(38,27), KX(38,27), BKX(38,27),
3 KY(38,27), BKY(38,27), BLX(38,27), BLY(38,27), IA(38,27),
4 I,J,IMAX,JMAX,ITER,CNTER,ERINF,IMAXM1,JMAXM1,XP

REAL XX,XY
INTEGER CNTER

C DECLARE OUT3 VARIABLES
C
DIMENSION A(4,16)

C
C INITIALIZE
C
MAP=1
ISHEET=1
JSHEET=1
M=1
L=8
LN=9
C
C SET ARRAY TO ZERO
C
107 CONTINUE
DO 101 M=1,4
DO 102 N=1,16
A(M1,N)=0.
102 CONTINUE
101 CONTINUE
100 CONTINUE
C
C SET VARIABLES FOR SHEET
C
LM1=L-7
LPL1=L+1
MPL1=M+1
MPL2=M+2
MPL3=M+3
MPL4=M+4
C
C TRANSFER ACCORDING TO MAP
C
IF (MAP.EQ.2) GO TO 110
IF (MAP.EQ.3) GO TO 111
IF (MAP.EQ.4) GO TO 112
IF (MAP.EQ.5) GO TO 113
IF (MAP.EQ.6) GO TO 114
IF (MAP.EQ.7) GO TO 115
IF (MAP.EQ.8) GO TO 116
IF (MAP.EQ.9) GO TO 117
C
C FILL ARRAY FOR MAP 1 AND WRITE
C
IF (ID1.EQ.1) GO TO 108
DO 103 L1=LM17,L
N=IABS(L1-LN)
M2=0
DO 104 M1=M,MPL3
M2=M2+1
IF (M1.GT.IMAX.OR.L1.GT.JMAX) GO TO 104
A(M2,N)=B(M1,L1)
104 CONTINUE
103 CONTINUE
GO TO 118
C FILL ARRAY FOR MAP 2 AND WRITE
C
IF (ID2.EQ.1) GO TO 108
DO 119 L1=LM17,L
N=IABS(L1-LN)
M2=0
DO 120 M1=M,MPL3
M2=M2+1
IF (M1.GT.IMAX.OR.L1.GT.JMAX) GO TO 120
A(M2,N)=H(M1,L1)
120 CONTINUE
119 CONTINUE
GO TO 118
C FILL ARRAY FOR MAP 3 AND WRITE
C
IF (ID3.EQ.1) GO TO 108
DO 121 L1=LM17,L
N=IABS(L1-LN)
M2=0
DO 122 M1=M,MPL3
M2=M2+1
IF (M1.GT.IMAX.OR.L1.GT.JMAX) GO TO 122
A(M2,N)=BFR(M1,L1)
122 CONTINUE
121 CONTINUE
GO TO 118
C FILL ARRAY FOR MAP 4 AND WRITE
C 112 IF (ID4.EQ.1) GO TO 108
   DO 123 L1=LM17,L
   N=IABS(L1-LN)
   M2=0
   DO 124 M1=M,MPL3
   M2=M2+1
   IF (M1.GT.IMAX.OR.L1.GT.JMAX) GO TO 124
   AM2,N=FR(M1,L1)
124 CONTINUE
123 CONTINUE
   GO TO 118
C
C FILL ARRAY FOR MAP 5 AND WRITE
C 113 IF (ID5.EQ.1) GO TO 108
   DO 125 L1=LM17,L
   N=IABS(L1-LN)
   M2=0
   DO 126 M1=M,MPL3
   M2=M2+1
   IF (M1.GT.IMAX.OR.L1.GT.JMAX) GO TO 126
   AM2,N=BFY(M1,L1)
   AM2,N+8=BFX(M1,L1)
126 CONTINUE
125 CONTINUE
   GO TO 118
C
C FILL ARRAY FOR MAP 6 AND WRITE
C 114 IF (ID6.EQ.1) GO TO 108
   DO 127 L1=LM17,L
   N=IABS(L1-LN)
   M2=0
   DO 128 M1=M,MPL3
   M2=M2+1
   IF (M1.GT.IMAX.OR.L1.GT.JMAX) GO TO 128
   AM2,N=FY(M1,L1)
   AM2,N+8=FX(M1,L1)
128 CONTINUE
127 CONTINUE
GO TO 118
C
C FILL ARRAY FOR MAP 7 AND WRITE
C
115 IF (ID7.EQ.1) GO TO 108
DO 129 L1=LM17,L
N=IABS(L1-LN)
M2=0
DO 130 M1=M,MPL3
M2=M2+1
IF (M1.GT.JMAX OR L1.GT.JMAX) GO TO 130
A(M2,N)=BKY(M1,L1)
A(M2,N+8)=BKX(M1,L1)
130 CONTINUE
129 CONTINUE
GO TO 118
C
C FILL ARRAY FOR MAP 8 AND WRITE
C
116 IF (ID8.EQ.1) GO TO 108
DO 131 L1=LM17,L
N=IABS(L1-LN)
M2=0
DO 132 M1=M,MPL3
M2=M2+1
IF (M1.GT.JMAX OR L1.GT.JMAX) GO TO 132
A(M2,N)=KY(M1,L1)
A(M2,N+8)=KX(M1,L1)
132 CONTINUE
131 CONTINUE
GO TO 118
C
C FILL ARRAY FOR MAP 9 AND WRITE
C
117 IF (ID9.EQ.1) GO TO 108
DO 133 L1=LM17,L
N=IABS(L1-LN)
M2=0
DO 134 M1=M,MPL3
M2=M2+1
IF (M1.GT.JMAX OR L1.GT.JMAX) GO TO 134
A(M2,N) = BLY(M1,L1)
A(M2,N+8) = BLX(M1,L1)
134 CONTINUE
133 CONTINUE
GO TO 118

C INCREMENT JSHEET
C
105 CONTINUE
IF (L .GT. JMAX) GO TO 106
L = L + 8
JSHEET = JSHEET + 1
LN = LN + 8
GO TO 107

C INCREMENT ISHEET
C
106 CONTINUE
L = 8
JSHEET = 1
LN = 9
IF (M + 4 .GT. JMAX) GO TO 108
M = M + 4
ISHEET = ISHEET + 1
GO TO 107

C INCREMENT MAP
C
108 CONTINUE
IF (MAP .EQ. 9) GO TO 109
ISHEET = 1
JSHEET = 1
M = 1
L = 8
LN = 9
MAP = MAP + 1
GO TO 107

C TRANSFER POINT FOR WRITE
C
118 CALL OUT3A (MAP, M, L, MPL1, MPL2, MPL3, LM17, ISHEET, JSHEET, MPL4, LPL1, A)
GO TO 105
109 RETURN
END

SUBROUTINE OUT3A (MAP, M, L, MPL1, MPL2, MPL3, LM17, ISHEET, JSHEET,
2 MPL4, LPL1, A)

C **********************************************
C SUBROUTINE OUT3A **********************************************
C **********************************************
C HIGHEST STATEMENT NUMBER IS 135
C OTHER NUMBERS AVAILABLE ARE 125 THRU 130
C
C DECLARE COMMON VARIABLES
C
COMMON H(38,27), BH(38,27), FX(38,27), FY(38,27),
2 BFY(38,27), FR(38,27), BKX(38,27),
3 KY(38,27), BKY(38,27), BLX(38,27), BLY(38,27), IA(38,27),
4 I, J, IMAX, JMAX, ITER, CNTER, ERINF, IMAIM1, JMAIM1, XP

REAL KX, KY
INTEGER CNTER

C DECLARE OUT3A VARIABLES
C
DIMENSION A(4,16)

C WRITE MAPS
C
WRITE (6,131)

131 FORMAT ('1',/////////)
IF (MAP.EQ.5 .OR. MAP.EQ.6 .OR. MAP.EQ.7 .OR. MAP.EQ.8 .OR. MAP.EQ.9)
2 GO TO 124

C WRITE NODE MAPS
C
C WRITE FIRST LINE
C
IF (MAP.EQ.1) WRITE (6,100) M, MPL3, LM17, L
100 FORMAT ("",6X,"ENTERED HEAD AT NODES (BH) - I FROM ',12,' TO ',12
2, ' - J FROM ',12,' TO ',12,23X,"MAP 1")
IF (MAP.EQ.2) WRITE (6,111) M, MPL3, LM17, L

111 FORMAT (' ',6X,'FINAL HEAD AT NODES (H) - I FROM ',I2,' TO ',I2,')
   2- J FROM ',I2,' TO ',I2,26X,'MAP 2')
   IF (MAP.EQ.3) WRITE (6,112) M,MPL3,LMT7,L
112 FORMAT (' ',6X,'ENTERED RECHARGE AT NODES (BFR) - I FROM ',I2,' TO ')
   2 ',I2,' - J FROM ',I2,' TO ',I2,18X,'MAP 3')
   IF (MAP.EQ.4) WRITE (6,113) M,MPL3,LMT7,L
113 FORMAT (' ',6X,'FINAL RECHARGE AT NODES (FR) - I FROM ',I2,' TO ')
   2 ',I2,' - J FROM ',I2,' TO ',I2,21X,'MAP 4')
C WRITE SECOND LINE
C WRITE (6,101) ISHEET,JSHEET
101 FORMAT (' ',85X,'SHEET ',I2,'-',I2)'
C WRITE THIRD LINE
C IF (MAP.EQ.1) WRITE (6,102)
102 FORMAT (' ',11X,'(12345 INDICATES HEAD SPECIFIED AS ZERO, ZERO IND'
   2ICATES NO HEAD SPECIFIED)*)
   IF (MAP.EQ.2) WRITE (6,103)
   IF (MAP.EQ.3) WRITE (6,114)
114 FORMAT (' ',11X,'(12345 INDICATES ZERO SPECIFIED, ZERO IS UNSPEC'
   2IRED, FLOW OUT IS NEGATIVE)*)
   IF (MAP.EQ.4) WRITE (6,115)
115 FORMAT (' ',11X,'(FLOW OUT OF SYSTEM IS NEGATIVE)*)
C WRITE HEADINGS
C WRITE (6,103)
103 FORMAT (' ')  
   WRITE (6,104) M,MPL1,MPL2,MPL3,MPL4
104 FORMAT (' ',6X,'J=',4X,'I=',',12,17X,12,17X,12,17X,12,17X,12)
   WRITE (6,103)
   WRITE (6,105) LPL1,LPL1
105 FORMAT (' ',6X,12,8X,*,18X,*,18X,*,18X,*,18X,*,18X,*,12)
   WRITE (6,106)
106 FORMAT (' ')  
C WRITE DATA
C DO 109 N=1,8
IN=IABS(L+1-N)
WRITE (6,107)
107 FORMAT (' ',16X,'|',18X,'|',18X,'|',18X,'|')
WRITE (6,103)
WRITE (6,108) IN,A(1,N),A(2,N),A(3,N),A(4,N),IN
WRITE (6,106)
109 CONTINUE
WRITE (6,110) M,MPL1,MPL2,MPL3,MPL4
110 FORMAT (' ',15X,12,17X,12,17X,12,17X,12)
RETURN
C BRANCH MAPS
C WRITE FIRST LINE
C
124 IF (MAP.EQ.5) WRITE (6,116) M,MPL3,LM17,L
116 FORMAT (' ',6X,'ENTERED FLOW IN BRANCHES (BFX AND BFY) - I FROM ',
212,' TO ',12,' - J FROM ',12,' TO ',12,11X,'MAP 5')
IF (MAP.EQ.6) WRITE (6,117) M,MPL3,LM17,L
117 FORMAT (' ',6X,'FINAL FLOW IN BRANCHES (FX AND FY) - I FROM ',12,' 2 TO ',12,' - J FROM ',12,' TO ',12,15X,'MAP 6')
IF (MAP.EQ.7) WRITE (6,118) M,MPL3,LM17,L
118 FORMAT (' ',6X,'ENTERED PERMEABILITY IN BRANCHES (BFX AND BFY) - I 2 FROM ',12,' TO ',12,' - J FROM ',12,' TO ',12,3X,'MAP 7')
IF (MAP.EQ.8) WRITE (6,119) M,MPL3,LM17,L
119 FORMAT (' ',6X,'FINAL PERMEABILITY IN BRANCHES (FX AND FY) - I FROM 2M ',12,' TO ',12,' - J FROM ',12,' TO ',12,7X,'MAP 8')
IF (MAP.EQ.9) WRITE (6,120) M,MPL3,LM17,L
120 FORMAT (' ',6X,'LENGTH (ENTERED) OF BRANCHES (BFX AND BFY) - I FROM 2M ',12,' TO ',12,' - J FROM ',12,' TO ',12,7X,'MAP 9')
C WRITE SECOND LINE
C
WRITE (6,101) ISHEET,JSHEET
C WRITE THIRD LINE
C
IF (MAP.EQ.5) WRITE (6,121)
121 FORMAT (' ',11X,'12345 IS 0 SPECIFIED, 0 IS UNSPECIFIED, FLOW DOW
2N OR TO LEFT IS NEGATIVE))
IF (MAP.EQ.6) WRITE (6,122)
122 FORMAT ('**','11X, '(FLOW DOWN OR TO LEFT IS NEGATIVE)' )
IF (MAP.EQ.7) WRITE (6,123)
123 FORMAT ('**','11X, '(12345 IS ZERO SPECIFIED, ZERO IS UNSPECIFIED)' )
IF (MAP.EQ.8.OR.MAP.EQ.9) WRITE (6,124)
C WRITE HEADINGS
C WRITE (6,103)
WRITE (6,104) M, MPL1, MPL2, MPL3, MPL4
WRITE (6,103)
WRITE (6,105) LPL1, LPL2
WRITE (6,132)
132 FORMAT ('**')
C WRITE DATA
C DO 133 N=1,8
IN=IABS(L+1-N)
WRITE (6,134) A(1,N), A(2,N), A(3,N), A(4,N)
134 FORMAT ('**','9X,' (',GL14.7,' ) (',GL14.7,' ) (',GL14.7,' ) (',GL14.7,' )')
WRITE (6,103)
WRITE (6,135) IN, A(1,N+8), A(2,N+8), A(3,N+8), A(4,N+8), IN
135 FORMAT ('**','6X,' ,GL14.7,' ) * (',GL14.7,' ) * (',GL14.7,' ) * (',GL14.7,' ) *
2(*,GL14.7,' ) * ',12)
WRITE (6,132)
133 CONTINUE
WRITE (6,110) M, MPL1, MPL2, MPL3, MPL4
RETURN
END
SUBROUTINE OUT3B (KK, FERROR, HERROR, MAXIT, KNTR)
C
C *****************************************************
C *****************************************************
C SUBROUTINE OUT3B *****************************************************
C *****************************************************
C HIGHEST STATEMENT NUMBER IS 112
C C DECLARE COMMON VARIABLES
C
COMMON H(38,27), BH(38,27), FX(38,27), BFH(38,27), FY(38,27),
FR(38,27), BFR(38,27), KX(38,27), BXK(38,27), FY(38,27), IA(38,27),
>J, IMAX, JMAX, ITER, CNTER, EINF, IMAX, JMAX, XP
REAL KX, KY
INTEGER CNTER

C WRITE HEADINGS
C
KK=KK+1
IF (KK .NE. 1) GO TO 112
WRITE (6,100)
100 FORMAT ('!', 'COMMENTS')
WRITE (6,101)
101 FORMAT ('!', '1X, 'OUTPUT WITH SUBROUTINE OUT3')
WRITE (6,102)
102 FORMAT ('!', '1X, ITER = ', ITPL1)
IF (CNTER .NE. 0 .OR. KNTR .NE. 0) GO TO 107
WRITE (6,103) FERROR
103 FORMAT (!', '11X,' ALL FLOW ERRORS LESS THAN VALUE SPECIFIED (' , G10.2) AND ALL HEAD ERRORS LESS THAN')
WRITE (6,104) HERROR, ITPL1
104 FORMAT ('!', '11X,' VALUE SPECIFIED (' , G10.3)' VALUES OF THESE ERRORS FOR FOLLOWING ITERATION (' , I2)' ARE')
WRITE (6,105)
105 FORMAT ('!', '11X,'LISTED BELOW.')
GO TO 106
106 CONTINUE
WRITE (6,108) FERROR
108 FORMAT ('!', '11X,' SOME FLOW ERRORS EXCEEDED VALUE SPECIFIED (' , G10.2), AND/OR SOME HEAD ERRORS')
WRITE (6,109) HERROR
109 FORMAT ('!', '11X,' EXCEEDED VALUE SPECIFIED (' , G10.3) ON ITERATION 2 PRECEDING MAXIMUM SPECIFIED. VALUES')
WRITE (6,110) MAXIT
110 FORMAT ('!', '11X,' OF THESE ERRORS FOR THE MAXIMUM SPECIFIED ITERATION 20N(' , I5)' ARE LISTED BELOW')
107 CONTINUE
WRITE (6,102)
WRITE (6,111)
**FORMAT** (*',11X,'ENTERED AND FINAL VALUES OF VARIABLES FOLLOW THE ERROR LISTING')

112 RETURN

END

**SUBROUTINE OUT3C** (KNTR, HEDERX, HEDERY)

C

C

C **SUBROUTINE OUT3C**

C

C

C **COMMON VARIABLE**

C

C COMMON H(38,27), BH(38,27), FX(38,27), BFX(38,27), FY(38,27),
2 BFY(38,27), FR(38,27), BFR(38,27), XX(38,27), BXX(38,27),
3 KY(38,27), BKY(38,27), BLX(38,27), BLX(38,27), IA(38,27),
4 I,J,IMAX,JMAX,ITER,CPNTER,ERINF,IMAXM1,JMAXM1,XP

REAL KX,KY

INTEGER CNTER

C

C WRITE HEADING ON FIRST SHEET

C

C IF (I.EQ.1.AND.J.EQ.1) LPK=1
C IF (I.EQ.1.AND.J.EQ.1) LPLIM=2
C IF (I.NE.1.OR.J.NE.1) GO TO 100
C WRITE (6,101)

101 FORMAT (*0')

WRITE (6,102)

102 FORMAT (*')

100 IF (LPK.NE.LPLIM) GO TO 103

WRITE (6,104)

104 FORMAT ('1',//'--------')

103 IF (I.EQ.1.AND.J.EQ.1) GO TO 111

GO TO 105

111 LPK=1

LPLIM=38

IF (I.EQ.1.AND.J.EQ.1) LPLIM=28

WRITE (6,106)

106 FORMAT (*',4X,'CUM. FLOW',36X,'CUM. HEAD')
WRITE (6,107)
107 FORMAT (' ',15X,'I',6X,'J',9X,'FLOW',4X,'ERRORS GT',7X,'X BRANCH',
     28X,'Y BRANCH ERRORS GT')
WRITE (6,108)
108 FORMAT (' ',32X,'ERROR',5X,'LIMIT',8X,'HEAD ERROR',6X,'HEAD ERROR,
     2 'LIMIT')
WRITE (6,102)
105 IF (J.NE.1) GO TO 109
WRITE (6,102)
LPK=LPK+1
109 WRITE (6,110) I,J,ERINF,CNTER,HEDERX,HEDERY,KNTR
110 FORMAT (' ',14X,12,5X,12,4X,14.7,2X,13,7X,14.7,2X,14.7,5X,13)
LPK=LPK+1
RETURN
END
SUBROUTINE OUT4
C ********** SUBROUTINE OUT4 **********
C HIGHEST STATEMENT NUMBER IS 103
C DECLARE COMMON VARIABLES
C COMMON H(38,27), BH(38,27), FX(38,27), BFX(38,27), FY(38,27),
     2 BFY(38,27), FR(38,27), BFR(38,27), KX(38,27), BKX(38,27),
     3 KY(38,27), BKY(38,27), BLX(38,27), BLY(38,27), IA(38,27),
     4 I,J,IMAX,JMAX,ITER,CNTER,ERINF,IMAXM1,JMAXM1,XP
     REAL KX,KY
     INTEGER CNTER
C PUNCH CARDS
C NCARD=IMAX*JMAX
WRITE (7,100) NCARD
100 FORMAT (110)
    DO 101 I=1,IMAX
    DO 102 J=1,JMAX
    WRITE (7,103) I,J,H(I,J),FX(I,J),FY(I,J),FR(I,J),KX(I,J),KY(I,J)
103 FORMAT (215,6E10.3)