5-1979

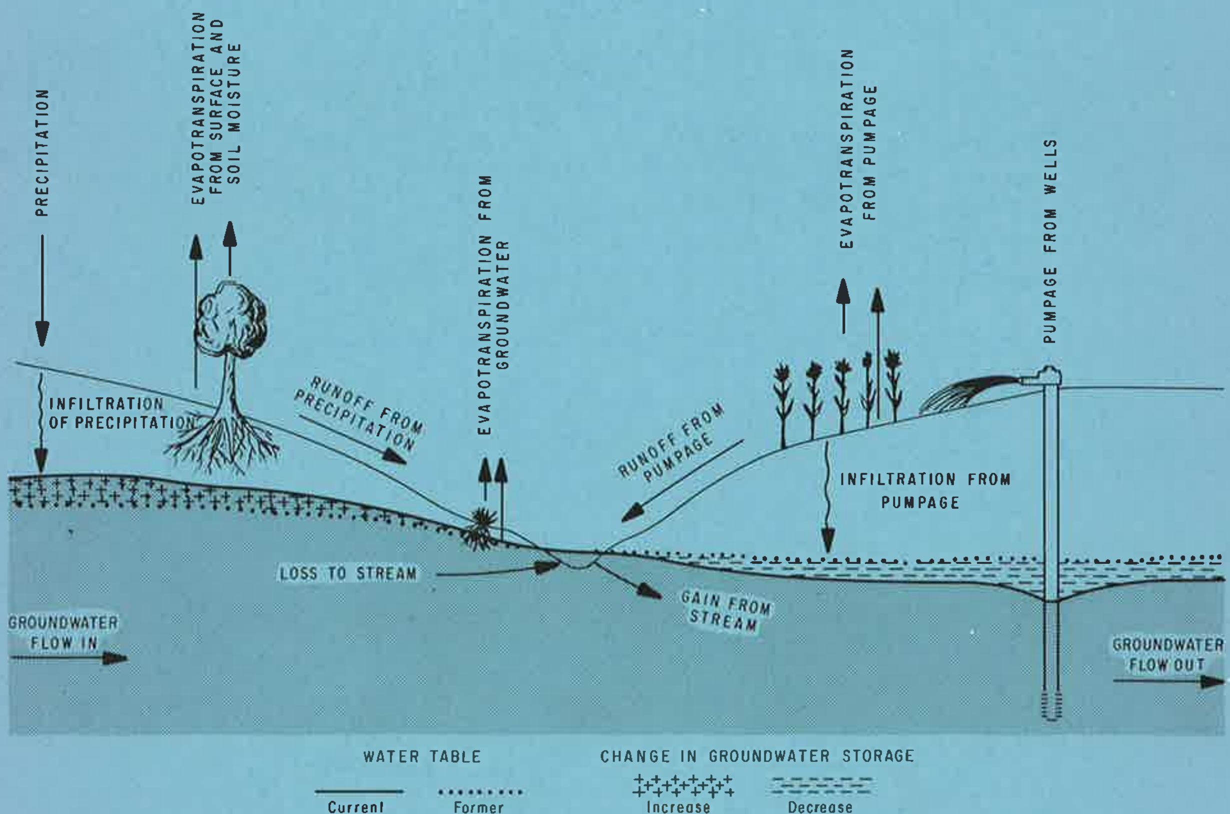
# Interpretive Study and Numerical Model of the Hydrogeology Upper Big Blue Natural Resources District.

Ralph E. Cady
*University of Nebraska-Lincoln*

Marilyn H. Ginsberg
*University of Nebraska-Lincoln*

**Interpretive Study and Numerical Model of the Hydrogeology**

# UPPER BIG BLUE NATURAL RESOURCES DISTRICT, NEBRASKA




Conservation and Survey Division
Institute of Agriculture and Natural Resources
University of Nebraska–Lincoln

# INTERPRETIVE STUDY AND NUMERICAL MODEL OF THE HYDROGEOLOGY UPPER BIG BLUE NATURAL RESOURCES DISTRICT

Ralph E. Cady and Marilyn H. Ginsberg

Prepared in cooperation with the
Upper Big Blue Natural Resources District

Conservation and Survey Division
Institute of Agriculture and Natural Resources
The University of Nebraska-Lincoln

May 1979

CONTENTS

Illustrations

Page



Table

PREFACE

Expansions of pumping from irrigation wells in the Upper Big Blue Natural Resources District (NRD) are reducing the quantity of the available groundwater supply. During the spring and early summer of 1974, proposals for studies relative to the understanding and management of the water resources of the Upper Big Blue River basin were formulated between the University of Nebraska and the Upper Big Blue NRD. This study is the result of a proposal by the Conservation and Survey Division, Institute of Agriculture and Natural Resources, of the University of Nebraska to determine the geohydrologic parameters and framework within the basin and to develop and apply numerical simulation methods for use in groundwater resource management. The project proposal was approved by the Upper Big Blue NRD Board of Directors on September 2, 1974. The University of Nebraska Board of Regents granted approval of the contract on November 22, 1974.

INTRODUCTION

The Upper Big Blue Natural Resources District (NRD) is located in southeast-central Nebraska (fig. 1). The NRD lies within the Big Blue River basin, has an area of 2,858 square miles, and comprises York County and parts of Butler, Seward, Saline, Fillmore, Polk, Hamilton, Clay, and Adams counties. Topographically the area is a broad loessial plain of low relief with local shallow depressions. Elevations range from a little more than 2,000 feet in the west to a little less than 1,400 feet in the east. Drainage is to the southeast and major streams are characterized by valleys having relatively broad flood plains and terraces.

Annual precipitation ranges from about 25 inches in the north western part of the area to about 28 inches in the southeastern part. About 57% of this precipitation falls during the irrigation season, May through August. Because of the desire to increase crop production, the number of irrigation wells has grown from 513 registered wells in 1950 to 9,433 at the end of 1977 (fig. 2a-2e). After the end of the 1977 irrigation season, water levels had declined an average of 14.82 feet and locally as much as 44.6 feet from predevelopment levels. From predevelopment to Spring 1977, the average decline was 11.98 feet. Figure 3a-3c consists of hydrographs of three observation wells whose records typify water-level conditions in much of the NRD. The Aurora well (fig. 3a) reflects the water level in an unconfined aquifer. The shallow observation well at York (fig. 3b) also

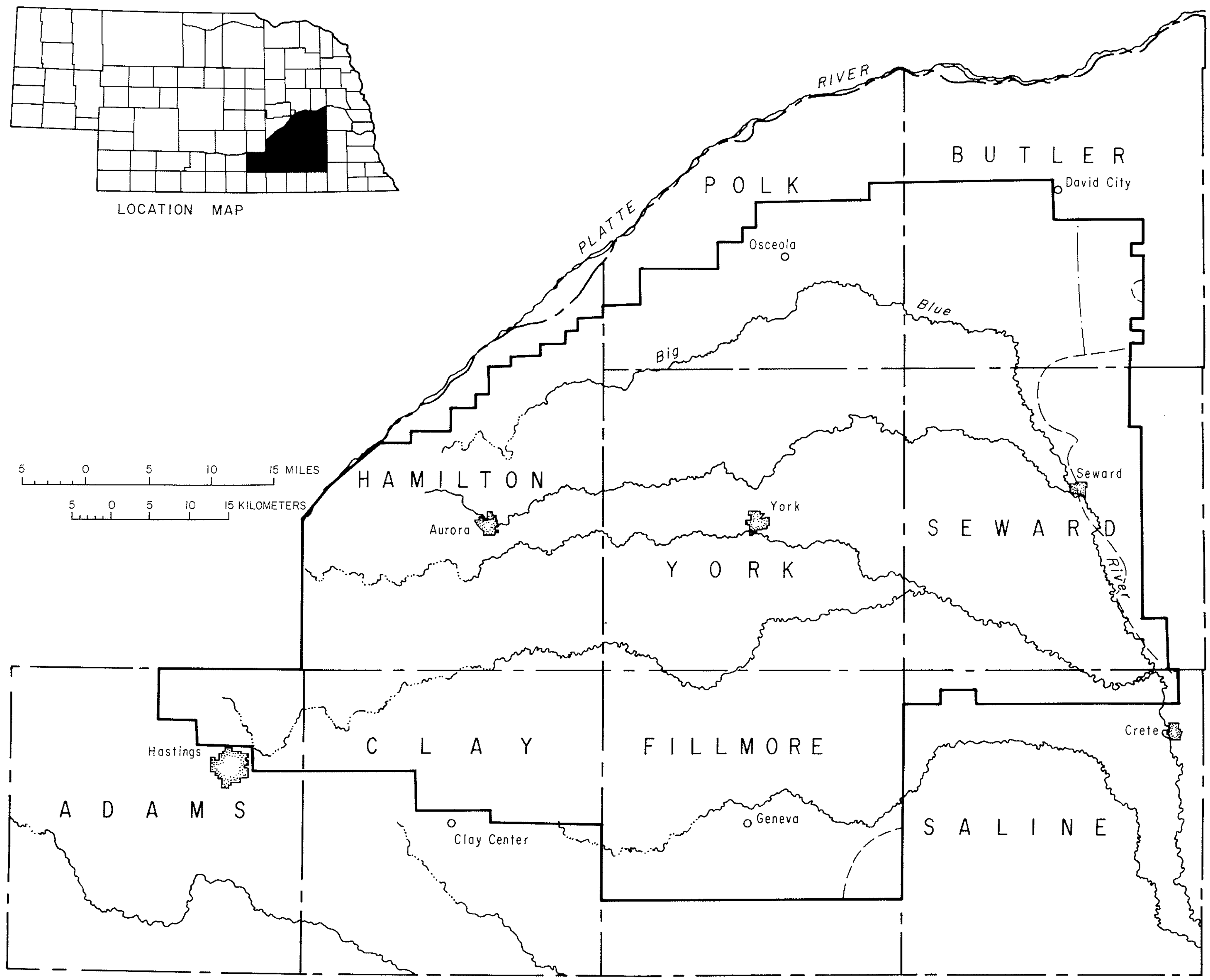



Figure 1-Location map

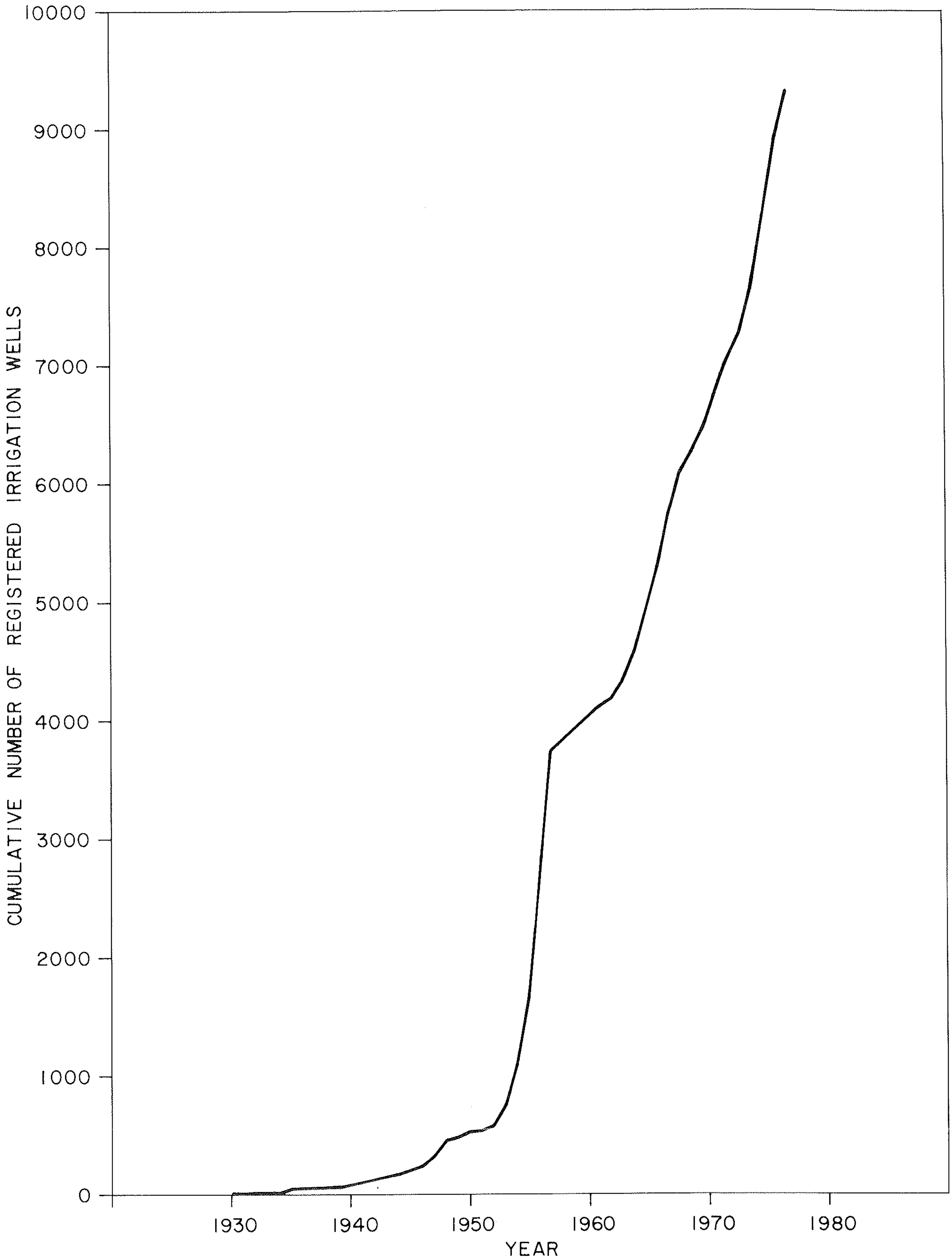



Figure 2a-Growth in number of irrigation wells in the Upper Big Blue NRD

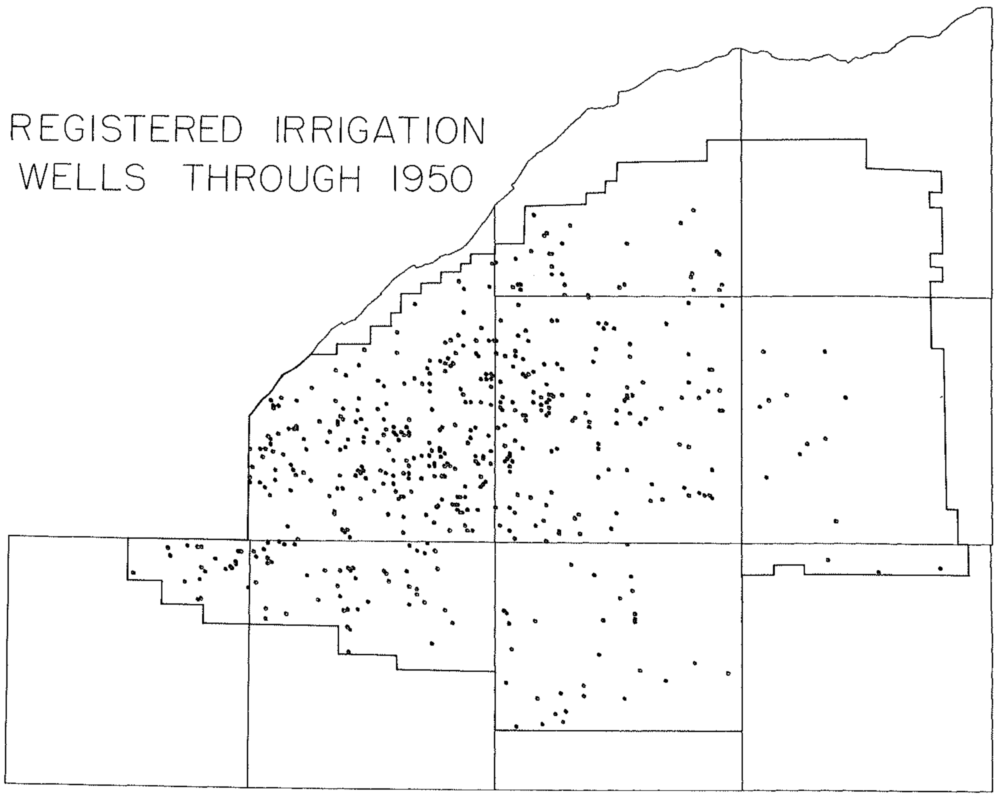



Figure 2b

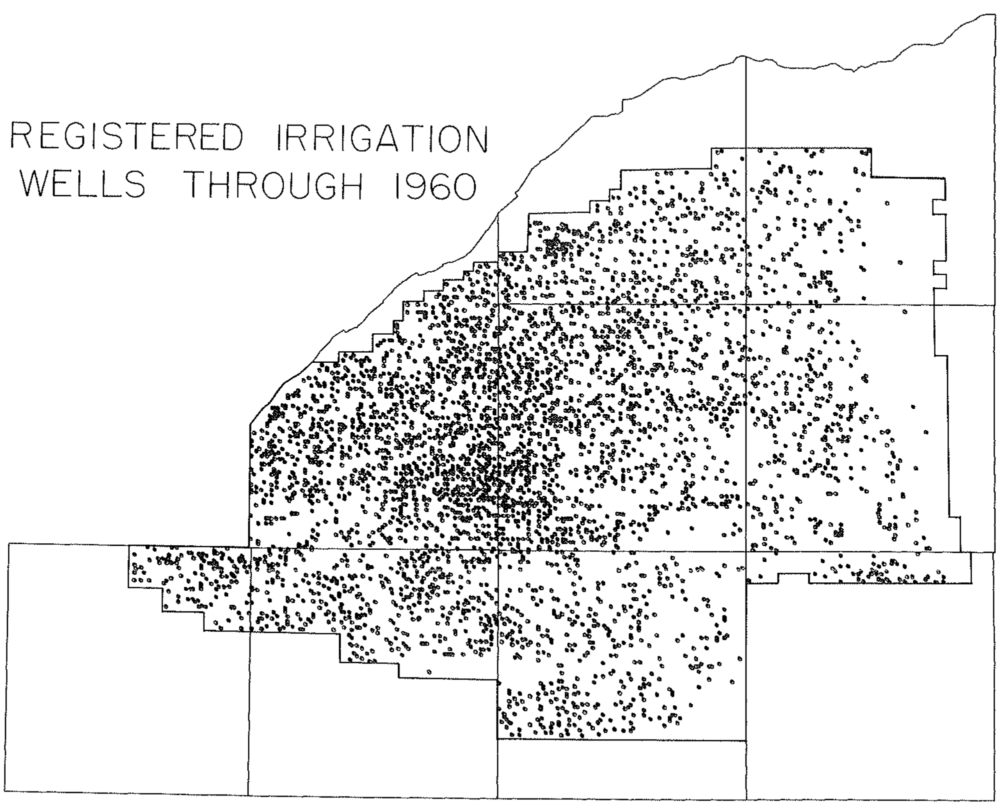



Figure 2c

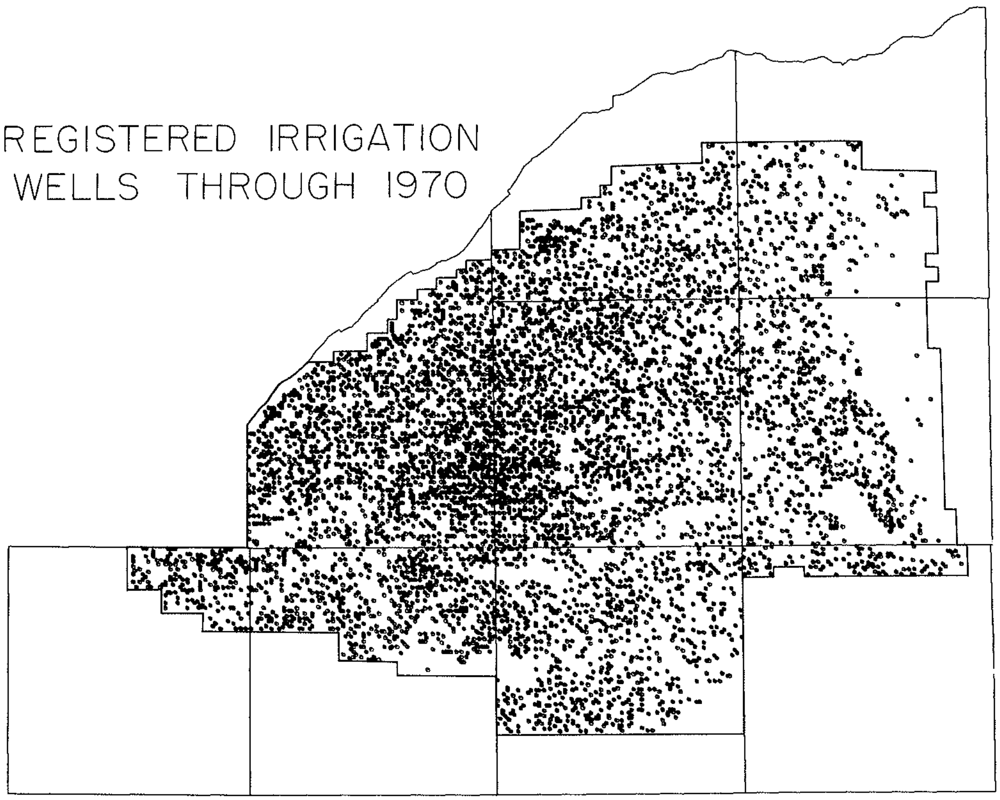



Figure 2d

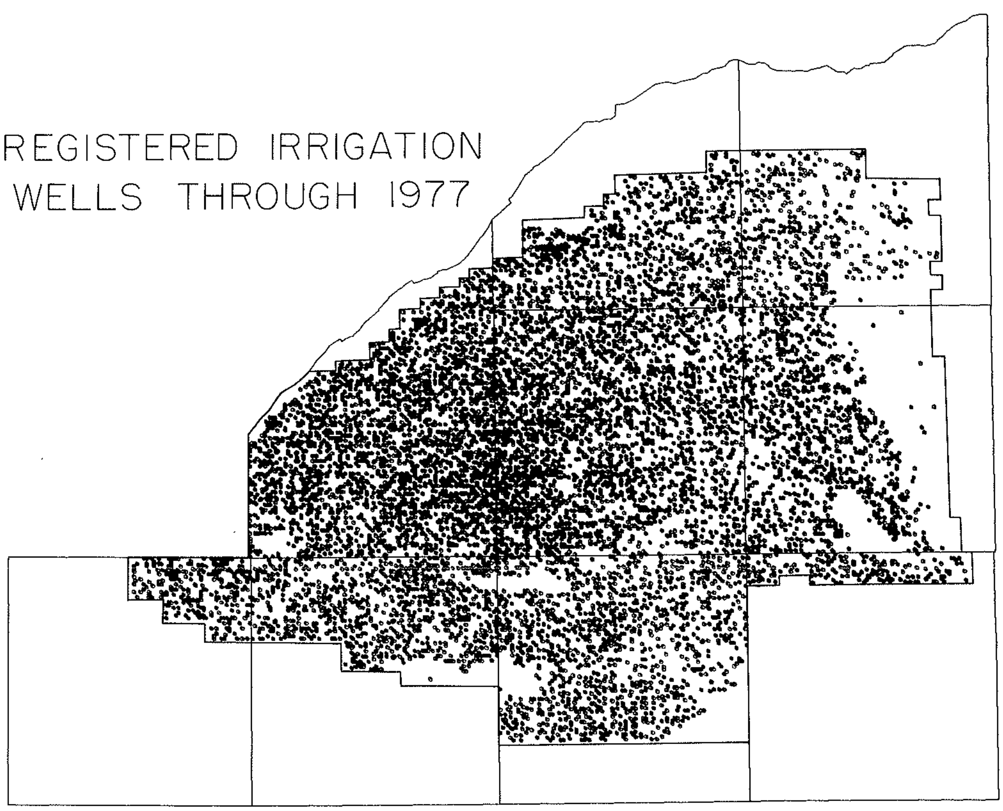



Figure 2e

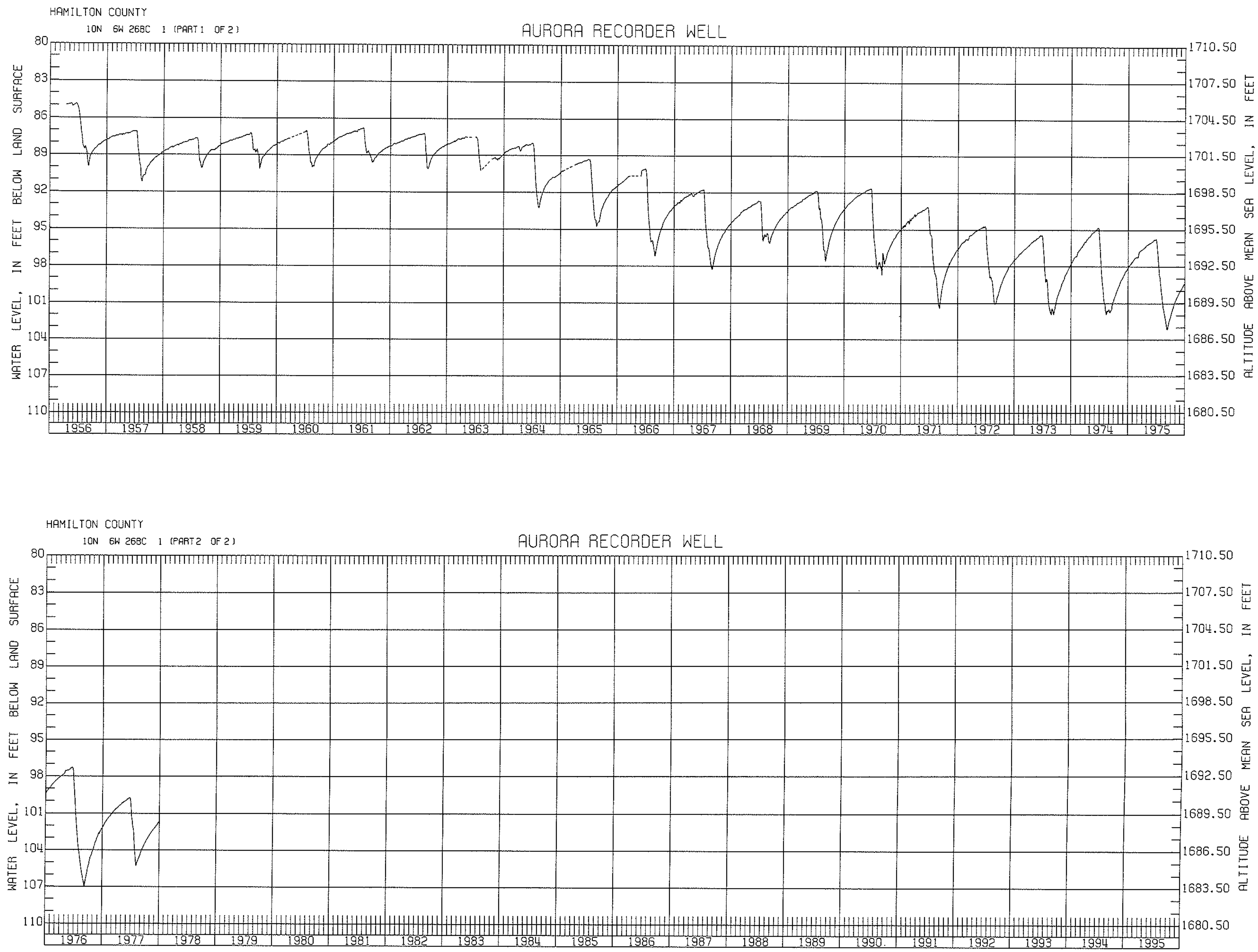



Figure 3a

**Figure 3-Hydrographs for observation wells**

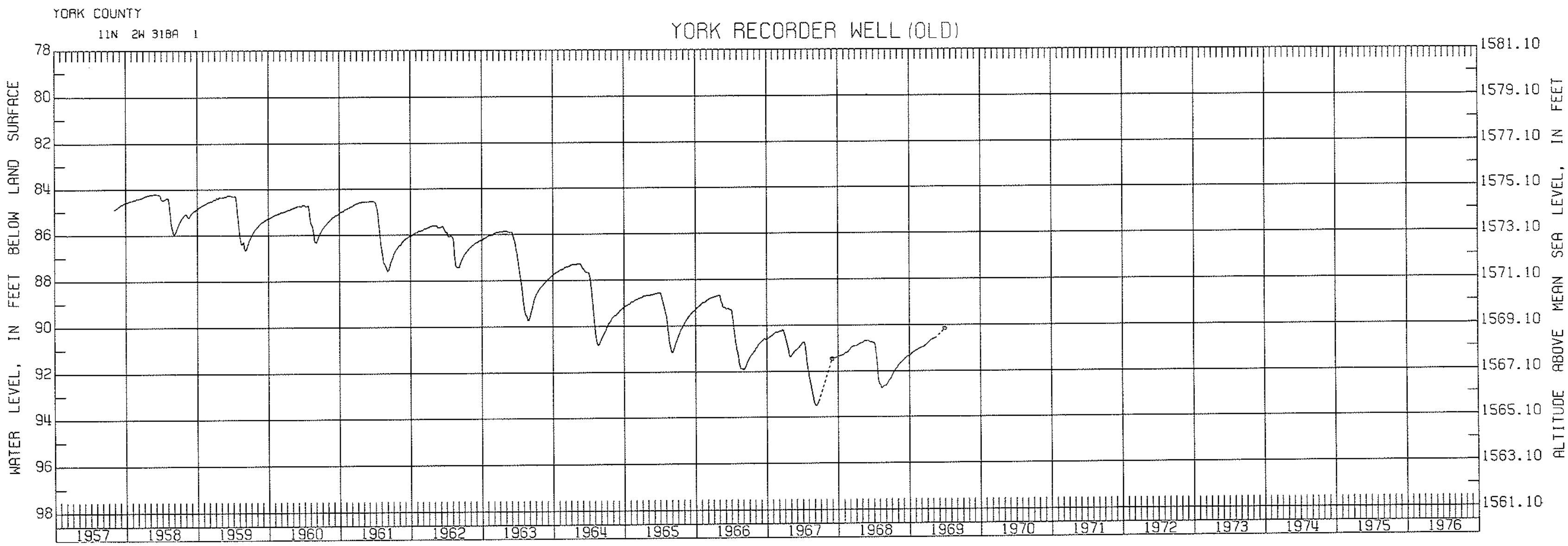



Figure 3b

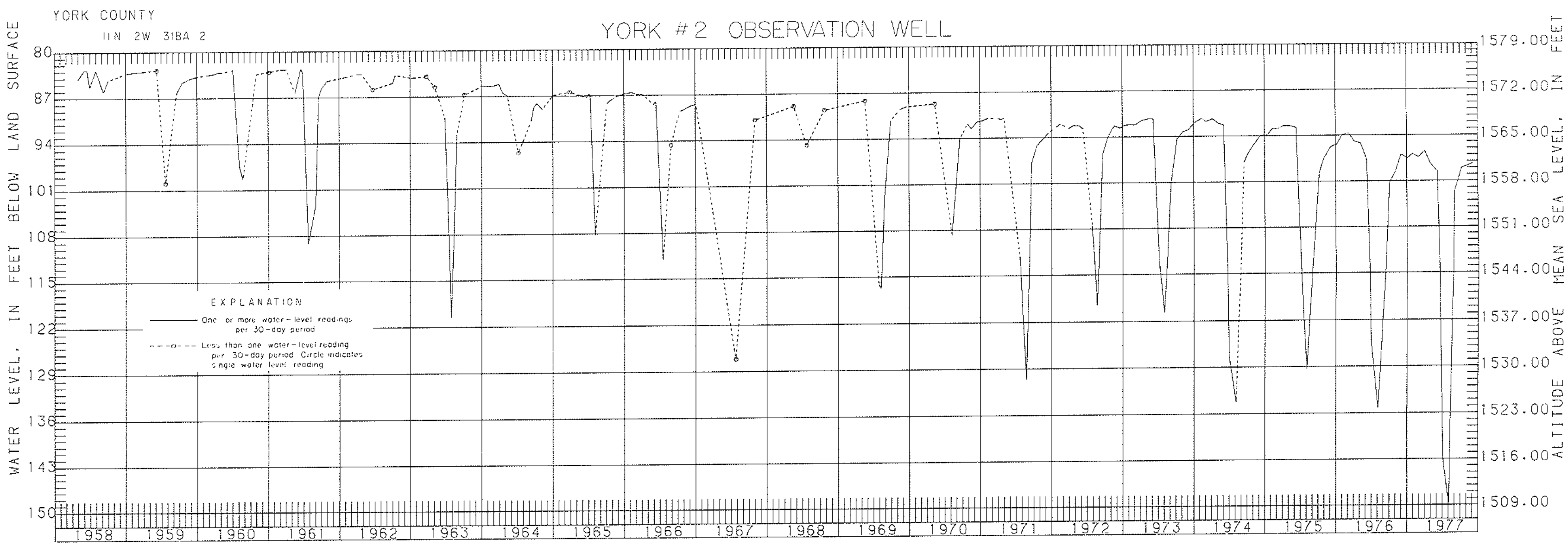



Figure 3c

shows the water level in an unconfined aquifer. Figure 3c shows the water level in a deeper observation well at York which penetrates a confined aquifer.

## SOURCES OF HYDROGEOLOGIC DATA

Data used for portrayals of water-table configuration, base of aquifer configuration and transmissivity of the principal aquifer came from the following sources:

1) Logs of test holes drilled as part of the U.S. Geological Survey - Conservation and Survey Division cooperative studies, (altitudes of water level and base of aquifer; transmissivity estimates based on texture of water-bearing materials);
2) Irrigation well completion reports (altitudes of water level and base of aquifer, transmissivity estimates based on measurements of well yield and water-level drawdown);
3) Logs of test holes drilled for siting irrigation and other privately owned wells, and logs of test holes drilled by the U.S. Corps of Engineers, U.S. Bureau of Reclamation and Nebraska State Highway department (altitudes of water level and base of aquifer);
4) Oil and gas stratigraphic tests (altitudes of base of aquifer);
5) Periodic water level measurements made as part of the U.S. Geological Survey - Conservation and Survey Division cooperative studies.

Test hole and water-level data collected as part of the U.S. Geological Survey - Conservation and Survey Division cooperative program were regarded as primary; all other data were considered to be supplemental.

## GEOLOGY

Bedrock of Cambrian to Cretaceous age and unconsolidated deposits of Tertiary (?) and Quaternary age underlie the NRD. Beginning in the east with the oldest and proceeding westward, five different bedrock units of Cretaceous age are successively in direct contact with the mantling unconsolidated deposits.

The Dakota Group, or oldest of the Cretaceous rock units, was deposited in a near-shore environment. It is composed of shales and sandstones. After this group was deposited, the sea encroached onto the land and during most, if not all, of the remainder of the Cretaceous, sedimentary material was deposited in a deeper-water marine environment. In order of decreasing age the rock units overlying the Dakota Group are the Graneros Shale, Greenhorn Limestone, Carlile Shale, and Niobrara Formation. At the end of Cretaceous time, the sea withdrew. Subsequently the Cretaceous rock units were tilted slightly to the northwest.

Early in Tertiary time, erosion removed much Cretaceous material to form an east-southeastward sloping surface. Deep valleys were cut into the bedrock surface.

In parts of these ancient valleys, Tertiary (?) sediments, mainly silt, overlie the bedrock. Originally these deposits may have filled the ancient valleys and even buried the uplands, but most were removed during a later erosional cycle. Those remaining are generally thickest in the deeper parts of the ancient valleys, and so subdue the relief of the surface on which the first Pleistocene (early Quaternary) sediments were deposited.

During Pleistocene time continental glaciers advanced into the central part of the study area, thereby damming the east-flowing streams. West of the ice margin sand, gravel, silt, and clay were deposited by debris-laden streams. Most of the debris was transported into the area by east-flowing streams that headed in the Rocky Mountains. The remainder was transported by a succession of south-flowing ice marginal streams.

Thick sequences of till (generally unconsolidated, unsorted and unstratified material deposited by and beneath a glacier) are present in the eastern and central parts of the NRD. Blanketing the till and related deposits is a layer of wind-deposited silt (loess). In parts of eastern Seward and south eastern Fillmore counties all or nearly all of the Pleistocene sequence consists of till and loess, there being no or only thin stream-deposited sediments.

The Big Blue River and its tributaries have eroded their valleys into the unconsolidated deposits. In a few places, in southeastern Seward and northeastern Saline counties, cutting has been deep enough to expose the bedrock.

GROUNDWATER RESERVOIR

"An aquifer is a formation, group of formations, or part of a formation that contains sufficient saturated permeable material to yield significant quantities of water to wells and springs." (Lohman et al, 1972).

The principal aquifer system underlying the Upper Big Blue NRD is composed of Pleistocene sands and gravels. Water is stored in and flows through connected intergranular pore spaces. Silts and clays interbedded with the sands and gravels are also considered to be part of the aquifer system because they make a small contribution by gravity drainage to wells screened in a lower sand or gravel.

Not considered to be part of the aquifer system are the thin layers of sand and gravel near the base of the Pleistocene deposits. These are separated from the major aquifer material by thick sequences of silt and clay and are tapped by very few wells. Rock materials underlying the base of the Pleistocene generally yield water in such small amounts or of such poor quality that they were not considered hydrologically important for this study.

Most of the NRD is in the eastern part of the South-Central Plains Region, which is bounded on the north by the Platte River Valley Region, on the east and southeast by the Southeast Nebraska Glacial Drift Region, and on the southwest and west by the Republican River and Dissected Plains Region (Reed, 1969). The principal aquifer beneath the NRD is hydraulically continuous with the aquifer beneath the NRDs bordering it on all sides except the east. Toward the east loess mantles till, which in turn rests on bedrock, and aquifer continuity is maintained only through sands and gravels in narrow eastward extensions of buried paleovalleys; thus the regional aquifer underlying some or most of the other NRDs to the west is virtually absent. The Ogallala Formation, a major water-bearing rock unit in much of central and western Nebraska, extends into the westernmost part of the Upper Big Blue NRD. It rests on the surface of the Cretaceous rocks and nowhere within the NRD is more than a few feet thick. It is hydrologically significant, however, in that part of the subsurface inflow from the west comes out of the thicker Ogallala beneath the NRD to the west.

The base of the principal aquifer is here defined as the base of the lowermost major Pleistocene sand and gravel. It is lower in areas of bedrock valleys and higher in regions of valley walls and bedrock highs (fig. 4).

In order to better understand the hydrologic system we will define and discuss "water table" and "piezometric surface." Figure 5 is a map showing the configuration of the surface defined by the water level in wells (average piezometric surface) tapping the principal aquifer beneath the Upper Big Blue NRD in the spring of 1975.

The upper boundary of an aquifer is either the water table or the bottom surface of a confining layer. Where it is a water table, the aquifer is said to be unconfined or nonartesian, and the static water level in wells may or may not coincide with the level of the water table. The water level in those wells open to the uppermost part of an unconfined aquifer coincides with the water table. The water level in wells open

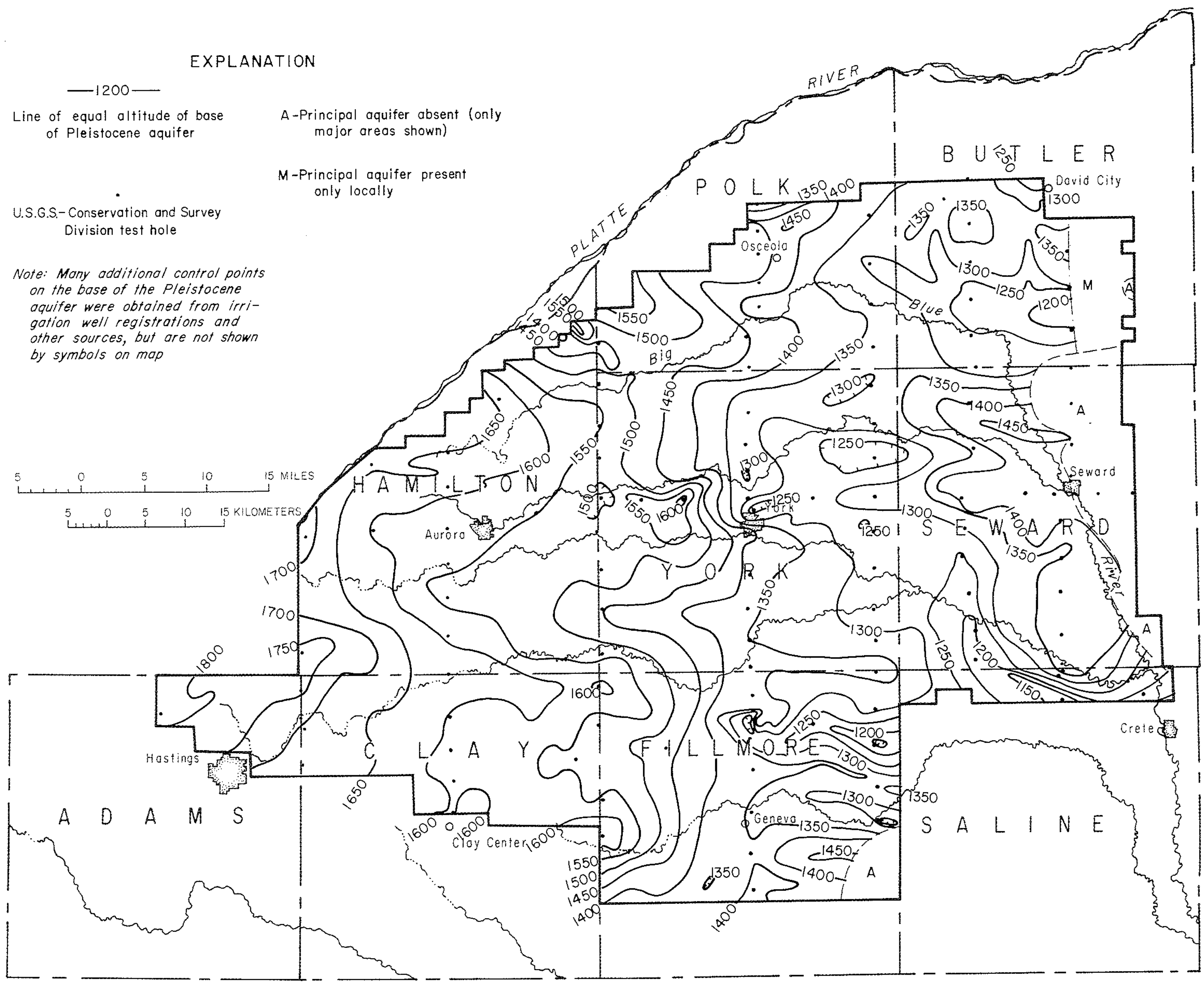



Figure 4-Base of Pleistocene aquifer

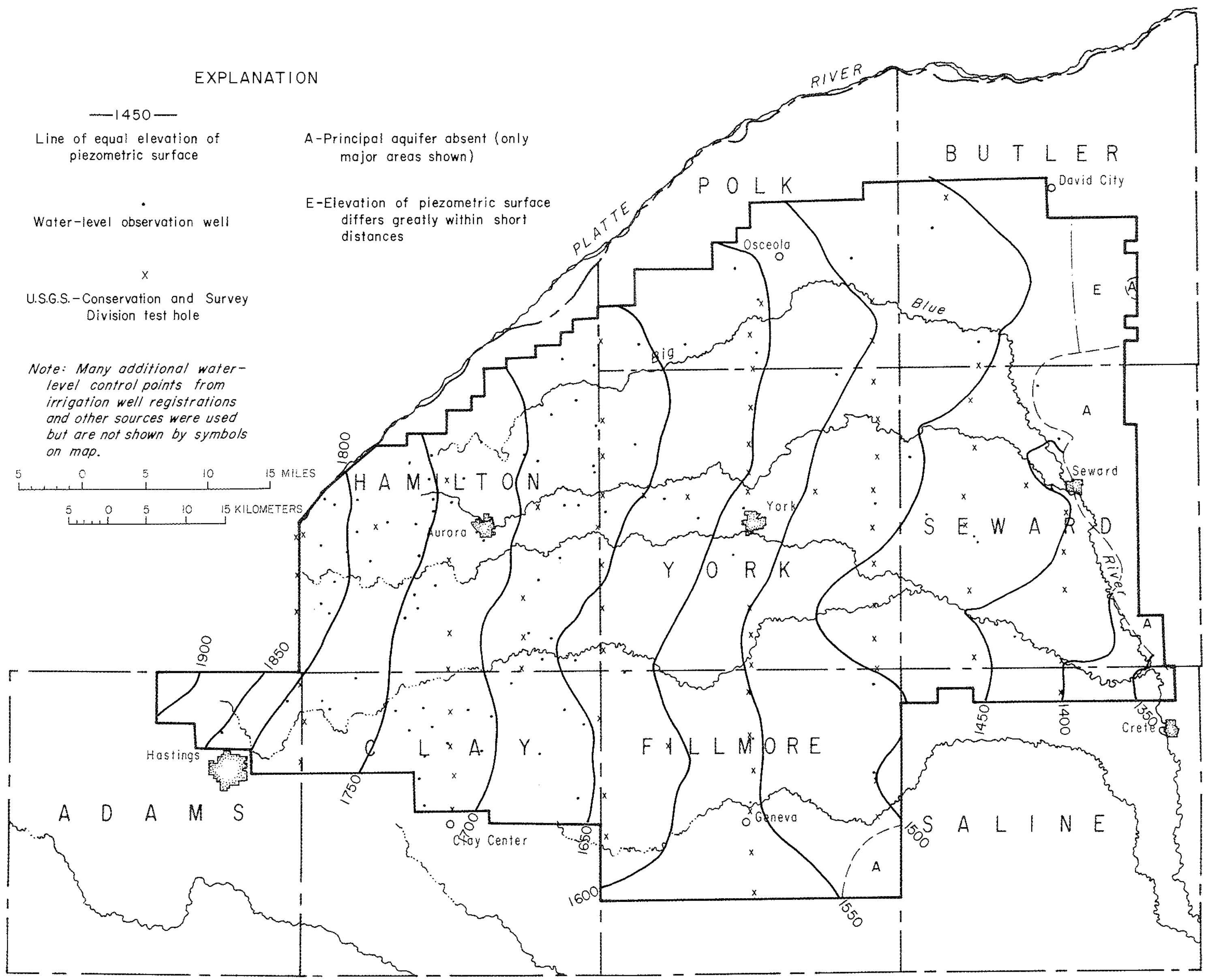



Figure 5-Average piezometric surface, Spring 1975

to lower parts of the aquifer may coincide or be somewhat higher or lower than the water table.

Where the upper boundary is a confining layer, no water table exists and the aquifer is said to be confined, or artesian. The static water level in wells tapping confined aquifers generally is somewhat higher than the top of the aquifer. The water level in these wells coincides with and is defined as the piezometric, or pressure, surface of the aquifer.

Since both unconfined and confined conditions exist, and the water level in most wells tapping the aquifer where it is unconfined reflects the pressure level deep in the aquifer, the surface depicted is referred to as the average (of vertical piezometric levels) piezometric surface. That terminology is used consistently throughout the report, even though in much of the area the water table and the surface depicted are virtually the same.

No water-level configuration is shown for the eastern part of Seward County and the southeastern part of Fillmore County, because the principal aquifer does not extend into those areas.

The relationship between groundwater and streams in the Upper Big Blue NRD is variable and dependent upon the local geologic conditions. Even the degree of hydraulic connection with different stream reaches may not be the same. The Platte River is in relatively good connection with the regional groundwater aquifer except north of Butler County. The Big Blue River is less well hydraulically connected and its North Branch loses connection with the aquifer a short distance west of Butler

County. In the western part of the NRD smaller streams such as Beaver Creek and Lincoln Creek are not in connection with the regional aquifer.

Another map (fig. 6) shows the configuration of the average piezometric surface before extensive development of the groundwater system. This predevelopment map represents conditions that prevailed in the early 1950s.

Figure 7 portrays the vertical distance between the piezometric surface and the base of the major aquifer in Spring 1975. Where the piezometric surface coincides with the water table, figure 7 is a saturated thickness map. Water levels are highest in the spring because they have recovered to their greatest extent from pumping during the previous irrigation season. The shaded parts of the map indicate areas where the average piezometric surface is above the top of the saturated zone. Comparison of this map with the map showing the base of the aquifer (fig. 4) shows that the aquifer is thickest in bedrock valleys and is thinner or lacking where it overlies bedrock highs. Where the Pleistocene sequence consists mostly of till or stratified silt and clay, aquifers occur mainly as discrete channel fills that were deposited by small streams.

Figure 8a, 8b is a map showing the transmissivity of the principal aquifer under conditions of the mid 1960s. "Transmissivity is the rate at which water of the prevailing kinematic viscosity is transmitted through a unit width of the aquifer under a unit hydraulic gradient . . . . It is equal to an integration of the hydraulic conductivity across the saturated part of the aquifer, perpendicular to flow paths." (Lohman et al, 1972).

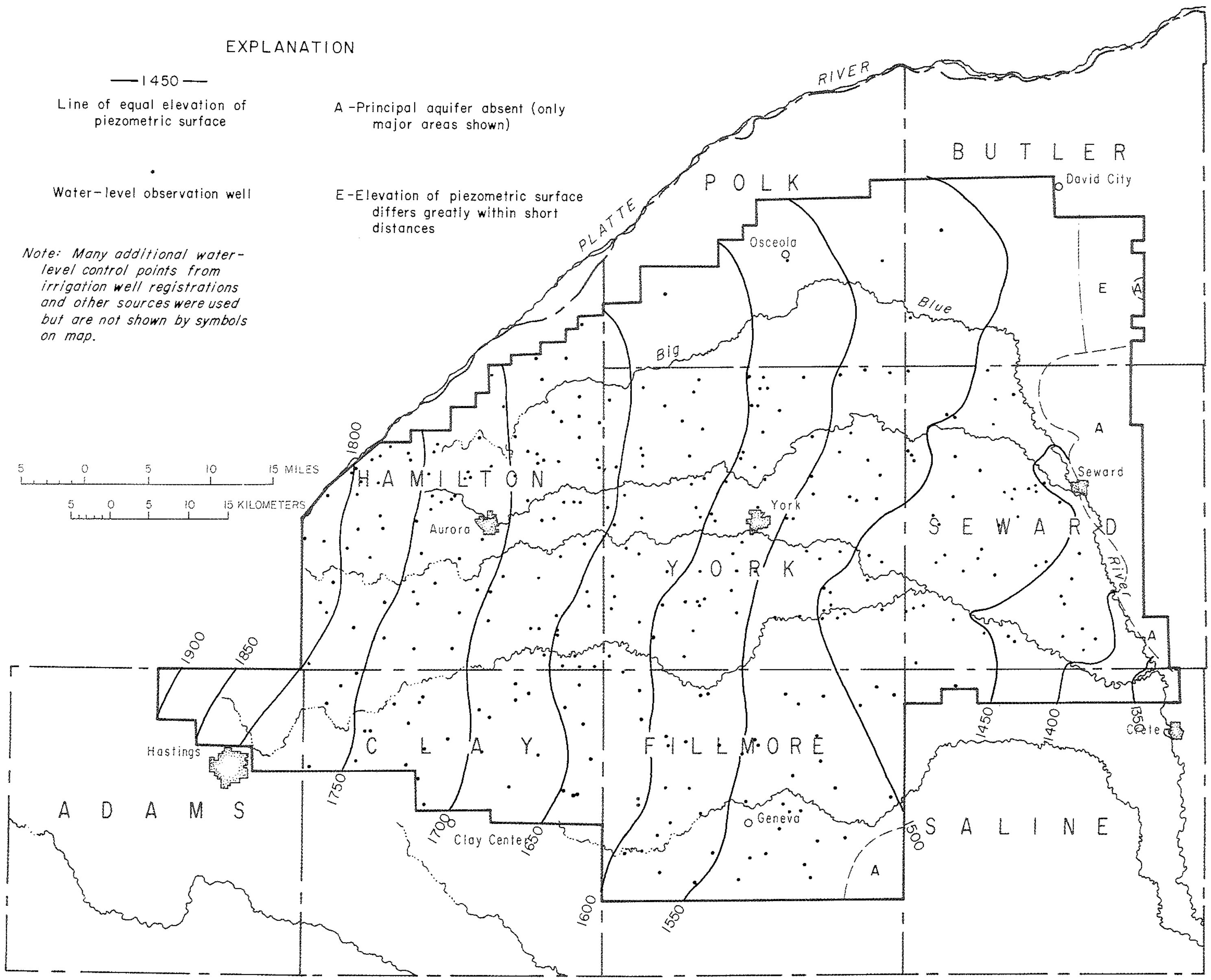



Figure 6-Predevelopment average piezometric surface

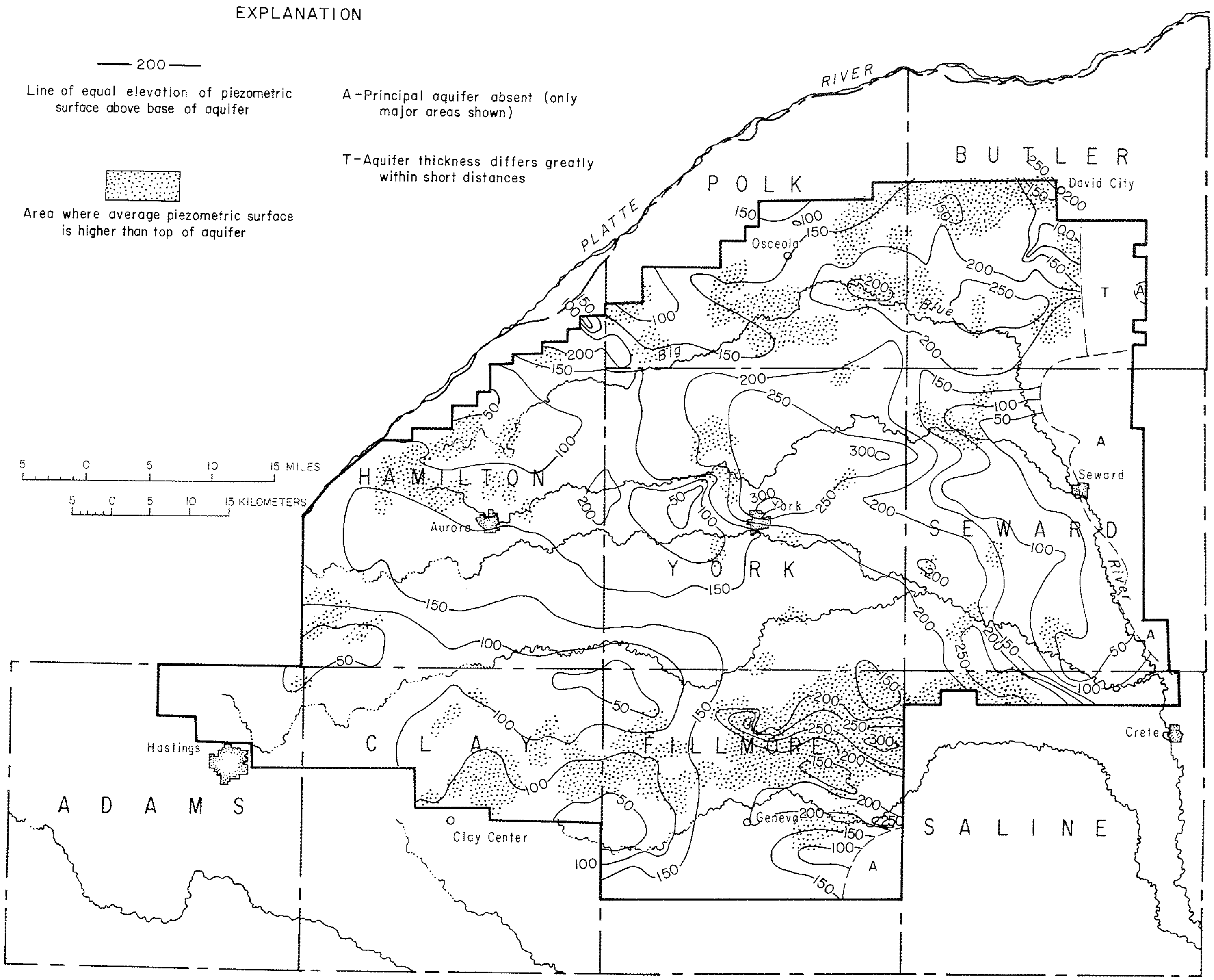



Figure 7-Elevation of piezometric surface above base of Pleistocene aquifer, Spring 1975

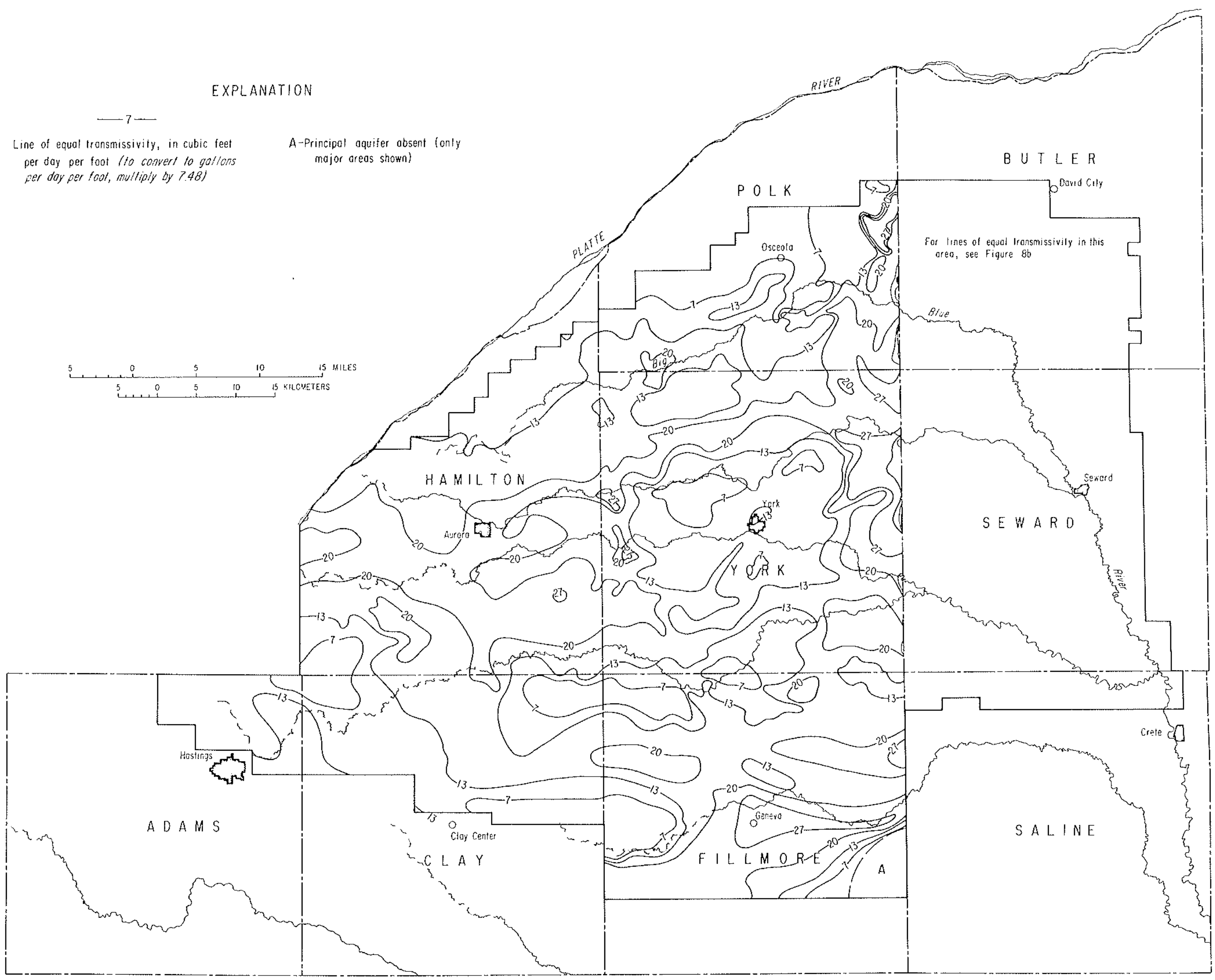



Figure 8a-Transmissivity of aquifer in the middle 1960's, exclusive of Butler, Seward and Saline counties

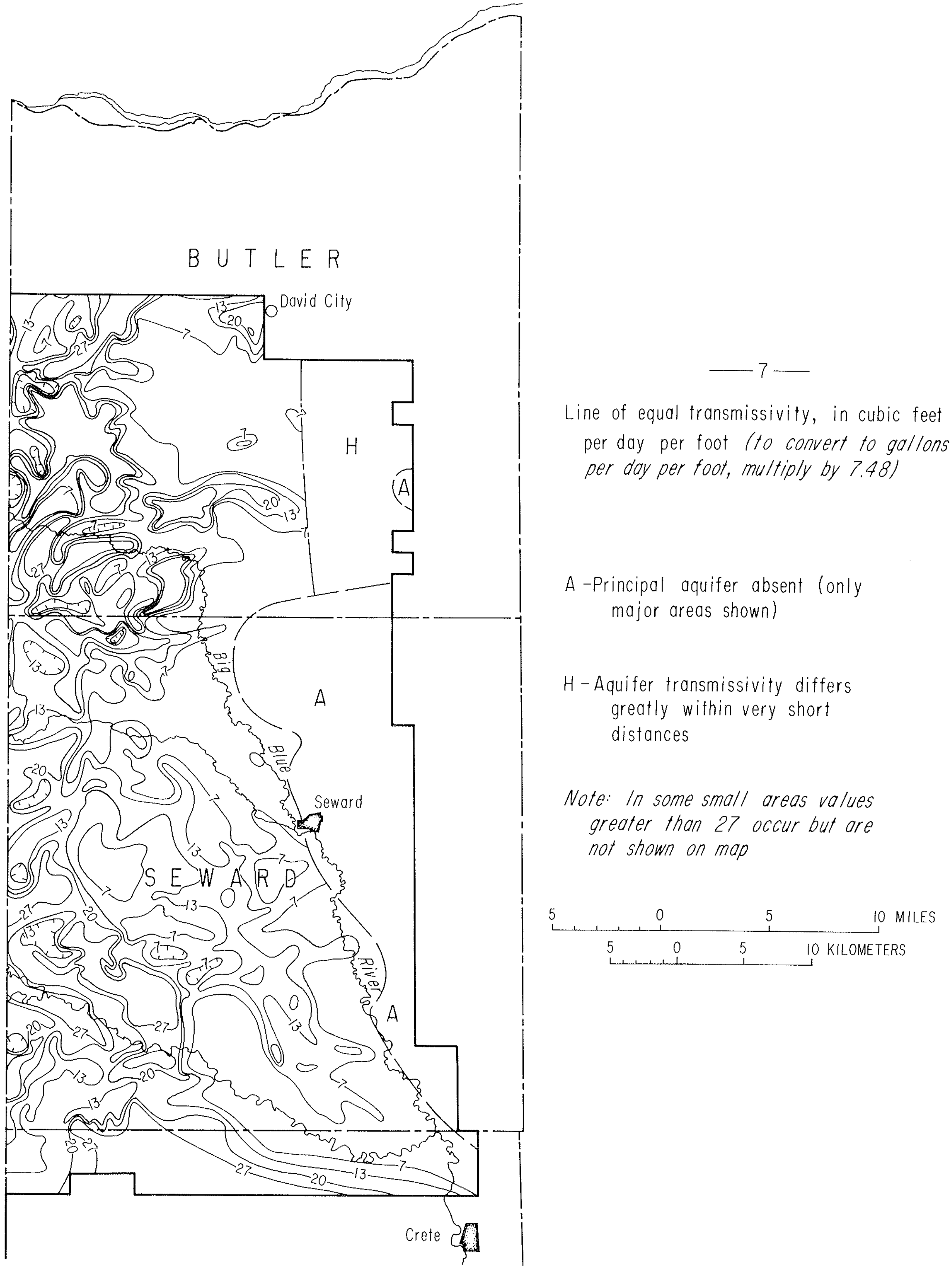



Figure 8b-Transmissivity of aquifer underlying Butler, Seward and Saline counties in middle 1960's

Transmissivity of sands and gravels is greater than that of silts and clays and the transmissivity of a thick sequence of saturated material is greater than that of a thinner sequence. Therefore, at a given site transmissivity decreases as saturated thickness decreases due to declining water levels.

Transmissivity values in this report were obtained by either of two methods. The first method utilized logs and samples of wells. A value of hydrologic conductivity was assigned to each unit of water-bearing material, multiplying those values by the thickness of the respective unit, and then summing the products. The value for hydraulic conductivity was based on particle size range, degree of sorting and silt content. The second method (after Logan, 1964) consisted of multiplying the specific capacity (reported yield in gallons per minute divided by drawdown of the water level) of a pumping well by 2000.

Comparison of figure 8a, 8b with the map showing the configuration of the bedrock surface shows that transmissivity tends to be somewhat higher in areas of bedrock valleys, where thick sequences of sand and gravel often occur. That relation is not direct because in some areas a thick silt or clay within the bedrock valley lowers the transmissivity. Areas of very low transmissivity are also related to high bedrock, where aquifers are thin or absent as in west-central Fillmore County, and to thick sequences of till as in eastern Butler County. Very low transmissivity characterizes much of the area labeled "Principal aquifer absent."

## HYDROLOGY

The complex nature of the geology in the Upper Big Blue River basin is the principal cause of the extreme variations of aquifer response to the stress resulting from pumping from wells.

A comparison of the hydrographs for the observations wells at York (fig. 3b, 3c) clearly shows substantial differences even though the wells are in the same location. The well represented by hydrograph 3c is screened at a greater depth than is the well represented by hydrograph 3b. The former shows conditions in the confined (lower) aquifer, the latter shows conditions in the unconfined (upper) aquifer.

Another example of the different water-level responses in wells that are completed in unconfined aquifers is shown by the hydrographs for the Shickley recorder well in Fillmore County and for the Aurora recorder well in Hamilton County. The Shickley well is completed in an aquifer which lies beneath an extensive silt layer overlain by sand and gravel. Recharge from precipitation and return flow from irrigation perches on the silt layer and some moves laterally rather than infiltrating to the lower aquifer. In the case of the Aurora well, infiltration of water from the land surface to the major aquifer in the vicinity of the well is through very fine material and is therefore quite slow. Since there is no sand or gravel overlying the fine-textured material, there is little lateral movement of infiltrating water at the site. Given enough time, most of the water drains to the principal aquifer.

## GROUNDWATER SIMULATION

The concern for efficient water-resources management has led to the development and application of numerous means for simulation of hydrologic systems and groundwater flow and the impact of current and potential stresses on those systems. The foundation for most of these methods is the hydrologic budget concept (fig. 9). A hydrologic budget accounts for water in much the same way that a bank book accounts for the money in a savings account. All of the gains to and losses from the system must be identified and determined in order for the budget to balance properly. Numerical simulations often consist of budget equations for specific locations or areas within the total area simulated. Equations describing the nature of groundwater flow are also incorporated into the budget equations in order to describe groundwater movement from one location to an adjacent location. The amount of groundwater at a particular site is related to the piezometric surface, which represents the elevation of the water table in an unconfined aquifer or the water pressure in a confined aquifer. Groundwater moves in response to the slope of the piezometric surface as long as the aquifer material contains conduits through which water may flow. The solution of the equations is the piezometric surface within the system.

Inputs to the simulation include values representing net discharge. Net discharge is the total of well discharges, recharge from precipitation, and irrigation return flow. Two discharge

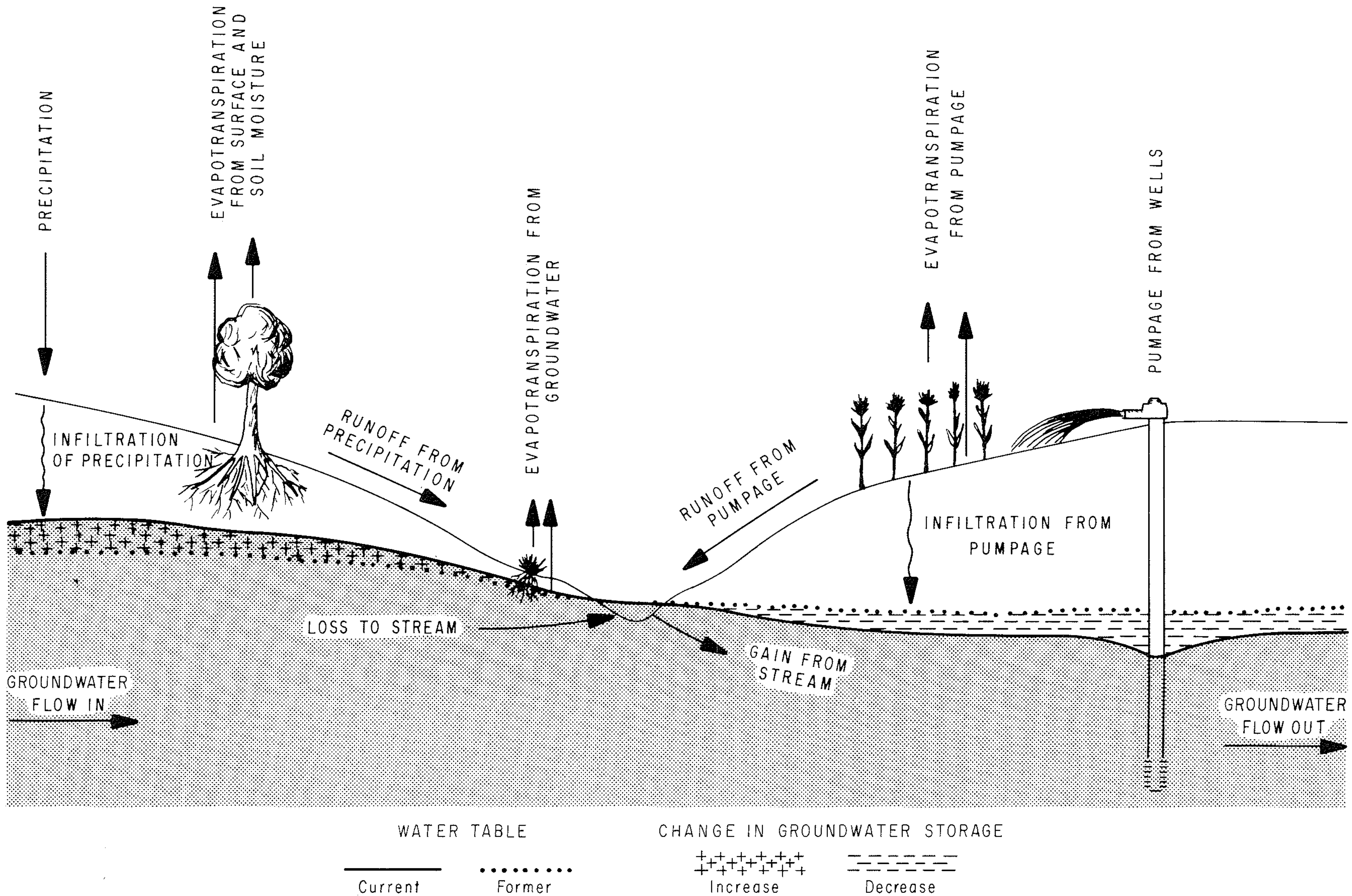



Figure 9-Hydrologic budget

periods were simulated for each year; an irrigation season (July and August) and a recovery season (September-June). Additional inputs are the hydrogeologic parameters (transmissivity, storage coefficient, stream-aquifer interconnection) and the water level at the start of the simulation period. Simulated stream baseflows are computed based upon water-level values throughout the simulation as aquifer-stream gains and losses are estimated. Since well yields are reduced as the water levels decline, a routine to estimate and simulate the reduction of well discharge due to declining water levels has been incorporated into the simulation program.

## DETERMINATION OF HISTORIC DISCHARGE, RECHARGE AND RETURN FLOW

Irrigation well registration records served as one of the sources of data necessary to simulate the historic development of groundwater irrigation within the NRD. The registered data include installation date, location, well capacity, irrigated acreage, static water level, and pumping water level. The dates and locations provide a history of water-use development throughout the NRD. Data indicating the quantity of water pumped by a well during a typical irrigation season are not noted on the well registration.

The data used to approximate the use of groundwater for irrigation within the NRD were obtained from two sources. J. M. Jess (1970) collected data for Seward County in 1968. Systematic collection of pumpage data within the NRD by Eugene K. Steele, Jr. of the U.S. Geological Survey provided data for Hamilton and York counties in 1969 through 1975 (Steele, 1971a and b), Seward County in 1971 through 1975 (Steele, 1973), Clay County in 1970 through 1975 (Steele, 1972), and Fillmore County in 1974 and 1975. The data that is not referenced is available as unpublished reports from the U.S. Geological Survey.

A rather small percentage of the total number of irrigation wells within each county was involved in the inventory since the number of wells in these counties is very large. The pumpage data include measured well discharges, irrigated acreages, and number of hours that each well was pumped for the season. Monthly precipitation data for the area were used in conjunction with the pumpage data in order to study relations between the quantities of water pumped for irrigation and the precipitation during the irrigation season. An empirical formula relating the number of hours that a well was pumped during an irrigation season with the June, July and August precipitation was developed. Although the correlation between the hours pumped and precipitation is not very good it was the best correlation of the relations tested.

Figure 10 is a graph of the well discharges measured by Steele and Jess versus the well discharge noted on the irrigation well registration form. If the measured discharge corresponded with the registered discharge, the data would plot as a 45 degree line passing through the origin. The graph indicates that there is no exact relationship between the measured discharge and the registered discharge. Therefore, a statistically based empirical relationship was derived in order to estimate realistic well discharges from the registered well discharges.

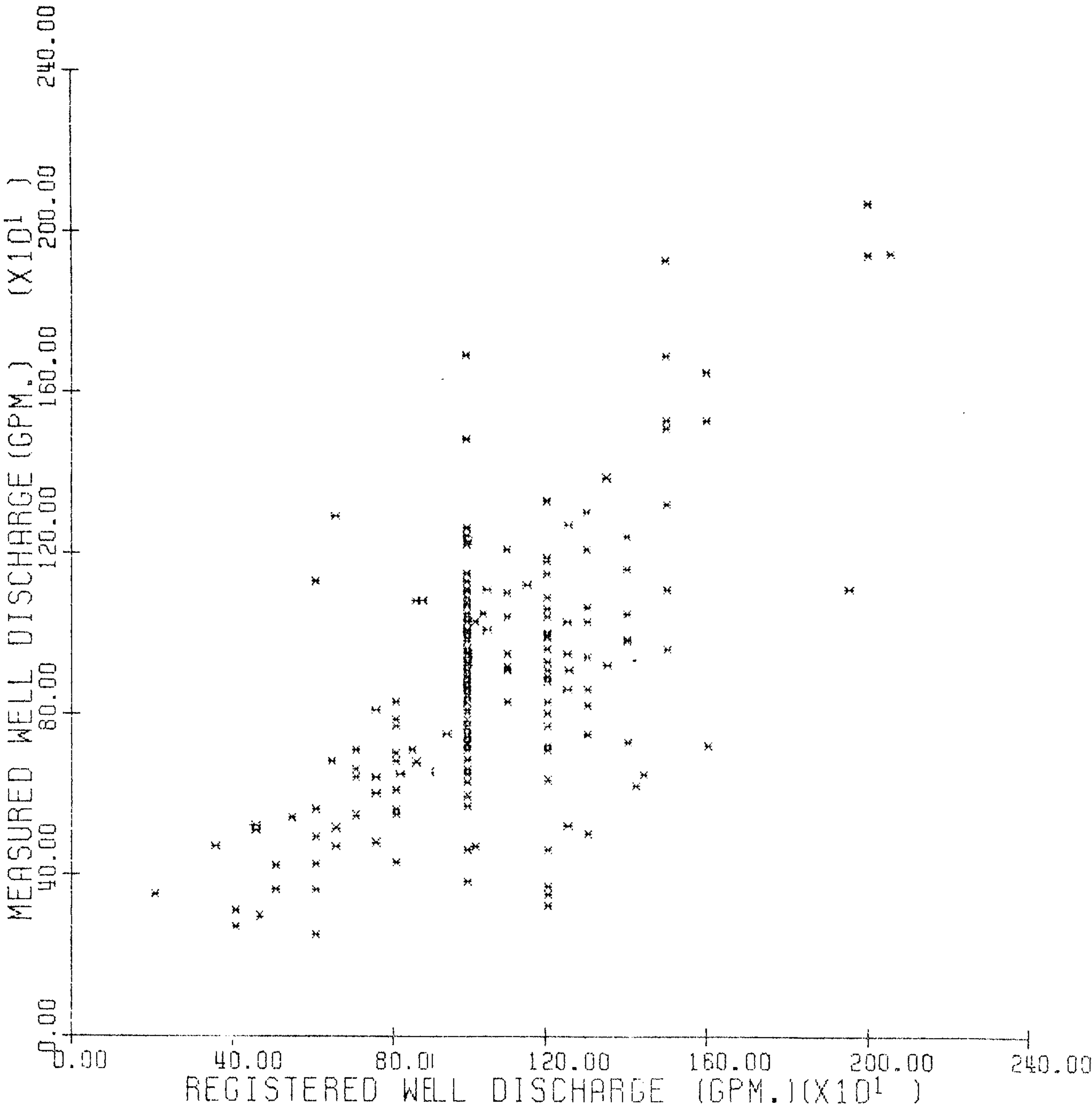



Figure 10--Registered vs measured discharge of irrigation wells.

Figure 11 is a graph of irrigated acreage determined by Jess and Steele plotted with respect to the irrigated acreage reported on the well registration. The scatter of the data indicates a poor statistical correlation between the measured irrigated acreage and the registered irrigated acreage. For this reason, an average irrigated acreage (88.7 acres) was assumed for each irrigation well.

Monthly precipitation data were utilized in order to estimate the duration of irrigation, natural recharge, and irrigation return flow. Data were furnished by the Nebraska Natural Resources Commission Natural Resources Data Bank and supplemented with data from the climatology office of the Conservation and Survey Division.

Natural recharge is the amount of water which infiltrates to the groundwater system from precipitation. Irrigation return flow is water which was applied to the land surface and eventually returns to the groundwater system by infiltration. Both of these quantities must be estimated since they are components of the groundwater system hydrologic budget but are not measured quantities. They are estimated by starting with either precipitation or applied irrigation water and deducting the components that are lost before reaching the groundwater system. Surface runoff and evapotranspiration are deducted from the precipitation to yield the potential natural recharge.

The estimated potential irrigation return flow is essentially that amount of applied water which is not evaporated or used by crops. However, depending upon climatic variations and the

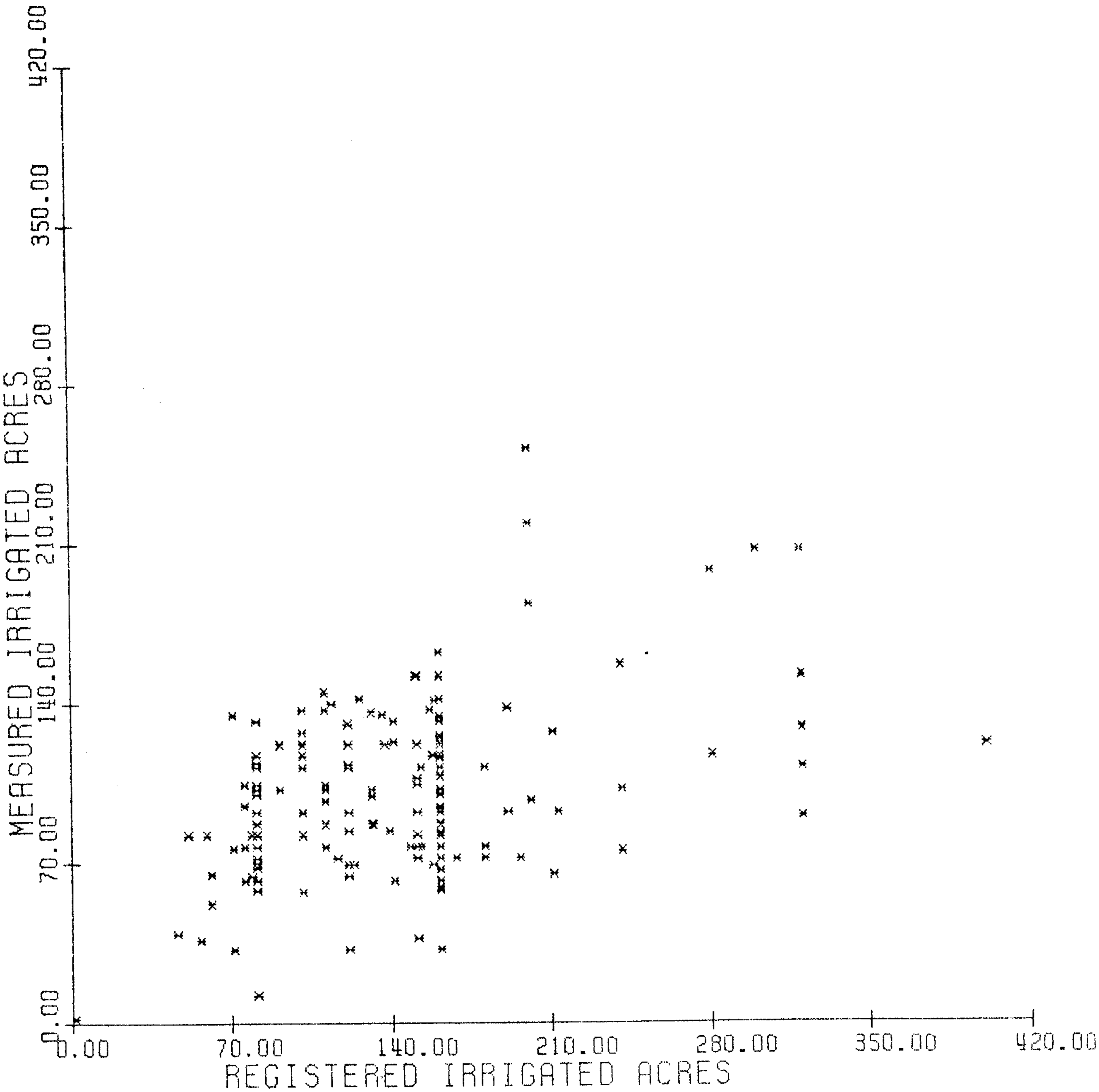
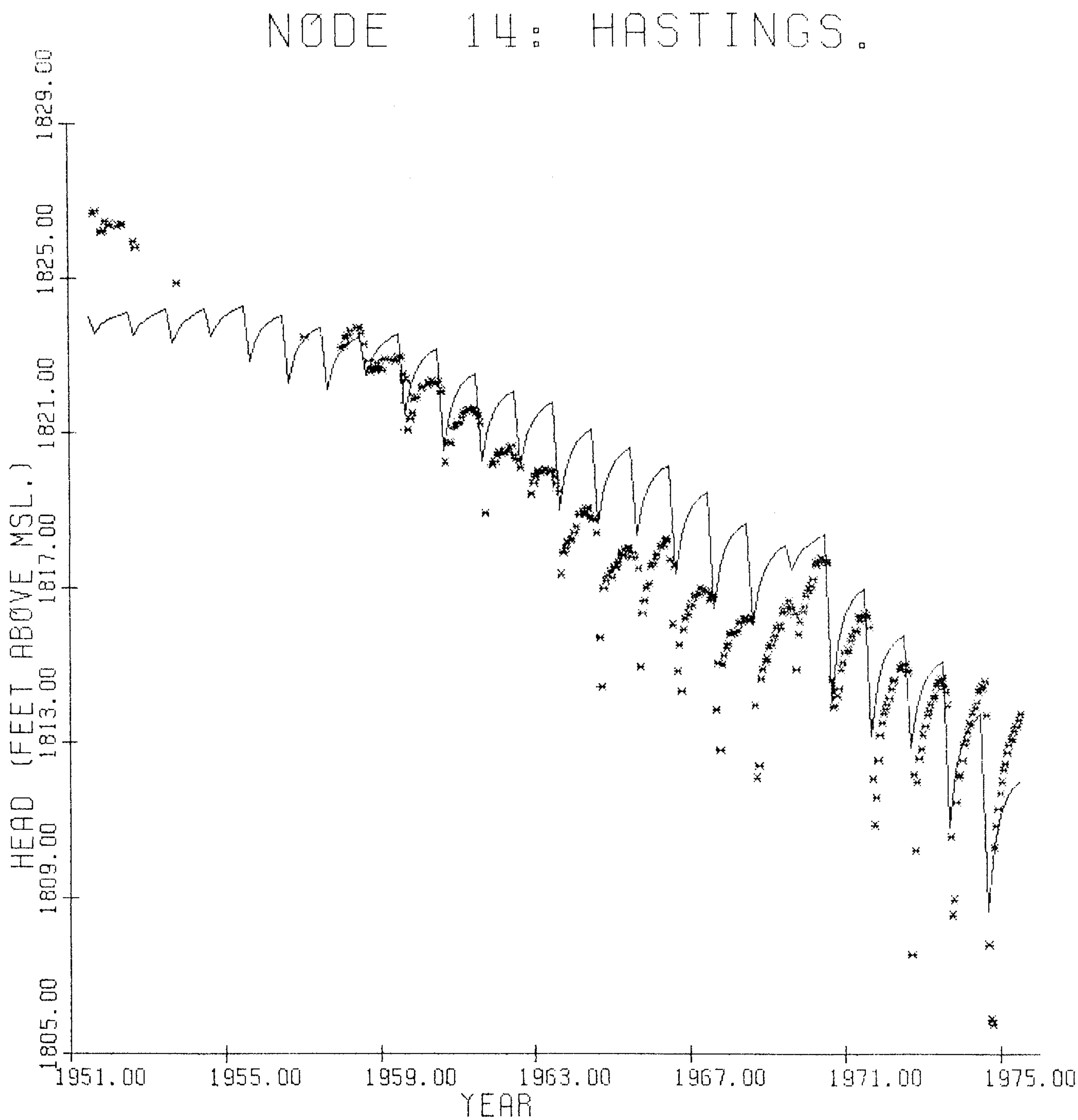



Figure 11--Registered vs measured acreage per irrigation well.

nature of the geology, some of the estimated potential natural recharge and the estimated potential irrigation return flow may not reach the groundwater system. Evapotranspiration and runoff values were estimated by methods detailed by E. G. Lappala (1978).

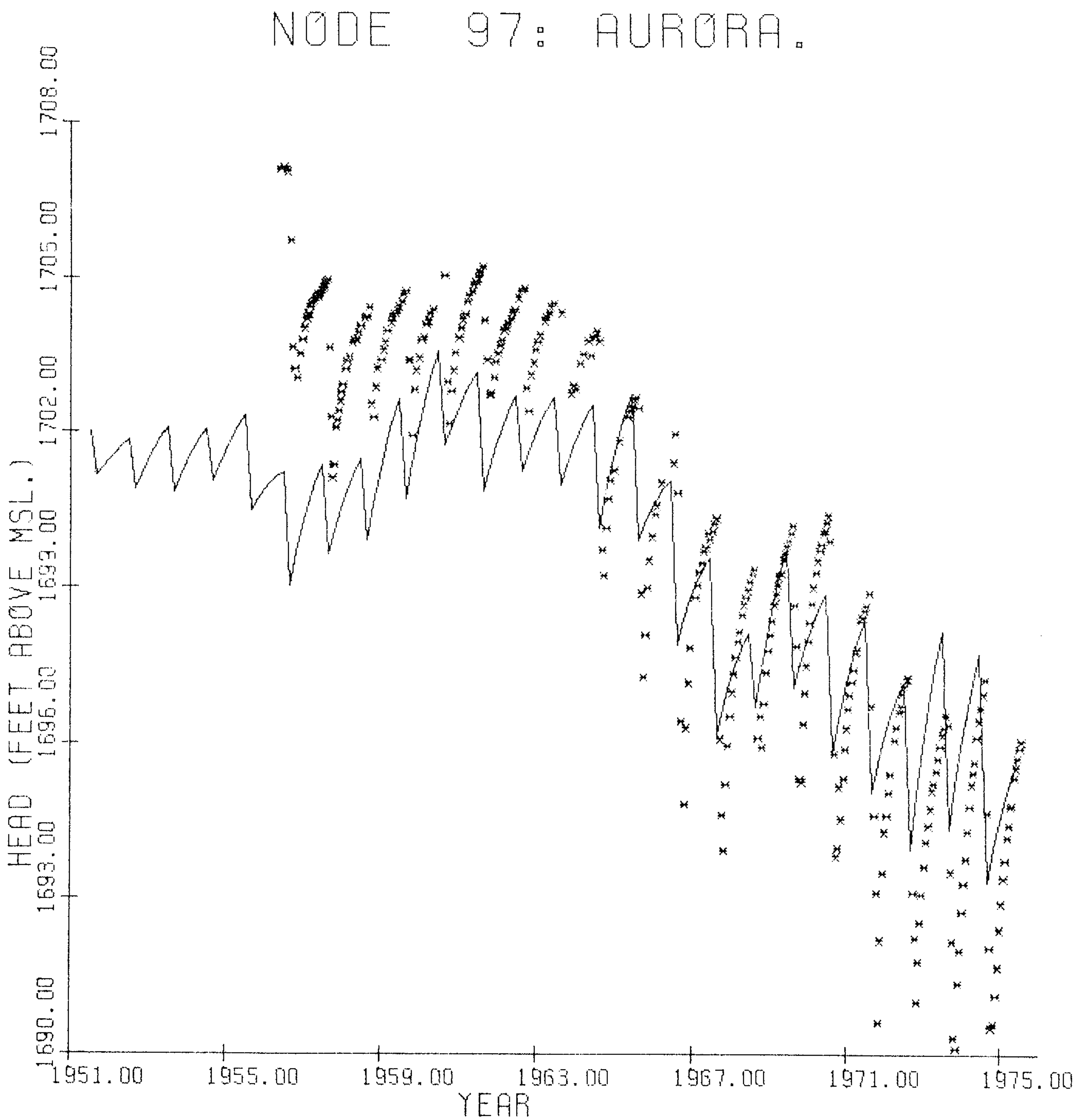
## MODEL CALIBRATION AND VERIFICATION

Simulations of the changes that occurred in the water level between Spring 1951 and Spring 1975 were the basis for calibration of the model through adjustment of the natural recharge and irrigation return flow, storage coefficient and stream-aquifer interconnection. Comparisons of calculated values versus observed well hydrographs (fig. 12a-12f), plus calculated water levels Spring 1975 (fig. 13) versus observed water levels Spring 1975 (fig. 14), indicated areas where parameters must be varied in order to approximate more closely the historical response of the groundwater system. The principal parameter that was adjusted was the proportionality factor relating estimated recharge from precipitation plus irrigation return flow with computed maximum recharge from precipitation plus irrigation return flow. This parameter has some physical basis since in some areas recharge-return flow is intercepted prior to reaching the major aquifer. The short-term storage coefficient values were adjusted where necessary in order to approximate drawdown of water levels during the irrigation seasons. The long-term specific yield was kept constant over the simulated area. The value simulated (0.25) represents a mean value for the typical aquifer material within the NRD. The parameter necessary for determining the time dependency of the delayed gravity response term was also kept constant



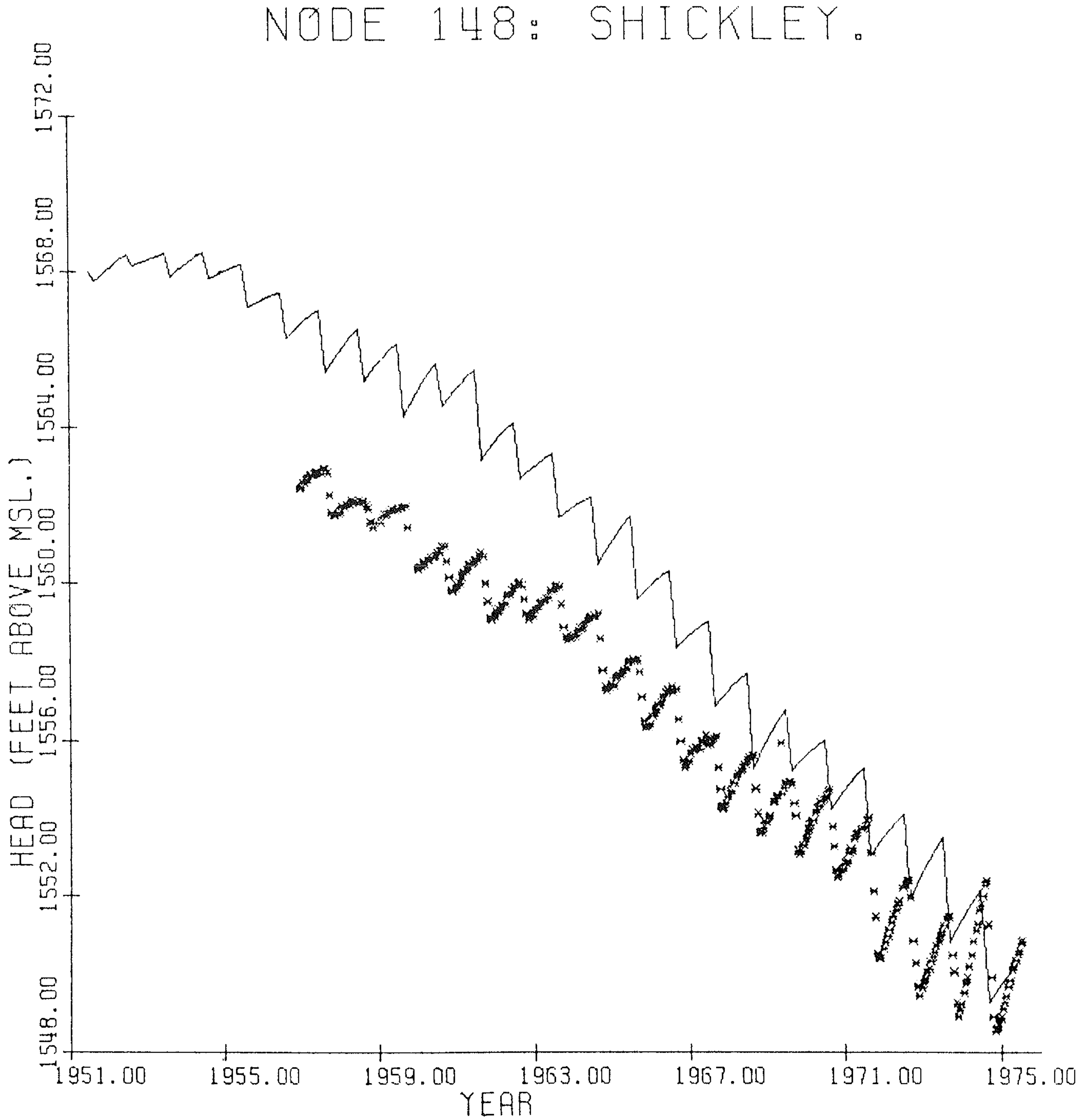

- Simulated values
* Observed values

Figure 12a.--Calculated and observed water levels at Hastings.



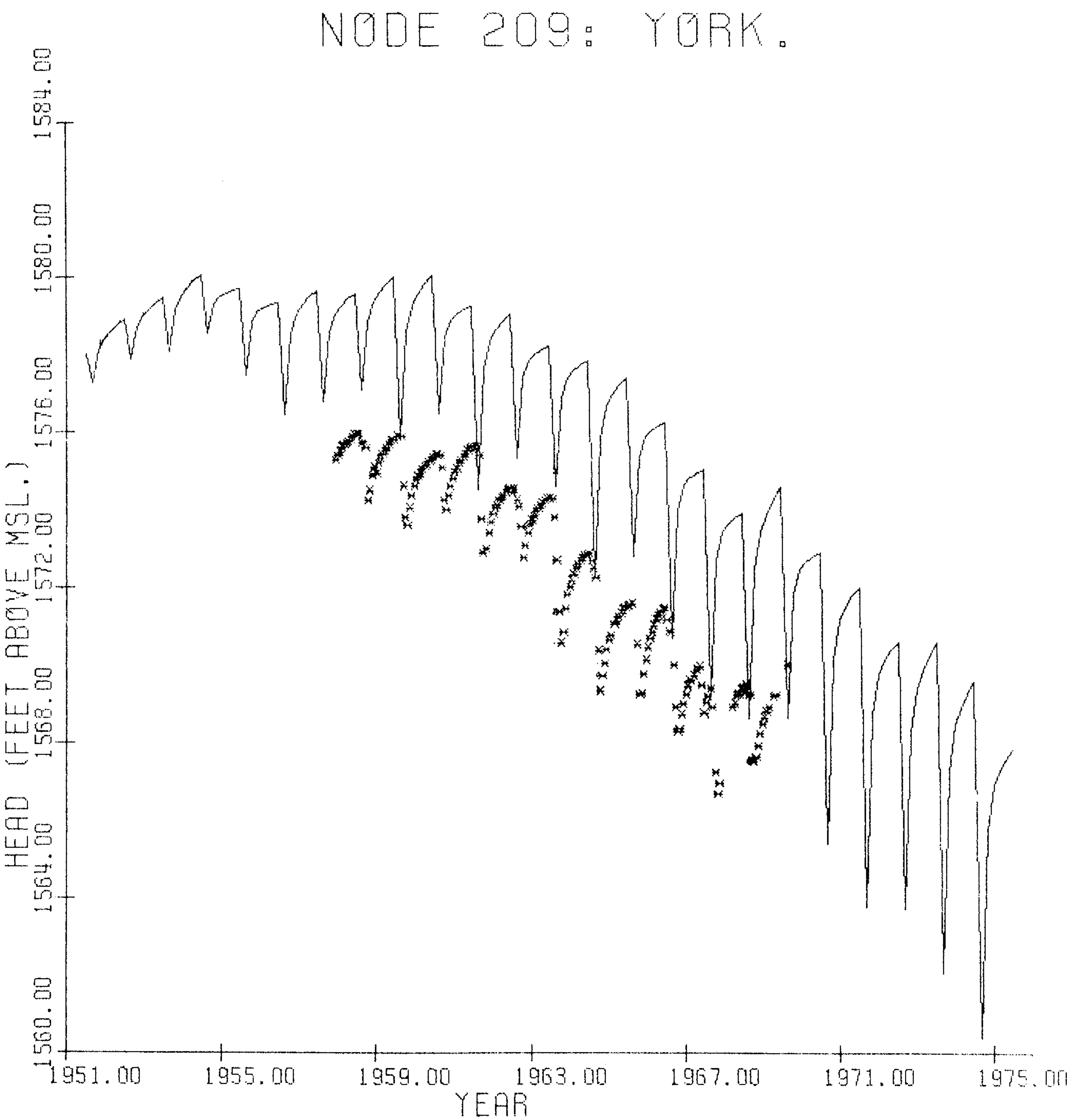

- Simulated values
* Observed values

Figure 12b.--Calculated and observed water levels at Aurora.



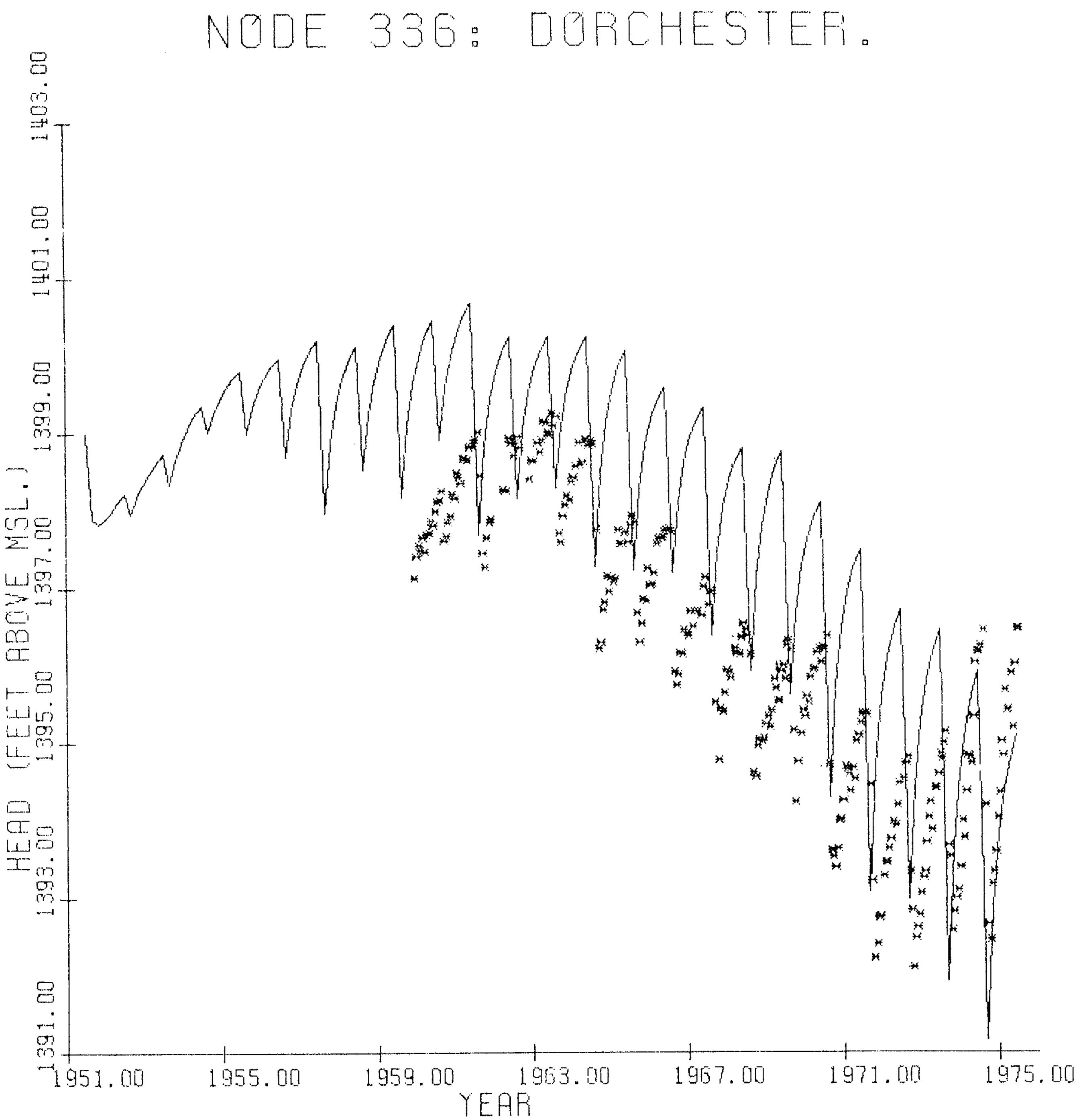

- Simulated values
* Observed values

Figure 12c.--Calculated and observed water levels at Shickley.




- Simulated Values
* Observed values

Figure 12d.--Calculated and observed water levels at York.




- Simulated values
* Observed values

Figure 12e.--Calculated and observed water levels at Dorchester.

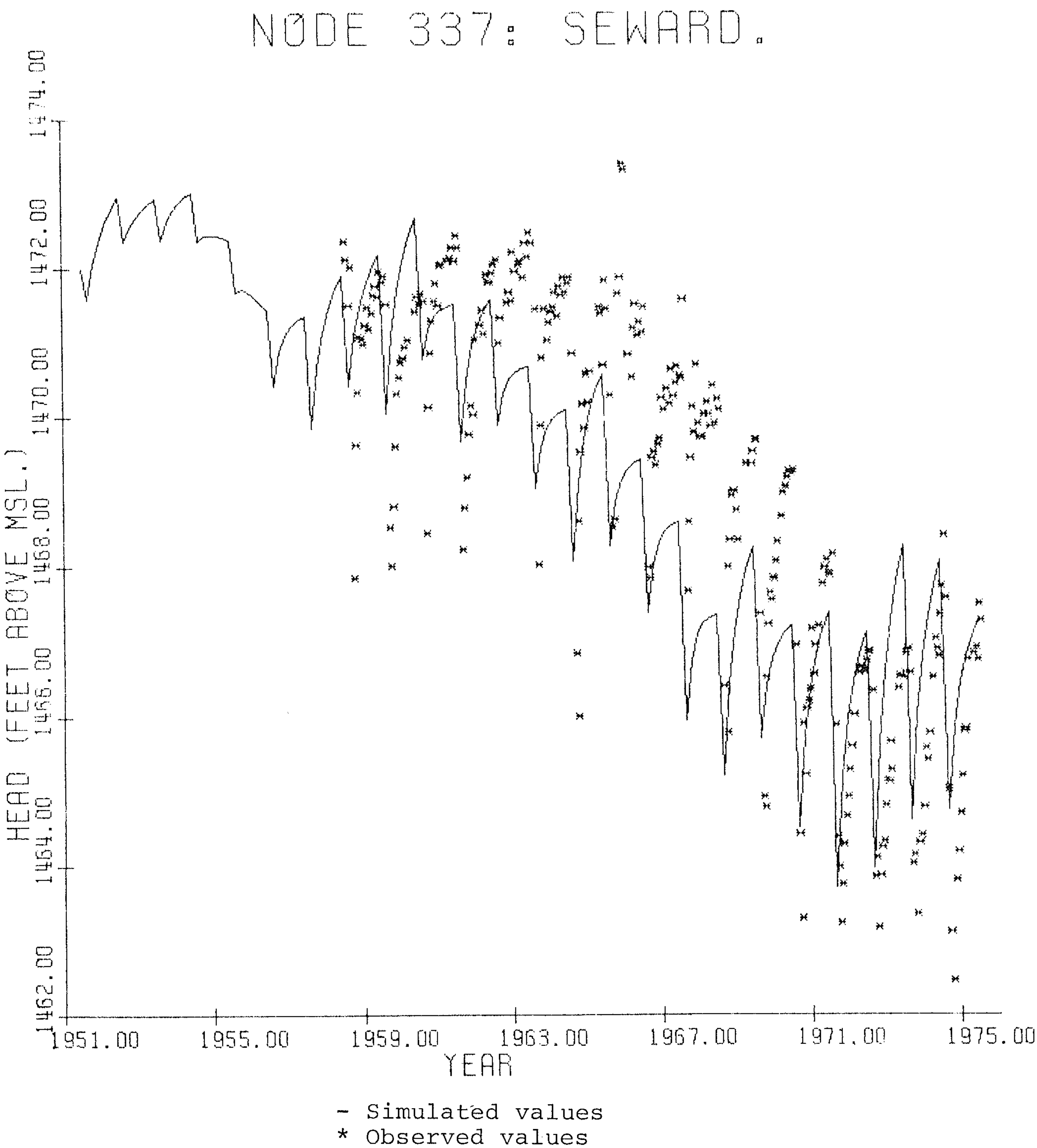



Figure 12f.--Calculated and observed water levels at Seward.

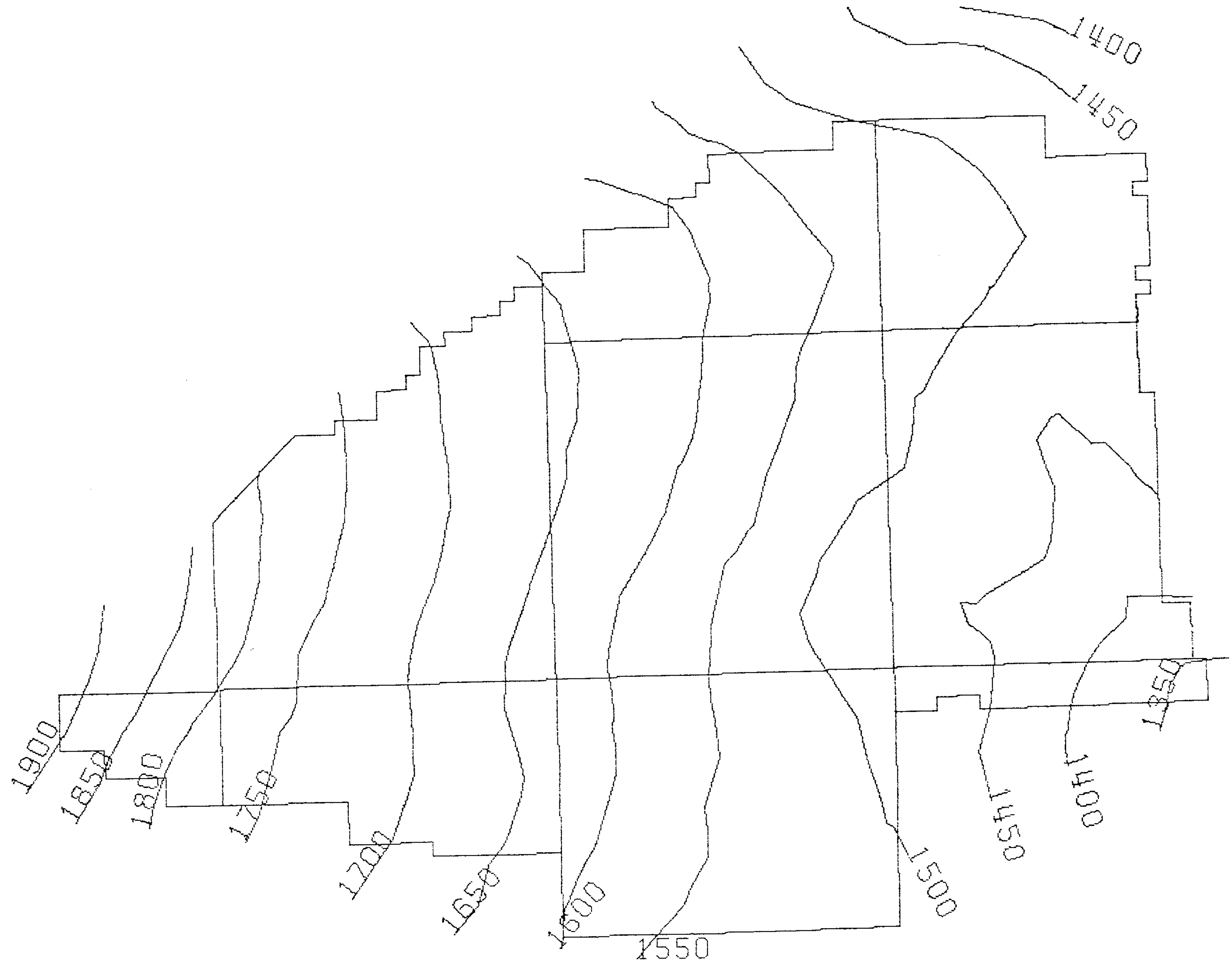



Figure 13.--Computed water levels, in feet above mean sea level, Spring 1975.

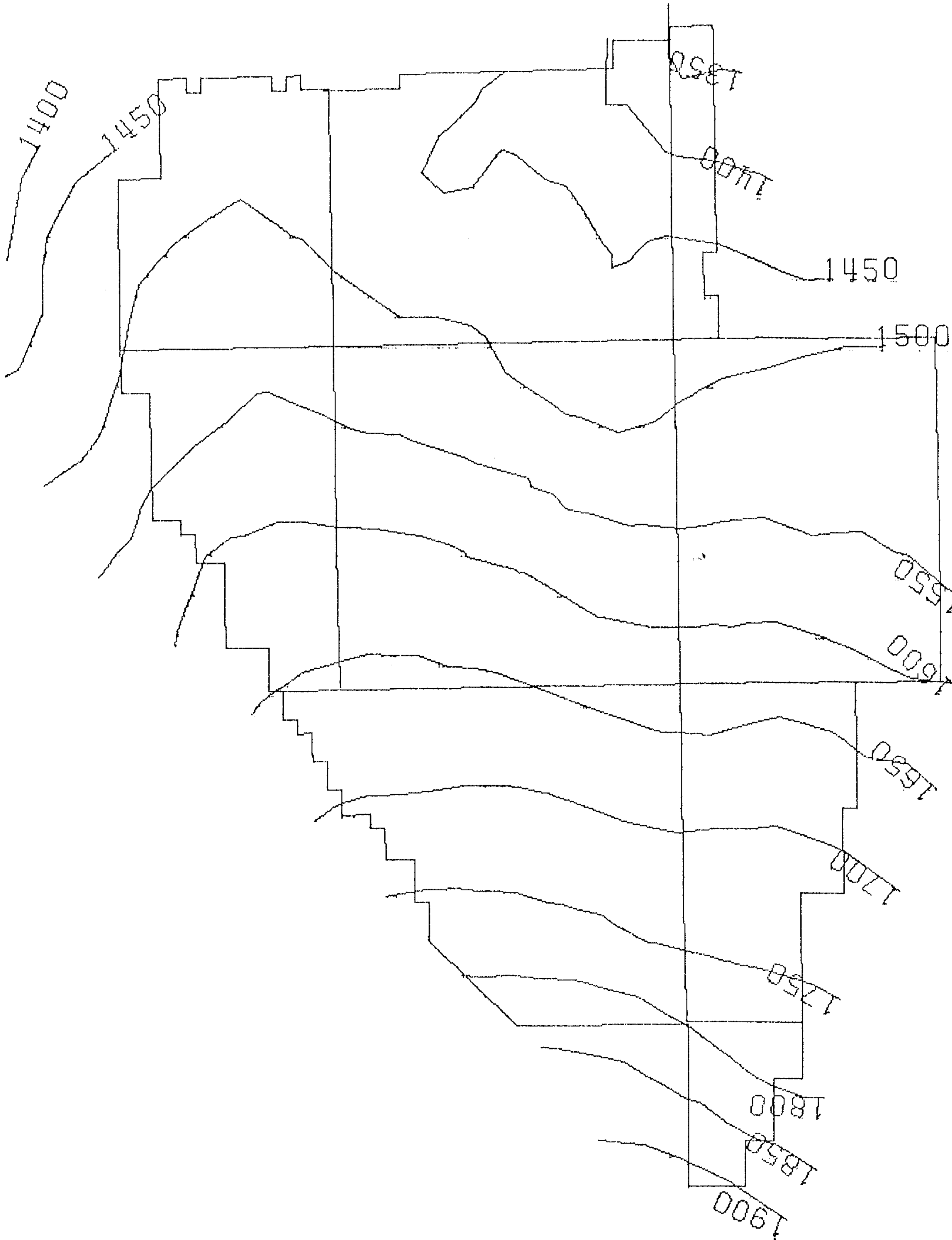



Figure 14.--Observed water levels in feet, Spring 1975.

throughout the simulation since data to estimate this term rigorously is not currently available.

Once a suitable fit was achieved between the model behavior and the historical response, the model was considered calibrated within the limits of the data and the approximate nature of the numerical simulation method.

DEVELOPMENT SCHEMES

As a means to display the utility of the simulation routine and to assess the impact of various management alternatives on hypothetical irrigation development schemes, a number of cases were simulated from Spring of 1975 through Spring of 2000. The results of these simulations are represented by areal water-level maps (figs. 15a-15r, in appendix 2), areal water-level decline maps (figs. 16a-16r, in appendix 2), and maps representing elevation of water level above the base of the aquifer (figs. 17a-17r, in appendix 2).

Development of the groundwater resource within the Upper Big Blue NRD has increased from year to year but not at a constant rate. It is impossible to predict accurately the future rate of development since so many unpredictable factors affect development of groundwater irrigation. Three different rates of irrigation well installation were chosen in order to evaluate the impact of irrigation development rate. The initial simulations restricted the development to the 1977 level as indicated by irrigation well registrations. The historical development pattern within the NRD was used in order to estimate a rate representing the long-term historical rate (0.070 wells per section per year) to be applied to the second series of simulations. A third set of simulations used a development rate representing the maximum historical irrigation well installation rate (0.185 wells per section per year). It was assumed that the maximum number of wells per section would never exceed six and that development at any location would not exceed the 1977 level. The purpose for the latter restriction is to control development in areas where irrigation has historically not flourished.

Allocation of groundwater for irrigation as an additional means for managing the groundwater resource has been addressed by this study. The first simulation in each of the three development rate series bases irrigation usage on estimated historical usage. The remaining three simulations for each of the three development rate series assume yearly allocations of 9, 12, and 15 acre-inches per acre per well with an average of 88.7 acres per well. All of the simulations assumed average precipitation throughout the simulation period.

Twelve combinations were simulated for completion of this phase--four discharge schemes for each of three development schemes (see table 1).

Figure 18 (in appendix 2) may be used with figures 17a-17r to evaluate changes of the elevation of the water level above the base of the aquifer from the Spring of 1975 under the various development schemes.

TABLE 1. INDEX TO OUTPUT PLOTS FOR 12 DEVELOPMENT SCHEMES

| Scheme Number | Development Rate (wells/section/yr.) | | | Irrigation Application (Acre-inches/acre) | | | | Output Plot Subfigure letter[1] | |
|---|---|---|---|---|---|---|---|---|---|
| | 0.000 | 0.070 | 0.185 | historic use | 9.0 | 12.0 | 15.0 | Spring 1990 | Spring 2000 |
| 01 | x | | | x | | | | a | b |
| 02 | x | | | | x | | | | c |
| 03 | x | | | | | x | | | d |
| 04 | x | | | | | | x | | e |
| 05 | | x | | x | | | | f | g |
| 06 | | x | | | x | | | h | i |
| 07 | | x | | | | x | | j | k |
| 08 | | x | | | | | x | l | m |
| 09 | | | x | x | | | | n | o |
| 10 | | | x | | x | | | | p |
| 11 | | | x | | | x | | | q |
| 12 | | | x | | | | x | | r |

[1]Each output subfigure letter designates the entire suite of plots for a specific scheme and date. For example: The water-level plot for Spring 1990 under scheme 06 is figure 15h, and the water-level decline and plots of elevation of the piezometric surface above base of aquifer are figures 16h and 17h, respectively.

APPENDIX 1

While the numerical simulation is accomplished within one computer program (PGM1), numerous additional programs were required to generate the input to the simulation program and process the output from the simulation program. The following programs were written for this project but their usefulness is not limited to it. Explanatory comments are included within the programs as a means of tracing the logic of the program without the need for interpreting the program code.

PGM1-NUMERICAL SIMULATION

The major program written for the project is the routine which performs the actual numerical simulation of transient areal groundwater flow. This program is an adaptation of ISOQUAD2 (Pinder, 1974) programmed by G. F. Pinder, E. O. Frind and P. C. Trescott. Numerous additions and modifications were required in order to meet the needs of regional groundwater simulation, since the initial program was designed for smaller time frame problems, such as well field design.

The most obvious alteration of the initial program is the adaptation to triangular finite elements. The simplicity of triangles in this application far outweighs the geometric flexibility of the isoparametric elements. Once the switch was made to triangular elements, additional programs were possible which further exploited the simplicity of the triangular elements. The plotting program (PGM3) and the apportioning program (PGM4) depend entirely on the utilization of triangular elements.

Triangular elements allow the element mesh to be altered with ease if there is need for better definition of the aquifer stress or response in a particular area.

The partial differential equations describing transient, areal, groundwater flow are approximated by a set of linear equations with groundwater head as the dependent variable. The Galerkin method (Pinder, 1974) is used to generate the approximate integral equations while triangular finite elements form the basis for the integrations. The simultaneous linear equations are solved directly through a two-step process using the Cholesky method (Pinder and Gray, 1977) for factorization of the associated coefficient matrix.

Various naturally occurring boundary conditions may be approximated by simulating known head, constant gradient, and no-flow boundaries. Streams may be simulated as known head nodes or as nodes with aquifer-stream leakage. Delayed drainage of aquifer materials may be simulated since the program also incorporates terms analagous to those proposed by Boulton and Pontin (1971). The transmissivity may be allowed to vary directly with the saturated thickness as a means of simulating unconfined conditions. Well discharges may be adjusted in order to account for reduced well capacity due to limited transmissivity and saturated thickness. Data necessary for an economic analysis of pumping costs and reduced discharge is output for assessment of this aspect of the various schemes.

The output for the simulation is essentially in two forms. One, the printed output, serves to document the input and results, and the second serves primarily as input to subsequent graphical output routines. Plots of head, drawdown, or saturated thickness may be generated in contour-map form. Water-level hydrographs may also be produced in order to display the response of the water level at one site as a function of time.

```
C-PGM1*****************************************************************
C *                                                                   *
C *  PROGRAM ORIGINATED AS ISOQUAD2 PROGRAMMED BY PINDER, FRIND AND   *
C *  TRESCOTT (PINDER, G.F., 1974.  A GALERKIN-FINITE ELEMENT MODEL   *
C *  FOR AQUIFER EVELUATION: WATER RESOURCES PROGRAM, PRINCETON       *
C *  UNIVERSITY, PRINCETON, NEW JERSEY 08540).  SUBSEQUENT ADDITIONS, *
C *  DELETIONS AND MODIFICATIONS TO THE ORIGINAL CODE ARE THE RESULT  *
C *  OF WORK BY RALPH CADY, CONSERVATION AND SURVEY DIVISION,         *
C *  UNIVERSITY OF NEBRASKA, LINCOLN, NEBRASKA, 68588.                *
C *                                                                   *
C *                                                                   *
C *******************************************************************
C *                                                                   *
C *                         APRIL 1978                                *
C *                                                                   *
C *  PURPOSE: TO SIMULATE TRANSIENT, AREAL GROUNDWATER FLOW USING     *
C *           TRIANGULAR FINITE ELEMENTS WITH A DIAGONALIZED TIME     *
C *           DERIVATIVE MATRIX.  INCLUDES OPTION FOR INCLUDING       *
C *           DELAYED YIELD BASED UPON BOULTON EQUATION.              *
C *******************************************************************
C
C *******************************************************************
C *  A........ELEMENT AREA.                                           *
C *  ALPH.... ALPHA FROM BOULTON EQUATION FOR DELAYED YIELD.          *
C *  BASE.....ELEVATION OF BASE OF AQUIFER FOR WATERTABLE PROBLEM.    *
C *  CHNG.....MULTIPLIER FOR INCREASING TIME STEP (DELT=CHNG*DELT).   *
C *  CS.......ARRAY CONTAINING DELAYED DRAINAGE TERMS (BOULTON).      *
C *  DDDN.....SATURATED THICKNESS OR DRAWDOWN.                        *
C *  DELT.....DURATION OF INITIAL TIME STEP.                          *
C *  DGX......X DERIVATIVE OF BASIS FUNCTION.                         *
C *  DGY......Y DERIVATIVE OF BASIS FUNCTION.                         *
C *  DISC.....AVERAGE WELL DISCHARGE FOR INDIVIDUAL NODES.            *
C *  DVAL.....ARRAY CONTAINING COEFFICIENT FOR DELAYED DRAINAGE.      *
C *  EA3......EXPONENTIAL DECAY TERM FOR DELAYED DRAINAGE.            *
C *  EDP......SAME AS EA3 EXCEPT FOR THE PREVIOUS TIME STEP.          *
C *  FACTOR...MULTIPLIER FOR X AND Y COORDINATES.                     *
C *  FQ.......SOURCE AND SINK LUMPED AT NODES (DISCHARGE IS NEGATIVE). *
C *  FTFM.....TRANSFORMATION FROM EXTERNAL NODE NUMBER TO INTERNAL    *
C *           NODE REPRESENTATION.                                    *
C *  HSAVE....ARRAY FOR OUTPUT OF HYDROGRAPH HEAD VALUES.             *
C *  HVAL.....INPUT VECTOR FOR LINEAR EQUATION SOLVER BASED UPON      *
C *           THE CHOLESKY (SQUARE ROOT) METHOD.                      *
C *  IDW......TOTAL NUMBER OF NODES DEWATERED EACH TIME STEP.         *
C *  IN.......ELEMENT INDICES (INPUT COUNTER-CLOCKWISE).              *
C *  IT.......CUMULATIVE NUMBER OF TIME STEPS.                        *
C *  ITP......ARRAY DESIGNATING NODE AT HEADWATER OF STREAM.          *
C *  ITCHNG...NUMBER OF TIME STEPS BETWEEN CHANGES IN DELT.           *
C *  ITRANS...NUMBER OF TIME STEPS BETWEEN UPDATE OF WATERTABLE TRANS. *
C *  IOUT.....NUMBER OF TIME STEPS BETWEEN PRINTED OUTPUT.            *
C *  IDISK....NUMBER OF TIME STEPS BETWEEN DISK OUTPUT.               *
C *  ITMAX....MAXIMUM PERMITTED NUMBER OF TIME STEPS.                 *
C *  JD.......ARRAY OF ACTIVE NODES IN ELEMENT.                       *
C *  KODI.....CONTROL CODES: 1 INITIATES PROCEDURE.                   *
C *                          0 SUPPRESSES PROCEDURE.                  *
```

```
C *    KODD...INCLUDE DELAYED DRAINAGE TERM (BOULTON).                      *
C *    KOD1...PRINT ELEMENT COEFFICIENT MATRICES.                           *
C *    KOD2...PRINT GLOBAL COEFFICIENT MATRIX.                              *
C *    KOD3...ASSUME STEADY STATE PROBLEM.                                  *
C *    KOD4...PRINT KNOWN VECTOR (IE. RIGHT HAND SIDE OF EQUATION).         *
C *    KOD5...READ NODAL TRANSMISSIVITY AND STORAGE COEFFICIENT.            *
C *    KOD6...CONSIDER WATERTABLE PROBLEM.                                  *
C *    KOD7...CALCULATE NODAL WEIGHTED AREA AND ESTIMATED STEADY            *
C *           RECHARGE AND WRITE TO DISK: FT11F001.                         *
C *    KOD8...ALLOW FOR STREAMS AS LEAKY NODES.                             *
C *    KOD9...INCLUDE NODAL RECHARGE-DISCHARGE.                             *
C *    KOD10..EXTERNAL TO INTERNAL NODE REPRESENTATION:                     *
C *           =0: CONDENSED WITH REMOVAL OF FIXED HEAD NODES.               *
C *           =1: TRANSFORMATION ON DEVICE FT02F001.                        *
C *    KOD11..PRINT NODAL DRAWDOWN VALUES.                                  *
C *    KOD12..WRITE COMPUTED HEAD VALUES TO DISK:FT12F001.                  *
C *    KOD13..WRITE COMPUTED DRAWDOWN TO DISK:FT12F001.                     *
C *    KOD14..INITIALIZE LISTS: 'HEADS', 'FIXED' AND 'VARI'.                *
C *    KOD15..UPDATE LISTS:'HEADS' AND 'VARI'.                              *
C *    KOD16..SAVE PARTICULAR NODAL HYDROGRAPHS ON DEVICE FT08F001.         *
C *    KODP...INTERNAL INDICATOR FOR PUMPING VS. NON-PUMPING PERIOD.        *
C *  L........INDEX INDICATING ELEMENT NUMBER.                              *
C *  LRC......CUMMULATIVE SUMMATION FOR CONSTANT HEAD VALUES.               *
C *  LR.......INDICATOR ARRAY FOR CONSTANT HEAD VALUES.                     *
C *  MOUTH....ARRAY DESIGNATING TAILWATER NODE OF A STREAM.                 *
C *  MV.......ADDRESS VECTOR FOR LINEAR ARRAYS SVAL AND HVAL.               *
C *           LOCATES DIAGONAL POSITIONS WITHIN EACH VECTOR.                *
C *  N........NUMBER OF DEGREES OF FREEDOM (NN-NS-NBK).                     *
C *  NBND.....NUMBER OF CONSTANT GRADIENT BOUNDARY NODES.                   *
C *  NBK......NUMBER OF NODES WITH HEAD-TIME DATA (NOT CONSTANT HEAD).      *
C *  NE.......NUMBER OF ELEMENTS.                                           *
C *  NEM......LOWER HALF HORIZONTAL PROFILE FOR SVAL AND HVAL.              *
C *  NN.......NUMBER OF NODES.                                              *
C *  NS.......NUMBER OF CONSTANT HEAD NODES.                                *
C *  NSAVE....TOTAL NUMBER OF NODAL HYDROGRAPHS SAVED ON UNIT 08.           *
C *  NSN......NUMBER OF STREAM NODES.                                       *
C *  NSS......NUMBER OF STREAM SYSTEMS.                                     *
C *  OLD......COMPUTED HEAD FROM PREVIOUS TIME STEP (CONDENSED).            *
C *  QBOUND...INITIAL FLUX ACROSS CONSTANT GRADIENT BOUNDARY.               *
C *  PBOUND...POINTER ARRAY FOR CONNECTING QBOUND AND SPECIFIC NODES.       *
C *  PSAVE....POINTER ARRAY FOR SAVING NODAL HYDROGRAPHS.                   *
C *  PVAL.....GLOBAL COEFFICIENT MATRIX FOR THE TIME DERIVATIVE.            *
C *  PE.......ELEMENT VECTOR ASSOCIATED WITH TIME DERIVATIVE.               *
C *  PTRANS...TRANSMISSIVITY ARRAY FOR NODES OR ELEMENTS.                   *
C *  RT.......VECTOR CONTAINING BOUNDARY CONDITIONS.                        *
C *  SMAX.....TOTAL SPECIFIC YIELD (LONG TERM).                             *
C *  SVAL.....GLOBAL COEFFICIENT MATRIX FOR THE SPACE DERIVATIVE.           *
C *  SE.......ELEMENT MATRIX ASSOCIATED WITH SPACE DERIVATIVE.              *
C *  STIME....ELAPSED TIME IN 'UNITT' UNITS.                                *
C *  STORAG...STORAGE COEFFICIENT.                                          *
C *  TIME.....SIMULATION PERIOD.                                            *
C *  U........CALCULATED HEAD (EXPANDED).                                   *
C *  UI.......INITIAL HEAD (EXPANDED).                                      *
C *  UNITL....LENGTH UNITS.                                                 *
```

```
C *   UNITT....TIME UNITS.                                                  *
C *   UO.......COMPUTED HEAD FROM PREVIOUS TIME STEP (EXPANDED).            *
C *   UPF......ARRAY FOR FLOW TO HEADWATER STREAM NODE FROM UPSTREAM.       *
C *   X .......X COORDINATE OF NODE.                                        *
C *   Y .......Y COORDINATE OF NODE.                                        *
C ***************************************************************************
C
C....MAIN PROGRAM FOR INITIATING OBJECT TIME DIMENSION STATEMENTS
      REAL*4     DGX(03),DGY(03),SE(03,03),PE(03)
C
C +---------------THE ENCLOSED CARDS ARE PROBLEM DEPENDENT--------------+
      INTEGER*2 IN(03,750),LR(400),LRC(400),NSTP(190,3),MV(400),
     1          PBOUND(40),ITP(05),MOUTH(05),FTFM(400),JD(3)
      DIMENSION SVAL(07100),HVAL(07100),FM(400),RT(400),PVAL(400),
     1          U(400),UI(400),UO(400),OLD(400),PTRANS(400),STORAG(400),
     2          BASE(400),DDDN(400),X(400),Y(400),CS(400),DVAL(400),
     3          SFLW(190),QLK(190),US(190),FQ(400),QBOUND(40),UPF(05)
      DIMENSION DISC(400)
      JNSS=05
      JNEL=750
      JNND=400
      JNMX=400
      JSTN=190
      JNEM=07100
      JBND=40
C
C==========================================================================
C
      CALL REDUCI (JNND,DISC,U,STORAG,PTRANS,FM,FQ,BASE,DDDN,LRC)
      CALL DSETI  (JNND,JNEL,JNMX,DVAL,CS,IN,STORAG,U,UO,JD)
      CALL UPDATI (JNND,FQ,UO,U,LR)
      CALL SFLUXI (JNND,JSTN,JNSS,ITP,MOUTH,NSTP,LR,UPF,QLK,US,UI,U,
     *             SFLW)
      CALL AREASI (JNND,JNEM,JSTN,JBND,PBOUND,QBOUND,BASE,U,UI,QLK,US,
     *             SVAL,NSTP,LR,SE,IN,DGX,DGY,DDDN,JNMX,PTRANS,
     *             JNEL,JD)
      CALL SHAPEI (JNND,X,Y,DGX,DGY,JD)
      CALL FACTI  (JNND,JNEM,HVAL,MV)
      CALL SOLVEI (JNND,JNEM,HVAL,FM,OLD,MV)
      CALL MNPROG (JNEL,JNND,JNEM,JBND,JNSS,JSTN,IN,LR,LRC,PBOUND,ITP,
     *             MOUTH,NSTP,MV,X,Y,UI,UO,U,OLD,US,PTRANS,STORAG,DDDN,
     *             BASE,FQ,RT,FM,QBOUND,DGX,DGY,QLK,SE,PE,SVAL,PVAL,
     *             HVAL,UPF,FTFM,CS,JNMX,DVAL,JD)
      STOP
      END
```

```
C-MAIN PROGRAM ROUTINE:
C ----------------------------------------------------------------------
      SUBROUTINE MNPROG (JNEL,JNND,JNEM,JBND,JNSS,JSTN,IN,LR,LRC,PBOUND,
     1                  ITP,MOUTH,NSTP,MV,X,Y,UI,UO,U,OLD,US,PTRANS,
     2                  STORAG,DDDN,BASE,FQ,RT,FM,QBOUND,DGX,DGY,QLK,
     3                  SE,PE,SVAL,PVAL,HVAL,UPF,FTFM,CS,JNMX,DVAL,JD)
      INTEGER*2 IN(03,JNEL),LR(JNND),LRC(JNND),PBOUND(JBND),ITP(JNSS),
     1          MOUTH(JNSS),NSTP(JSTN,3),MV(JNND),LRT(20),FTFM(JNND),
     2          PSAVE(19),JD(03),LV(4,5)
      DIMENSION X(JNND),Y(JNND),UI(JNND),UO(JNND),U(JNND),OLD(JNND),
     1          US(JSTN),PTRANS(JNMX),STORAG(JNMX),DDDN(JNND),CS(JNND),
     2          BASE(JNND),FQ(JNND),RT(JNND),QBOUND(JBND),HVAL(JNEM),
     3          QLK(JSTN),UPF(JNSS),HSAVE(019),DVAL(JNND),PVAL(JNND),
     4          TITLE(20),FM(JNND),SVAL(JNEM),STREAM(13),KOD(16)
      REAL*4    DGX(03),DGY(03),SE(03,03),PE(03)
      EQUIVALENCE (KOD(1),KOD1),(KOD(2),KOD2),(KOD(3),KOD3),
     *(KOD(4),KOD4),(KOD(5),KOD5),(KOD(6),KOD6),(KOD(7),KOD7),
     *(KOD(8),KOD8),(KOD(9),KOD9),(KOD(10),KOD10),(KOD(11),KOD11),
     *(KOD(12),KOD12),(KOD(13),KOD13),(KOD(14),KOD14),(KOD(15),KOD15),
     *(KOD(16),KOD16)
      COMMON/AREA1/KOD,NN,NBK,NBND,MOUT,UNITT,STIME,DELT,DELT1,KODP
      DATA START/4HTRIA/,START2/4HNAME/,NST/0/,NEL/0/,EA3/0./,YES/3HYES/
C
C.....READ TITLE AND CONTROL DATA.
   10 READ(05,2130) TITLE
      WRITE(6,2050) TITLE
      IF(TITLE(1).EQ.START) GO TO 20
      IF(TITLE(1).EQ.START2) GO TO 560
      GO TO 10
   20 READ(05,2130) TITLE
      READ(05,2000) UNITL,UNITT
      WRITE(6,2140)
      WRITE(6,2150)
      WRITE(6,2160) TITLE
      WRITE(6,2550) UNITL,UNITT
      WRITE(6,2390)
      READ(05,2170) TITLE,NN
      WRITE(6,2540) TITLE,NN
      READ(05,2170) TITLE,NE
      WRITE(6,2540) TITLE,NE
      READ(05,2180) TIME,DELT,CHNG,ITMAX,ITCHNG,ITRANS,IOUT,IDISK
      WRITE(6,2190) TIME,UNITT,DELT,CHNG,ITMAX,ITCHNG,ITRANS,IOUT,IDISK
      DELT1=DELT
C
C.....INITIALIZE CONSTANTS.
      IT=0
      DELTP=DELT
      STIME=0.0
C
C.....READ OPTION CODES AND PRINT OPTIONS.
      READ(05,2010) TITLE,DUM
      KODD=0
      IF(DUM.EQ.YES) KODD=1
      WRITE(6,2510)
```

```
      READ(05,2180) SMAX,ALPH
      WRITE(6,2520) TITLE,DUM,KODD
      IF(KODD.EQ.1) WRITE(6,2690) SMAX,ALPH
      DO 30 I=1,16
      READ(05,2010) TITLE,DUM
      KOD(I)=0
      IF(DUM.EQ.YES) KOD(I)=1
   30 WRITE(6,2530) TITLE,DUM,I,KOD(I)
      IF(KOD16.LE.0) GO TO 50
C
C.....READ NODES FOR OUTPUT HYDROGRAPHS.
      READ(05,2170) TITLE,NSAVE
      WRITE(6,2230)
      WRITE(6,2540) TITLE,NSAVE
      KOD16=KOD16*NSAVE
      IF(KOD16.LE.0) STOP
      DO 40 I=1,KOD16
      READ(05,2170) TITLE,PSAVE(I)
   40 WRITE(6,2540) TITLE,PSAVE(I)
C
C.....READ CONSTANT GRADIENT BOUNDARY NODES & INITIAL BOUNDARY FLUX.
   50 READ(05,2170) TITLE,NBND
      WRITE(6,2230)
      WRITE(6,2540) TITLE,NBND
      IF(NBND.LE.0) GO TO 60
      READ(05,2560)(PBOUND(I),QBOUND(I),I=1,NBND)
      WRITE(6,2270)(PBOUND(I),QBOUND(I),I=1,NBND)
C
C.....READ NODAL PARAMETERS.
   60 READ(05,2180) FACTOR
      IF(KOD5.EQ.1) GO TO 80
      READ(10,2020)(J,X(J),Y(J),UI(J),BASE(J),I=1,NN)
      GO TO 90
   80 READ(10,2030)(J,X(J),Y(J),UI(J),BASE(J),PTRANS(J),STORAG(J),
     *      I=1,NN)
   90 END FILE 10
      WRITE(6,2200)
      WRITE(6,2210) FACTOR
      WRITE(6,2220)
      WRITE(6,2270)(J,X(J),J,Y(J),J=1,NN)
      DO 100 I=1,NN
      X(I)=X(I)*FACTOR
      Y(I)=Y(I)*FACTOR
      UO(I)=UI(I)
      U(I)=UI(I)
      DDDN(I)=U(I)-BASE(I)
      IF(DDDN(I).LE..01) DDDN(I)=.01
      LR(I)=0
  100 RT(I)=0.0
      WRITE(6,2240)
      WRITE(6,2270)(I,UI(I),I=1,NN)
      WRITE(6,2040)
      WRITE(6,2270)(I,BASE(I),I=1,NN)
      IF(KOD5.NE.1) GO TO 120
```

```
      WRITE(6,2260)
      WRITE(6,2270)(I,PTRANS(I),I=1,NN)
      WRITE(6,2250)
      WRITE(6,2270)(I,STORAG(I),I=1,NN)
      IF(KOD5.NE.1.OR.KOD6.NE.1)  GO TO 120
      DO 110 I=1,NN
  110 PTRANS(I)=PTRANS(I)/DDDN(I)
  120 IF(KOD6.NE.1.OR.NBND.LE.0) GO TO 140
      DO 130 I=1,NBND
      J=PBOUND(I)
  130 QBOUND(I)=QBOUND(I)/DDDN(J)
C
C.....READ ELEMENT INDICES (INPUT COUNTER-CLOCKWISE).
  140 READ(09,2350) LV
      DO 150 J=1,5
      I=LV(1,J)
      IF(I.LE.0) GO TO 160
      NEL=NEL+1
      DO 150 JJ=1,3
  150 IN(JJ,I)=LV(JJ+1,J)
      GO TO 140
  160 IF(NEL.EQ.NE) GO TO 170
      WRITE(6,2310) NEL,NE
      STOP
  170 END FILE 09
      WRITE(6,2320)
      WRITE(6,2330)(L,(IN(I,L),I=1,3),L=1,NE)
      IF(KOD5.NE.0) GO TO 180
C
C.....READ ELEMENT TRANSMISSIVITY AND STORAGE.
      READ(05,2280)(L,STORAG(L),PTRANS(L),K=1,NE)
      WRITE(6,2290)
      WRITE(6,2270)(L,STORAG(L),L=1,NE)
      WRITE(6,2300)
      WRITE(6,2270)(L,PTRANS(L),L=1,NE)
C
C.....READ CONSTANT HEAD BOUNDARY NODES (INDICATED BY LR=3).
C
C     LR(I)= 0 : NODE WITH UNKNOWN HEAD-TIME DATA.
C     LR(I)= 3 : PERMANENT CONSTANT HEAD NODE.
C     LR(I)= 4 : NODE WITH A PRIORI HEAD-TIME DATA.
C     LR(I)=-1 : FLOWING STREAM NODE (LEAKY).
C     LR(I)=-2 : DEPLETED STREAM NODE (LEAKY).
C
  180 WRITE(6,2170)
      READ(05,2170) TITLE,NS
      WRITE(6,2340)
      WRITE(6,2540) TITLE,NS
      IF(NS.EQ.0) GO TO 220
  190 READ(05,2350)(LRT(I),I=1,20)
      K=0
      DO 200 I=1,20
      IF(LRT(I).EQ.0) GO TO 210
      K=I
```

```
      J=LRT(I)
      NST=NST+1
      IF(J.LE.NN) GO TO 200
      WRITE(6,2360) J
      STOP
  200 LR(J)=3
  210 WRITE(6,2370)(LRT(I),I=1,K)
      IF(K.EQ.20) GO TO 190
      IF(NST.NE.NS) WRITE(6,2380) NST,NS
  220 IF(KOD8.NE.1) GO TO 270
C
C.....READ STREAM SYSTEMS AND STREAM NODES.
      READ(05,2170) TITLE,NSS
      WRITE(6,2230)
      WRITE(6,2540) TITLE,NSS
      IF(NSS.LE.0) GO TO 270
      WRITE(6,2090)
      DO 230 J=1,NSS
      READ(05,2110)(I,ITP(I),MOUTH(I),UPF(I)),STREAM
  230 WRITE(6,2100)(I,ITP(I),MOUTH(I),UPF(I)),STREAM
      READ(05,2170) TITLE,NSN
      WRITE(6,2230)
      WRITE(6,2540) TITLE,NSN
      IF(NSN.LE.0) STOP
      WRITE(6,2070)
      DO 240 K=1,NSN
      READ(05,2060) I,(NSTP(I,J),J=1,3),QLK(I),US(I)
      II=NSTP(I,1)
      WRITE(6,2080) I,(NSTP(I,J),J=1,3),QLK(I),US(I),U(II)
  240 LR(II)=-1
      DO 250 I=1,NSN
      II=NSTP(I,1)
      J= NSTP(I,3)
      IF(J.EQ.0) GO TO 250
      JJ=NSTP(J,1)
      DUM=.5*SQRT((X(II)-X(JJ))**2+(Y(II)-Y(JJ))**2)
      RT(I)=RT(I)+DUM
      RT(J)=RT(J)+DUM
  250 CONTINUE
      DO 260 J=1,NSN
  260 QLK(J)=QLK(J)*RT(J)
      MOUT=0
      CALL SFLUX (NSN,NSS,MOUT)
C
C.....READ A-PRIORI NODES (HEAD-TIME RELATIONSHIP DESIGNATED).
  270 READ(05,2170) TITLE,NBK
      WRITE(6,2230)
      WRITE(6,2540) TITLE,NBK
      IF(NBK.EQ.0) GO TO 290
      DO 280 J=1,NBK
      READ(5,2350) JJ
  280 LR(JJ)=4
  290 LRC(1)=0
      IF(LR(1).GT.0) LRC(1)=1
```

```
      DO 300 J=2,NN
      LRC(J)=LRC(J-1)
      IF(LR(J).GT.0) LRC(J)=LRC(J)+1
  300 CONTINUE
C
C.....SUM CONSTANT BOUNDARY NODES.
      N=NN-LRC(NN)
      WRITE(6,2230)
      WRITE(6,2390)
      WRITE(6,2400) NN,NS,NBK,N,NE,NSS,NSN,NBND,NSAVE
C
C.....SET UP INTERNAL REPRESENTATION OF NODES; EITHER CONDENSED OR
C     TRANSFORMED FROM AN OPTIMIZATION ROUTINE (DEVICE FT02F001).
      IF(KOD10.EQ.1) GO TO 340
      NEM=N
      DO 330 I=1,NN
      IF(LR(I).GT.0) GO TO 320
      FTFM(I)=I-LRC(I)
      GO TO 330
  320 NEM=NEM+1
      FTFM(I)=NEM
  330 CONTINUE
      GO TO 350
  340 READ(02,2350)(FTFM(I),I=1,NN)
      END FILE 02
  350 DO 360 I=1,NN
      II=FTFM(I)
  360 LRC(II)=I
      WRITE(06,2650)
      WRITE(06,2660)(I,LR(I),FTFM(I),I,LRC(I),I=1,NN)
C
C.....GENERATE ARRAY STRUCTURE FOR SOLUTION.
      DO 370 I=1,N
  370 LRC(I)=1
      DO 400 L=1,NE
      DO 380 I=1,3
  380 JD(I)=IN(I,L)
      DO 400 I=1,3
      II=JD(I)
      IF(LR(II).GT.0) GO TO 400
      IR=FTFM(II)
      II=I+1
      IF(II.GT.3) GO TO 400
      DO 390 J=II,3
      JJ=JD(J)
      IF(LR(JJ).GT.0) GO TO 390
      JC=FTFM(JJ)
      K=JC-IR+1
      IF(K.GT.LRC(JC)) LRC(JC)=K
      K=IR-JC+1
      IF(K.GT.LRC(IR)) LRC(IR)=K
  390 CONTINUE
  400 CONTINUE
C
```

```
C.....SET UP MV ARRAY (LENGTH OF LEFT HORIZONTAL HALF BANDS).
      MV(1)=1
      DO 410 I=2,N
  410 MV(I)=MV(I-1)+LRC(I-1)
      MV(N+1)=MV(N)+LRC(N)
      NEM=MV(N+1)-1
      WRITE(6,2610)NEM
      IF(NEM.LE.JNEM) GO TO 420
      WRITE(6,2620) NEM,JNEM
      STOP
C
C.....CALCULATE OLD FROM UI, THE INITIAL HEAD VALUES.
  420 DO 430 I=1,NN
      II=FTFM(I)
  430 OLD(II)=UI(I)
C
C.....CREATE AND COMPUTE PVAL MATRIX.
      DO 440 I=1,NN
  440 PVAL(I)=0.0
      IF(KOD3.EQ.1) GO TO 550
      L=0
  450 L=L+1
      DO 460 I=1,3
  460 JD(I)=IN(I,L)
C
C.....THE ARRAY JD NOW CONTAINS THE INDICES OF THE NODES IN ELEMENT L.
C
C.....COMPUTE VALUE OF SHAPE FUNCTION DERIVATIVES AND AREA.
C ***********************************************************************
      CALL SHAPE (L,A)
C ***********************************************************************
      IF(KOD5.EQ.1) GO TO 480
      STOR=STORAG(L)
      GO TO 500
  480 ST=0.0
      DO 490 I=1,3
      II=JD(I)
  490 ST=ST+STORAG(II)
  500 DO 510 I=1,3
      IF(KOD5.NE.1) GO TO 510
      II=JD(I)
      STOR=0.25*(ST+STORAG(II))
  510 PE(I)=STOR*A/3.0
C
C.....PRINT ELEMENT COEFFICIENT MATRICES.
      IF(KOD1.NE.1) GO TO 530
      WRITE(6,2430) L
      DO 520 I=1,3
  520 WRITE(6,2420) I,PE(I)
      WRITE(6,2440)
C
C.....THIS COMPLETES THE ELEMENT STIFFNESS MATRIX.
C=======================================================================
C.....ASSEMBLY OF GLOBAL COEFFICIENT MATRIX PVAL.
```

```
  530 DO 540 I=1,3
      JDI=JD(I)
      IF(LR(JDI).GT.0) GO TO 540
      IR=FTFM(JDI)
      PVAL(IR)=PVAL(IR)+PE(I)
  540 CONTINUE
      IF(L.LT.NE) GO TO 450
C
C.....FORM TIME DELAY VECTOR DVAL.
      IF(KODD.EQ.1) CALL DFORM (NN,NE,SMAX,KOD5)
C
C.....CALCULATE NODAL CONTRIBUTING AREAS AND ESTIMATE STEADY RECHARGE.
  550 IF(KOD7.EQ.1) CALL AREAS (NE,NEM,NSN)
      KOD7=0
      GO TO 570
C
C.....READ LISTED INPUT DATA FROM DISK.
  560 END FILE 05
      READ(3,2130) SVAL,RT,CS,EA3
      READ(4) X,Y,BASE
      READ(4) PTRANS,QLK,US,QBOUND,FBOUND,FTFM
      READ(4) MV,NSTP,DVAL,PVAL,KODD,SMAX,ALPH
      READ(4) IN
      READ(4) UI,STORAG,UPF,ITP,MOUTH,CHNG,DELT1,NN,NE,N,NBND,NBK,NSN,
     *        NSS,NEM,ITRANS,ITCHNG,TIME,ITMAX,UNITT,UNITL,IDISK,IOUT,
     *        KOD,NSAVE,PSAVE
      END FILE 04
      READ(1) U,OLD
      READ(1) DDDN
      READ(1) UO,LR,DELT,DELTP,STIME,IT
      WRITE(6,2470) STIME,UNITT,IT,DELT
      WRITE(6,2270)(I,U(I),I=1,NN)
  570 IF(KOD16.LE.0) GO TO 590
      DO 580 I=1,NSAVE
      II=PSAVE(I)
  580 HSAVE(I)=U(II)
      WRITE(8,2350) KOD16,PSAVE
      WRITE(8,2130) STIME,HSAVE
C
C.....READ IN INITIAL NODAL DISCHARGES AND/OR A-PRIORI HEAD VALUES.
  590 IF(KOD9.EQ.1.OR.NBK.GT.0) CALL INITL
C
C.....INITIALIZE TIME CHECKING OPTION (FACILITY DEPENDENT).
      CALL CPTIME (K)
      CPMAX=0.0
      CPT=FLOAT(K)*0.00002604
      CPRE=CPT
      IF(TITLE(1).NE.START2) GO TO 1000
      IF(MOD(IT,ITRANS).NE.0.OR.IT.EQ.0) GO TO 1200
C     ==================================================================
C
C.....BEGIN CALCULATIONS FOR GLOBAL COEFFICIENT MATRICES.
 1000 IF(KOD6.NE.1) GO TO 1020
C
```

```
C.....ADJUST WATERTABLE TRANSMISSIVITY.
      DO 1010 I=1,NN
      DDDN(I)=U(I)-BASE(I)
 1010 IF(DDDN(I).LE..01) DDDN(I)=.01
C
C.....CLEAR ARRAYS.
 1020 DO 1030 J=1,NEM
 1030 SVAL(J)=0.0
      DO 1040 I=1,NN
 1040 RT(I)=0.0
C
C.....INITIALIZE CONSTANTS.
      A3=1./DELT
      L=0
C======================================================================
C.....BEGIN CALCULATION OF ELEMENT COEFFICIENT MATRICES.
 1050 L=L+1
      DO 1060 I=1,3
 1060 JD(I)=IN(I,L)
C
C.....THE ARRAY JD NOW CONTAINS THE INCIDENCES OF THE NODES IN L.
C **********************************************************************
      CALL SHAPE (L,A)
C **********************************************************************
C
C.....SELECT PARAMETER VALUES.
      IF(KOD5.EQ.1) GO TO 1080
      TR=PTRANS(L)
      GO TO 1100
 1080 TR=0.0
      DUM=1.0
      DO 1090 II=1,3
      I=JD(II)
      IF(KOD6.EQ.1) DUM=DDDN(I)
 1090 TR=TR+PTRANS(I)*DUM
      TR=TR/3.0
 1100 DO 1110 I=1,3
      DO 1110 J=I,3
      SE(I,J)=TR*(DGX(I)*DGX(J)+DGY(I)*DGY(J))
 1110 SE(J,I)=SE(I,J)
C
C.....PRINT ELEMENT COEFFICIENT MATRICES.
      IF(KOD1.NE.1) GO TO 1130
      WRITE(6,2410) L
      DO 1120 I=1,3
 1120 WRITE(6,2420) I,(SE(I,J),J=1,3)
      WRITE(6,2440)
C
C.....THIS COMPLETES THE ELEMENT STIFFNESS MATRIX.
C======================================================================
C.....ASSEMBLY OF GLOBAL COEFFICIENT MATRIX SVAL.
 1130 DO 1180 I=1,3
      JDI=JD(I)
      IF(LR(JDI).GT.0) GO TO 1150
```

```
      IR=FTFM(JDI)
      DO 1140 J=I,3
      JDJ=JD(J)
      IF(LR(JDJ).GT.0) GO TO 1140
      JC=FTFM(JDJ)
      K=MV(JC)+JC-IR
      IF(JC.LT.IR) K=MV(IR)+IR-JC
      SVAL(K)=SVAL(K)+SE(I,J)
 1140 CONTINUE
      GO TO 1180
 1150 DO 1160 J=1,3
      JC=JD(J)
      IF(LR(JC).GT.0) GO TO 1160
      RT(JC)=RT(JC)+SE(I,J)*U(JDI)
 1160 CONTINUE
      IF(LR(JDI).NE.3) GO TO 1180
      DO 1170 J=1,3
      JDJ=JD(J)
 1170 RT(JDI)=RT(JDI)+SE(I,J)*U(JDJ)
 1180 CONTINUE
      IF(L.LT.NE) GO TO 1050
      IF(KOD14.EQ.1) GO TO 1560
C
C.....CALCULATIONS FOR GLOBAL COEFFICIENT MATRIX COMPLETE.
      DO 1190 I=1,NN
      IF(LR(I).EQ.3) WRITE(6,2570) RT(I),FQ(I),I
 1190 CONTINUE
      IF(KOD2.NE.1) GO TO 1200
      WRITE(6,2450)
      WRITE(6,2460)(RT(I),I=1,NN)
C
C.....BEGIN TIME LOOP.
 1200 A3=1./DELT
      CALL EMOD (ALPH,A3,EA3,EDP)
C
C.....COMPUTE COMPLETE COEFFICIENT MATRIX.
      DO 1210 I=1,NEM
 1210 HVAL(I)=SVAL(I)
      DO 1220 I=1,N
      K=MV(I)
 1220 HVAL(K)=HVAL(K)+PVAL(I)*A3
      IF(KODD.EQ.0) GO TO 1240
      DO 1230 I=1,NN
      IF(LR(I).GT.0) GO TO 1230
      IR=FTFM(I)
      II=MV(IR)
      HVAL(II)=HVAL(II)+DVAL(I)*EDP
 1230 CONTINUE
 1240 IF(KOD2.NE.1) GO TO 1270
C
C.....PRINT GLOBAL COEFFICIENT MATRIX.
      DO 1260 I=1,N
      WRITE(6,2630) I
      II=I+MV(I)-MV(I+1)+1
```

```
      J=MV(I)+I
 1250 JJ=II+8
      IF(JJ.GT.I)JJ=I
      WRITE(6,2640)(K,HVAL(J-K),K=II,JJ)
      II=JJ+1
      IF(II.LE.I) GO TO 1250
 1260 WRITE(6,2440)
C
C.....DECOMPOSE LEFT-HAND ARRAY HVAL.
C *************************************************************************
 1270 CALL FACTR (N,FACT,KOD2)
C *************************************************************************
C
C.....CALCULATE VECTOR OF KNOWN VALUES (RIGHT-HAND SIDE).
 1280 DO 1290 I=1,N
 1290 FM(I)=0.0
      IF(KOD8.EQ.0) GO TO 1310
C
C.....IF STREAMS TREATED AS LEAKY, ALTER KNOWN VECTOR TO ACCOUNT FOR
C      AMOUNT OF LEAKAGE BASED UPON LATEST HEAD VALUES.
      DO 1300 I=1,NSN
      II=NSTP(I,1)
      IF(LR(II).NE.-1) GO TO 1300
      IR=FTFM(II)
      DUM=US(I)-U(II)
      IF(DUM.GT.1.0) DUM=1.0
      FM(IR)=DUM*QLK(I)
 1300 CONTINUE
 1310 DO 1320 I=1,NN
      IF(LR(I).GT.0) GO TO 1320
      IR=FTFM(I)
      FM(IR)=FM(IR)-RT(I)+DVAL(I)*(OLD(IR)*EDP-CS(I)*EA3)+KOD9*FQ(I)
 1320 CONTINUE
      IF(NBND.EQ.0) GO TO 1350
      DO 1340 J=1,NBND
      I=PBOUND(J)
      L=FTFM(I)
      IF(KOD6.EQ.0) GO TO 1330
      DUM=U(I)-BASE(I)
      IF(DUM.LT.0.0) DUM=0.0
      FM(L)=FM(L)+QBOUND(J)*DUM
      GO TO 1340
 1330 FM(L)=FM(L)+QBOUND(J)
 1340 CONTINUE
 1350 DO 1360 I=1,N
 1360 FM(I)=FM(I)+PVAL(I)*A3*OLD(I)
      IF(KOD4.NE.1) GO TO 1370
      WRITE(6,2120)
      WRITE(6,2460)(FM(J),J=1,N)
C
C.....SOLVE FOR NEW HEAD VALUES; RETURNED AS VECTOR 'OLD'.
C *************************************************************************
 1370 CALL SOLVE (N,FACT)
C *************************************************************************
```

```
      IT=IT+1
      STIME=STIME+DELT
      IDW=0
      MOUT=MOD(IT,IOUT)
      DO 1390 I=1,NN
      UO(I)=U(I)
      IF(LR(I).GT.0) GO TO 1380
      II=FTFM(I)
      U(I)=OLD(II)
      IF(U(I).GT.BASE(I)) GO TO 1390
      IDW=IDW+1
      IF(MOUT.EQ.0) WRITE(06,2680) I
      GO TO 1390
 1380 U(I)=UC(I)
 1390 DDDN(I)=UI(I)-U(I)
      IF(IDW.GT.0) WRITE(6,2700) IDW,IT
C
C.....UPDATE DELAYED YIELD TERMS.
      CALL CSMOD (NN,EA3,EDP)
      IF(KOD16.LE.0) GO TO 1420
C
C.....OUTPUT OF HEADS FOR NODAL HYDROGRAPHS.
      DO 1410 I=1,NSAVE
      II=PSAVE(I)
 1410 HSAVE(I)=U(II)
      WRITE(8,2130) STIME,HSAVE
 1420 IF(MOUT.NE.0) GO TO 1430
C
C.....PRINT CURRENT NODAL HEAD VALUES.
      WRITE(6,2440)
      WRITE(6,2440)
      WRITE(6,2440)
      WRITE(6,2470) STIME,UNITT,IT,DELT
      WRITE(6,2480)
      WRITE(6,2270)(I,U(I),I=1,NN)
      IF(KOD11.NE.1) GO TO 1430
C
C.....PRINT NODAL DRAWDOWN VALUES.
      WRITE(6,2670)
      WRITE(6,2270)(I,DDDN(I),I=1,NN)
 1430 IF(IDISK.EQ.0) GO TO 1450
      MDISK=MOD(IT,IDISK)
      IF(MDISK.NE.0) GO TO 1450
      IF(KOD12.NE.1) GO TO 1440
C
C.....WRITE NODAL HEAD VALUES ON DEVICE FT12F001.
      WRITE(12,2580) STIME,IT
      WRITE(12,2590)(I,U(I),I=1,NN)
 1440 IF(KOD13.NE.1) GO TO 1450
C
C.....WRITE NODAL DRAWDOWN VALUES ON DEVICE FT12F001.
      WRITE(12,2600) STIME,IT
      WRITE(12,2590)(I,DDDN(I),I=1,NN)
 1450 IF(KOD8.EQ.1) CALL SFLUX (NSN,NSS,MOUT)
```

```
C
C.....ELAPSED TIME CHECK.
      DELTP=DELT
      IF(STIME.LT.TIME) GO TO 1460
      WRITE(6,2490) STIME
      GO TO 1510
 1460 IF(IT.LT.ITMAX) GO TO 1470
      WRITE(6,2500) IT
      GO TO 1510
C
C.....INCREASE LENGTH OF TIME STEP?
 1470 IF(MOD(IT,ITCHNG).NE.0) GO TO 1480
      DELT=CHNG*DELT
 1480 IF(KOD3.EQ.0) GO TO 1500
      IF(KOD6.EQ.1) GO TO 1000
      IF(KOD8) 1510,1510,1280
 1500 MKOD=0
      IF(NBK.GT.0.OR.KOD9.EQ.1) CALL UPDATE
      IF(MOD(IT,ITRANS).EQ.0.AND.KOD6.EQ.1) MKOD=1
C
C.....DETERMINE AMOUNT OF MACHINE TIME REMAINING AND STOP IF THERE IS
C     NOT ENOUGH TO SAFELY COMPLETE ANOTHER TIME STEP.
      CALL CPTIME (K)
      CPT=FLOAT(K)*0.00002604
      DUM=CPRE-CPT
      CPRE=CPT
      IF(CPMAX.LT.DUM) CPMAX=DUM
      IF(CPT.LT.CPMAX) GO TO 1510
      IF(MKOD.EQ.1) GO TO 1000
      IF(DELT.NE.DELTP) GO TO 1200
      GO TO 1280
C
C.....OUTPUT OF FINAL VALUES.
 1510 END FILE 08
      WRITE(6,2710) CPMAX
      WRITE(6,2440)
      WRITE(6,2470) STIME,UNITT,IT,DELT
      WRITE(6,2480)
      WRITE(6,2270)(I,U(I),I=1,NN)
      DO 1520 I=1,NN
 1520 DDDN(I)=UI(I)-U(I)
      IF(KOD11.NE.1) GO TO 1530
      WRITE(6,2670)
      WRITE(6,2270)(I,DDDN(I),I=1,NN)
 1530 IF(KOD12.NE.1) GO TO 1540
      WRITE(12,2580) STIME,IT
      WRITE(12,2590)(I,U(I),I=1,NN)
 1540 IF(KOD13.NE.1) GO TO 1550
      WRITE(12,2600) STIME,IT
      WRITE(12,2590)(I,DDDN(I),I=1,NN)
 1550 END FILE 12
      IF(KOD15.NE.1) STOP
C
C.....WRITE UPDATED LISTS TO DEVICES 01 AND 03 FOR INPUT NEXT RUN.
```

```
      REWIND 01
 1560 WRITE(1) U,OLD
      WRITE(1) DDDN
      WRITE(1) UO,LR,DELT,DELTP,STIME,IT
      END FILE 01
      REWIND 03
      WRITE(3,2130) SVAL,RT,CS,EA3
      END FILE 03
      IF(KOD14.NE.1) STOP
C
C.....WRITE CONSTANT LISTS TO DEVICE FT04F001 FOR INPUT LATER.
      KOD14=0
      WRITE(4) X,Y,BASE
      WRITE(4) PTRANS,QLK,US,QBOUND,PBOUND,FTFM
      WRITE(4) MV,NSTP,DVAL,PVAL,KODD,SMAX,ALPH
      WRITE(4) IN
      WRITE(4) UI,STORAG,UPF,ITP,MOUTH,CHNG,DELTI,NN,NE,N,NBND,NBK,NSN,
     *         NSS,NEM,ITRANS,ITCHNG,TIME,ITMAX,UNITT,UNITL,IDISK,IOUT,
     *         KOD,NSAVE,PSAVE
      END FILE 04
      RETURN
C ***********************************************************************
C **************** MAIN PROGRAM INPUT-OUTPUT FORMATS.*****************
C ***********************************************************************
 2000 FORMAT(76X,A4)
 2010 FORMAT(19A4,A1,A3)
 2020 FORMAT(I4,2F10.0,2F8.0)
 2030 FORMAT(I4,2F10.0,4F8.0)
 2040 FORMAT(////,11X,15HBASE OF AQUIFER/11X,15(1H-)//11X,6(4HNODE,5X,5H
     2VALUE,5X))
 2050 FORMAT(11X,20A4)
 2060 FORMAT(4I5,2G10.0)
 2070 FORMAT(11X,'DEPLETABLE STREAM NODES',/11X,13(1H-)/,11X,'INDEX ',
     *'NODE  UP  DOWN  LEAK-FACTOR  STREAM-HEAD  AQUIFER-HEAD'/)
 2080 FORMAT(11X,4I5,3F12.2)
 2090 FORMAT(11X,'DEPLETABLE STREAMS',/11X,18(1H-),/11X,'STREAM NUMBER',
     *'  HEAD-WATER  MOUTH  IN-FLOW TO HEAD-WATER'/)
 2100 FORMAT(11X,I7,10X,I3,6X,I3,1PE20.3,10X,13A4)
 2110 FORMAT(3I5,F10.0,13A4)
 2120 FORMAT(1H0,10X,15HKNOWN VECTOR FM/11X,15(1H-)/)
 2130 FORMAT(20A4)
 2140 FORMAT(1H1,40X,'  T R I A D    ',//33X,'GROUND-WATER FLOW ANALYSIS
     1',//33X,'WITH TRIANGULAR ELEMENTS',//11X,80(1H*)////)
 2150 FORMAT(1H ,30X,'  DIFFUSION EQUATION, APRIL 1978.')
 2160 FORMAT(11X,80(1H*)//11X,20A4//11X,80(1H*)///)
 2170 FORMAT(16A4,2A2,2A1,I10)
 2180 FORMAT(70X,G10.0)
 2190 FORMAT(/////11X,'TIME PARAMETERS',/11X,15(1H-)//11X,'SIMULATION PE
     1RIOD',F39.2/11X,'INITIAL TIME STEP IN ',A4,F31.6/11X,
     2'MULTIPLIER FOR INCREASING TIME STEP',F21.3/11X,'MAXIMUM PERMITTED
     3 NUMBER OF TIME STEPS',I18/11X,'NUMBER OF TIME STEPS BETWEEN CHANG
     4ES IN DELT',I12/,11X,'NUMBER OF TIME STEPS BETWEEN UPDATES OF TRAN
     5S',I11,/11X,'NUMBER OF TIME STEPS BETWEEN PRINTED OUTPUT',I13,
     6 /11X,'NUMBER OF TIME STEPS BETWEEN OUTPUT TO DISK',I13//)
```

```
2200 FORMAT(1H1,10X,16HNODE COORDINATES/11X,16(1H-)/)
2210 FORMAT(11X,33HMULTIPLICATION FACTOR FOR X AND Y,1PE10.3/)
2220 FORMAT(1H ,11X,'LOCATION    X           LOCATION    Y'//)
2230 FORMAT(1H0,//////)
2240 FORMAT(/////,11X,'INITIAL HEAD',/11X,12(1H-)//11X,6('NODE',5X,'VAL
    1UE',5X))
2250 FORMAT(/////,11X,'NODAL STORAGE COEFFICIENT:',/11X,26(1H-)//11X,
    16('NODE',5X,'VALUE',5X))
2260 FORMAT(/////,11X,'NODAL TRANSMISSIVITY',/11X,20(1H-)//11X,6('NODE'
    1,5X,'VALUE',5X))
2270 FORMAT(/(11X,6(I4,2X,1PE10.3,3X)))
2280 FORMAT(I3,2F6.0)
2290 FORMAT(/////11X,'ELEMENT STORAGE COEFFICIENT:',/11X,28(1H-)//11X,
    1 6('ELEMENT',2X,'VALUE',5X))
2300 FORMAT(/////11X,'ELEMENT TRANSMISSIVITY:     :',/11X,23(1H-)//11X,
    1 6('ELEMENT',2X,'VALUE',5X))
2310 FORMAT(////11X,'NUMBER OF ELEMENTS READ(',I5,') DOES NOT AGREE',
    1'WITH NE=',I5)
2320 FORMAT(////10X,4('ELEMENT',3X,1H/,3(1H-),'CORNERS',3(1H-),1H/,2X))
2330 FORMAT(/(10X,4(I5,4X,3I5,3X)))
2340 FORMAT(/////11X,19HCONSTANT HEAD NODES/11X,19(1H-))
2350 FORMAT(20I4)
2360 FORMAT(11X,10(1H*),18HCONSTANT HEAD NODEI4,37HDOES NOT EXIST - EXE
    1CUTION TERMINATED10(1H*))
2370 FORMAT(11X,20I5)
2380 FORMAT(1H0,10X,10(1H*),34HNUMBER OF CONSTANT HEAD NODES READI6,33
    1HDISAGREES WITH NUMBER ANTICIPATEDI6,10(1H*))
2390 FORMAT(///11X,19HFINITE ELEMENT DATA/11X,19(1H-)/)
2400 FORMAT(1H ,10X,'NUMBER OF -  NODES',T60,I8/T22,'-  CONSTANT HEAD',
    *' NODES',T60,I8/T22,'-  A-PRIORI HEAD NODES',T60,I8/T22,'-  DEGREE
    *S OF FREEDOM',T60,I8/T22,'-  ELEMENTS',T60,I8/T22,'-  STREAM',
    *' SYSTEMS',T60,I8/T22,'-  STREAM NODES',T60,I8/T22,'-  CONSTANT',
    *' GRADIENT BOUNDARY NODES',T60,I8/T22,'-  OUTPUT HYDROGRAPHS',
    *T60,I8//)
2410 FORMAT(////,10X,7HELEMENT,I4,5X,16HSTIFFNESS MATRIX/)
2420 FORMAT (I5,1P8E15.6)
2430 FORMAT (1H ,9X,7HELEMENT,I4,5X,14HSTORAGE MATRIX/)
2440 FORMAT (//)
2450 FORMAT (/////,11X,'RT COEFFICIENT MATRIX',/11X,21(1H-)//)
2460 FORMAT (11X,1P10E12.3)
2470 FORMAT (1H ,///10X,'COMPUTED HEAD',/11X,13(1H-)//11X,'ELAPSED TIME
    1',1PE13.2,1X,A4,//11X,'TIME STEP',I16,//11X,'DELT',E21.2)
2480 FORMAT(1H ,//11X,6('NODE',5X,'VALUE',5X))
2490 FORMAT (/////11X,10(1H*),41HEXECUTION TERMINATED ON TIME-ELAPSED T
    1IME1PE12.4,1X,A4,10(1H*))
2500 FORMAT (////11X,10(1H*),42HEXECUTION TERMINATED ON TIME STEPS AT S
    1TEPI10,10(1H*))
2510 FORMAT(///11X,15HKODI PARAMETERS/11X,15(1H-)/)
2520 FORMAT(11X,19A4,A1,A3,5X,'KODD    =',I1)
2530 FORMAT(11X,19A4,A1,A3,5X,'KOD(',I2,')=',I1)
2540 FORMAT(11X,16A4,2A2,2A1,I10)
2550 FORMAT(11X,'ALL LENGTH UNITS IN ',A4/,11X,'ALL TIME UNITS IN ',A4)
2560 FORMAT(I5,G10.0)
2570 FORMAT(5X,1PE12.4,' DIVERTED AND ',E12.4,' DISCHARGED FROM CONSTAN
```

```
     *T HEAD NODE (',I3,').')
2580 FORMAT(//1X,'COMPUTED HEAD'/,'TIME=',1PE13.2,/'ITERATIONS=',I5)
2590 FORMAT(1X,I4,5X,1PE10.3)
2600 FORMAT(//1X,'COMPUTED DRAWDOWN'/,'TIME=',1PE13.2,/'ITERATIONS=',
     * I5)
2610 FORMAT(//11X,'HORIZONTAL LEFT PROFILE OF SOLUTION MATRICES = ',I8)
2620 FORMAT(//11X,'PROFILE (',I8,') EXCEEDS ARRAY DIMENSION (',I8,').')
2630 FORMAT(01X,'EQUATION ',I5/)
2640 FORMAT(9(I4,1PE10.2))
2650 FORMAT(1H1,///11X,'TRANSFORMATIONS:',/11X,15(1H-),//11X,'EXTERNAL
     *TO INTERNAL',25X,'INTERNAL TO EXTERNAL',/11X,'NODE  TYPE  INTERNAL
     * REPRESENTATION',10X,'INTERNAL REPRESENTATION  NODE',/)
2660 FORMAT(11X,I4,2X,I4,13X,I4,30X,I4,09X,I4)
2670 FORMAT(/////,11X,'COMPUTED DRAWDOWN',/11X,17(1H-),//11X,6('NODE',
     * 5X,'VALUE',5X))
2680 FORMAT(11X,'AQUIFER DEWATERED AT NODE(',I4,').')
2690 FORMAT(16X,'TOTAL SPECIFIC YIELD = ',1PE10.3,' AND ALPHA = ',
     *E10.3)
2700 FORMAT(11X,I5,' NODES DEWATERED TIME STEP (',I5,').')
2710 FORMAT(1H1,//11X,'MAXIMUM CPU FOR ONE TIMESTEP:',1PE10.3,' SEC.')
     END
```

```
C-DSET: IMPLEMENTS BOULTON DELAYED YIELD FUNCTION.
C ------------------------------------------------------------------------
      SUBROUTINE DSETI (JNND,JNEL,JNMX,DVAL,CS,IN,STORAG,U,UO,JD)
      INTEGER*2 JD(03),IN(03,JNEL)
      DIMENSION STORAG(JNMX),DVAL(JNND),CS(JNND),U(JNND),UO(JNND)
      DATA EMIN/-180.2/
      TINV=1./3.
      DO 10 I=1,JNND
      DVAL(I)=0.0
   10 CS(I)=0.0
      RETURN
      ENTRY DFORM (NN,NE,SMAX,KOD5)
      DO 70 L=1,NE
      DO 20 I=1,3
   20 JD(I)=IN(I,L)
      CALL SHAPE (L,A)
      IF(KOD5.EQ.1) GO TO 30
      STOR=STORAG(L)
      GO TO 50
   30 ST=0.0
      DO 40 I=1,3
      IR=JD(I)
   40 ST=ST+STORAG(IR)
   50 DO 60 I=1,3
      IR=JD(I)
      IF(KOD5.EQ.0) GO TO 60
      STOR=0.25*(ST+STORAG(IR))
   60 DVAL(IR)=DVAL(IR)+(SMAX-STOR)*A*TINV
   70 CONTINUE
      RETURN
      ENTRY CSMOD (NN,EA3,EDP)
      DO 80 I=1,NN
   80 CS(I)=CS(I)*EA3+(U(I)-UO(I))*EDP
      RETURN
      ENTRY EMOD (ALPH,A3,EA3,EDP)
      EA3=0.0
      EXPA=-ALPH/A3
      IF(EXPA.GT.EMIN) EA3=EXP(EXPA)
      EDP=A3*(1.0-EA3)
      RETURN
      END
```

```
C-UPDATE: INPUTS NODAL DISCHARGE AND A-PRIORI HEAD DATA PLUS CONTROLS
C         MAXIMUM LENGTH OF TIMESTEP TO SIMULATE THESE CONDITIONS.
C ------------------------------------------------------------------------
      SUBROUTINE UPDATI (JNND,FQ,UO,U,LR)
      DIMENSION FQ(JNND),UO(JNND),U(JNND),KOD(16)
      INTEGER*2 LR(JNND)
      COMMCN/AREA1/KOD,NN,NBK,NBND,MOUT,UNITT,STIME,DELT,DELT1,KODP
      RETURN
      ENTRY INITL
C
C.....READ THE STARTING TIME FOR FIRST DISCHARGE PERIOD AND IRRIGATED
C     AREA PER WELL.  IF THE AREA IS ZERO, IRRIGATION IS NOT INCLUDED
C     IN THIS DISCHARGE PERIOD.
      IF(KOD(9).EQ.1) READ(07,1030) PINC,IYEAR,APUMP
C
C.....READ THE AVERAGE WELL DISCHARGE FOR EACH NODE.
      IF(KOD(9).EQ.1) CALL DISCIN (NN)
C
C.....READ THE STARTING TIME FOR ANY 'A-PRIORI' NODES.
      IF(NBK.GT.0)    READ(05,1030) TINC
C
C.....CHECK WHETHER DISCHARGE DATA OR 'A-PRIORI' DATA WILL CHANGE AND
C     IF SO WHEN.  IF EITHER IS ALTERED PRIOR TO START OF NEXT
C     TIME STEP, DO SO AT THIS TIME.  IF EITHER IS ALTERED DURING
C     THE NEXT TIME STEP, ALTER THE LENGTH OF THE TIME STEP TO
C     ALLOW THE ALTERATION TO BE ACCOMPLISHED BETWEEN TIME STEPS.
      ENTRY UPDATE
      IF(KOD(9).NE.1) GO TO 60
      IF(NBK.EQ.0) GO TO 10
      IF(PINC.GT.TINC) GO TO 60
   10 DUM=STIME+DELT
      IF(PINC.GE.DUM) RETURN
      IF(PINC.LE.STIME) GO TO 20
      DELT=PINC-STIME
      RETURN
   20 KODP=0
      IF(APUMP.GT.0.0) KODP=1
      WRITE(6,1060) PINC,UNITT,KODP
      READ(07,1020)(FQ(I),I=1,NN)
      IF(KODP.NE.1) GO TO 40
      IF(MOUT.NE.0) GO TO 30
      WRITE(6,1050)
      WRITE(6,1040)(I,FQ(I),I=1,NN)
C
C.....IF AN IRRIGATION PERIOD, CHECK FOR REDUCED WELL DISCHARGES.
   30 CALL REDUCE (NN,KOD(5),KOD(6),PINC,UNITT,IYEAR,MOUT)
C
C.....READ TIME FOR ALTERING THIS DISCHARGE AND WHETHER THIS SUBSEQUENT
C     PERIOD INCLUDES IRRIGATION.
   40 READ(07,1030) PINC,IYEAR,APUMP
      IF(MOUT.NE.0) GO TO 50
      WRITE(6,1050)
      WRITE(6,1040)(I,FQ(I),I=1,NN)
   50 DELT=DELT1
```

```
      DUM=STIME+DELT
      IF(PINC.GE.DUM) GO TO 60
      DELT=PINC-STIME
      IF(DELT.LE.0.0) GO TO 20
   60 IF(NBK.EQ.0) RETURN
      DUM=STIME+DELT
      IF(TINC.GT.DUM) RETURN
      IF(TINC.LE.STIME) GO TO 70
      DELT=TINC-STIME
      RETURN
   70 DO 80 I=1,NN
      IF(LR(I).NE.4) GO TO 80
C
C.....READ 'A-PRIORI' NODE AND HEAD.
      READ(5,1000) J,DUM
      UO(J)=U(J)
      U(J)=DUM
      IF(I.EQ.J) GO TO 80
      WRITE(6,1010) I,J
      STOP
   80 CONTINUE
C
C.....READ TIME WHEN NEXT SEQUENCE OF 'A-PRIORI' DATA IS TO BE INPUT.
C     AND ALTER TIME STEP IF NECESSARY.
      READ(5,1030) TINC
      IF(TINC.LE.STIME) GO TO 70
      DELT=DELT1
      DUM=STIME+DELT
      IF(TINC.LT.DUM) DELT=TINC-STIME
      RETURN
C
C.....INPUT AND OUTPUT FORMATS:
 1000 FORMAT(I3,F10.0)
 1010 FORMAT(11X,'INPUT VARIABLE J =',I5,' DOES NOT AGREE WITH I =',I5)
 1020 FORMAT(20A4)
 1030 FORMAT(F20.0,I5,F20.0)
 1040 FORMAT(/(11X,6(I4,2X,1PE10.3,3X)))
 1050 FORMAT(/////,11X,'NODAL DISCHARGE ARRAY (FQ): RECHARGE (+),',
     *' DISCHARGE (-).'/,11X,56(1H-)//11X,6('NODE',5X,'VALUE',5X))
 1060 FORMAT(11X,'DISCHARGE PERIOD STARTING WITH TIME=',F10.2,1X,A4,
     *', KODP=',I1)
      END
```

```
C-SFLUX: CALCULATES NODAL FLUX TO STREAM, STREAM BASEFLOWS AND
C        DETERMINES WHETHER STREAM IS 'DRY' OR 'WET'.
C ---------------------------------------------------------------------
      SUBROUTINE SFLUXI (JNND,JSTN,JNSS,ITP,MOUTH,NSTP,LR,UPF,QLK,
     *                   US,UI,U,SFLW)
      DIMENSION UPF(JNSS),QLK(JSTN),US(JSTN),UI(JNND),U(JNND),
     *          SFLW(JSTN)
      INTEGER*2 ITP(JNSS),MOUTH(JNSS),NSTP(JSTN,3),LR(JNND)
      RETURN
      ENTRY SFLUX (NSN,NSS,MOUT)
      IF(MOUT.EQ.0) WRITE(6,1000)
      DO 10 J=1,NSN
      JJ=NSTP(J,1)
      DU=U(JJ)-US(J)
      IF(DU.LT.-1.0) DU=-1.0
   10 SFLW(J)=QLK(J)*DU
      IF(MOUT.EQ.0) WRITE(6,1010)(NSTP(J,1),SFLW(J),J=1,NSN)
      DO 90 NOS=1,NSS
      I=ITP(NOS)
      SFLW(I)=SFLW(I)+UPF(NOS)
   20 BFSUM=0.0
   30 IF(I.EQ.0) GO TO 80
      K=NSTP(I,1)
      NUP=NSTP(I,2)
      NDN=NSTP(I,3)
      IF(LR(K).LT.-2) GO TO 60
      SFLW(I)=SFLW(I)+BFSUM
      BFSUM=SFLW(I)
      IF(NUP.EQ.0) GO TO 40
      LR(K)=-6
      I=NUP
      GO TO 20
   40 IF(BFSUM.LE.0.0) GO TO 50
      LR(K)=-1
      GO TO 70
   50 SFLW(I)=0.0
      BFSUM=0.0
      LR(K)=-2
      GO TO 70
   60 LR(K)=-1
      SFLW(I)=SFLW(I)+BFSUM
      BFSUM=SFLW(I)
      GO TO 40
   70 IF(I.EQ.MOUTH(NOS)) GO TO 80
      I=NDN
      GO TO 30
   80 IF(MOUT.EQ.0) WRITE(6,1020) NOS,BFSUM
   90 CONTINUE
      IF(MOUT.NE.0) GO TO 100
      WRITE(6,1030)
      WRITE(6,1040)(NSTP(I,1),LR(NSTP(I,1)),I=1,NSN)
      WRITE(6,1050)
  100 RETURN
C
```

```
C.....INPUT-OUTPUT FORMATS:
 1000 FORMAT(1H0,///T50,'STREAM FLOW'//,11X,'NODE   STREAMFLOW'/)
 1010 FORMAT(/(11X,6(I4,2X,1PE10.3,3X)))
 1020 FORMAT(/11X,'BASEFLOW FOR STREAM  ', I3,'  EQUALS ',1PE12.4)
 1030 FORMAT(///11X,'LR=-2 IF STREAM IS DEWATERED AND LR=-1 IF STREAM IS
     * STILL LIVE.'//,11X,10(2X,'NODE',2X,'LR',2X))
 1040 FORMAT(/(11X,10(2X,I4,2X,I2,2X)))
 1050 FORMAT(///)
      END
```

```
C-AREAS: CALCULATES NODAL CONTRIBUTING AREA AND ESTIMATES STEADY
C        RECHARGE.
C ---------------------------------------------------------------------
      SUBROUTINE AREASI (JNND,JNEM,JSTN,JBND,PBOUND,QBOUND,BASE,U,UI,
     *                   QLK,US,SVAL,NSTP,LR,SE,IN,DGX,DGY,DDDN,JNMX,
     *                   PTRANS,JNEL,JD)
      REAL*4     DGX(03),DGY(03),SE(03,03)
      INTEGER*2 NSTP(JSTN,3),PBOUND(JBND),LR(JNND),JD(03),IN(3,JNEL)
      DIMENSION U(JNND),UI(JNND),QLK(JSTN),US(JSTN),SVAL(JNEM),
     1          BASE(JNND),QBOUND(JBND),PTRANS(JNMX),DDDN(JNND),KOD(16)
      COMMON/AREA1/ KOD,NN,NBK,NBND,MOUT,UNITT,STIME,DELT,DELT1
      RETURN
      ENTRY AREAS (NE,NEM,NSN)
      DO 10 I=1,NEM
   10 SVAL(I)=0.0
      DO 80 L=1,NE
      DO 20 I=1,3
   20 JD(I)=IN(I,L)
      IF(KOD(5).EQ.1) GO TO 30
      TR=PTRANS(L)
      GO TO 50
   30 TR=0.0
      DUM=1.0
      DO 40 K=1,3
      I=JD(K)
      IF(KOD(6).EQ.1) DUM=DDDN(I)
   40 TR=TR+PTRANS(I)*DUM
      TR=TR/3.0
   50 CALL SHAPE (L,A)
      DO 60 I=1,3
      K=JD(I)
      SVAL(K)=SVAL(K)+A/3.0
      DO 60 J=I,3
   60 SE(I,J)=TR*(DGX(I)*DGX(J)+DGY(I)*DGY(J))
C
C.....FILL LOWER HALF OF SE ARRAY
      DO 70 I=2,3
      I1=I-1
      DO 70 J=1,I1
   70 SE(I,J)=SE(J,I)
      DO 80 I=1,3
      JDI=JD(I)+NN
      DO 80 J=1,3
      JDJ=JD(J)
   80 SVAL(JDI)=SVAL(JDI)+SE(I,J)*U(JDJ)
      WRITE(6,1030)
      WRITE(6,1010)(I,SVAL(I+NN),I=1,NN)
      IF(NBND.LE.0) GO TO 120
      IF(KOD(6).NE.1) GO TO 100
      DO 90 I=1,NBND
      J=PBOUND(I)
      TEM=SVAL(J+NN)-QBOUND(I)*(U(J)-BASE(J))
      WRITE(6,1060) J,SVAL(J+NN),TEM
   90 SVAL(J+NN)=TEM
```

```
      GO TO 120
  100 DO 110 I=1,NBND
      J=PBOUND(I)
      TEM=SVAL(J+NN)-QBOUND(I)
      WRITE(6,1060) J,SVAL(J+NN),TEM
  110 SVAL(J+NN)=TEM
  120 IF(KOD(8).NE.1) GO TO 140
      DO 130 I=1,NSN
      J=NSTP(I,1)
      IF(LR(J).NE.-1) GO TO 130
      DH=UI(J)-US(I)
      IF(DH.LT.-1.0) DH=-1.0
      TEM=SVAL(NN+J)+QLK(I)*DH
      WRITE(6,1040) J,SVAL(NN+J),TEM
      SVAL(NN+J)=TEM
  130 CONTINUE
  140 DO 150 I=1,NN
      IF(SVAL(I).EQ.0.0) GO TO 150
      SVAL(NN+I)=SVAL(NN+I)/SVAL(I)
  150 CONTINUE
      WRITE(6,1000)
      WRITE(6,1010)(I,SVAL(I),I=1,NN)
      WRITE(6,1020)
      WRITE(6,1010)(I,SVAL(NN+I),I=1,NN)
      WRITE(11,1050)(I,SVAL(I),SVAL(NN+I),I=1,NN)
      END FILE 11
      RETURN
C
C.....INPUT-OUTPUT FORMATS:
 1000 FORMAT(/////,11X,'NODAL WEIGHTED AREA : (L**2)',/11X,27(1H-),
     *//11X,6('NODE',5X,'VALUE',5X))
 1010 FORMAT(/(11X,6(I4,2X,1PE10.3,3X)))
 1020 FORMAT(/////,11X,'ESTIMATED STEADY RECHARGE : (L/T)',/11X,33(1H-)
     *,//11X,6('NODE',5X,'VALUE',5X))
 1030 FORMAT(/////,11X,'ESTIMATED NODAL RECHARGE : (L**3/T)',/11X,
     *35(1H-),//11X,6('NODE',5X,'VALUE',5X))
 1040 FORMAT(10X,I5,10X,' STREAM NODE: RECHARGE SET FROM ',1PE12.4,
     *' TO ',E12.4)
 1050 FORMAT(I5,1P2E15.6)
 1060 FORMAT(10X,I5,10X,' BOUNDARY FLUX NODE: RECHARGE SET FROM ',
     *1PE12.4,' TO',E12.4)
      END
```

```
C-SHAPE: DETERMINES BASIS FUNCTION DERIVATIVES AND INTEGRALS.
C ------------------------------------------------------------------------
      SUBROUTINE SHAPEI (JNND,X,Y,DGX,DGY,JD)
      DIMENSION X(JNND),Y(JNND),XI(05),YI(05)
      REAL*4    DGX(03),DGY(03)
      INTEGER*2 JD(03)
      DATA TH/0.333/
      RETURN
      ENTRY SHAPE (L,A)
      XM=0.0
      YM=0.0
      DO 10 I=1,3
      II=JD(I)
      XI(I)=X(II)
      XM=XM+XI(I)
      YI(I)=Y(II)
   10 YM=YM+YI(I)
      XM=XM*TH
      YM=YM*TH
      DO 20 I=1,3
      XI(I)=XI(I)-XM
   20 YI(I)=YI(I)-YM
      XI(4)=XI(1)
      YI(4)=YI(1)
      XI(5)=XI(2)
      YI(5)=YI(2)
      A=0.0
      DO 30 I=1,3
   30 A=A+XI(I)*(YI(I+1)-YI(I+2))
      A=0.5*A
      IF(A.LE.0.0) WRITE(6,1000) L,A
      AA=0.5/(SQRT(A))
      DO 40 I=1,3
      I1=I+1
      I2=I+2
      DGX(I)=AA*(YI(I1)-YI(I2))
   40 DGY(I)=AA*(XI(I2)-XI(I1))
      RETURN
C
C.....INPUT-OUTPUT FORMATS:
 1000 FORMAT(//11X,'AREA OF ELEMENT ',I4,'= ',1PE10.3)
      END
```

```
C-FACTOR:
C ---------------------------------------------------------------------
      SUBROUTINE FACTI (JNND,JNEM,HVAL,MV)
      DIMENSION HVAL(JNEM)
      INTEGER*2 MV(JNND)
      REAL*8 SUM,TEMP
      EPSLN=1.0E-37
      RETURN
      ENTRY FACTR (N,FACT,KOD2)
C
C.....CHOLESKY FACTORIZATION OF MATRIX HVAL.  ADJUST ARRAY IN AN ATTEMPT
C      TO AVOID UNDERFLOWS.
      FACT=1./EPSLN
      ISMALL=0
      DO 70 I=1,N
      KIS=MV(I)
      KID=MV(I+1)-1-KIS
      KR=KID
      SUM=HVAL(KIS)*FACT
   10 IF(KR.LE.0) GO TO 40
      J=I-KR
      KL=KID-KR
      KIJ=KIS+KR
      TEMP=HVAL(KIJ)*FACT
      KJS=MV(J)
      KJD=MV(J+1)-1-KJS
      IF(KJD.LT.KL) KL=KJD
      IF(KL.LE.0) GO TO 30
      DO 20 K=1,KL
   20 TEMP=TEMF-HVAL(KIJ+K)*HVAL(KJS+K)
   30 HVAL(KIJ)=TEMP*HVAL(KJS)
C
C     IF(ABS(HVAL(KIJ)).LT.EPSLN.AND.HVAL(KIJ).NE.0.0) ISMALL=ISMALL+1
C
      SUM=SLM-HVAL(KIJ)**2
      KR=KR-1
      GO TO 10
   40 IF(SUM.LE.0.0) GO TO 50
      HVAL(KIS)=1./DSQRT(SUM)
      IF(KOD2.EQ.0) GO TO 70
C
C.....PRINT GLOBAL COEFFICIENT MATRIX.
   50 WRITE(6,1020) I
      II=I-KID
      J=KIS+I
   60 JJ=II+8
      IF(JJ.GT.I) JJ=I
      WRITE(6,1010)(K,HVAL(J-K),K=II,JJ)
      II=JJ+1
      IF(II.LE.I) GO TO 60
      WRITE(6,1030)
      IF(SUM.LE.0.0) GO TO 80
   70 CONTINUE
C     WRITE(6,1040) ISMALL,EPSLN
```

```
      RETURN
   80 WRITE(6,1000) I,SUM
      STOP
C
C.....INPUT-OUTPUT FORMATS:
 1000 FORMAT(//11X,'FACTORIZATION FAILS ON EQUATION ',I5,' WITH SUM = ',
     * 1PE15.6)
 1010 FORMAT(9(I4,1PE10.2))
 1020 FORMAT(01X,'EQUATION ',I5/)
 1030 FORMAT(/)
 1040 FORMAT(//11X,'FACTORIZATION SUCCESSFUL WITH ',I5,' TERMS LESS ',
     *'THAN EPSILON OF ',1PE10.3)
      END
```

```
C-SOLVE:
C ----------------------------------------------------------------------------------
      SUBROUTINE SOLVEI (JNND,JNEM,HVAL,FM,U,MV)
      DIMENSION HVAL(JNEM),FM(JNND),U(JNND)
      INTEGER*2 MV(JNND)
      REAL*8 SUM
      RETURN
      ENTRY SOLVE (N,FACT)
C
C.....FOWARD SUBSTITUTION FOR INTERMEDIATE SOLUTION AND STORE AS FM.
      DO 20 I=1,N
      SUM=FM(I)*FACT
      KI=MV(I)
      KS=KI+1
      KF=MV(I+1)-1
      KD=KI+I
      IF(KI.EQ.KF) GO TO 20
      DO 10 K=KS,KF
   10 SUM=SUM-HVAL(K)*FM(KD-K)
   20 FM(I)=SUM*HVAL(KI)
C
C.....BACK SUBSTITUTION AND STORE SOLUTION IN VECTOR U.
      DO 40 II=1,N
      I=N+1-II
      KI=MV(I)
      KD=KI+I
      KS=KI+1
      KF=MV(I+1)-1
      U(I)=FM(I)*HVAL(KI)
      IF(KF.EQ.KI) GO TO 40
      DO 30 K=KS,KF
   30 FM(KD-K)=FM(KD-K)-HVAL(K)*U(I)
   40 CONTINUE
      RETURN
      END
```

```
C-REDUCTION OF DISCHARGE IF SATURATED THICKNESS BECOMES CRITICAL.
C -----------------------------------------------------------------------
      SUBROUTINE REDUCI (JNND,DISC,U,STORAG,PTRANS,DUMI,FQ,BASE,HRS,NUM)
      DIMENSION DISC(JNND),STORAG(JNND),PTRANS(JNND),FQ(JNND),U(JNND),
     *          BASE(JNND),DUMI(JNND),HRS(JNND)
      INTEGER*2 NUM(JNND)
      DATA FACT1/158.22/,FACT2/-7.59/
      RETURN
C
C.....READ INITIAL AVERAGE WELL DISCHARGE FOR EACH NODE.
      ENTRY  DISCIN (NN)
      READ(15,1000)(DISC(I),I=1,NN)
      WRITE(14,1010)
      WRITE(14,1020)(I,DISC(I),I=1,NN)
      RETURN
C
C.....CHECK FOR REDUCED WELL DISCHARGE DUE TO LIMITING SATURATED
C     THICKNESS UTILIZING THE COOPER-JACOB APPROXIMATION TO THE
C     THEIS EQUATION.  THE EFFECT OF REDUCED SATURATED THICKNESS
C     ADJACENT TO THE WELLS IS ESTIMATED BY THE JACOB CORRECTION:
C         SA=SWT-(SWT**2)/(2*SATT); SOLVING FOR SWT.
      ENTRY REDUCE (NN,KOD5,KOD6,PINC,UNITT,IYEAR,MOUT)
      IF(KOD5.NE.1) GO TO 70
      IF(KOD6.NE.1) GO TO 80
C
C.....READ NUMBER OF WELLS ASSOCIATED WITH EACH NODE (IN TENTHS).
      READ(13,1060)(NUM(I),I=1,NN)
      DO 10 I=1,NN
   10 DUMI(I)=0.1*NUM(I)
      IF(MOUT.NE.0) GO TO 20
      WRITE(14,1070) IYEAR,PINC,UNITT
      WRITE(14,1080)(I,DUMI(I),I=1,NN)
      WRITE(14,1030)
   20 DO 60 I=1,NN
C
C.....IF NODAL DISCHARGE OR NUMBER OF WELLS IS ZERO, SKIP THIS NODE.
      IF(FQ(I).GE.0.0.OR.NUM(I).LE.0) GO TO 50
C
C.....CALCULATE AVERAGE NET DISCHARGE PER WELL AND SATURATED THICKNESS.
      FQQ=FQ(I)/DUMI(I)
      SATT=U(I)-BASE(I)
C
C.....CALCULATE TRANSMISSIVITY.
      TRANS=PTRANS(I)*SATT
      IF(TRANS.LE.0.0.OR.SATT.LE.0.0) GO TO 40
C
C.....CALCULATE COOPER-JACOB APPROXIMATION TO DRAWDOWN IN AN EQUIVALENT
C     CONFINED AQUIFER WITH NODAL AVERAGE WELL DISCHARGE.
      DUM=.0086*TRANS/STORAG(I)
      DUM=105.88*ALOG10(DUM)
      SA=DISC(I)*DUM/TRANS
      S2=0.5*SATT
C
C.....IF AQUIFER WILL BE DEWATERED BY THIS AVERAGE WELL DISCHARGE.
```

```
C     THE DISCHARGE MUST BE REDUCED, SO JUMP TO STATEMENT NUMBER 20.
      IF(SA.GT.S2) GO TO 30
C
C.....ADJUST DRAWDOWN FOR DEWATERING ADJACENT TO WELL.
      SWT=SATT-SQRT(SATT*(SATT-SA-SA))
      DUMI(I)=U(I)-SWT
      HRS(I)=FACT2*FQQ/DISC(I)
      GO TO 60
C
C.....IF AVERAGE DISCHARGE CAN NOT BE MET:
C     REDUCE DISCHARGE TO MAXIMUM ATTAINABLE UNDER GIVEN AQUIFER
C     CONDITIONS.  ADJUST NET VOLUME FLUX IF IT CAN NOT BE MET BY
C     INCREASING THE DURATION OF PUMPING TO A MAXIMUM OF 50 DAYS
C     AT THE CALCULATED MAXIMUM RATE OF DISCHARGE.
   30 Q=S2*TRANS/DUM
      TOT=-Q*FACT1
      IF(TOT.LT.FQQ) TOT=FQQ
      RATIO=100.0*TOT/FQQ
      FQ(I)=TOT*DUMI(I)
      DUMI(I)=U(I)-SATT
      HRS(I)=FACT2*TOT/Q
      IF(MOUT.NE.0) GO TO 60
C
C.....OUTPUT NODE, RATIO OF CALCULATED NET VOLUME FLUX TO INITIAL NET
C     VOLUME FLUX (AS A PERCENT), ADJUSTED DISCHARGE AND SATURATED
C     THICKNESS AT START OF PUMPING PERIOD.
      WRITE(14,1040) I,RATIO,Q,SATT,TRANS
      GO TO 60
   40 RATIO=0.0
      IF(MOUT.EQ.0) WRITE(14,1040) I,RATIO,Q,SATT,TRANS
      IF(FQ(I).LT.0.0) FQ(I)=0.0
C
C.....NODES WITHOUT DISCHARGE FROM WELLS WILL EXPERIENCE NO REDUCTION
C     OF THE DISCHARGE DUE TO LIMITING SATURATED THICKNESS.
   50 DUMI(I)=U(I)
      HRS(I)=0.0
   60 CONTINUE
      IF(MOUT.NE.0) RETURN
C
C.....OUTPUT THE ESTIMATED PUMPING WATER LEVELS.
      WRITE(14,1050)
      WRITE(14,1020)(I,DUMI(I),I=1,NN)
      WRITE(14,1110)
      WRITE(14,1020)(I,HRS(I),I=1,NN)
      RETURN
   70 WRITE(14,1090)
      RETURN
   80 WRITE(14,1100)
      RETURN
C
C.....INPUT-OUTPUT FORMATS:
 1000 FORMAT(20A4)
 1010 FORMAT(//11X,'NODAL AVERAGE INITIAL DISCHARGES (GAL./MIN.)',//11X,
     *       6(4HNODE,5X,5HVALUE,5X))
```

```
1020 FORMAT(/(11X,6(I4,2X,1PE10.3,3X)))
1030 FORMAT(/11X,'ADJUSTED DISCHARGES:                 ',/11X,'NODE ',3X,
    *'% OF INITIAL TOTAL FLUX    GPM     SATURATED THICKNESS       ',
    *'TRANSMISSIVITY (SQFT/DAY).',/)
1040 FORMAT(11X,I4,F20.1,10X,1PE10.3,5X,E10.3,10X,E10.3)
1050 FORMAT(//11X,'NODAL PUMPING HEADS',//11X,6(4HNODE,5X,5HVALUE,5X))
1060 FORMAT(40A2)
1070 FORMAT(//11X,'DISCHARGE REDUCTICN FOR YEAR',I5,/11X,'TIME =',
    *1PE10.3,A4,'.',//11X,'WELLS PER NODE:',//11X,6(4HNODE,5X,
    *5HVALUE,5X))
1080 FORMAT(/(11X,6(I4,2X,F10.1,3X)))
1090 FORMAT(/11X,'PERMEABILITY AND STORAGE INPUT FOR ELEMENTS.')
1100 FORMAT(/11X,'NOT A WATER-TABLE SIMULATION.')
1110 FORMAT(//11X,'DURATION OF PUMPING (HOURS):',//11X,
    *        6(4HNODE,5X,5HVALUE,5X))
     END
```

PGM2-NUMBERING OF NODE POINTS

Due to the large number of nodes in many simulations of this type, judicious numbering of the nodes may save considerable amounts of computer time and computer storage requirements. A routine to number the nodes efficiently for use within the simulation routine would relieve the modeler of this laborious task.

This program is based upon an article by Gibbs, Poole and Stockmeyer (1976). The nodes must initially be numbered to provide reference for the grid network and the nodal data. This numbering system must be done by the modeler, but may be arbitrary as long as there are no missing numbers. This program will then generate a new numbering system aimed at computational efficiency. This new numbering system will be used within the numerical simulation program (PGM1) by using a transformation array produced during PGM2. The initial numbering serves for any reference to the nodes except during the phase within PGM1, which forms and solves the nodal equations.

Input to this progam consists of the element indices data set and known-head node numbers. Output of the new numbering system is arranged in the initial numbering order; the new number for the node initially numbered one is the first number in the output data set. The printed output contains the new numbering system and additional information necessary for proper dimensioning of certain arrays within the numerical simulation program (PGM1).

```
C-PGM2*********************************************************************
C *   PROGRAMMED BY RALPH CADY                                            *
C *                 CONSERVATION AND SURVEY DIVISION                      *
C *                 UNIVERSITY OF NEBRASKA                                *
C *                 LINCOLN, NEBRASKA, 68588                              *
C *********************************************************************
C *********************************************************************
C *   REORDERING SCHEME FOR FINITE ELEMENT NODE SEQUENCING BASED UPON  *
C *   ARTICLE WRITTEN BY NORMAN E. GIBBS, WILLIAM G. POOLE, JR. AND    *
C *   PAUL K. STOCKMEYER OF THE DEPARTMENT OF MATHEMATICS, COLLEGE OF  *
C *   WILLIAM AND MARY, WILLIAMSBURG, VIRGINIA, 23185.  ARTICLE:       *
C *       'AN ALGORITHM FOR REDUCING THE BANDWIDTH AND PROFILE         *
C *        OF A SPARSE MATRIX', SOCIETY OF INDUSTRIAL AND APPLIED      *
C *        MATHEMATICS JOURNAL OF NUMERICAL MATHEMATICS, VOLUME 13,    *
C *        NUMBER 2, APRIL 1976, PAGES 236-250.                        *
C *                                                                    *
C *********************************************************************
C *   PROFILE DETERMINED IS THE HORIZONTAL PROFILE TO THE RIGHT OF THE *
C *   DIAGONAL.  THE PROFILE IS AFFECTED BY REVERSE NUMBERING OF THE   *
C *   NODES.  IF HORIZONTAL PROFILE TO THE LEFT OF THE DIAGONAL IS     *
C *   TO BE MINIMIZED, A CHARACTER IS INCLUDED ON THE FIRST DATA CARD. *
C *                                                                    *
C *********************************************************************
C *********************************************************************
C     THE SEQUENCE FOR THE DATA CARDS IS AS FOLLOWS:
C
C        FIRST CARD:  NUMBER OF NODES IN COL 1-4  (RIGHT JUSTIFIED).
C                     IF LEFT PROFILE IS TO BE MINIMIZED, PLACE 'LEFT'
C                     IN COLUMNS 6-9.
C
C        SECOND CARD SET:  CONSTANT HEAD NODES (I4 FORMAT STARTING IN
C                          COLUMN ONE, FINAL I4 GROUP MUST BE BLANK OR
C                          ZERO TO DENOTE END OF THIS DATA SET).
C
C        THIRD CARD SET:  ELEMENT SET, FIVE ELEMENTS PER CARD:
C                         FORMAT(20I4). FIRST I4 IS ELEMENT NUMBER,
C                         FOLLOWED BY THREE VERTICES LISTED COUNTER-
C                         CLOCKWISE.
C
C        THE FINAL CARD MUST BLANK IF THE PRECEDING ELEMENT CARD WAS
C                         ENTIRELY FILLED.
C
C *********************************************************************
      IMPLICIT INTEGER*2 (A-Z)
      INTEGER*4 JNDS,JARY
      COMMON JNDS,JARY
C *                                                                    *
C *   THE FOLLOWING CARDS MUST BE REDIMENSIONED IF SIZE OF PROBLEM     *
C *   IS TO BE ALTERED:                                                *
C *                                                                    *
      DIMENSION LR(400),LRC(400),LEVV(400),LEVU(400),LEVEL(400),
     *          NUM(400),LL(06000),PNT(06000)
      JARY=06000
      JNDS=400
```

```
C ****************************************************************************
      CALL BLDLVI (PNT,LEVEL,LL)
      CALL DEGREI (PNT,LEVU,LL)
      CALL MNPGM  (LR,LRC,LEVV,LEVU,NUM,LL,PNT)
      STOP
      END
```

C-MAIN PROGRAM

```
C ---------------------------------------------------------------------
      SUBROUTINE MNPGM (LR,LRC,LEVV,LEVU,NUM,LL,PNT)
      IMPLICIT INTEGER*2 (A-Z)
      INTEGER*4 JNDS,JARY,LFT,BAND,IDEG,ISUM
      COMMON JNDS,JARY,DMAX,DMIN,LMAX,LMIN
      DATA LFT/4HLEFT/
      DIMENSION LR(JNDS),LRC(JNDS),LEVV(JNDS),LEVU(JNDS),NUM(JNDS),
     *          LL(JARY),PNT(JARY),NCH(20),NO(100),NH(100),NL(100),
     *          LV(4,5)
      READ(5,1000) NN,BAND
      DO 10 I=1,NN
   10 LR(I)=0
   20 READ(5,1010)(NCH(I),I=1,20)
      DO 30 I=1,20
      II=NCH(I)
      IF(II.EQ.0) GO TO 40
   30 LR(II)=1
      GO TO 20
   40 LRC(1)=LR(1)
      DO 50 I=2,NN
   50 LRC(I)=LRC(I-1)+LR(I)
      END FILE 05
      NI=NN-LRC(NN)
      WRITE(6,1130) NI
      DO 60 I=1,NN
      IF(LR(I).GT.0) GO TO 60
      NINT=I-LRC(I)
      WRITE(6,1120) I,NINT
   60 CONTINUE
      DO 70 I=1,JARY
   70 PNT(I)=0
      IVOID=NI+1
   80 READ(8,1020) LV
      DO 120 II=1,5
      L=LV(1,II)
      IF(L.EQ.0) GO TO 130
      DO 120 JJ=2,4
      J=LV(JJ,II)
      IF(LR(J).GT.0) GO TO 120
      J=J-LRC(J)
      DO 110 KK=2,4
      IF(JJ.EQ.KK) GO TO 110
      I=LV(KK,II)
      IF(LR(I).GT.0) GO TO 110
      I=I-LRC(I)
      IP=J
   90 PRE=IP
      IP=PNT(IP)
      IF(IP.EQ.0) GO TO 100
      IF(LL(IP).EQ.I) GO TO 110
      GO TO 90
  100 PNT(PRE)=IVOID
      LL(IVOID)=I
```

```
      IVCID=IVCID+1
  110 CONTINUE
  120 CONTINUE
      GO TC 80
  130 IVCID=IVCID-1
      END FILE 08
      WRITE(6,1040) IVOID
C
C.....DETERMINE DEGREE OF NODE AND STORE AS LL(NODE).  ALSO DETERMINE
C      NODES WITH MINIMUM AND MAXIMUM DEGREES.
      KSTAT=1
      CALL DEGREE (NI,KSTAT)
C
C.....SET 'IV' TO NODE WITH MINIMUM DEGREE AND CREATE 'IV' LEVEL
C      STRUCTURE.
      IV=LMIN
      CALL BLDLVL (LEVV,IV,LMXV,WMXV,NI)
      WRITE(6,1050) IV,LMXV,WMXV
C
C.....CHECK FOR 'IU' WITHIN HIGHEST LEVEL OF 'IV' STRUCTURE STARTING
C      WITH LOWEST DEGREE.
  140 KODW=0
      DM2=DMIN
  150 DO 160 I=1,NI
      IF(LEVV(I).NE.LMXV) GO TO 160
      IF(LL(I).GT.DM2) GO TO 160
      IU=I
      LEVV(I)=-LEVV(I)
      DM2=LL(I)
      GO TC 170
  160 CONTINUE
      DM2=DM2+1
      IF(DM2.GT.DMAX) GO TO 210
      GO TC 150
  170 CALL BLDLVL (LEVU,IU,LMXU,WMXU,NI)
      WRITE(6,1060) IU,LMXU,WMXU
C
C.....CHECK LEVEL STRUCTURE OF 'IU'.
      IF(KODW.EQ.1) GO TO 180
      IMNU=IU
      WMN=WMXU
      KODW=1
C
C.....IF DEPTH OF 'IU' IS GREATER THAN DEPTH OF 'IV' LEVEL STRUCTURE,
C      SET 'IV' TO 'IU' AND REGENERATE 'IU' LEVELS.
  180 IF(LMXU.LT.LMXV) GO TO 200
      IF(WMXU.GE.WMXV) GO TO 200
      WMXV=WMXU
      LMXV=LMXU
      KODICH=1
      DO 190 I=1,NI
  190 LEVV(I)=LEVU(I)
      IV=IU
      WRITE(6,1050) IV,LMXV,WMXV
```

```
      GO TO 140
C
C......SELECT 'IU' AS ENDPOINT WITH MINIMUM WIDTH LEVEL STRUCTURE.
  200 IF(WMXU.GE.WMN) GO TO 150
      IMNU=IU
      WMN=WMXU
      GO TO 150
  210 DO 220 I=1,NI
      IF(LEVV(I).LT.0)LEVV(I)=-LEVV(I)
  220 CONTINUE
      IU=IMNU
      CALL BLDLVL (LEVU,IU,LMXU,WMXU,NI)
      WRITE(6,1060) IU,LMXU,WMXU
C
C.....SET UP 'IU' LEVEL STRUCTURE.
      DO 230 I=1,NI
  230 LEVU(I)=LMXU-LEVU(I)+1
      KODICH=0
      IF(LL(IU).LT.LL(IV))KODICH=1
C
C.....DELETE NODES WITH LEVU EQUAL TO LEVV.
      NELM=0
      DO 240 I=1,NI
      IF(LEVV(I).NE.LEVU(I)) GO TO 240
      NELM=NELM+1
      LL(I)=0
  240 CONTINUE
      WRITE(6,1070) NELM
C
C.....LOCATE AND RANK DISJOINT CONNECTED COMPONENTS.
  250 IDCC=0
      DO 260 I=1,NI
      IF(LL(I).NE.0) LL(I)=1
  260 CONTINUE
      II=1
      ICMAX=0
  270 DO 280 I=II,NI
      IS=I
      IF(LL(I).GT.0) GO TO 290
  280 CONTINUE
      GO TO 340
  290 IC=0
      ISTART=IS
  300 IC=IC+1
      LL(IS)=-1
  310 IS=PNT(IS)
      IF(IS.EQ.0) GO TO 320
      LIS=LL(IS)
      IF(LL(LIS).NE.1) GO TO 310
      LL(LIS)=2
      GO TO 310
  320 DO 330 I=1,NI
      IS=I
      IF(LL(IS).EQ.2) GO TO 300
```

```
  330 CONTINUE
      II=ISTART+1
      IF(IC.LE.ICMAX) GO TO 270
      ICMAX=IC
      MSTART=ISTART
      GO TO 270
  340 IF(ICMAX.EQ.0) GO TO 470
      IDCC=IDCC+1
C
C.....'NO','NH','NL' ARRAYS CONTAIN THE NUMBER OF NODES ON EACH LEVEL.
C     'NO' CONSISTS ONLY OF NODES WITH DETERMINED LEVEL (LEVV=LEVU).
C     'NH' CONSISTS OF 'NO' PLUS NODES OF DISJOINT CONNECTED COMPONENT
C     USING LEVV LEVELS.
C     'NL' CONSISTS OF 'NO' PLUS NODES OF DISJOINT CONNECTED COMPONENT
C     USING LEVU LEVELS.
      DO 350 I=1,LMXV
      NH(I)=0
      NL(I)=0
  350 NO(I)=0
C
C.....FLAG ELEMENTS OF DISJOINT CONNECTED COMPONENT BY SETTING LL=1.
      IS=MSTART
  360 LL(IS)=1
  370 IS=PNT(IS)
      IF(IS.EQ.0) GO TO 380
      LIS=LL(IS)
      IF(LL(LIS).NE.-1) GO TO 370
      LL(LIS)=2
      GO TO 370
  380 DO 390 I=1,NI
      IS=I
      IF(LL(IS).EQ.2) GO TO 360
  390 CONTINUE
C
C.....FORM NH,NL,NO ARRAYS.
      DO 400 I=1,NI
      IF(LL(I).LT.0) GO TO 400
      L1=LEVV(I)
      L2=LEVU(I)
      NH(L1)=NH(L1)+1
      NL(L2)=NL(L2)+1
      IF(LL(I).EQ.0) NO(L1)=NO(L1)+1
  400 CONTINUE
      LLM=0
      LHM=0
      DO 410 I=1,LMXV
C
C.....DETERMINE MAXIMUM WIDTH OF 'NH' AND 'NL', CONSIDERING LEVELS
C     WHICH WOULD BE CONTRIBUTED TO BY SELECTING EITHER OF THE
C     LEVEL STRUCTURES.
      IF(NH(I).GT.LHM.AND.NH(I).GT.NO(I)) LHM=NH(I)
      IF(NL(I).GT.LLM.AND.NL(I).GT.NO(I)) LLM=NL(I)
  410 CONTINUE
      IF(LHM-LLM) 430,420,450
```

```
  420 IF(WMXV.GT.WMXU) GO TO 450
  430 IF(IDCC.EQ.1) KODREV=1-KODICH
      DO 440 I=1,NI
      IF(LL(I).LE.0) GO TO 440
      LEVU(I)=LEVV(I)
      LL(I)=0
      NELM=NELM+1
  440 CONTINUE
      GO TO 470
  450 IF(IDCC.EQ.1) KODREV=KODICH
      DO 460 I=1,NI
      IF(LL(I).LE.0) GO TO 460
      LEVV(I)=LEVU(I)
      LL(I)=0
      NELM=NELM+1
  460 CONTINUE
  470 WRITE(6,1010) NELM
      DO 480 I=1,NI
      IF(LL(I).NE.0) GO TO 250
  480 CONTINUE
      WRITE(6,1090)
C
C.....RENUMBER NODES STARTING WITH 'IV' AFTER CONSIDERING DEGREES OF
C     BOTH 'IU' AND 'IV'.  WORK INCREASING LEVELS: SELECT A NODE 'W'
C     WITH MINIMUM NUMBER AND ADJACENT UNNUMBERED NODES OF SAME LEVEL.
C     NUMBER THESE BY INCREASING MODIFIED DEGREE.  IF THERE ARE ANY
C     UNNUMBERED NODES REMAINING ON THIS LEVEL, NUMBER THESE BY
C     INCREASING MODIFIED DEGREE.  SELECT A NODE 'W' WITH MINIMUM
C     NUMBER AND UNNUMBERED ADJACENT NODES ON NEXT LEVEL.  NUMBER
C     THESE BY INCREASING MODIFIED DEGREE AND SO ON.
      KSTAT=0
      CALL DEGREE (NI,KSTAT)
      IF(KODICH.EQ.0) GO TO 500
      DO 490 I=1,NI
  490 LEVV(I)=LMXV-LEVV(I)+1
      ITEMP=IV
      IV=IU
      IU=ITEMP
  500 DO 510 I=1,NI
      LEVU(I)=-1
  510 NUM(I)=0
      WRITE(6,1010) IV
      NUM(1)=IV
      LRC(IV)=1
      LEVU(IV)=0
      NUMC=1
      NUMX=1
      LEVEL=1
      LEVEF=1
      NUMS=1
      NULS=1
      NULF=1
      CALL MODDEG (IV)
  520 DO 560 I=NUMS,NUMX
```

```
      JJ=NUM(I)
      IF(LEVV(JJ).NE.LEVEL) GC TO 560
  530 IDEG=DMAX+1
      J=JJ
  540 J=PNT(J)
      IF(J.EQ.0) GO TC 550
      II=LL(J)
      IF(LEVU(II).EQ.0)      GO TO 540
      IF(LEVV(II).NE.LEVEF) GC TO 540
      ISUM=LL(II)
      IF(ISUM.GT.IDEG) GO TO 540
      IDEG=ISUM
      W=II
      GC TC 540
  550 IF(IDEG.GT.DMAX) GO TO 560
      NUMC=NUMC+1
      NUM(NUMC)=W
      LEVU(W)=0
      LRC(W)=NUMC
      CALL MCDDEG (W)
      GO TC 530
  560 CONTINUE
      IF(NUMC.EQ.NI)      GO TO 610
      IF(LEVEL.EQ.LEVEF) GO TC 570
      LEVEL=LEVEF
      NULS=NULF+1
      NULF=NUMC
      NUMS=NULS
      NUMX=NULF
      GO TO 520
  570 IF(NUMC.NE.NUMX) GO TO 590
      IDEG=DMAX+1
      DO 580 I=1,NI
      IF(LEVV(I).NE.LEVEL) GC TO 580
      IF(LEVU(I).EQ.0)      GO TO 580
      IF(LL(I).GE.IDEG) GC TO 580
      IDEG=LL(I)
      W=I
  580 CCNTINUE
      IF(IDEG.GT.DMAX) GO TO 600
      NUMC=NUMC+1
      NUM(NUMC)=W
      LEVU(W)=0
      LRC(W)=NUMC
      CALL MCDDEG (W)
  590 NUMS=NUMX+1
      NUMX=NUMC
      GO TO 520
  600 LEVEF=LEVEF+1
      NULF=NUMC
      NUMS=NULS
      NUMX=NULF
      GO TO 520
  610 IF(BANC.NE.LFT) GC TO 620
```

```
      IF(KCDREV) 650,650,630
  620 IF(KODREV) 630,630,650
  630 DO 640 I=1,NI
  640 NUM(I)=LRC(I)
      GO TO 670
  650 DO 660 I=1,NI
  660 NUM(I)=NI+1-LRC(I)
C
C.....DETERMINE THE HALF-BANDWIDTH AND PROFILE.
  670 PRFL=0
      PRFR=0
      BDWTH=0
      DO 680 I=1,NI
  680 LRC(I)=1
      DO 710 I=1,NI
      DR=1
      LI=I
  690 LI=PNT(LI)
      IF(LI.EQ.0) GO TO 700
      LIS=LL(LI)
      DJ=NUM(LIS)-NUM(I)+1
      DI=-DJ+2
      IF(DJ.GT.DR) DR=DJ
      IF(DI.GT.LRC(I)) LRC(I)=DI
      GO TO 690
  700 PRFR=PRFR+DR
      IF(DR.GT.BDWTH) BDWTH=DR
  710 CONTINUE
      DO 720 I=1,NI
  720 PRFL=PRFL+LRC(I)
      IF(BAND.EQ.LFT) GO TO 730
      WRITE(06,1080)
      GO TO 740
  730 WRITE(06,1100)
  740 WRITE(06,1110) BDWTH,PRFR,PRFL
      ICH=NI
      IVH=0
      DO 760 I=1,NN
      IF(LR(I).EQ.1) GO TO 750
      IVH=IVH+1
      LRC(I)=NUM(IVH)
      GO TO 760
  750 ICH=ICH+1
      LRC(I)=ICH
  760 CONTINUE
      WRITE(6,1030)(I,LRC(I),I=1,NN)
      WRITE(10,1010)(LRC(I),I=1,NN)
      STOP
C*************************************************************************
C.....INPUT-OUTPUT FORMATS:
C*************************************************************************
 1000 FORMAT(I4,1X,A4)
 1010 FORMAT (20I4)
 1020 FORMAT (20I4)
```

```
1030 FORMAT ('  NODE (',I3,') RELABELED TO (',I3,').')
1040 FORMAT ('  NUMBER OF LINK LIST UNITS FILLED =',I6)
1050 FORMAT ( / ,' IV SET TO',I4,' WITH LEVEL DEPTH=',I4,' AND WIDTH=',
    *        I4//)
1060 FORMAT ('  IU SET TO',I4,' WITH LEVEL DEPTH=',I4,' AND WIDTH=',
    *        I4)
1070 FORMAT (/I5,' NODES ELIMINATED ')
1080 FORMAT(1H1,'  RIGHT HORIZONTAL PROFILE MINIMIZED.')
1090 FORMAT ('  BEGIN TO RENUMBER')
1100 FORMAT(1H1,'  LEFT HORIZONTAL PROFILE MINIMIZED.')
1110 FORMAT (1H ,'  HALFBAND WIDTH =',I3/,'   RIGHT HORIZONTAL PROFILE
    * = ',I6/,'   LEFT HORIZONTAL PROFILE   = ',I6///)
1120 FORMAT(11X,I5,10X,I5)
1130 FORMAT(11X,'TOTAL FREE NODES = ',I5,//10X,'EXTERNAL',6X,
    *'INTERNAL')
     END
```

```
C-BUILD LEVEL STRUCTURES.
C -----------------------------------------------------------------------
      SUBROUTINE BLDLVI (PNT.LEVEL.LL)
      IMPLICIT INTEGER*2 (A-Z)
      INTEGER*4 JNDS.JARY
      COMMON JNDS.JARY
      DIMENSION LEVEL(JNDS).PNT(JARY).LL(JARY)
      RETURN
      ENTRY BLDLVL (LEVEL.IV.LMX.WMX.NI)
      DO 10 I=1.NI
   10 LEVEL(I)=0
      LEV=0
      LEVEL(IV)=1
      WMX=1
   20 LEV=LEV+1
      WIDTH=0
   30 DO 50 I=1.NI
      IF(LEVEL(I).NE.LEV) GO TO 50
      IS=I
   40 IS=PNT(IS)
      IF(IS.EQ.0) GO TO 50
      LIS=LL(IS)
      IF(LEVEL(LIS).NE.0) GO TO 40
      WIDTH=WIDTH+1
      LEVEL(LIS)=LEV+1
      GO TO 40
   50 CONTINUE
      IF(WIDTH.GT.WMX) WMX=WIDTH
C
C.....CHECK FOR FILLED STRUCTURE.
      DO 60 I=1.NI
      IF(LEVEL(I).EQ.0) GO TO 20
   60 CONTINUE
      LMX=LEV+1
      RETURN
      END
```

```
C-DETERMINE DEGREES
C ----------------------------------------------------------------
      SUBROUTINE DEGREI (PNT,LEVU,LL)
      IMPLICIT INTEGER*2 (A-Z)
      INTEGER*4 JNDS,JARY,ISUM
      COMMON JNDS,JARY,DMAX,DMIN,LMAX,LMIN
      DIMENSION PNT(JARY),LL(JARY),LEVU(JNDS)
      RETURN
C
C.....DETERMINE DEGREE OF NODE AND STORE AS LL(NODE).
      ENTRY DEGREE (NI,KSTAT)
      DO 20 I=1,NI
      ISUM=0
      IP=I
   10 IP=PNT(IP)
      IF(IP.EQ.0) GO TO 20
      ISUM=ISUM+1
      GO TO 10
   20 LL(I)=ISUM
      IF(KSTAT.EQ.0) RETURN
      DMAX=LL(1)
      DMIN=LL(1)
      LMIN=1
      LMAX=1
C
C.....DETERMINE MAXIMUM DEGREE (DMAX) AND MINIMUM DEGREE (DMIN).
      DO 40 I=1,NI
      IF(LL(I).GE.DMIN) GO TO 30
      LMIN=I
      DMIN=LL(I)
      GO TO 40
   30 IF(LL(I).LE.DMAX) GO TO 40
      DMAX=LL(I)
      LMAX=I
   40 CONTINUE
      WRITE(6,100) DMAX,LMAX,DMIN,LMIN
      RETURN
      ENTRY NODDEG (I)
      J=I
   50 J=PNT(J)
      IF(J.EQ.0) GO TO 60
      JJ=LL(J)
      IF(LEVU(JJ).EQ.-1) LEVU(JJ)=1
      GO TO 50
   60 J=I
   70 J=PNT(J)
      IF(J.EQ.0) RETURN
      K=LL(J)
      KJ=K
      ISUM=0
   80 KJ=PNT(KJ)
      IF(KJ.EQ.0) GO TO 90
      KK=LL(KJ)
      IF(LEVU(KK).EQ.-1) ISUM=ISUM+1
```

```
      GO TO 80
   90 LL(K)=ISUM
      GO TO 70
  100 FORMAT ('  DMAX=',I3,' FOR I=',I3,', DMIN=',I3,' FOR I=',I5//)
      END
```

PGM3-PLOTTING AREAL DISTRIBUTION OF CONTINUOUS DATA AND RESULTS

The display of data and results is important in hydrogeologic simulation since evaluation and interpretation must be an integral part of the simulation process and any subsequent decision making that may consider the results of the simulation. Most often used as a means for the areal display of a continuous variable is a map consisting of lines of equal value for the particular variable. A contour map depicting configuration of the land surface is a display of this type. Elevation of water levels, elevation of the base of the aquifer, water-level decline, thickness of aquifer, and tranmissivity are other variables that also may be displayed in this manner.

The triangular finite-element method inherently defines a continuous distribution of the dependent variable based upon the element basis functions (Pinder and Gray, 1977) and the values for the dependent variable at the nodes. Since this form of distribution is used during the simulation, a graphical display duplicating this distribution is desirable for a number of reasons: first, areas where the element configurations do not allow for a suitable representation of the variable become apparent, and second, the various schemes normally used to interpolate values between point values are not necessary with a variable that is defined continuously. This second point may greatly reduce computation time and computer costs.

If the third dimension (out of the plane of the plot) represents the value of the variable, a variable defined by the nodal values and the triangular-element basis functions is a plane passing through the three nodal values. The distribution of equal values through this plane will normally be straight lines. If the value to be mapped is identically equal to the value of the variable at only one node in the element, the value will be mapped as solely that node point. If all three element nodal values are identically the value to be mapped, the distribution of the value will be the entire element.

The program consists basically of five parts: input of element mesh, determination of adjacent elements, input of node values, generation of interconnected equal-value line segments, and plotting and labeling the interconnected line segments. Considerable time is required initially to determine the adjacent elements; however, once this has been accomplished for a particular element mesh configuration, this step may be skipped. Normally a large number of plots will be generated for one element mesh.

The input to the program consists of the element mesh configuration, the nodal coordinates, format for the nodal values, nodal values, title of the plot, scale (inches of plot per input coordinate unit), nodal multiplication factor (to change units, etc.), and the code indicating whether the adjacent element determination must be made.

Output additional to the plot consists of a printout that summarizes the plot parameters and indicates the progress of the plotting routine.

```
C-PGM3*****************************************************************
C *   PROGRAMMED BY RALPH CADY                                        *
C *                  CONSERVATION AND SURVEY DIVISION                 *
C *                  UNIVERSITY OF NEBRASKA                           *
C *                  LINCOLN, NEBRASKA, 68588                         *
C *********************************************************************
C *                                                                   *
C *   THIS PROGRAM GENERATES A PLOT IN CONTOUR MAP FORM FOR A         *
C *   CONTINUOUS VARIABLE WITH VALUES SPECIFIED FOR EACH NODE.        *
C *   THE INITIAL RUN FOR ANY ELEMENT CONFIGURATION REQUIRES          *
C *   EXTRA COMPUTATION TIME IN ORDER TO GENERATE THE ARRAY WHICH     *
C *   DESIGNATES ADJACENT ELEMENTS.  ONCE THIS ARRAY IS FORMED, IT    *
C *   IS RECALLED FOR ANY SUBSEQUENT PLOTS.                           *
C *********************************************************************
      DIMENSION TITLE(020),X(400),Y(400),VAL(400),VEX(200),VEY(200),
     *          FMT(20)
      INTEGER*2 IN(3,750),EC(3,750),ISET(3),JSET(3)
      LOGICAL*1 LIN(750),T,F
      COMMON XMIN,YMIN,XMAX
      DATA NN/400/,NE/747/,FACT/1.0/,T/.TRUE./,ISET/2,3,1/,JSET/3,1,2/
      F=.NOT.T
      NE1=NE-1
C
C.....READ IN THE CODE WHICH DESIGNATES WHETHER ADJACENT ELEMENT ARRAY
C     IS TO BE FORMED (A '0' SIGNIFIES THAT THE ARRAY EXISTS AND DOES
C     NOT NEED TO BE RECOMPUTED).
      READ(05,540) KODEC
C
C.....READ IN THE TRIANGULAR ELEMENT INDICES.
      READ(02,510) IN
      ENDFILE 02
      IF(KODEC.EQ.0) GO TO 100
C
C.....FORM THE ADJACENT ELEMENT ARRAY.
      DO 10 L=1,NE
      DO 10 J=1,3
   10 EC(J,L)=0
      DO 90 LA=1,NE1
      LO=LA+1
      DO 80 LB=LO,NE
      DO 70 I=1,3
      L1=IN(I,LA)
      DO 70 J=1,3
      L3=IN(J,LB)
      IF(L1.NE.L3) GO TO 70
      II=ISET(I)
      JJ=JSET(J)
      L2=IN(II,LA)
      L4=IN(JJ,LB)
      IF(L2.NE.L4) GO TO 70
      M=0
   20 M=M+1
      IF(M.GT.3) GO TO 30
      IF(EC(M,LA).NE.0) GO TO 20
```

```
      EC(M,LA)=LB
      GO TO 40
   30 WRITE(6,610) LA,(EC(KK,LA),KK=1,3),LB
   40 M=0
   50 M=M+1
      IF(M.GT.3) GO TO 60
      IF(EC(M,LB).NE.0) GO TO 50
      EC(M,LB)=LA
      GO TO 80
   60 WRITE(6,610) LB,(EC(KK,LB),KK=1,3),LA
      GO TO 80
   70 CONTINUE
   80 CONTINUE
   90 CONTINUE
      WRITE(06,600)(L,(EC(I,L),I=1,3),L=1,NE)
      WRITE(08,620) EC
  100 IF(KODEC.EQ.0) READ(08,620) EC
      ENDFILE 08
C
C.....READ IN THE NODAL COORDINATES.
      READ(01,500)(I,X(I),Y(I),J=1,NN)
      ENDFILE 01
C
C.....DETERMINE THE MAXIMUM AND MINIMUM VALUES OF THE NODAL COORDINATES.
      XMIN=X(1)
      XMAX=X(1)
      YMIN=Y(1)
      DO 110 I=2,NN
      IF(XMAX.LT.X(I))XMAX=X(I)
      IF(XMIN.GT.X(I))XMIN=X(I)
      IF(YMIN.GT.Y(I))YMIN=Y(I)
  110 CONTINUE
      CALL INITP (VEX,VEY,TITLE)
      CALL VXYI (X,Y,VAL)
C
C.....READ THE CARD WHICH CONTAINS THE TITLE FOR THE PLOT.
  120 READ(05,520,END=430) TITLE
C
C.....READ THE CONTOUR INTERVAL, X SCALE, Y SCALE, VALUE CONVERSION
C     FACTOR AND NUMBER OF DECIMAL POINTS.
      READ(05,530) CINT,SCX,SCY,FAC,NDP
C
C.....READ CARD WITH FORMAT FOR DATA TO BE PLOTTED.
      READ(05,520)FMT
C
C.....READ THE DATASET CONTAINING THE NODAL VALUES FOR THE VARIABLE TO
C     BE PLOTTED.
      READ(03,FMT) VAL
      IF(FAC.EQ.0.0) FAC=FACT
      IF(SCX.EQ.0.0) SCX=FACT
      IF(SCY.EQ.0.0) SCY=FACT
      VMAX=VAL(1)*FAC
      VMIN=VAL(1)*FAC
C
```

```
C.....CONVERT THE VALUES TO DESIRED UNITS AND DETERMINE THE MAXIMUM AND
C     MINIMUM VALUE FOR THE VARIABLE.
      DO 130 I=2,NN
      VAL(I)=VAL(I)*FAC
      IF(VMAX.LT.VAL(I)) VMAX=VAL(I)
      IF(VMIN.GT.VAL(I)) VMIN=VAL(I)
  130 CONTINUE
C
C.....DETERMINE THE MAXIMUM CONTOUR, MINIMUM CONTOUR AND TOTAL NUMBER OF
C     CONTOURS TO BE PLOTTED.
      CMAX=CINT*AINT(VMAX/CINT)
      CMIN=CINT*(1.0+AINT(VMIN/CINT))
      NC=1+(CMAX-CMIN)/CINT
      WRITE(06,550) NC,CMAX,CMIN,CINT,NDP
      IF(NC.LT.1) STOP
      CALL PLOTT (SCY)
      CON=CMAX
C
C.....LOCATE CONTOUR LINES ONE VALUE AT A TIME.
  140 IF(CON.LT.CMIN) GO TO 380
      DO 160 L=1,NE
      LIN(L)=F
      DO 160 I=1,3
      N=IN(I,L)
      IF(VAL(N).LE.CON) GO TO 160
      DO 150 J=1,3
      M=IN(J,L)
      IF(VAL(M).GT.CON) GO TO 150
C
C.....FLAG ELEMENTS WHICH CONTAIN THE VALUE FOR THIS CONTOUR.
      LIN(L)=T
  150 CONTINUE
  160 CONTINUE
C
C.....SELECT ELEMENT ALONG BOUNDARY.
      LAST=1
  170 DO 270 L=LAST,NE
      LAST=L
      IF(.NOT.LIN(L)) GO TO 270
      LL=L
      IF(EC(3,L).NE.0) GO TO 270
      IF(EC(2,L).NE.0) GO TO 200
      DO 190 I=1,3
      K=0
      II=IN(I,L)
      IL=EC(1,L)
      DO 180 J=1,3
      JJ=IN(J,IL)
      IF(II.EQ.JJ) K=K+1
  180 CONTINUE
      IF(K.NE.0) GO TO 190
      I1=I-1
      IF(I.EQ.1) I1=3
      I3=I+1
```

```
      IF(I.EQ.3) I3=1
      I2=I
      IF(VAL(I1).GT.CON.AND.VAL(I2).LE.CON) GO TO 310
      I1=I
      I2=I3
      IF(VAL(I1).GT.CON.AND.VAL(I2).LE.CON) GO TO 310
  190 CONTINUE
      GO TO 270
C
C.....FIND BOUNDARY EDGE.
  200 DO 220 I=1,3
      M=I
      K=0
      II=IN(I,L)
      DO 210 IJ=1,2
      IL=EC(IJ,L)
      IF(IL.EQ.0) GO TO 210
      DO 210 J=1,3
      JJ=IN(J,IL)
      IF(II.EQ.JJ) K=K+1
  210 CONTINUE
      IF(K.EQ.2) GO TO 230
  220 CONTINUE
      GO TO 270
  230 IF(M.LT.3) GO TO 240
      I1=IN(1,L)
      I2=IN(2,L)
      GO TO 260
  240 IF(M.LT.2) GO TO 250
      I1=IN(3,L)
      I2=IN(1,L)
      GO TO 260
  250 I1=IN(2,L)
      I2=IN(3,L)
  260 IF(VAL(I1).GT.CON.AND.VAL(I2).LE.CON) GO TO 310
  270 CONTINUE
      DO 280 L=1,NE
      LL=L
      IF(LIN(L)) GO TO 290
  280 CONTINUE
      CON=CON-CINT
      GO TO 140
  290 DO 300 I=1,3
      I1=IN(I,LL)
      J=I+1
      IF(I.EQ.3) J=1
      I2=IN(J,LL)
      IF(VAL(I1).GT.CON.AND.VAL(I2).LE.CON) GO TO 310
  300 CONTINUE
C
C.....SET UP COORDINATES OF CONTOUR LINE SEGMENTS.
  310 CALL VXY (CON,I1,I2,VX,VY)
      VEX(1)=VX
      VEY(1)=VY
```

```
      IVEC=1
  320 DO 330 I=1,3
      IS2=IN(I,LL)
      II=I+1
      IF(I.EQ.3) II=1
      IF2=IN(II,LL)
      IF(VAL(IS2).LE.CON.AND.VAL(IF2).GT.CON) GO TO 340
  330 CONTINUE
  340 CALL VXY (CON,IS2,IF2,VX,VY)
      IF(VX.EQ.VEX(IVEC).AND.VY.EQ.VEY(IVEC)) GO TO 350
      IVEC=IVEC+1
      VEX(IVEC)=VX
      VEY(IVEC)=VY
C
C.....RESET FLAG FOR ELEMENT JUST COMPLETED.
  350 LIN(LL)=F
C
C.....SEARCH FOR FLAGGED ELEMENT WITH SIDE IF2-IS2.
      DO 370 I=1,3
      L=EC(I,LL)
      IF(L.EQ.0) GO TO 370
      IF(.NOT.LIN(L)) GO TO 370
      DO 360 K=1,3
      IS=IN(K,L)
      KK=K+1
      IF(K.EQ.3) KK=1
      IF=IN(KK,L)
      IF(IS.NE.IF2.OR.IF.NE.IS2) GO TO 360
      LL=L
      GO TO 320
  360 CONTINUE
  370 CONTINUE
      IF(IVEC.EQ.1) GO TO 170
C
C.....PLOT LINE SEGMENTS WITH CONTOUR VALUE LABEL.
      CALL PLOTVE (CON,IVEC,SCX,SCY,NDP)
      GO TO 170
  380 IB=0
C
C.....PLOT CONTINUOUS BOUNDARY.
  390 READ(04,580,END=400) BOUNDX,BOUNDY
      IB=IB+1
      VEX(IB)=BOUNDX
      VEY(IB)=BOUNDY
      GO TO 390
  400 IB=IB+1
      VEX(IB)=VEX(1)
      VEY(IB)=VEY(1)
      REWIND 04
      CALL BNDPLT (IB,SCX,SCY)
C
C.....PLOT LINE SEGMENT BOUNDARY.
  410 READ(07,590,END=420)(VEX(I),VEY(I),I=1,2)
      CALL BNDPLT (2,SCX,SCY)
```

```
      GO TO 410
  420 XX=(XMAX-XMIN)*SCX+05.0
      ENDFILE 03
C
C.....END OF THIS MAP, RESET ORIGIN.
      CALL ENDPLT (XX)
      REWIND 07
      WRITE(06,560)
      GO TO 120
C
C.....END OF JOB.
  430 CALL ENDJOB
      WRITE(06,570)
C
C.....FORMATS
  500 FORMAT(I4,2F10.6)
  510 FORMAT(5(4X,3I4))
  520 FORMAT(20A4)
  530 FORMAT(4F10.0,I10)
  540 FORMAT(I1)
  550 FORMAT(11X,'NUMBER OF CONTOURS = ',I10,/11X,'MAXIMUM CONTOUR     =
     *',1PE10.3,/11X,'MINIMUM CONTOUR     = ',E10.3,/11X,'CONTOUR INTERVA
     *L    = ',E10.3,/11X,'CALCOMP # OF D.P.  = ',I10)
  560 FORMAT(/11X,'PLOT COMPLETED.')
  570 FORMAT(/11X,'PLOTTING COMPLETED.')
  580 FORMAT(6X,A4,1X,A4)
  590 FORMAT(4X,4(1X,A4))
  600 FORMAT(5(5X,I3,2X,I4,I4,I4))
  610 FORMAT(5X,I5,3X,3I4,I8)
  620 FORMAT(40A2)
      STOP
      END
```

```
C-VXY: CALCULATES THE LOCATION OF A PARTICULAR VALUE OF THE VARIABLE
C       ON THE ELEMENT SIDE BETWEEN NODES I1 AND I2.
C ------------------------------------------------------------------
      SUBROUTINE VXYI (X,Y,VAL)
      DIMENSION X(400),Y(400),VAL(400)
      COMMON XMIN,YMIN,XMAX
      RETURN
      ENTRY VXY (CON,I1,I2,VX,VY)
      VX=(CON*(X(I1)-X(I2))+X(I2)*VAL(I1)-X(I1)*VAL(I2))/(VAL(I1)-
     *     VAL(I2))
      VY=(CON*(Y(I1)-Y(I2))+Y(I2)*VAL(I1)-Y(I1)*VAL(I2))/(VAL(I1)-
     *     VAL(I2))
      RETURN
      END
```

```
C-INITIATES AND CALLS THE COMMANDS FOR THE ACTUAL PLOTTING.
C -----------------------------------------------------------------
      SUBROUTINE INITP (VX,VY,TITLE)
      DIMENSION VX(200),VY(200),TITLE(20),IBUF(2500)
      COMMON XMIN,YMIN,XMAX
      DATA BLK/4H    /
      CALL PLOTS (IBUF(1),10000)
      RETURN
      ENTRY PLOTVE (CON,ITOT,SCX,SCY,NDP)
      WRITE(6,500) CON,ITOT
      DO 10 I=1,ITOT
      VX(I)=(VX(I)-XMIN)*SCX+2.0
   10 VY(I)=(VY(I)-YMIN)*SCY+1.0
      I=1
   20 I=I+1
      IF(I.GT.ITOT) GO TO 30
      DY=VY(I)-VY(1)
      DX=VX(I)-VX(1)
      IF(DX.EQ.0.0.AND.DY.EQ.0.0) GO TO 20
      ANG=ATAN2(DY,DX)
      ANG=ANG*180.0/3.1416
      IF(ANG.GT.+90.0) ANG=ANG-180.
      IF(ANG.LT.-90.0) ANG=ANG+180.
      GO TO 40
   30 ANG=90.0
   40 CALL NUMBER (VX(1),VY(1),0.14,CON,ANG,NDP)
      CALL PLOT (VX(1),VY(1),3)
      DO 50 I=2,ITOT
   50 CALL PLOT (VX(I),VY(I),2)
      RETURN
      ENTRY BNDPLT (IB,SCX,SCY)
      DO 60 I=1,IB
      VX(I)=(VX(I)-XMIN)*SCX+2.0
   60 VY(I)=(VY(I)-YMIN)*SCY+1.0
      CALL PLOT (VX(1),VY(1),3)
      DO 70 I=2,IB
   70 CALL PLOT (VX(I),VY(I),2)
      RETURN
      ENTRY PLOTT (SCY)
      DO 80 I=1,20
      N=I
      IF(TITLE(I).EQ.BLK) GO TO 90
   80 CONTINUE
   90 N=4*N
      HN=0.07*AINT(SCY/0.05)
      CALL SYMBOL (1.0,1.0,HN,TITLE,90.0,N)
      RETURN
C
C.....END PLOT AND RESET ORIGIN.
      ENTRY ENDPLT (XX)
      CALL PLOT (XX,0.0,-3)
      RETURN
C
C.....END OF JOB.  CALL FOR END OF PLOTTING.
```

```
      ENTRY ENDJOB
      CALL PLOT (0.0,0.0,999)
      RETURN
C
C.....INPUT-OUTPUT FORMATS:
  500 FORMAT(/11X,'CONTOUR =',F10.3,' WITH ITOT = ',I10)
      END
```

## PGM4-APPORTIONMENT OF IRRIGATION WELLS TO ADJACENT NODES

Many simulation routines in groundwater hydrology were developed initially for the purpose of local site investigations. One example of this might be simulation of a proposed well-field design. In such simulations the grid of nodes is constructed so that wells are represented by certain nodes and other node points are located between adjacent wells. The smaller the spacing between nodes, the more accurate the representation of water levels. However, if a simulation is to encompass an area as large as and with as many wells as the Upper Big Blue NRD, expanding the methods used for site investigations becomes impractical because the number of wells is so large.

Well discharges must be apportioned to specific nodes (equations). The finite-element method's basis functions may be used to accomplish this apportionment. Since at any point located within an element, the basis functions sum to one, the product of the basis function for a node and the well discharge is a consistent means for dividing the well stresses between the nodes. The following FORTRAN IV program uses the finite-element mesh and the basis functions to apportion the wells to the nodes. The basis functions are also used to determine whether a well is located inside or outside of any particular element. The program calculates the number of wells for any node for any year and the average well discharge associated with any node. For a discussion of the well-discharge values, see section "Determination of Historic Discharge, Recharge and Return Flow."

```
C-PGM4*****************************************************************
C *    PROGRAMMED BY RALPH CADY                                        *
C *                  CONSERVATION AND SURVEY DIVISION                  *
C *                  UNIVERSITY OF NEBRASKA                            *
C *                  LINCOLN, NEBRASKA, 68588                          *
C *******************************************************************
C
C *******************************************************************
C *                                                                  *
C *    PROGRAM FOR LOCATING A POINT WITH GIVEN GLOBAL COORDINATES    *
C *    IN THE ELEMENT GRID USING THE TRIANGULAR ELEMENT AREA         *
C *    COORDINATES.  THE WELL IS THEN APPORTIONED TO THE NODES       *
C *    BY THE AREA COORDINATE BASIS FUNCTIONS.                       *
C *                                                                  *
C *******************************************************************
C
C
C.....MAIN PROGRAM FOR INITIATING OBJECT TIME DIMENSION STATEMENTS
C
C
C +---------------THE ENCLOSED CARDS ARE PROBLEM DEPENDENT-------------+
C
      DIMENSION Y(400),X(400),XMN(750),YMN(750),XMX(750),YMX(750),
     1          WNN(400,27),DNUM(400),DSUM(400)
      INTEGER*2 IN(03,750)
      JYRS=27
      JNEL=750
      JNND=400
C
C
      CALL MNPROG (JNEL,JNND,IN,X,Y,XMN,XMX,YMN,YMX,JYRS,WNN,DNUM,DSUM)
C
      STOP
      END
```

```
C-MAIN PROGRAM:
C ------------------------------------------------------------------
      SUBROUTINE MNPROG (JNEL,JNND,IN,X,Y,XMN,XMX,YMN,YMX,JYRS,WNN,DNUM,
     *                   DSUM)
C
      DIMENSION X(JNND),Y(JNND),XMN(JNEL),XMX(JNEL),YMN(JNEL),YMX(JNEL),
     1          WNN(JNND,JYRS),DNUM(JNND),DSUM(JNND)
      INTEGER*2 IN(03,JNEL),YD,MD,LV(4,5)
      DATA NDATE/-1/,NTOT/0/,NOUT/0/,NEL/0/,WELLS/0.0/,PAVG/860./,
     *     C1/147.0/,C2/0.707/,NBLK/1H /,NTREP/0/
C
C.....READ NUMBER OF ELEMENTS, NUMBER OF NODES, INITIAL YEAR
C     AND FINAL YEAR.
      READ(5,590) NE,NN,IPREV,IEND
C
C.....READ NODAL COORDINATES.
      READ(8,510)(J,X(J),Y(J),K=1,NN)
      IYRS=IEND-IPREV+1
C
C.....READ ELEMENT CONFIGURATIONS.
   10 READ(07,580) LV
      DO 20 I=1,5
      L=LV(1,I)
      IF(L.LE.0) GO TO 30
      NEL=NEL+1
      DO 20 J=1,3
   20 IN(J,L)=LV(J+1,I)
      GO TO 10
   30 IF(NEL.EQ.NE) GO TO 40
      WRITE(6,600) NEL,NE
      STOP
   40 DO 50 I=1,NN
      DNUM(I)=0.0
      DSUM(I)=0.0
      DO 50 J=1,IYRS
   50 WNN(I,J)=0.0
C
C.....DETERMINE MAXIMUM AND MINIMUM COORDINATES FOR EACH ELEMENT.
      DO 70 L=1,NE
      I=IN(1,L)
      XMAX=X(I)
      XMIN=X(I)
      YMAX=Y(I)
      YMIN=Y(I)
      DO 60 J=2,3
      I=IN(J,L)
      IF(X(I).GT.XMAX) XMAX=X(I)
      IF(X(I).LT.XMIN) XMIN=X(I)
      IF(Y(I).GT.YMAX) YMAX=Y(I)
      IF(Y(I).LT.YMIN) YMIN=Y(I)
   60 CONTINUE
      XMN(L)=XMIN
      XMX(L)=XMAX
      YMN(L)=YMIN
```

```
      YMX(L)=YMAX
   70 WRITE(6,610) L,(IN(I,L),I=1,03),XMN(L),XMX(L),YMN(L),YMX(L)
C
C.....BEGIN LOCATING WELLS WITHIN ELEMENTS.
   80 NDATE=NDATE+1
   90 L=0
C
C.....READ WELL COORDINATES, INITIAL YEAR, AND REGISTERED DISCHARGE.
      READ(2,620,END=160) XW,YW,YD,REG,NREP
      NTOT=NTOT+1
      IYW=YD+1900
      IF(IYW.GT.IEND) GO TO 80
      IF(IYW.LT.IPREV) IYW=IPREV
  100 LL=L+1
      IF(LL.GT.NE) GO TO 120
      DO 110 IL=LL,NE
      L=IL
      IF(XW.LT.XMN(L)) GO TO 110
      IF(XW.GT.XMX(L)) GO TO 110
      IF(YW.LT.YMN(L)) GO TO 110
      IF(YW.GT.YMX(L)) GO TO 110
      GO TO 130
  110 CONTINUE
  120 NOUT=NOUT+1
      GO TO 90
  130 CONTINUE
      I=IN(1,L)
      J=IN(2,L)
      K=IN(3,L)
C
C.....CALCULATE AREA COORDINATES FOR WELL LOCATION.  IF ANY OF THE AREA
C     COORDINATES IS LESS THAN ZERO, THE WELL IS NOT LOCATED WITHIN THAT
C     ELEMENT.  A2 IS THE INVERSE OF THE ELEMENT AREA.  AI, AJ AND AK
C     ARE THE AREA COORDINATES.  DNUM AND DSUM ARE USED TO DETERMINE THE
C     AVERAGE ESTIMATED WELL DISCHARGE FOR THE NODE.
      A2=X(I)*(Y(J)-Y(K))+X(J)*(Y(K)-Y(I))+X(K)*(Y(I)-Y(J))
      IF(A2.LE.0.0) WRITE(6,540) L
      A2=1.0/A2
      AI=(XW*(Y(J)-Y(K))+X(J)*(Y(K)-YW)+X(K)*(YW-Y(J)))*A2
      IF(AI.LT.0.0) GO TO 100
      AJ=(X(I)*(YW-Y(K))+XW*(Y(K)-Y(I))+X(K)*(Y(I)-YW))*A2
      IF(AJ.LT.0.0) GO TO 100
      AK=(X(I)*(Y(J)-YW)+X(J)*(YW-Y(I))+XW*(Y(I)-Y(J)))*A2
      IF(AK.LT.0.0) GO TO 100
      IIY=IYW-IPREV+1
C
C.....ADJUST REGISTERED WELL DISCHARGE USING RELATIONSHIP FROM
C     WATER-USE DATA.  ACCUMULATE THE DISCHARGES ON A NODE BY NODE
C     BASIS IN ORDER TO ESTIMATE AN AVERAGE WELL DISCHARGE ASSOCIATED
C     WITH EACH NODE.
C
      IF(REG.LE.0.0) GO TO 140
      OBS=C1+C2*REG
      DNUM(I)=DNUM(I)+AI
```

```
      DSUM(I)=DSUM(I)+AI*OBS
      DNUM(J)=DNUM(J)+AJ
      DSUM(J)=DSUM(J)+AJ*OBS
      DNUM(K)=DNUM(K)+AK
      DSUM(K)=DSUM(K)+AK*OBS
C
C.....IF THIS IS A REPLACEMENT WELL, DO NOT ADD THIS TO NODAL TOTALS.
  140 IF(NREP.EQ.NBLK) GO TO 150
      NTREP=NTREP+1
      GO TO 90
  150 WNN(I,IIY)=WNN(I,IIY)+AI
      WNN(J,IIY)=WNN(J,IIY)+AJ
      WNN(K,IIY)=WNN(K,IIY)+AK
      GO TO 90
  160 DNTOT=0.0
      DSTOT=0.0
      DO 190 J=1,NN
      DNTOT=DNTOT+DNUM(J)
      DSTOT=DSTOT+DSUM(J)
      DAVG=0.0
      IF(DNUM(J).GT.0.0) DAVG=DSUM(J)/DNUM(J)
      DSUM(J)=DAVG
      DO 170 I=2,IYRS
  170 WNN(J,I)=WNN(J,I)+WNN(J,I-1)
C
C.....ROUND CUMULATIVE NUMBER OF WELLS PER NODE TO THE NEAREST TENTH.
      DO 180 I=1,IYRS
      IW=WNN(J,I)*10
      DEL=IW
      DEL=WNN(J,I)*10-DEL
      IF(DEL.GT.0.5)IW=IW+1
  180 WNN(J,I)=IW*0.1
  190 WELLS=WELLS+WNN(J,IYRS)
      IS=IPREV-1901
C
C.....WRITE CUMULATIVE TENTHS OF WELLS PER NODE FOR EACH YEAR TO
C     UNIT FT10F001.
      DO 220 I=1,IYRS
      IC=0
      IS=IS+1
      IST=1
      IFI=25
  200 IC=IC+1
      DO 210 II=IST,IFI
      IJ=II-IST+1
  210 IN(1,IJ)=WNN(II,I)*10
      WRITE(10,630) IS,IC,(IN(1,K),K=1,IJ)
      IST=IFI+1
      IFI=IFI+25
      IF(IST.GT.NN) GO TO 220
      IF(IFI.GT.NN) IFI=NN
      GO TO 200
  220 CONTINUE
      NTOTD=DNTOT
```

```
      NLOC=WELLS
      DAVG=DSTOT/DNTOT
      WRITE(6,550)(I,WNN(I,IYRS),I=1,400)
      WRITE(6,500) NTOT,NLOC,NOUT,NDATE,NTREP,NTOTD,DAVG
C
C.....WRITE AVERAGE ESTIMATED WELL DISCHARGE FOR EACH NODE
C     TO UNIT FT03F001.
      WRITE(03,570) DSUM
      WRITE(06,560)
      WRITE(06,550)(I,DSUM(I),I=1,NN)
      RETURN
C
C.....INPUT AND OUTPUT FORMATS:
  500 FORMAT(1H1,'NUMBER OF WELLS READ =',I5,/' TOTAL WELLS IN NRD=',
     *I5,/,' TOTAL OUT OF NRD =',I5,/' TOTAL LATER THAN 1977 =',I5,
     */' TOTAL REPLACEMENT WELLS =',I5,/' TOTAL WITH REG. DISC. =',
     *I5,/' AVG. DISCHARGE =',F10.0)
  510 FORMAT (I4,2F10.0)
  520 FORMAT (1H0,10X,17HINVALID CHARACTER,2X,A1,5X,7HELEMENT,I5)
  530 FORMAT (1H0,10X,25HINCOMPLETE DATA - ELEMENT,I5)
  540 FORMAT (1H0,10X,'AREA OF ELEMENT ',I3,' NEGATIVE OR ZERO')
  550 FORMAT (8(I5,F10.1))
  560 FORMAT(11X,'AVERAGE ESTIMATED WELL DISCHARGE FOR EACH NODE.',//)
  570 FORMAT(20A4)
  580 FORMAT(20I4)
  590 FORMAT(5I5)
  600 FORMAT(11X,'NUMBER OF ELEMENTS READ=',I5,'WHILE NE=',I5)
  610 FORMAT(I4,5X,3I4,10X,4(2X,F10.7))
  620 FORMAT(2A4,I2,A4,A1)
  630 FORMAT(I2,I3,25A2)
      END
```

## PGM5-GENERATION OF NATURAL RECHARGE, IRRIGATION WATER USE, AND RETURN FLOW ESTIMATES

As mentioned in the main body of the report, the data collected in the NRD by Eugene K. Steele, Jr., of the U.S. Geological Survey served as a basis for estimating the quantity of ground-water pumped for irrigation. These estimates are used in this program to generate nodal values for irrigation water use. Duration of pumping was determined from June, July, and August precipitation. Well-capacity values determined in PGM4 were used to estimate the total amount of water pumped for irrigation by a well associated with that particular node. Potential recharge was estimated from monthly precipitation data and mean monthly evapotranspiration estimates. Potential irrigation return flow was estimated in a manner similar to the estimate of potential recharge with the addition of the estimated applied irrigation water. An option is included which will allow the irrigation discharges to be determined by allocation rather than historical usage.

```
C-PGM5*************************************************************************
C *    PROGRAMMED BY RALPH CADY                                               *
C *                    CONSERVATION AND SURVEY DIVISION                       *
C *                    UNIVERSITY OF NEBRASKA                                 *
C *                    LINCOLN, NEBRASKA, 68588                               *
  *************************************************************************
  *                                                                           *
C *    THIS PROGRAM GENERATES RECHARGE FROM PRECIPITATION,                    *
C *    IRRIGATION WELL DISCHARGE, MAXIMUM IRRIGATION RETURN-FLOW              *
C *    BASED UPON PRECIPITATION, AVERAGE POTENTIAL EVAPOTRANSPIRATION,        *
C *    ROW CROP CONSUMPTIVE USE AND WATER USE DATA.  AN ALLOCATION            *
C *    SCHEME MAY BE SIMULATED BY SELECTING '1' AS THE VALUE FOR              *
C *    'KODA' FOLLOWED BY THE INCHES ALLOCATED FOR THE SEASON.                *
  *                                                                           *
  *************************************************************************
      DIMENSION PCP(16,12)
      CONV=1./365.25
      IYEAR=0
C
C.....READ THE TOTAL NUMBER OF NODES, NUMBER OF PRECIP STATIONS,
C     ALLOCATION CODE AND INCHES ALLOCATED FOR EACH PUMPING SEASON.
      READ(05,1000) NN,NPS,KODA,AMT
      WRITE(6,1002) NN,NPS,KODA,AMT
      END FILE 05
      AMT=AMT*CONV
      CALL FORMI (NN,NPS,CONV,KODA)
      CALL PREI (NPS,CONV)
   10 READ(03,1001,END=60) PINC,IY,KS,KF,APUMP
      IF(PINC.EQ.0.0) GO TO 10
   20 IF(IY.EQ.IYEAR) GO TO 30
      CALL PRECIP (IYEAR,IY,PCP)
      GO TO 20
   30 CALL ZERO (APUMP)
      KDK=KF-KS+1
      IF(KDK.LT.0) KDK=13-KS+KF
      AMTM=AMT/KDK
      K=0
      KY=KS
   40 IF(KY.LE.12) GO TO 50
      KY=1
      CALL PRECIP (IYEAR,IY,PCP)
   50 CALL FORM (KY,PCP,APUMP,AMTM)
      K=K+1
      KY=KY+1
      IF(K.LT.KDK) GO TO 40
      CALL QOUT (KDK,IY,PINC,APUMP)
      GO TO 10
   60 STOP
 1000 FORMAT(3I5,F10.0)
 1001 FORMAT(F8.1,A4,I3,2X,I3,F10.1)
 1002 FORMAT(11X,'NUMBER OF NODES :',T60,I10,/11X,'NUMBER OF PRECIP',
     *' STATIONS :',T60,I10,/11X,'CODE FOR ALLOCATION (1) OR HISTORIC ',
     *'TENDANCY (0):',T60,I10,/11X,'INCHES OF ALLOCATION :',T60,F10.2)
      END
```

```
C-PRECIPITATION.
C -------------------------------------------------------------------
      SUBROUTINE PREI (NPS,CON)
      DIMENSION PAVG(16,12),PCP(16,12)
      KOD=0
      CONV=CON
      READ(02,1000)(IM,J,(PAVG(J,K),K=1,12),I=1,NPS)
      DO 10 K=1,12
      DO 10 J=1,NPS
   10 PAVG(J,K)=CONV*PAVG(J,K)
      RETURN
      ENTRY PRECIP (IYEAR,IY,PCP)
      IF(KOD.EQ.1) GO TO 40
      READ(02,1000,END=30) (IYEAR,J,(PCP(J,K),K=1,12),I=1,NPS)
      DO 20 K=1,12
      DO 20 J=1,NPS
   20 PCP(J,K)=CONV*PCP(J,K)
      RETURN
   30 KOD=1
   40 IYEAR=IY
      DO 50 K=1,12
      DO 50 J=1,NPS
   50 PCP(J,K)=PAVG(J,K)
      RETURN
 1000 FORMAT(A4,I3,1X,12F6.2)
      END
```

```
C-FORM RECHARGE,DISCHARGE & RETURN FLOW.
C ------------------------------------------------------------------
      SUBROUTINE FORMI (NN,NPS,CONV,KODA)
      DIMENSION PCP(16,12),FQ(400),REPRE(400),REPUM(400),W(12),
     *          C1(2),C2(2),C3(2),DISC(400)
      INTEGER*2 NW(400),JSTA(400)
      DATA W/.12,.24,.66,1.31,2.14,4.20,6.82,6.68,3.82,1.25,.42,.18/,
     *     F1/2.0/,F2/10.0/,SL/0.65/,C1/26.27,25.74/,C2/139.8,136.9/,
     *     C3/998.5,862.0/,DFACT/6.3248/
   10 READ(08,1004) DISC
C
C.....DFACT CONVERTS GPM. TO  CFD/DAY.
      DO 20 I=1,NN
   20 DISC(I)=DISC(I)*DFACT
      CONP=CONV**(1.-SL)
      F1=F1*CONV
      F2=F2*CONV
      DO 30 I=1,12
   30 W(I)=W(I)*CONV
C
C.....READ STATION NUMBERS FOR EACH NODE.
      READ(01,1000) (J,JSTA(J),I=1,NN)
      END FILE 01
      RETURN
      ENTRY ZERO (APUMP)
      DO 40 I=1,NN
      NW(I)=0
      FQ(I)=0.0
      REPRE(I)=0.0
   40 REPUM(I)=0.0
      IF(APUMP.NE.0.0) READ(04,1001) NW
      RETURN
      ENTRY FORM (K,PCP,APUMP,AMT)
      DO 80 I=1,NN
      JS=JSTA(I)
      T1=PCP(JS,K)
   50 IF(T1.GT.F1)T1=F1
      T2=PCP(JS,K)
      IF(T2.GT.F2)T2=F2
      T2=T2-F1
      IF(T2.LT.0.0)T2=0.0
      T1=T1+CONP*T2**SL
   60 P1=T1-W(K)
      IF(P1.LT.0.0) P1=0.0
      IF(APUMP.LE.0.0) GO TO 80
      IF(K.NE.7.AND.K.NE.8) GO TO 80
      IF(KODA.EQ.1) GO TO 70
      KK=K-6
      PUMP=C1(KK)-C2(KK)*PCP(JS,6)-C3(KK)*PCP(JS,K)
      IF(PUMP.LT.0.0) PUMP=0.0
C
C.....DISCHARGE (CFD) IS EQUAL TO DAYS TIMES CFD/DAY.
      PUMP=DISC(I)*PUMP
      AMT=PUMP/APUMP
```

```
   70 P1P=AMT+T1-W(K)-P1
      IF(P1P.LT.0.0) P1P=0.0
      FQ(I)=FQ(I)+AMT*APUMP
      REPUM(I)=REPUM(I)+P1P
   80 REPRE(I)=REPRE(I)+P1
      RETURN
      ENTRY QOUT (KDK,IY,PINC,APUMP)
      FACT=1./KDK
      FSUM=0.0
      RPRE=0.0
      RPUM=0.0
      DO 90 I=1,NN
      FQ(I)=FQ(I)*FACT
      REPRE(I)=REPRE(I)*FACT
      REPUM(I)=REPUM(I)*FACT
      FSUM=FSUM+FQ(I)
      RPRE=RPRE+REPRE(I)
   90 RPUM=RPUM+REPUM(I)
      WRITE(07,1002) IY,PINC,APUMP
      WRITE(07,1003)(I,REPRE(I),REPUM(I),FQ(I),NW(I),I=1,NN)
      FSUM=FSUM*PINC/NN
      RPRE=RPRE*PINC*12/NN
      RPUM=RPUM*PINC*12/NN
      WRITE(06,1005) IY,PINC,APUMP,RPRE,RPUM,FSUM
      RETURN
 1000 FORMAT(I3,42X,I5)
 1001 FORMAT(05X,25A2)
 1002 FORMAT(1X,3A4)
 1003 FORMAT(I4,3A4,2X,A2)
 1004 FORMAT(20A4)
 1005 FORMAT(11X,'YEAR:',A4,1P5E10.3)
      END
```

PGM6-FORMATION OF NET FLUXES

The simulation program requires that recharge, discharges, and return flows be added algebraically to form a net discharge for each node. Flux in addition to that calculated during PGM5 may be incorporated during this program. These additional fluxes may include municipal or industrial use, artificial recharge, or similar fluxes that may be estimated without knowing the head in the aquifer. Flux components sensitive to head must be incorporated into the numerical simulation program. Aquifer-stream flux and constant-gradient boundary flux are calculated within the simulation program for this reason. The added flux may be constant (steady) or vary with time. In the event that discharge periods and added flux periods do not coincide (for the time variant case), additional net discharge periods will be formed in order to merge the discharge and added flux periods. The resultant output data set is compatible with the input requirements of the numerical simulation program. Printed output consists of summary information for the discharge periods and for any added flux periods.

```
C-PGM6*********************************************************************
C *    PROGRAMMED BY RALPH CADY                                            *
C *                  CONSERVATION AND SURVEY DIVISION                      *
C *                  UNIVERSITY OF NEBRASKA                                *
C *                  LINCOLN, NEBRASKA, 68588                              *
C *********************************************************************
C
C *********************************************************************
C *                                                                      *
C *    THIS PROGRAM GENERATES THE NET DISCHARGE VALUES FOR EACH NODE     *
C *    BASED UPON INPUT VALUES FOR RECHARGE, IRRIGATION WITHDRAWAL,      *
C *    IRRIGATION RETURN FLOW AND ANY ADDITIONAL FLUX FOR THAT NODE.     *
C *    RECHARGE AND RETURN FLOW ARE DELAYED UNTIL THE FOLLOWING PERIOD   *
C *    OF DISCHARGE IF THE DURATION OF THE PERIOD IS LESS THAN 100 DAYS.*
C *    ADDITIONAL FLUXES MAY BE INCLUDED IN THE NET FLUX BY SPECIFYING   *
C *    THE NODE NUMBER AND THE FLUX (DISCHARGE IS NEGATIVE) ONE NODE AT *
C *    A TIME.  THE FLUX MAY BE STEADY OR VARY WITH TIME DEPENDING UPON *
C *    THE OPTION SPECIFIED.  THE DURATION OF THE PERIOD FOR THE ADDED   *
C *    FLUXES IS REQUIRED IF STEADY FLUX IS NOT SPECIFIED.               *
C *                                                                      *
C *                                                                      *
C *                                                                      *
C *    INPUT CARDS: FIRST CARD-  RIGHT JUSTIFY THE NUMBER OF NODES IN    *
C *                              COLUMNS 1-4.  IF THE DISCHARGE VALUES   *
C *                              FOR THE CONSTANT HEAD NODES ARE NOT     *
C *                              DESIRED ENTER A '1' IN COLUMN 8.        *
C *                              THE CONSTANT HEAD NODE DISCHARGES ARE   *
C *                              NOT REQUIRED FOR INPUT TO THE ROUTINE   *
C *                              WHICH PERFORMS THE SIMULATION.          *
C *                 SECOND SET-  CONSTANT HEAD NODES.  ONE BLANK FIELD   *
C *                              MUST BE INCLUDED TO END THIS SET.       *
C *                              A BLANK CARD MAY REPLACE THIS SET IF    *
C *                              THERE IS NOT A '1' IN COLUMN 8 OF       *
C *                              THE FIRST CARD.                         *
C *                 THIRD SET  - NEEDED IF THERE WILL BE ADDED FLUXES.   *
C *                              SPECIFY 'YES ' IN COLUMNS 1-4 IF THE    *
C *                              ADDED FLUXES WILL NOT VARY WITH TIME.   *
C *                              SPECIFY 'NO  ' IN COLUMNS 1-4 IF THE    *
C *                              ADDED FLUXES WILL VARY WITH TIME.       *
C *                 FORTH SET -  THE FIRST CARD WILL CONTAIN 'TIME=' IN *
C *                              COLUMNS 1-5 FOLLOWED BY THE DURATION    *
C *                              OF THE FLUXES IN DAYS LOCATED IN        *
C *                              COLUMNS 11-25.  THE REMAINDER OF THE    *
C *                              SET CONSISTS OF THE NODE CARDS.  THE    *
C *                              NODE NUMBER IS LOCATED RIGHT JUSTIFIED *
C *                              IN COLUMNS 6-10 FOLLOWED BY THE ADDED   *
C *                              FLUX (DISCHARGE NEGATIVE) IN COLUMNS    *
C *                              11-25.                                  *
C *                                                                      *
C *                                                                      *
C *    IF THERE ARE ADDITIONAL PERIODS OF ADDED FLUX, REPEAT THE FORTH   *
C *    CARD SET UNTIL THE LAST FLUX PERIOD IS SPECIFIED.                 *
C *********************************************************************
C
      DIMENSION PK(400),ANODE(400),FQ(400),DL(400),DSET(400),FQ0(400)
      INTEGER*2 NUM(400),LR(400),LRT(20)
```

```
      COMMON NN,TFLX,TDUR,KODR,KODS,TTOT
      DATA PTIME/0.0/,AREA/0.0/,CONV/12.0/,KODC/0/
      TTOT=0.0
      READ(5,200) NN,IDUM
      IF(IDUM.EQ.1) KODC=1
      WRITE(06,210) NN,KODC
      DO 10 I=1,NN
      DSET(I)=0.0
   10 LR(I)=0
C
C.....READ IN THE CONSTANT HEAD NODES.
   20 READ(5,200) LRT
      DO 30 I=1,20
      II=LRT(I)
      IF(II.LE.0) GO TO 40
      IF(KODC.EQ.1) LR(II)=1
   30 CONTINUE
      GO TO 20
C
C.....INITIALIZE ADDED FLUX PARAMETERS IF ANY.
   40 CALL ADFLXI (DSET)
C
C.....READ NODAL CONTRIBUTING AREAS AND RECHARGE FACTORS.
   50 READ(01,220)(I,ANODE(I),PK(I),J=1,NN)
      END FILE 01
      DO 60 I=1,NN
      FQ(I)=0.0
      IF(LR(I).NE.0) GO TO 60
      AREA=AREA+ANODE(I)
   60 CONTINUE
C
C.....READ SIMULATION YEAR, DURATION OF FLUX PERIOD AND IRRIGATED AREA
C     PER WELL IF THIS IS AN IRRIGATION PERIOD.
   70 READ(02,230,END=110) IYR,TDIS,APUMP
      TQE=TTCT+TDIS
      KODP=0
      IF(APUMP.GT.0.0) KODP=1
      RESUM=0.0
      RFSUM=0.0
      WELSUM=0.0
      NWELLS=0
      DO 80 I=1,NN
C
C.....READ NODAL FLUX PARAMETERS (COMPUTED DURING PGM5).
      READ(02,250) J,REPRE,REPUM,DISCH,NUM(J)
      IF(LR(J).NE.0) GO TO 80
      NWELLS=NWELLS+NUM(J)
      WELLS=DISCH*NUM(J)*0.1
      WELSUM=WELSUM+WELLS
      FQ(J)=-WELLS+DL(J)*PTIME/TDIS+DSET(J)
      DL(J)=PK(J)*REPRE*ANODE(J)
      RESUM=RESUM+DL(J)
      DUM=PK(J)*REPUM*APUMP*NUM(J)*0.1
      DL(J)=DL(J)+DUM
```

```
      RFSUM=RFSUM+DUM
C
C.....IF TIME PERIOD IS LESS THAN 100 DAYS DELAY RECHARGE AND
C      IRRIGATION RETURN FLOW.
      IF(TDIS.LT.100.0) GO TO 80
      FQ(J)=FQ(J)+DL(J)
      DL(J)=0.0
   80 CONTINUE
      NWELLS=NWELLS*0.10
      AWELLS=NWELLS*APUMP
      PUSUM=WELSUM
      IF(AWELLS.EQ.0.) AWELLS=1.00
      WELSUM=TDIS*CONV*WELSUM/AWELLS
      REFLOW=TDIS*CONV*RFSUM/AWELLS
      REINF=TDIS*CONV*RESUM/AREA
      WRITE(6,260) IYR,TDIS,RESUM,REINF,RFSUM,REFLOW,PUSUM,WELSUM,NWELLS
      PTIME=TDIS
C
C.....READ ADDITIONAL FLUXES IF NECESSARY.
   90 IF(TTOT.EQ.TFLX) CALL ADFLUX
      TMIN=TQE
      TDUM=TFLX+TDUR
C
C.....ADJUST DURATION OF PERIOD IF NECESSARY FOR ADDED FLUXES.
      IF(TMIN.GT.TDUM.AND.KODR.NE.1) TMIN=TDUM
      TIME=TMIN-TTOT
      DO 100 I=1,NN
  100 FQ0(I)=FQ(I)+DSET(I)
      WRITE(3,240) TTOT,IYR,APUMP
      WRITE(3,270) FQ0
      IF(KODP.EQ.1) WRITE(4,280) NUM
      TTOT=TTOT+TIME
C
C.....IF PERIOD WAS ADJUSTED FOR ADDED FLUXES, CREATE ADDITIONAL
C      PERIODS TO COMPLETE THE DURATION OF THE DISCHARGE PERIOD.
      IF(TTOT.LT.TQE) GO TO 90
      GO TO 70
  110 WRITE(3,240) TTOT,IYR,APUMP
      STOP
C
C.....INPUT AND OUTPUT FORMATS:
  200 FORMAT(20I4)
  210 FORMAT(1H1,10X,'NUMBER OF NODES:',I5,/11X,'KODC IS SET TO ',I1,
     *////42X,'TOTAL',/11X,'PUMPING    TOTAL     AVERAGE    ',
     *'RETURN    AVERAGE    TOTAL',/6X,'YEAR  PERIOD   RECHARGE  ',
     *'RECHARGE    FLOW     RETURN    PUMPAGE    AVERAGE',
     *5X,'TOTAL NUMBER OF WELLS',/23X,'RATE',16X,
     *'RATE',6X,'FLOW',6X,'RATE',4X,'PUMPAGE',/12X,'(DAYS)',
     *3(5X,'(CFD)      (IN.)'))
  220 FORMAT (I4,52X,2E10.3)
  230 FORMAT(I5,2A4)
  240 FORMAT(F20.4,I5,F20.1)
  250 FORMAT(I4,3A4,2X,A2)
  260 FORMAT(5X,I4,1P7E10.3,I20)
```

```
270 FORMAT(20A4)
280 FORMAT(40A2)
290 FORMAT(I5,E10.0)
    END
```

```
C-ADDITIONAL FLUXES.
      SUBROUTINE ADFLXI (DSET)
      DIMENSION DSET(NN)
      COMMON NN,TFLX,TDUR,KODR,KODS,TTOT
      DATA YES/4HYES /,TCHAR/4HTIME/
      KODR=0
      TFLX=0.0
      KODS=0
      READ(5,100,END=30) STEADY
      IF(STEADY.EQ.YES) KODS=1
      WRITE(7,110) STEADY,KODS
      IF(KODS) 10,10,20
   10 READ(5,120) DUM,IDUM,TDUR
   20 RETURN
   30 KODR=1
      RETURN
      ENTRY ADFLUX
      IF(KODR.EQ.1) RETURN
      DO 40 I=1,NN
   40 DSET(I)=0.0
      DTOT=0.0
      ITOT=0
   50 READ(5,120,END=70) DUM,IDUM,VALUE
      IF(DUM.NE.TCHAR) GO TO 60
      IF(KODS.EQ.1) GO TO 50
      TFLX=TFLX+TDUR
      TDUR=VALUE
      WRITE(7,130) TFLX,ITOT,DTOT
      WRITE(7,140) TDUR
      IF(TFLX.LE.TTOT) GO TO 50
      RETURN
   60 DSET(IDUM)=VALUE
      ITOT=ITOT+1
      DTOT=DTOT+VALUE
      GO TO 50
   70 WRITE(7,130) TFLX,ITOT,DTOT
      KODR=1
      RETURN
C
C.....INPUT AND OUTPUT FORMATS:
  100 FORMAT(A4)
  110 FORMAT(/11X,'STEADY FLUX? ',A4,', KODS =',I2)
  120 FORMAT(A4,1X,I5,G15.0)
  130 FORMAT(11X,'STARTING TIME =',1PE10.3,' WITH ',I4,' NODES WITH A ',
     *'CUMULATIVE RECHARGE = ',E10.3)
  140 FORMAT(11X,'DURATION OF PERIOD = ',1PE10.3)
      END
```

PGM7-SIMULATION OF PROJECTED IRRIGATION DEVELOPMENT

The purpose of this program is to generate the array containing the number of irrigation wells associated with each node based upon various development constraints. Input consists of the historical array of wells per node, contributing area for each node, starting and ending years for output of number of wells per node, maximum allowable well density, well develoment rate, and maximum level of development over the present level. If the simulation time period includes years with historical data, those data are included in the output array. Development commences with the earliest year without historical data. The latest historical data are used as the base for the development process. For each successive year, the development rate is multiplied by the nodal areas to obtain the number of new wells each year for each node. This number is added to the number of wells for each node. In the event that the number of wells for any node exceeds the product of the maximum well density and the nodal area, the number of wells is reduced to the product and the node shut off from further development. The other restraint requires that the number of wells associated with any node not exceed the product of the latest historical number of wells and the maximum development level multiplier. This is necessary in order to restrict the development which would otherwise be forced in areas that historically have been shown to be incapable of supporting high-capacity irrigation wells.

Output from this program is used as input to PGM5 for generating the recharge, discharge, and return-flow values.

```
C-PGM7*************************************************************************
C *    PROGRAMMED BY RALPH CADY                                               *
C *                  CONSERVATION AND SURVEY DIVISION                         *
C *                  UNIVERSITY OF NEBRASKA                                   *
C *                  LINCOLN, NEBRASKA, 68588                                 *
C *****************************************************************************
C *****************************************************************************
C *                                                                           *
C *    THIS PROGRAM GENERATES THE ARRAY WHICH CONTAINS THE NUMBER OF          *
C *    IRRIGATION WELLS ASSOCIATED WITH EACH NODE FOR VARIOUS                 *
C *    HYPOTHETICAL DEVELOPMENT SCHEMES.  INPUT CONSISTS OF THE               *
C *    HISTORICAL IRRIGATION WELL DATA, DEVELOPMENT RATE, MAXIMUM             *
C *    WELL DENSITY AND MAXIMUM DEVELOPMENT LEVEL OVER PRESENT LEVEL.         *
C *                                                                           *
C *****************************************************************************
      DIMENSION AREA(400)
      INTEGER*2 NW(400),NO(400)
      DATA NN/400/,KODW/0/,ATOT/0.0/
      CONVA=1./(5280.*5280.)
C
C.....READ STARTING YEAR, ENDING YEAR, MAXIMUM DENSITY (WELLS/SECTION),
C     WELL DEVELOPMENT RATE (WELLS/SECTION/YEAR), AND THE MAXIMUM
C     DEVELOPMENT OVER THE PRESENT FOR EACH NODE (DIMENSIONLESS).
      READ(05,1005) IYS,IYF,DMAX,DINC,DEVM
      READ(04,1004) AREA
      DO 10 I=1,NN
   10 ATOT=ATOT+AREA(I)
      ATOT=ATOT*CONVA
      WRITE(6,1010) IYS,IYF,DMAX,DINC,DEVM,ATOT
      DINC=DINC*CONVA*10.
      DMAX=DMAX*CONVA*10.
      DO 90 IYR=IYS,IYF
      NSUM=0
      IF(KODW.EQ.1) GO TO 50
   20 READ(03,1001,END=30) IYW,(NW(I),I=1,25)
      READ(03,1003)(NW(I),I=26,NN)
      IYW=1900+IYW
      IF(IYW.NE.IYR) GO TO 20
      GO TO 70
   30 KODW=1
      DO 40 I=1,NN
   40 NO(I)=NW(I)*DEVM
   50 DO 60 I=1,NN
      WMAX=DMAX*AREA(I)
      IF(NO(I).LT.WMAX) WMAX=NO(I)
      NW(I)=NW(I)+DINC*AREA(I)
      IF(NW(I).GT.WMAX) NW(I)=WMAX
   60 CONTINUE
   70 DO 80 I=1,NN
   80 NSUM=NSUM+NW(I)
      NSUM=0.1*NSUM
      AVGN=NSUM/ATOT
      WRITE(6,1006) IYR,NSUM,AVGN
   90 WRITE(7,1003) NW
```

```
      STOP
1001  FORMAT(I2,3X,25A2)
1003  FORMAT(5X,25A2)
1004  FORMAT(56X,E10.3)
1005  FORMAT(2I5,3F10.0)
1006  FORMAT(11X,I4,' TOTAL # OF WELLS = ',I10,' WHICH = ',F10.2,
     *' WELLS PER SECTION.')
1010  FORMAT(11X,'STARTING YEAR: ',15X,I5,/11X,'ENDING YEAR:',18X,I5,
     */11X,'MAXIMUM WELL DENSITY:',4X,1PE10.3,' WELLS/SECTION.',/11X,
     *'WELL DEVELOPMENT RATE:',3X,E10.3,' WELLS/SECTION/YR.',/11X,
     *'MAXIMUM DEVELOPMENT',/11X,'OVER HISTORIC:',11X,E10.3,/11X,
     *'MODELED AREA:',12X,E10.3,/////)
      END
```

APPENDIX 2

Simulated Water levels

Simulated water-level declines from Spring 1975

Simulated elevation of water level above base of aquifer

Elevation of water level above base of aquifer, Spring 1975

SIMULATED WATER LEVELS

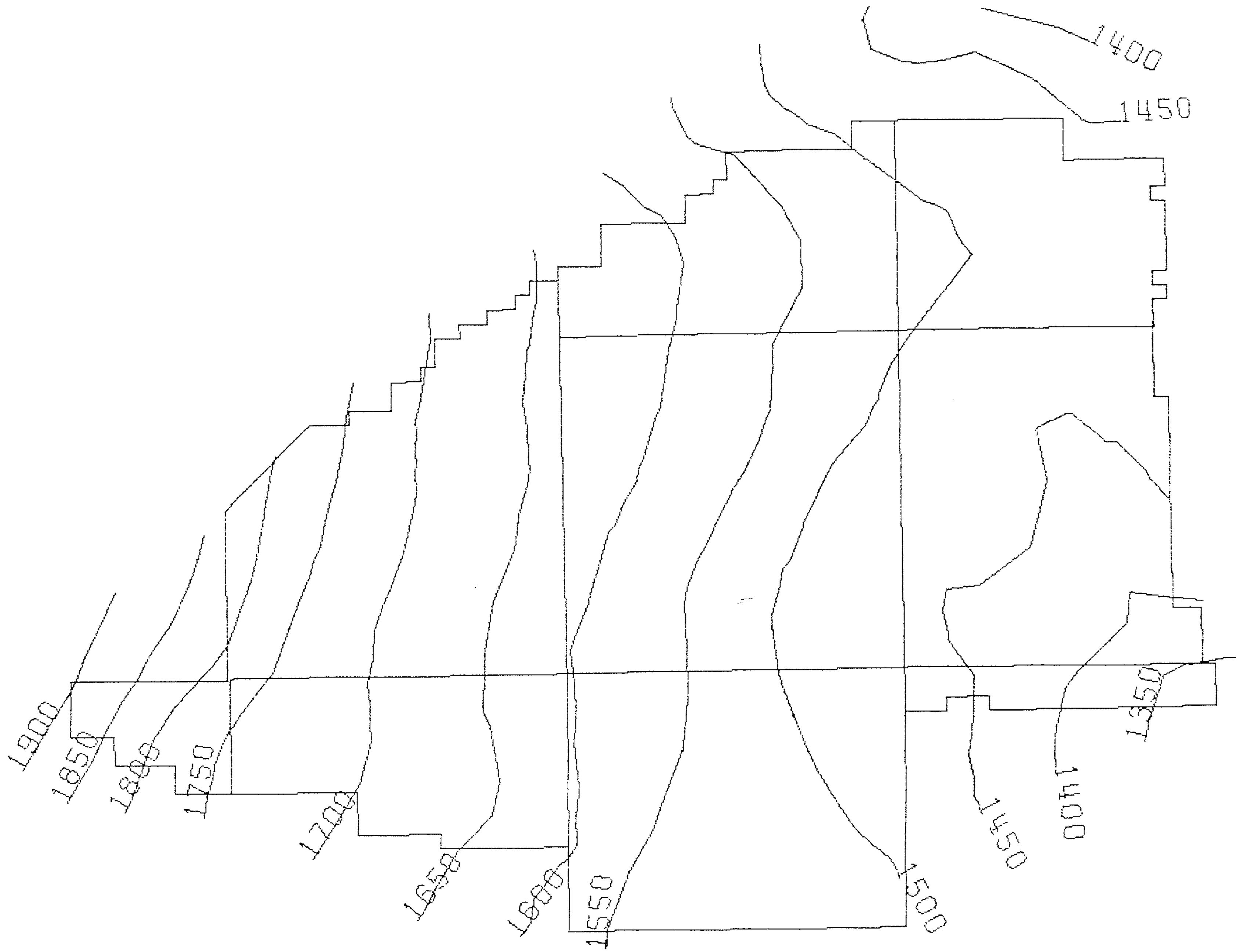



Figure 15a.--Simulated water levels, in feet, under scheme 01 conditions, Spring 1990.

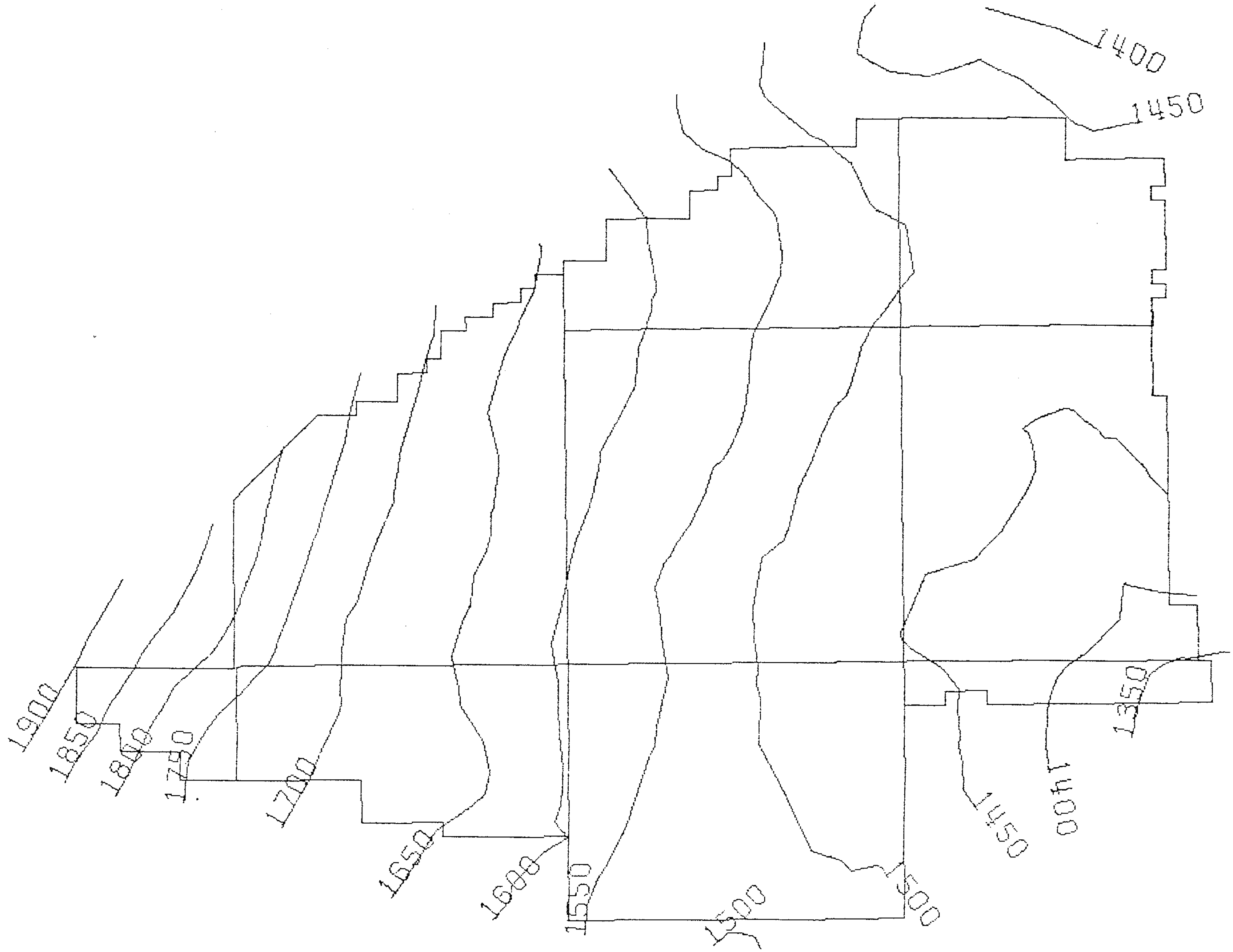



Figure 15b.--Simulated water levels, in feet, under scheme 01 conditions, Spring 2000.

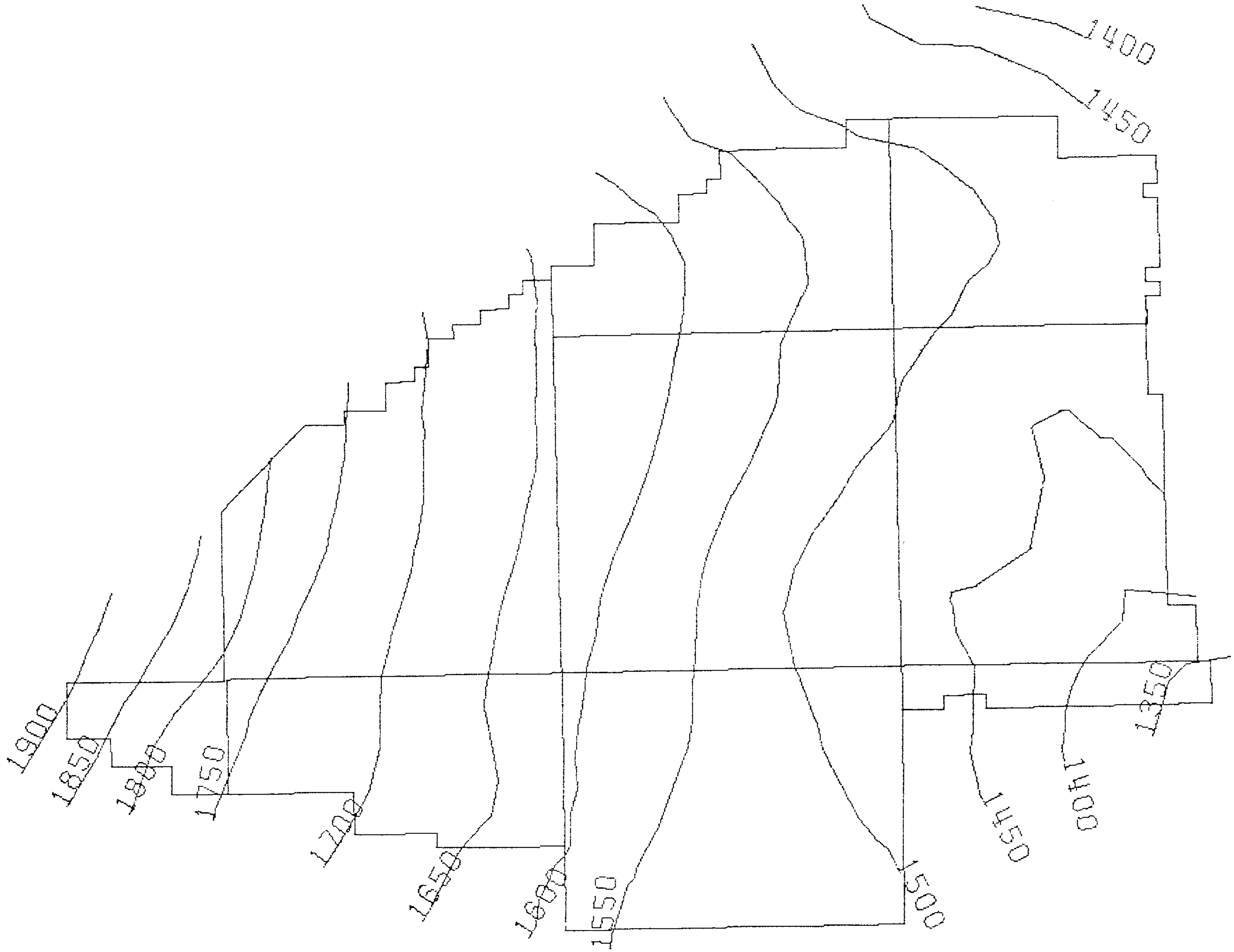



Figure 15c.--Simulated water levels, in feet, under scheme 02 conditions, Spring 2000.

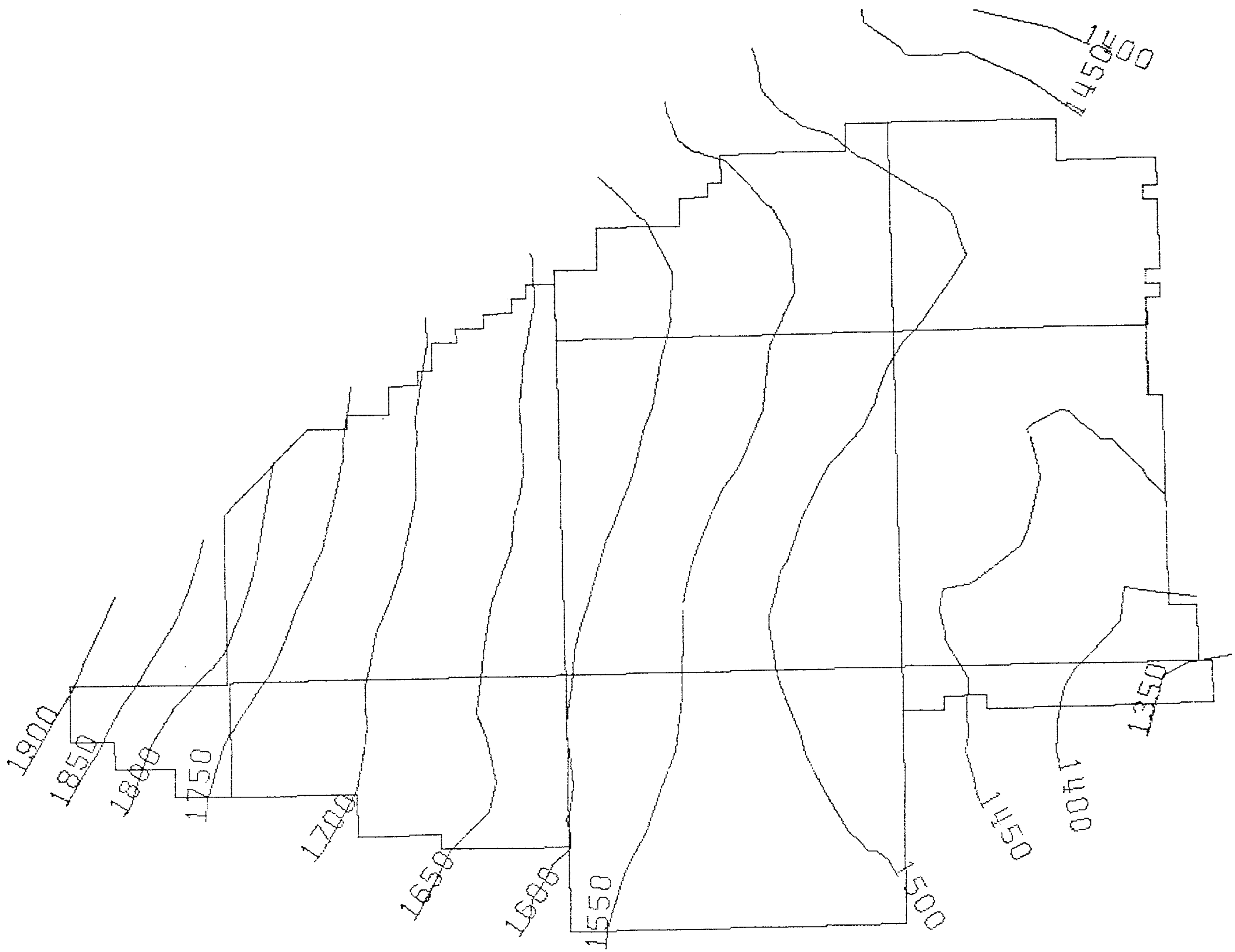



Figure 15d.--Simulated water levels, in feet, under scheme 03 conditions, Spring 2000.

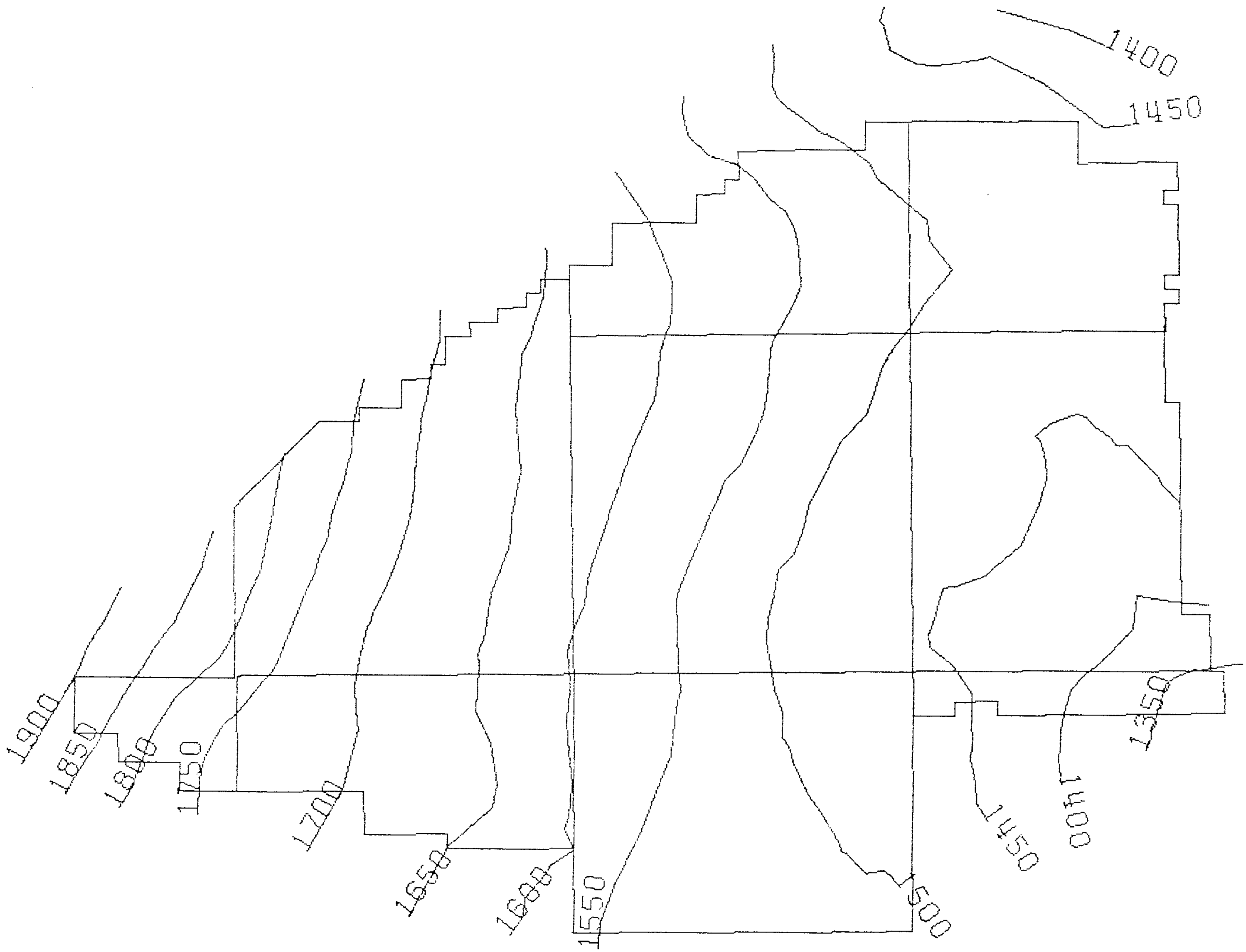



Figure 15e.--Simulated water levels, in feet, under scheme 04 conditions, Spring 2000.

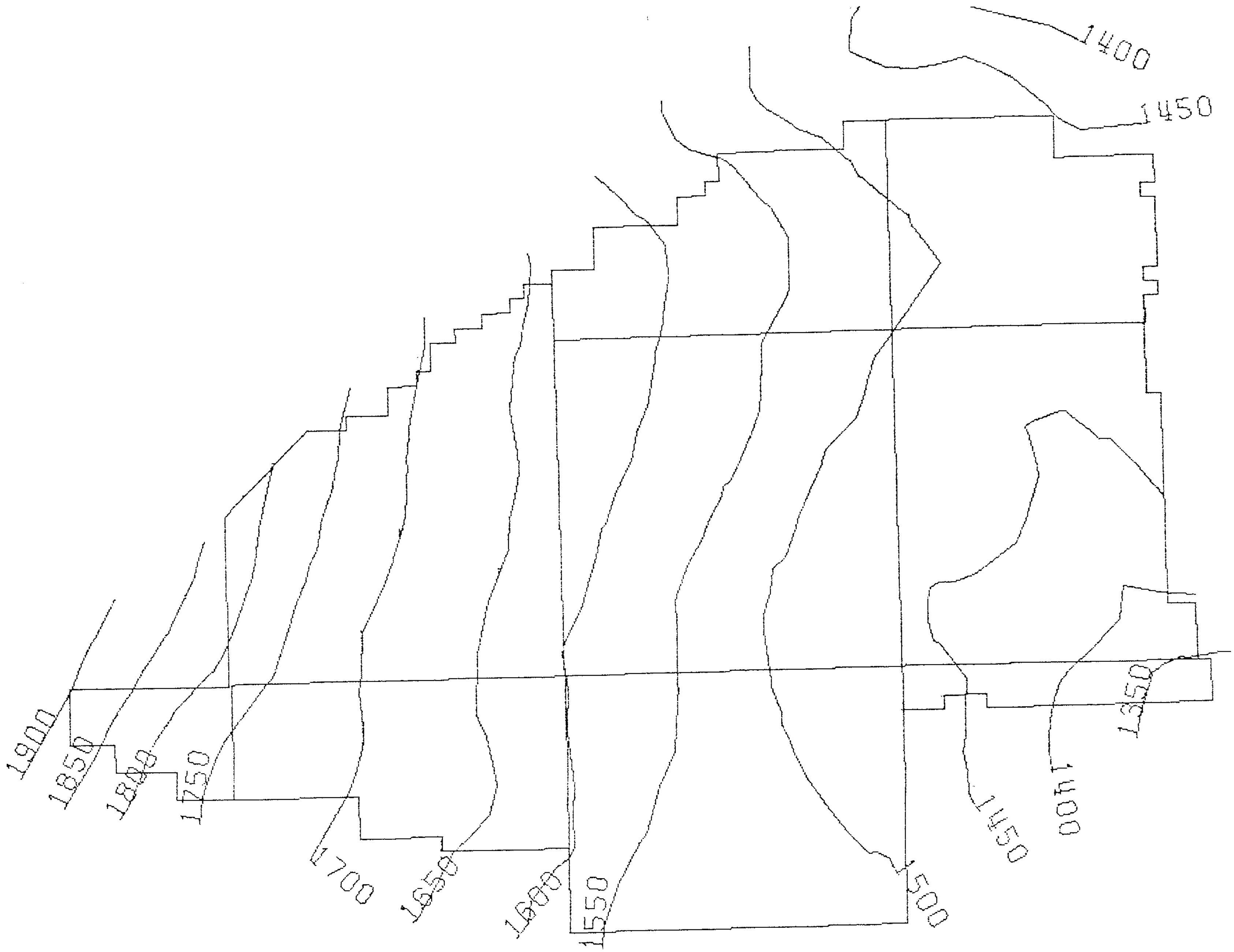



Figure 15f.--Simulated water levels, in feet, under scheme 05 conditions, Spring 1990.


Figure 15g.--Simulated water levels, in feet, under scheme 05 conditions, Spring 2000.

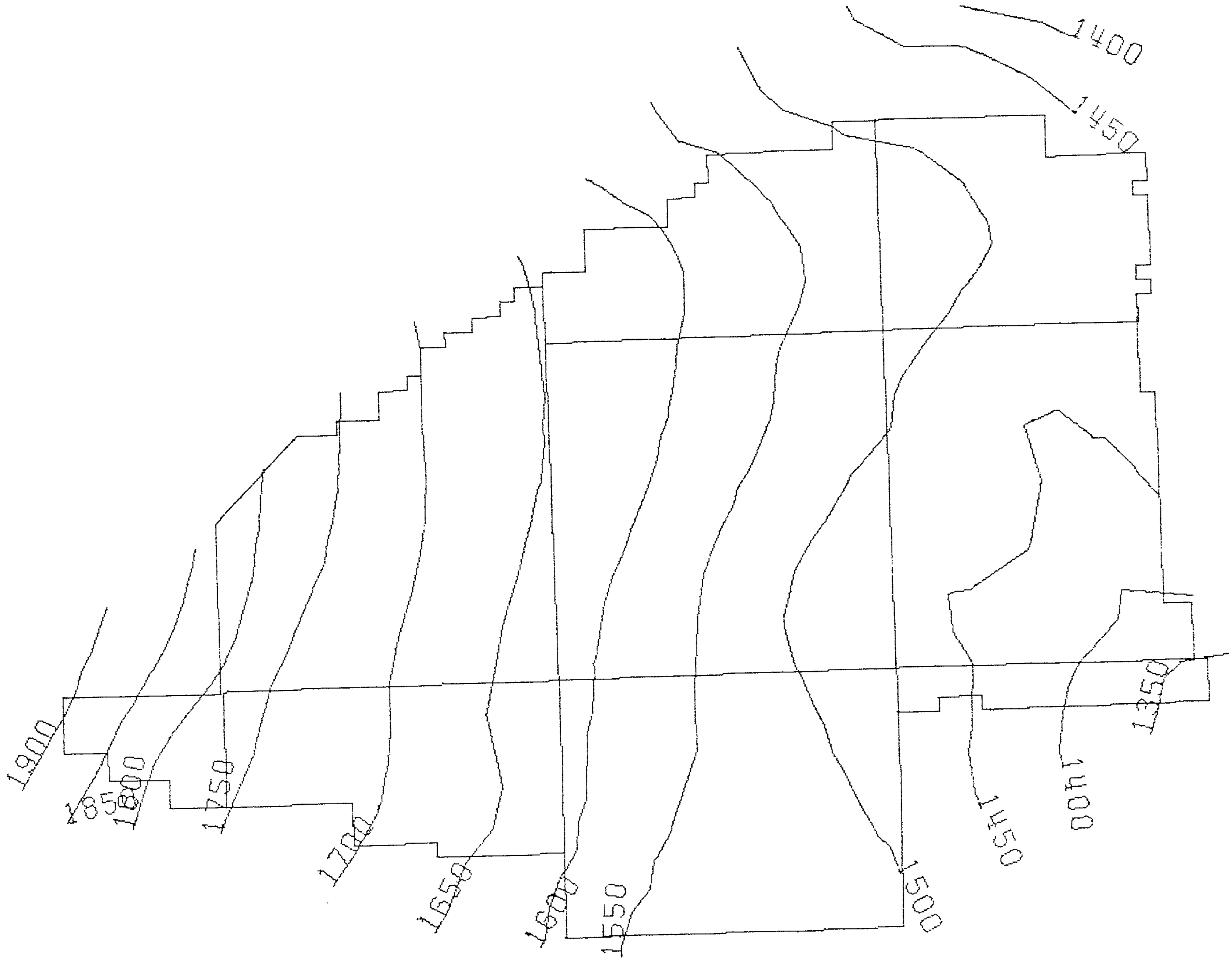



Figure 15h.--Simulated water levels, in feet, under scheme 06 conditions, Spring 1990.

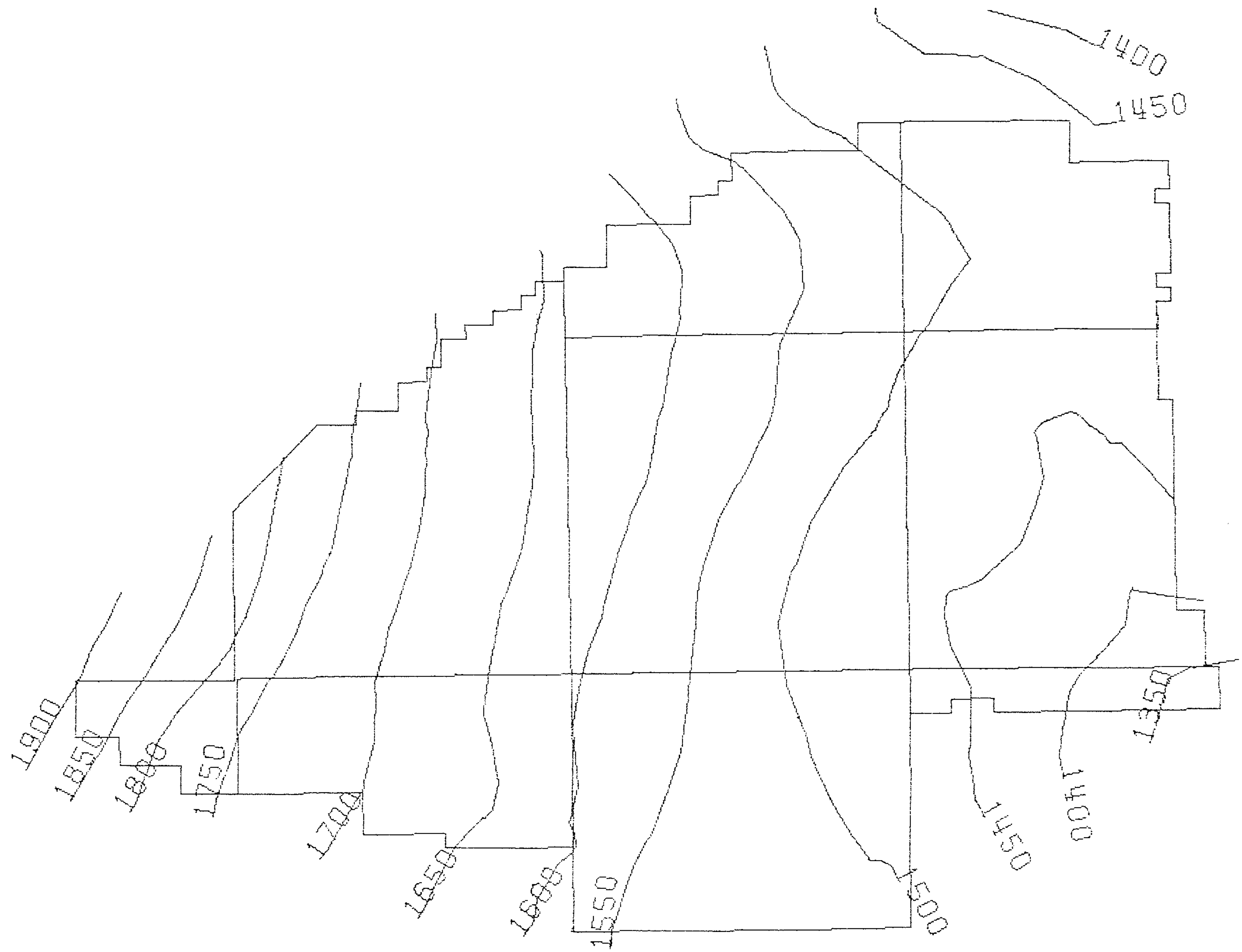



Figure 15i.--Simulated water levels, in feet, under scheme 06 conditions, Spring 2000.

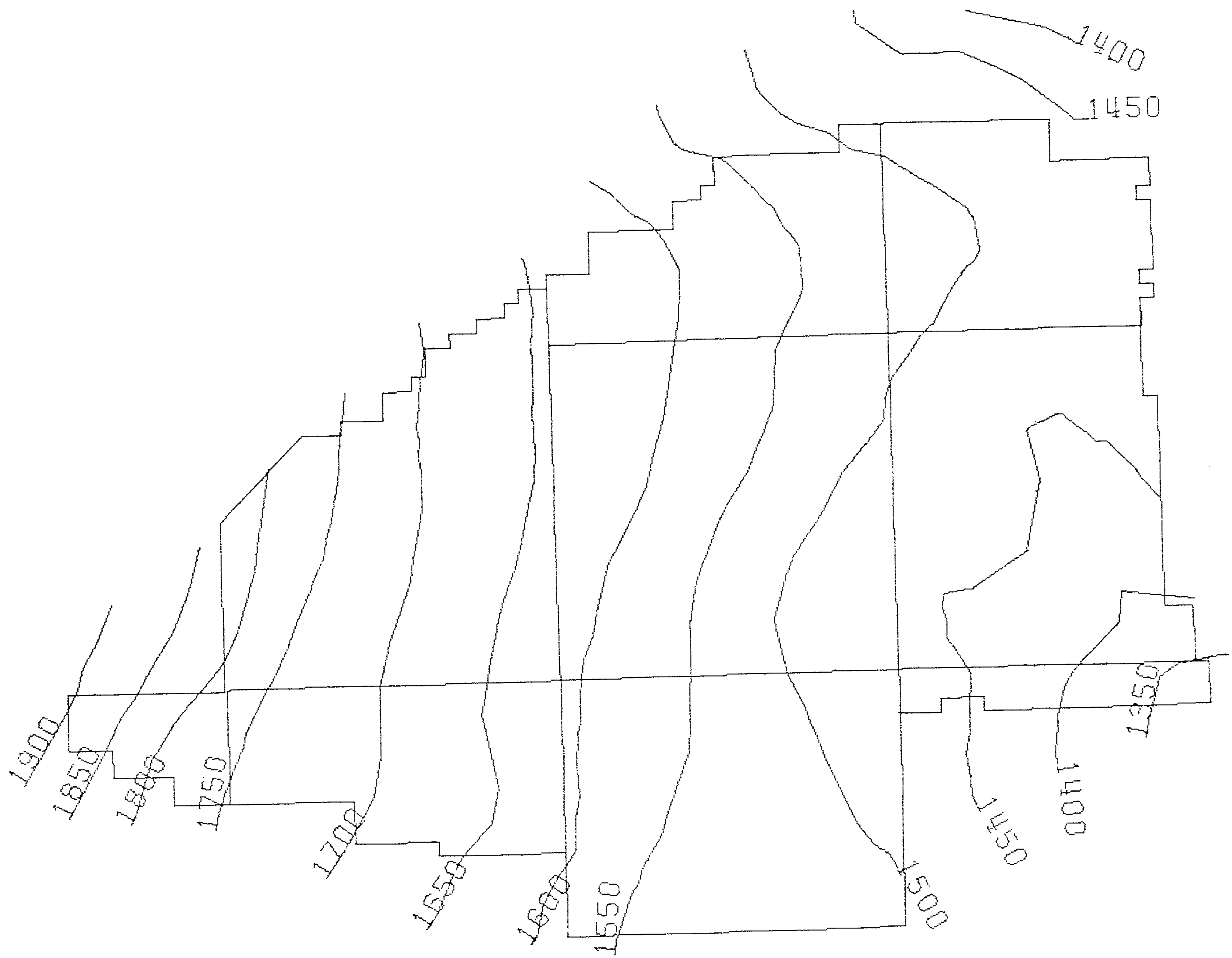



Figure 15j.--Simulated water levels, in feet, under scheme 07 conditions, Spring 1990.

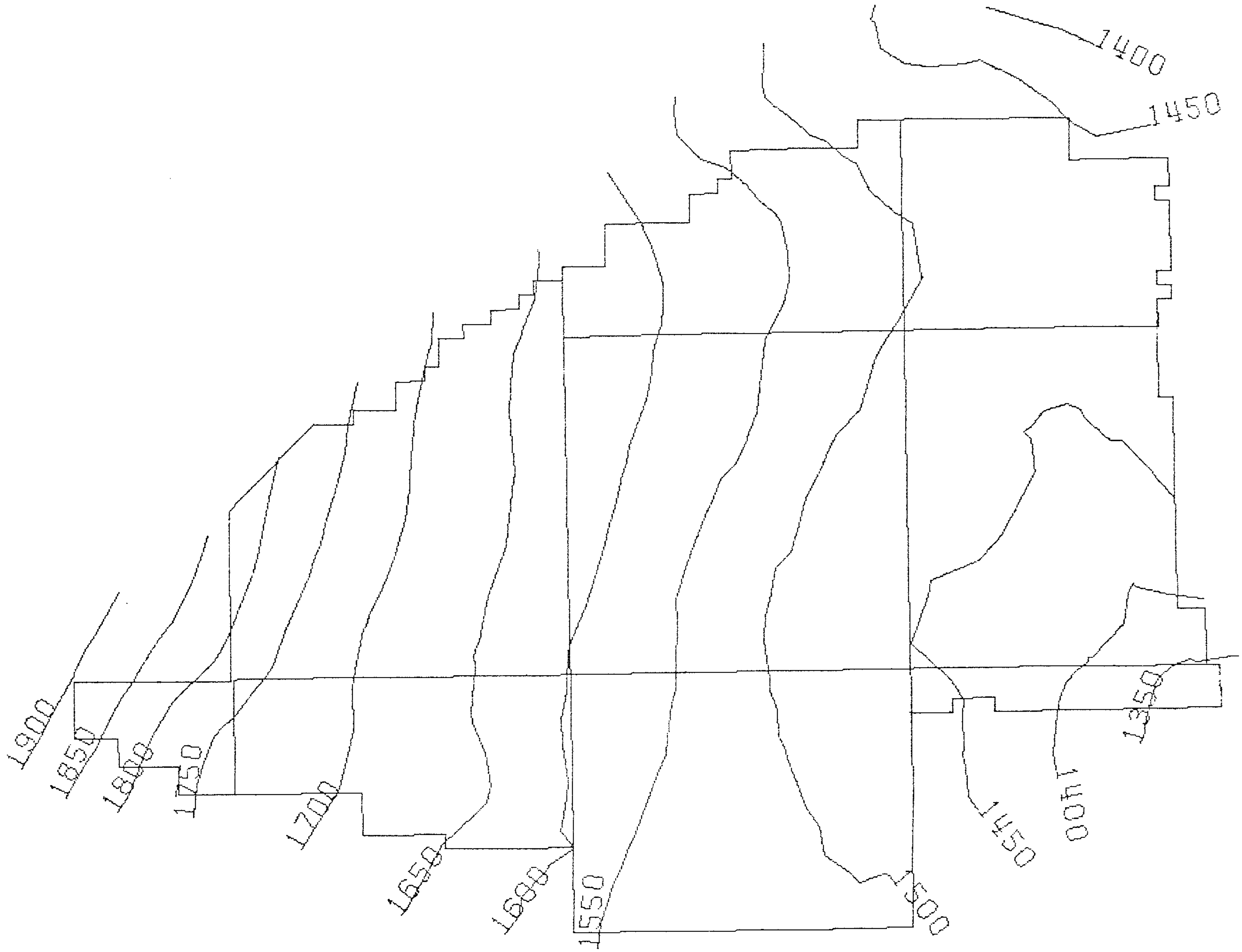



Figure 15k.--Simulated water levels, in feet, under scheme 07 conditions, Spring 2000.

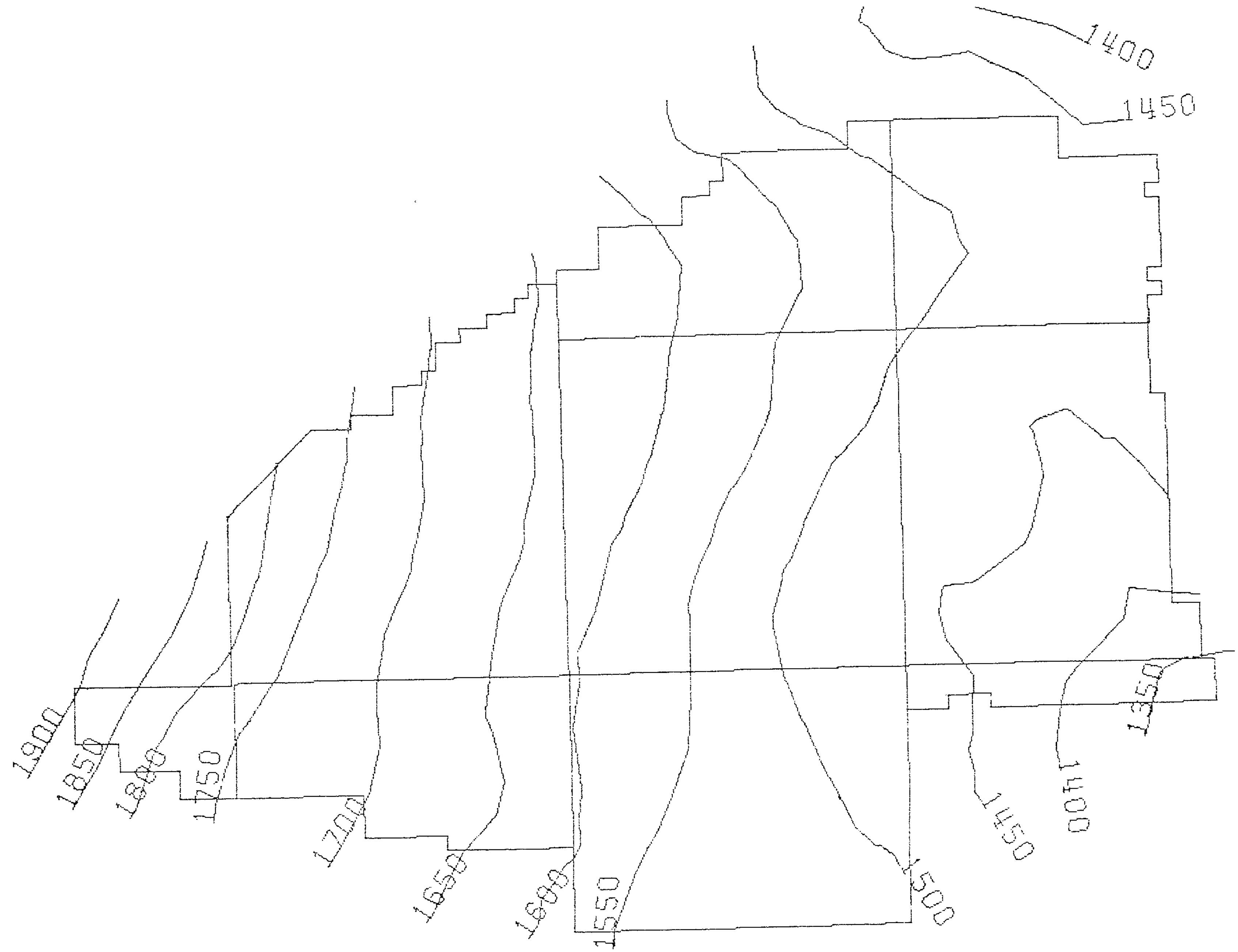



Figure 15ℓ.--Simulated water levels, in feet, under scheme 08 conditions, Spring 1990.

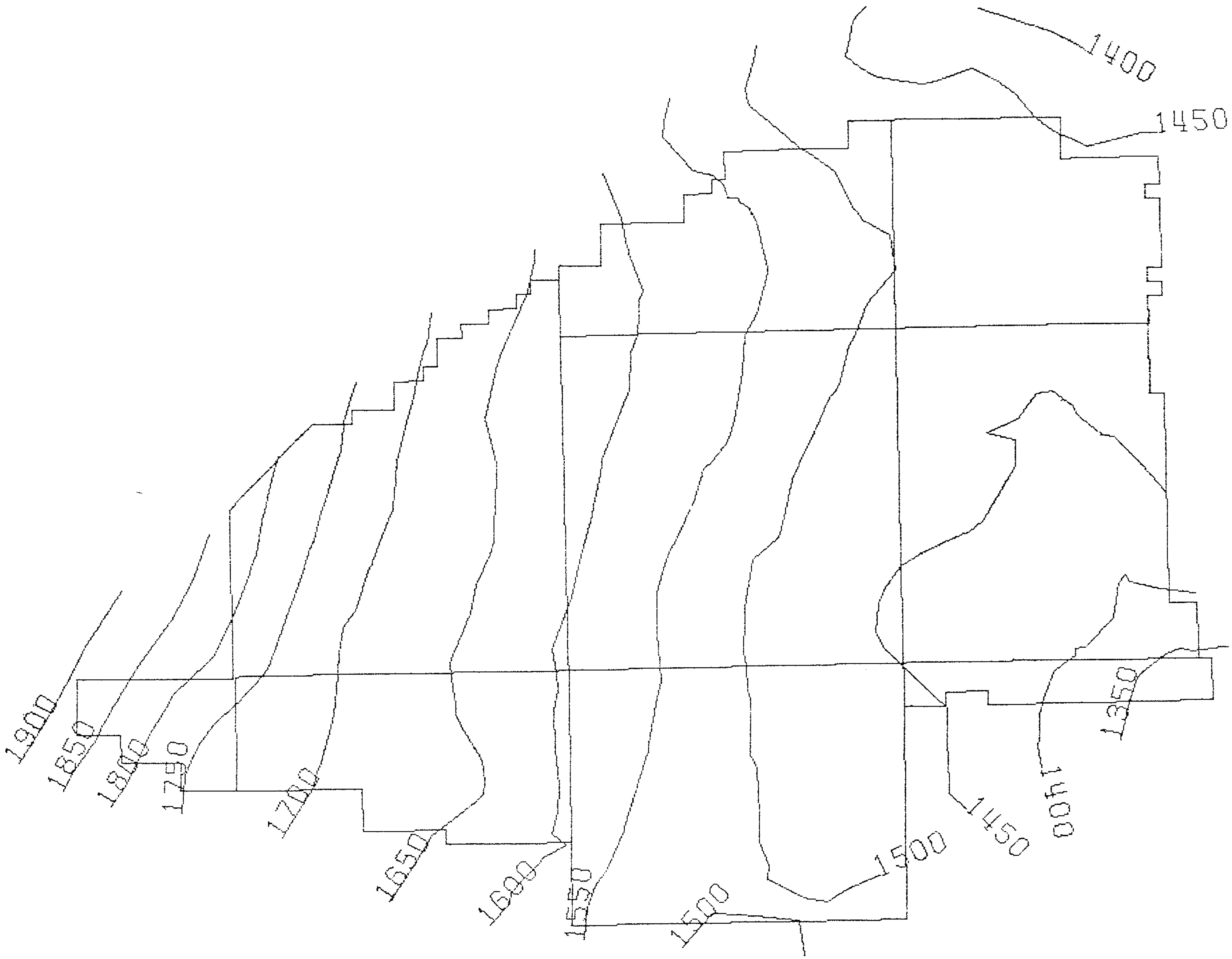



Figure 15m.--Simulated water levels, in feet, under scheme 08 conditions, Spring 2000.


Figure 15n.--Simulated water levels, in feet, under scheme 09 conditions, Spring 1990.

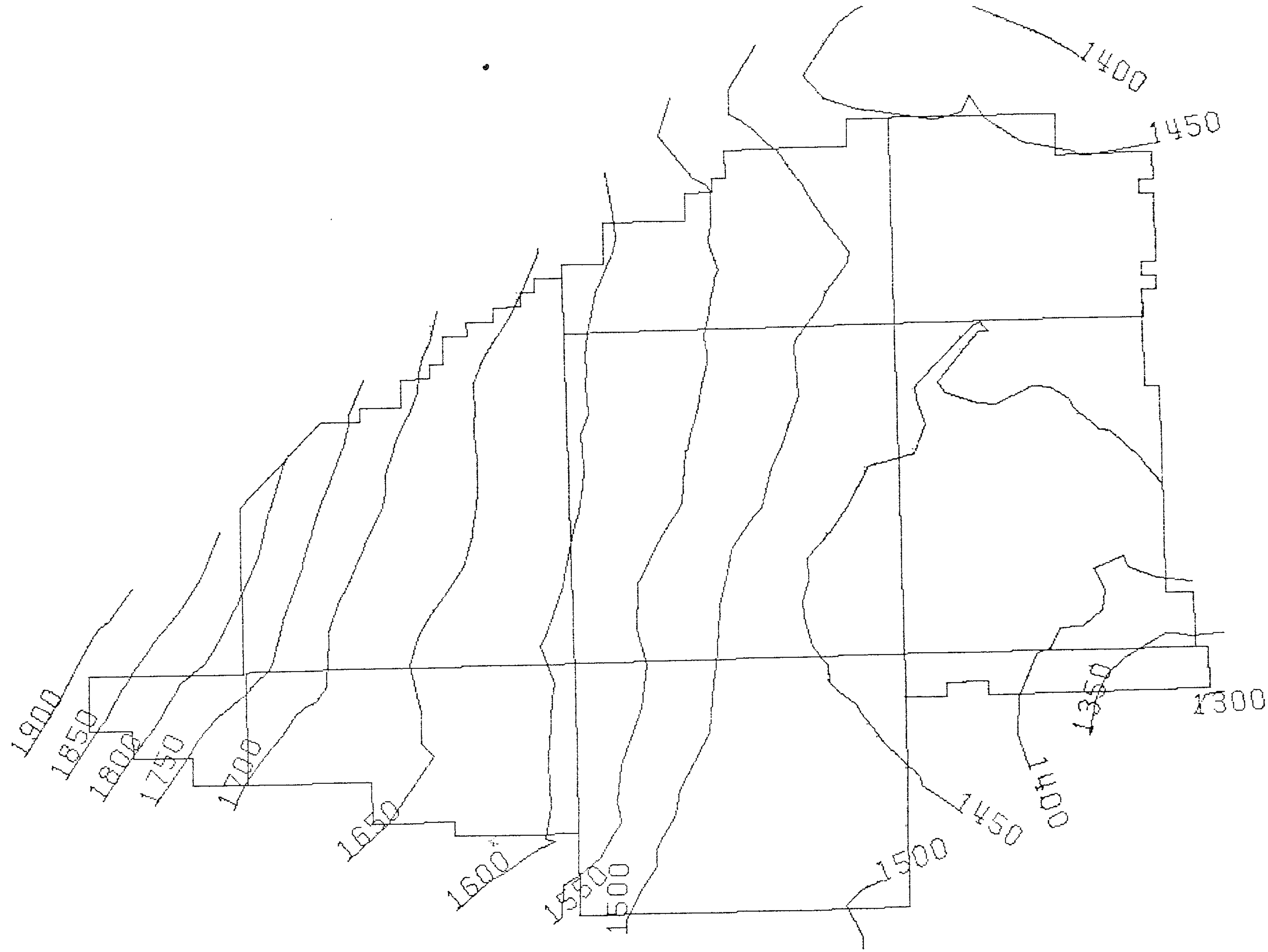



Figure 15o.--Simulated water levels, in feet, under scheme 09 conditions, Spring 2000.

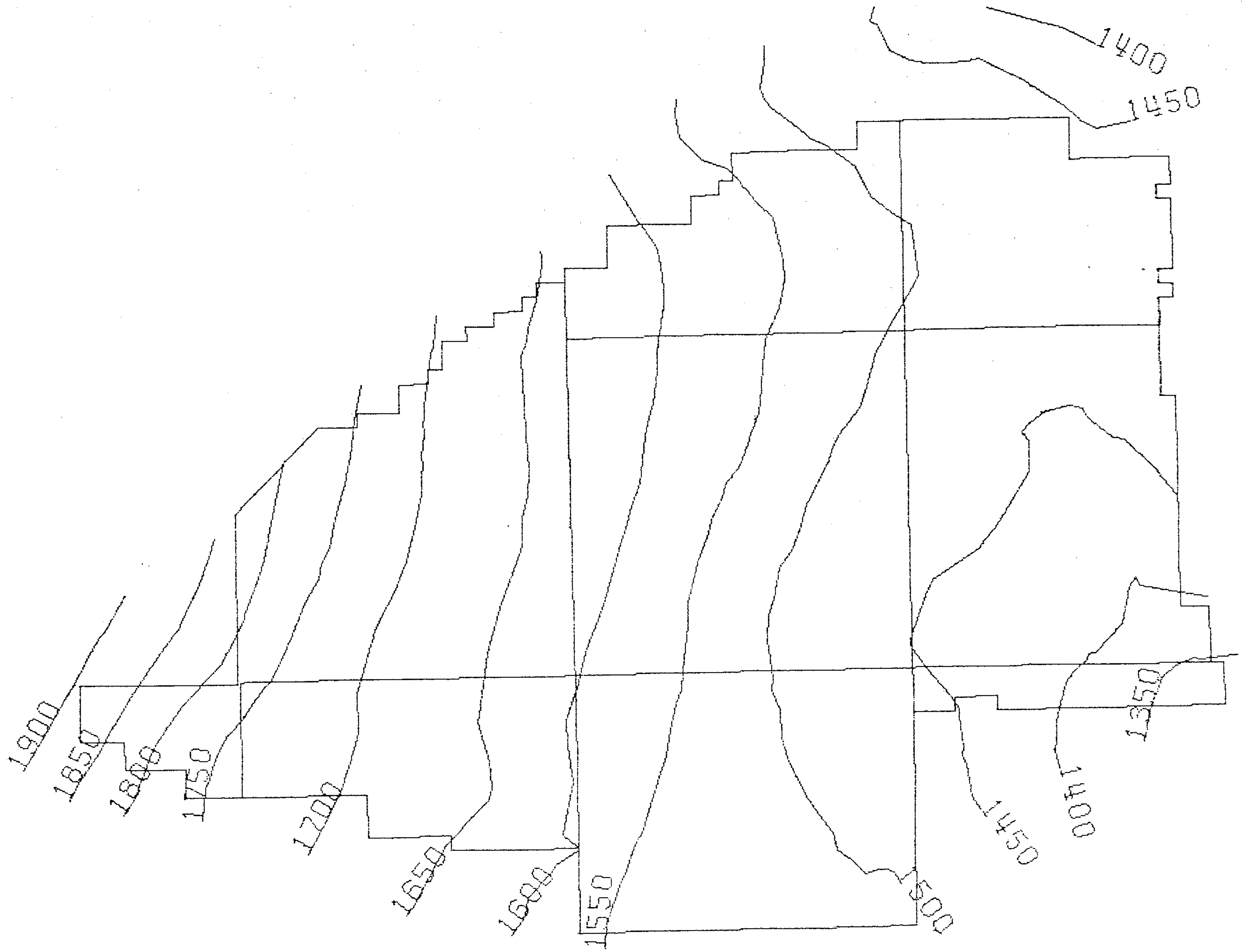



Figure 15p.--Simulated water levels, in feet, under scheme 10 conditions, Spring 2000.

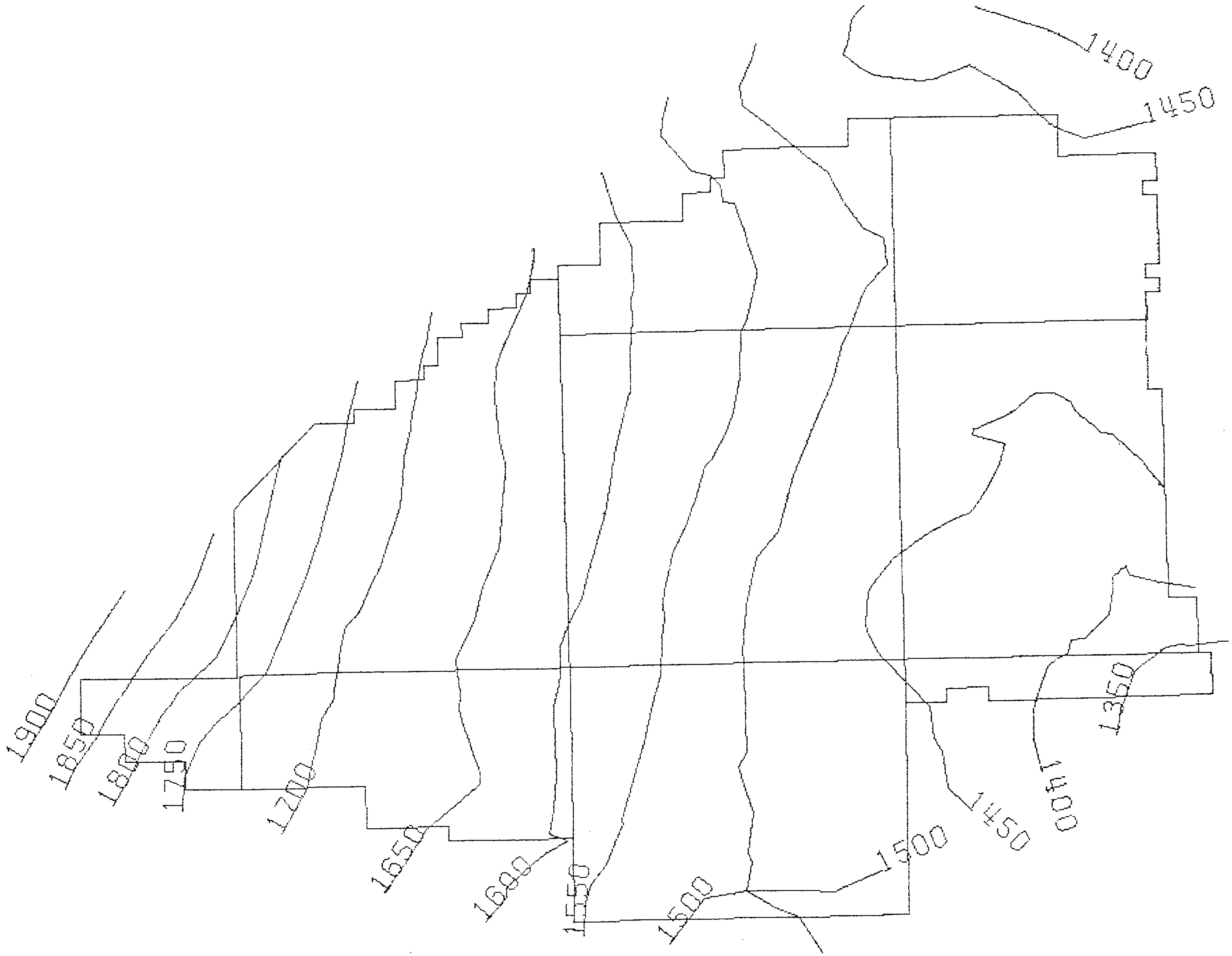



Figure 15q.--Simulated water levels, in feet, under scheme 11 conditions, Spring 2000.

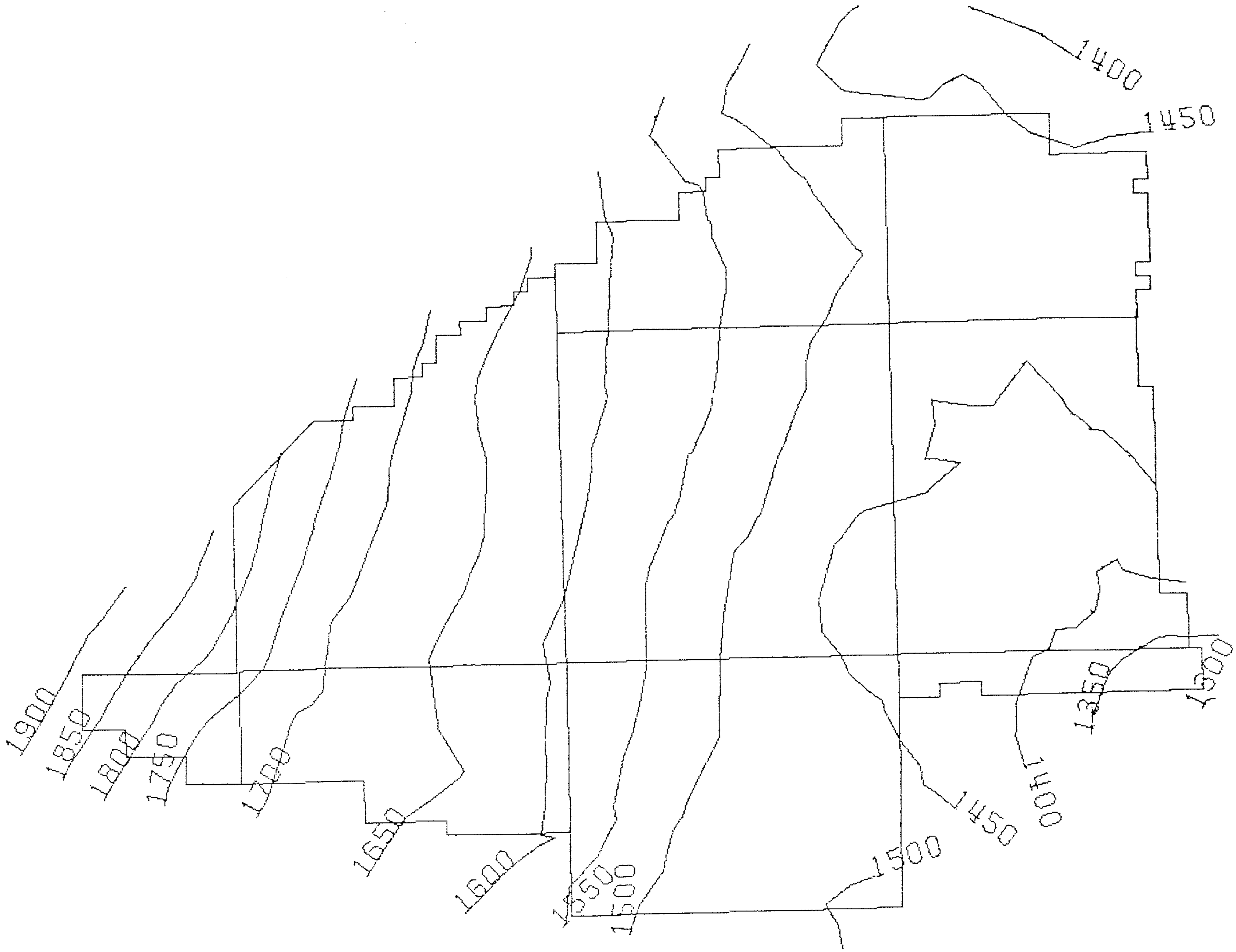



Figure 15r.--Simulated water levels, in feet, under scheme 12 conditions, Spring 2000.

SIMULATED WATER-LEVEL DECLINES FROM SPRING 1975

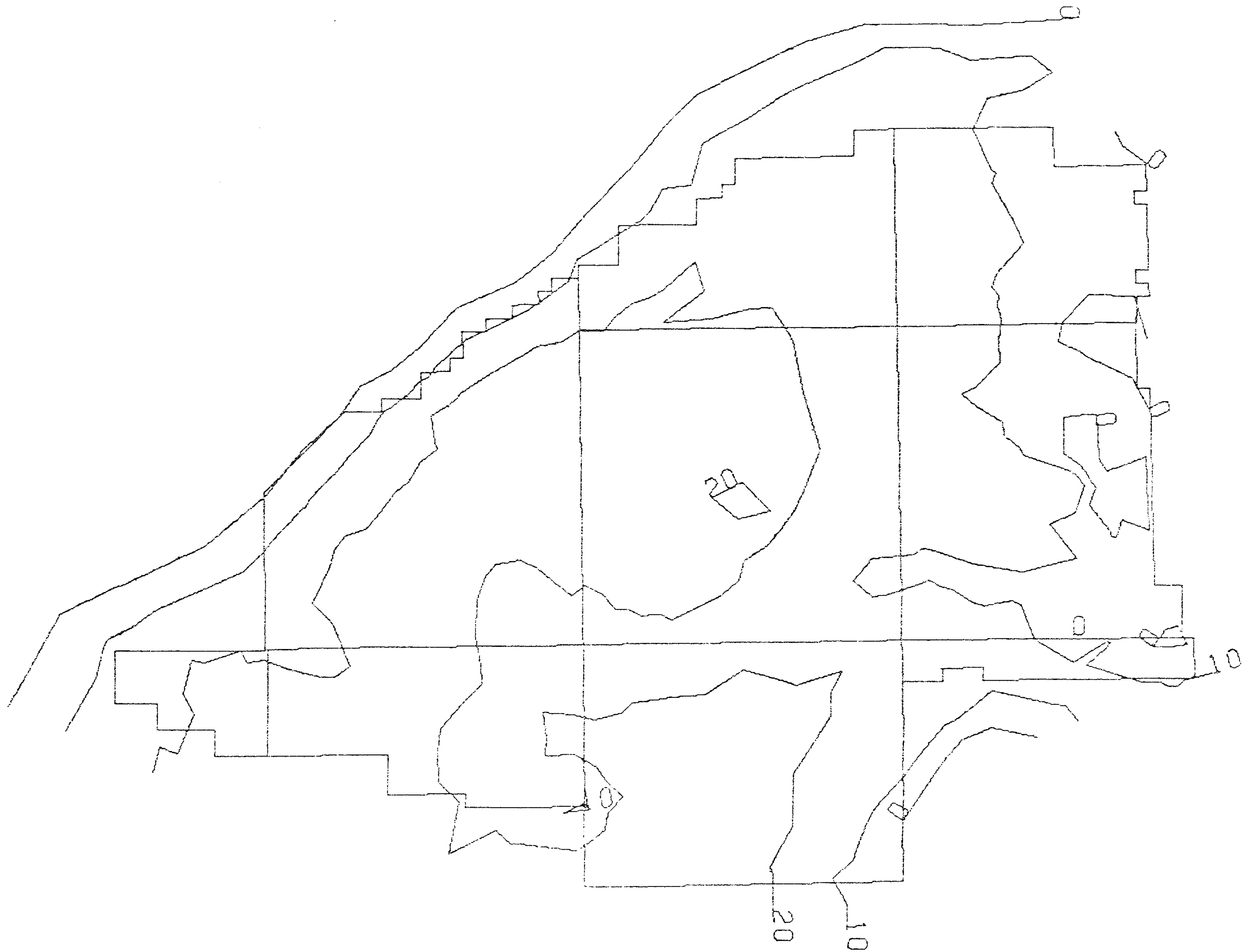



Figure 16a.--Simulated water-level decline from Spring 1975, in feet, under scheme 01 conditions, Spring 1990.

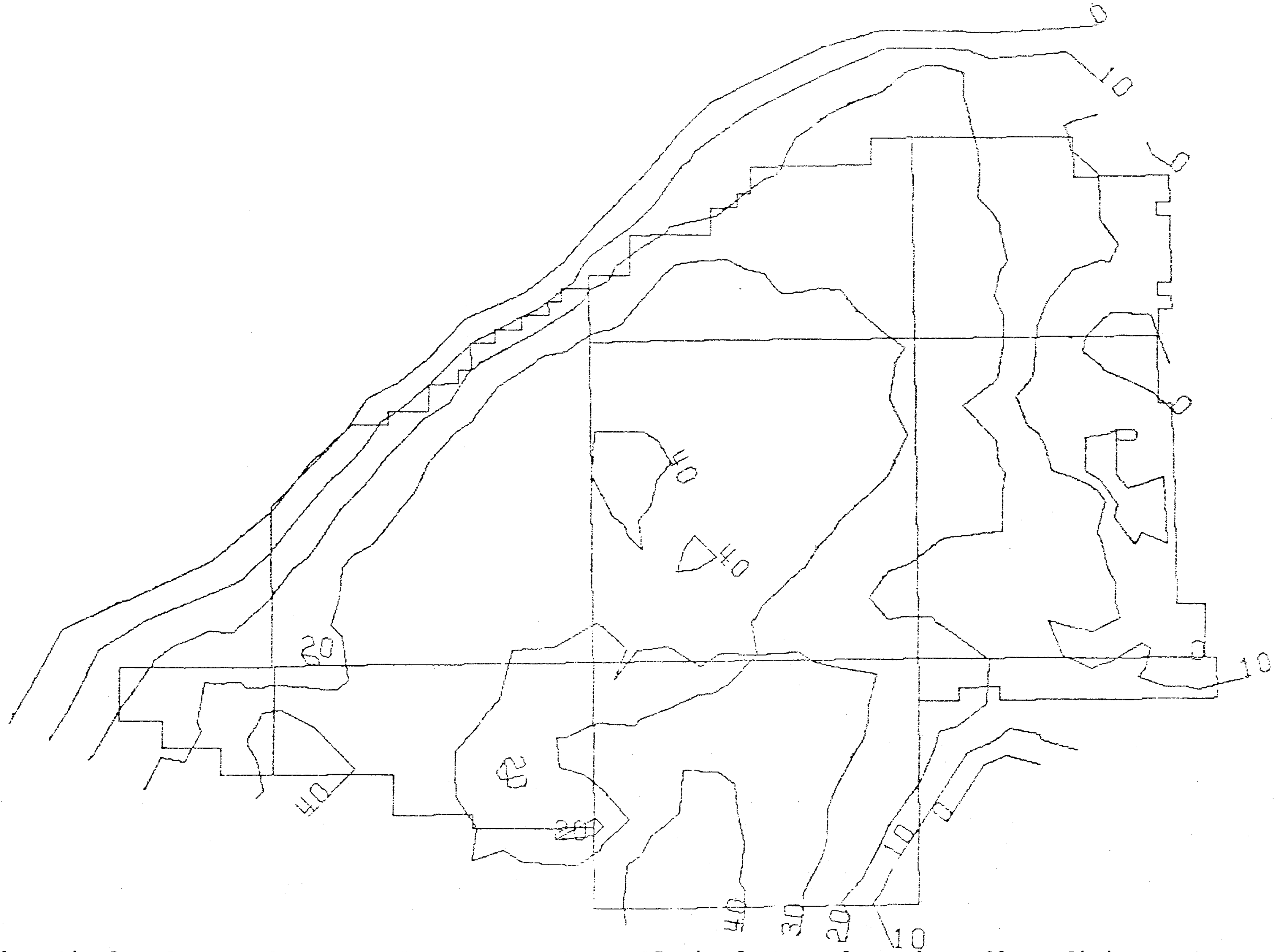



Figure 16b.--Simulated water-level decline from Spring 1975, in feet, under scheme 01 conditions, Spring 2000.

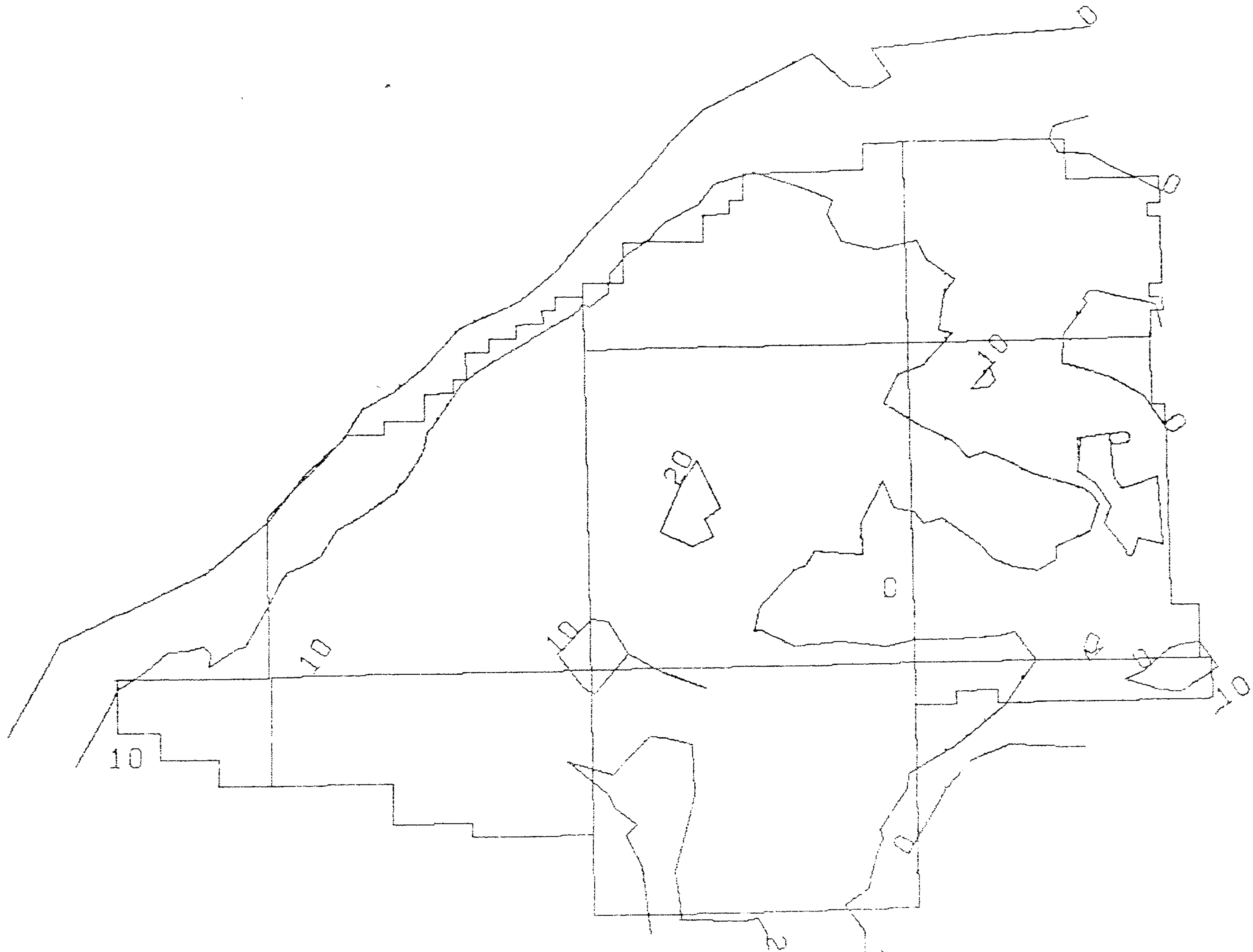



Figure 16c.--Simulated water-level decline from Spring 1975, in feet, under scheme 02 conditions, Spring 2000.

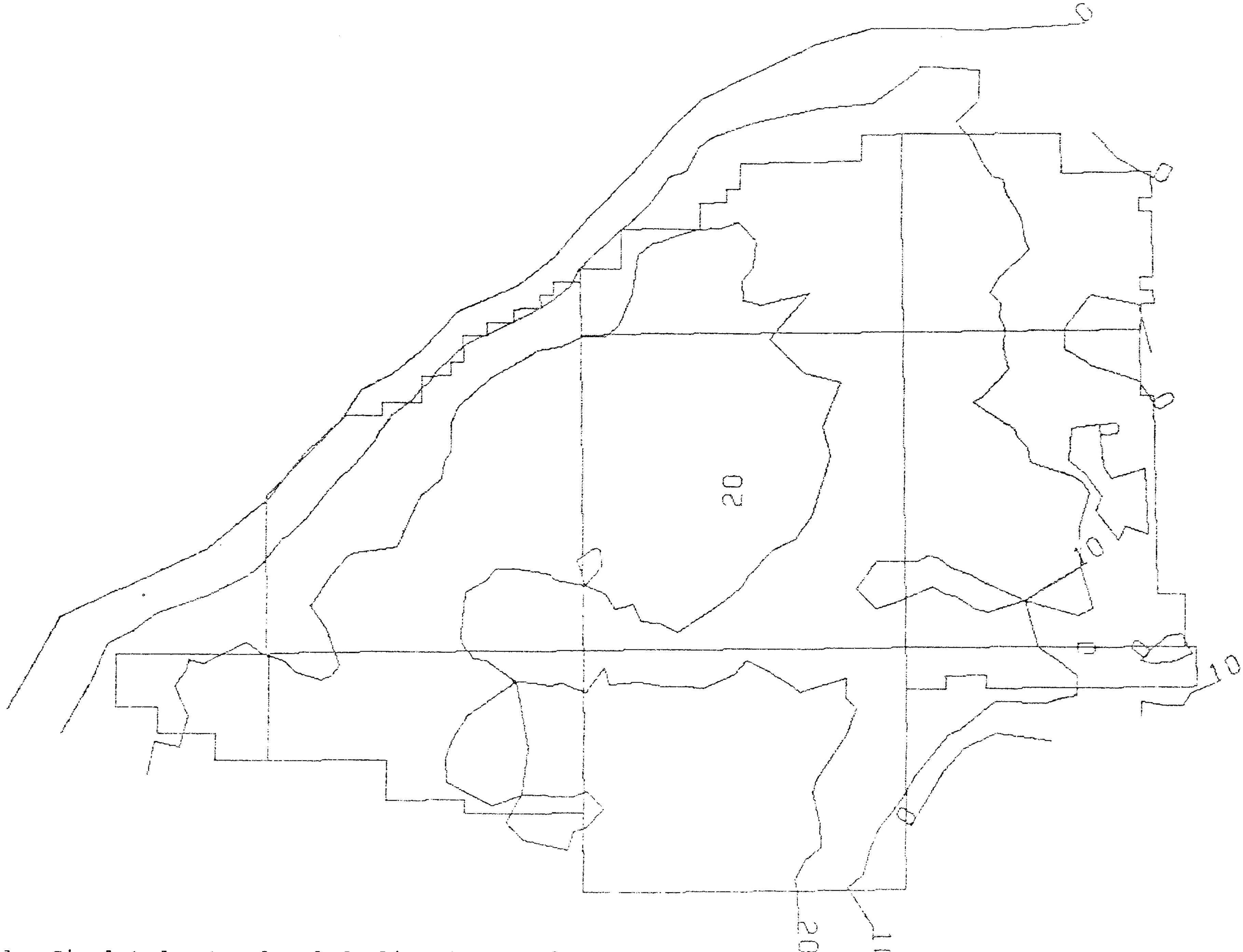



Figure 16d.--Simulated water-level decline from Spring 1975, in feet, under scheme 03 conditions, Spring 2000.

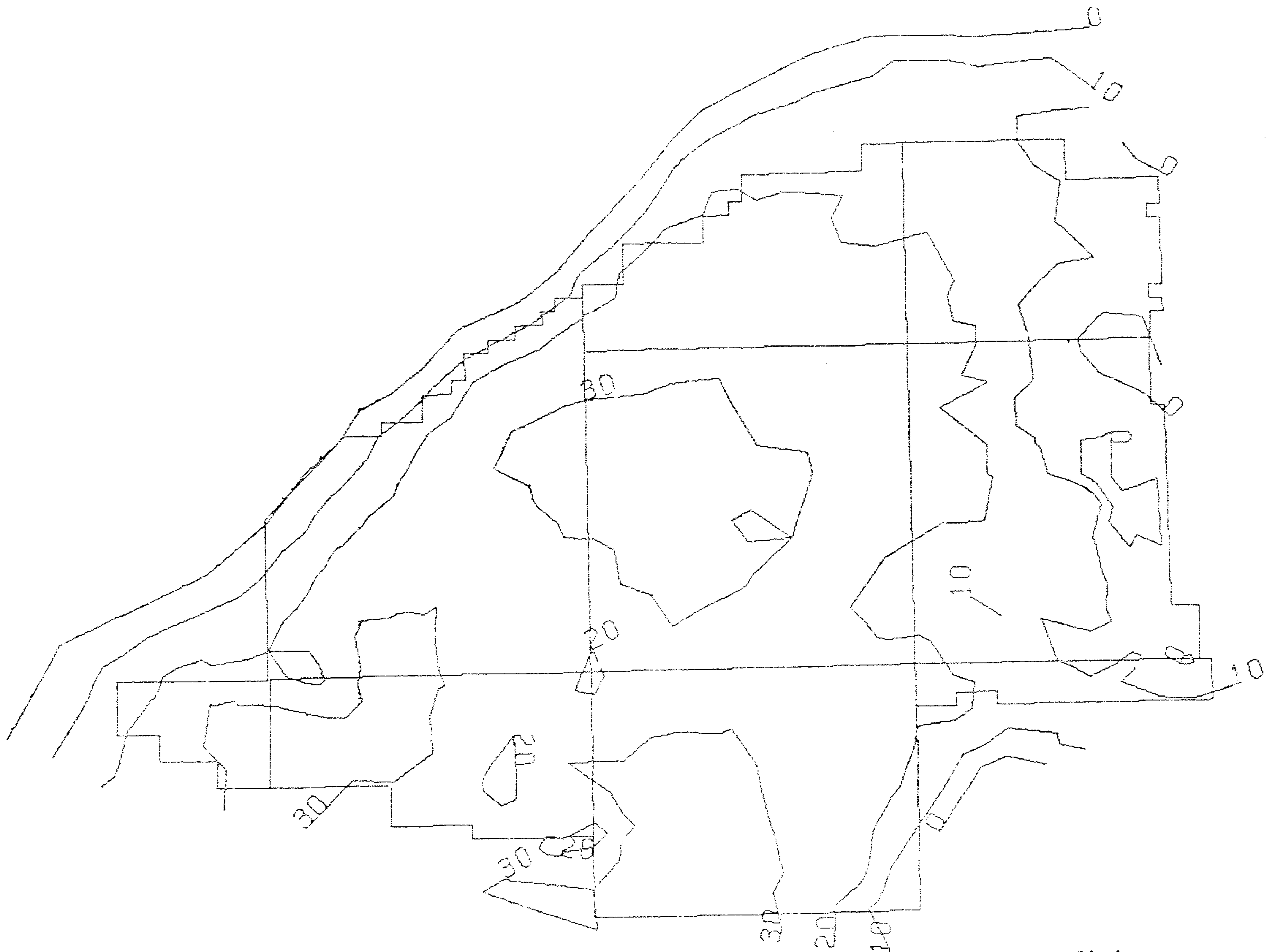



Figure 16e.--Simulated water-level decline from Spring 1975, in feet, under scheme 04 conditions, Spring 2000.

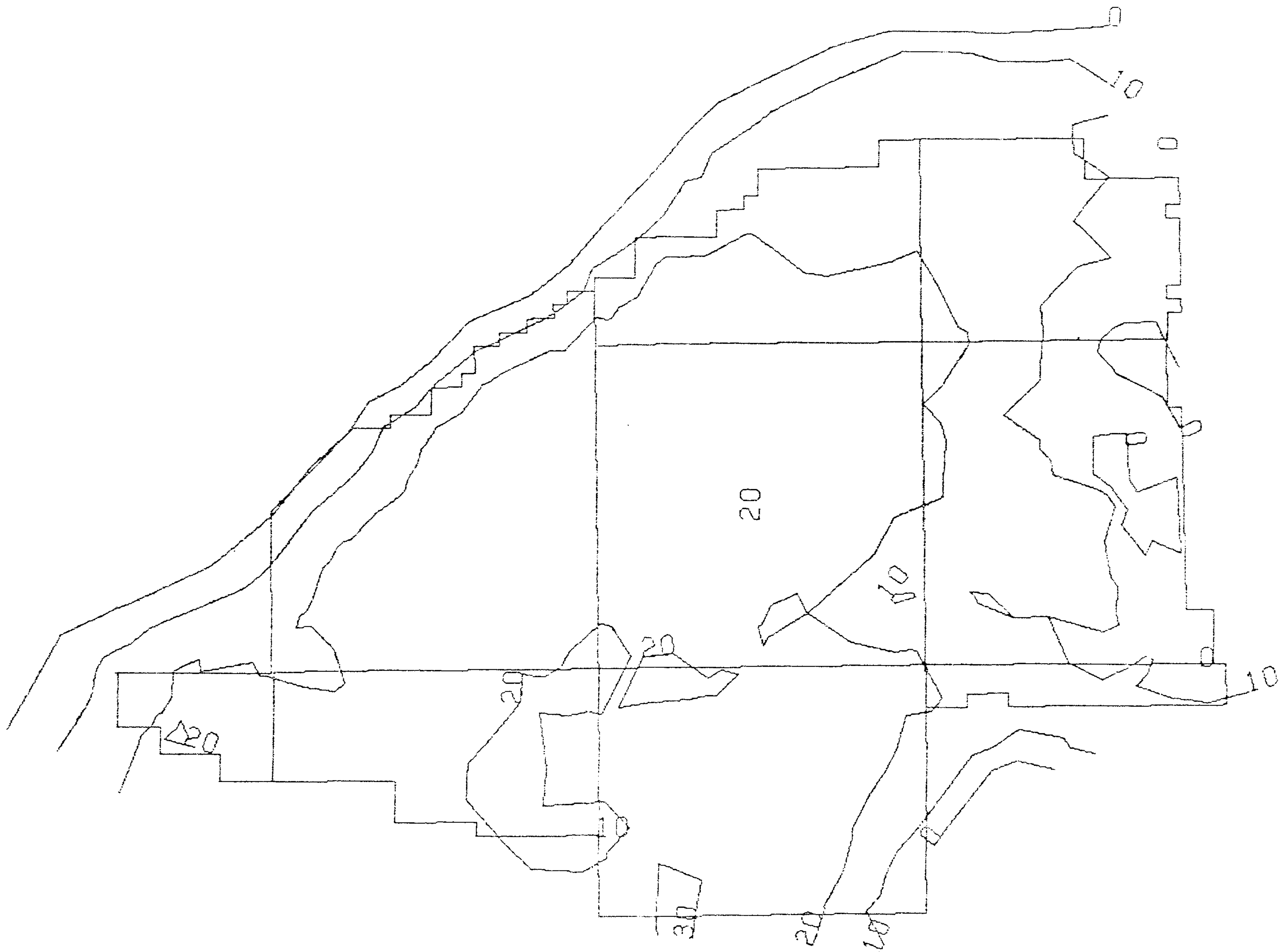



Figure 16f.--Simulated water-level decline from Spring 1975, in feet, under scheme 05 conditions, Spring 1990.

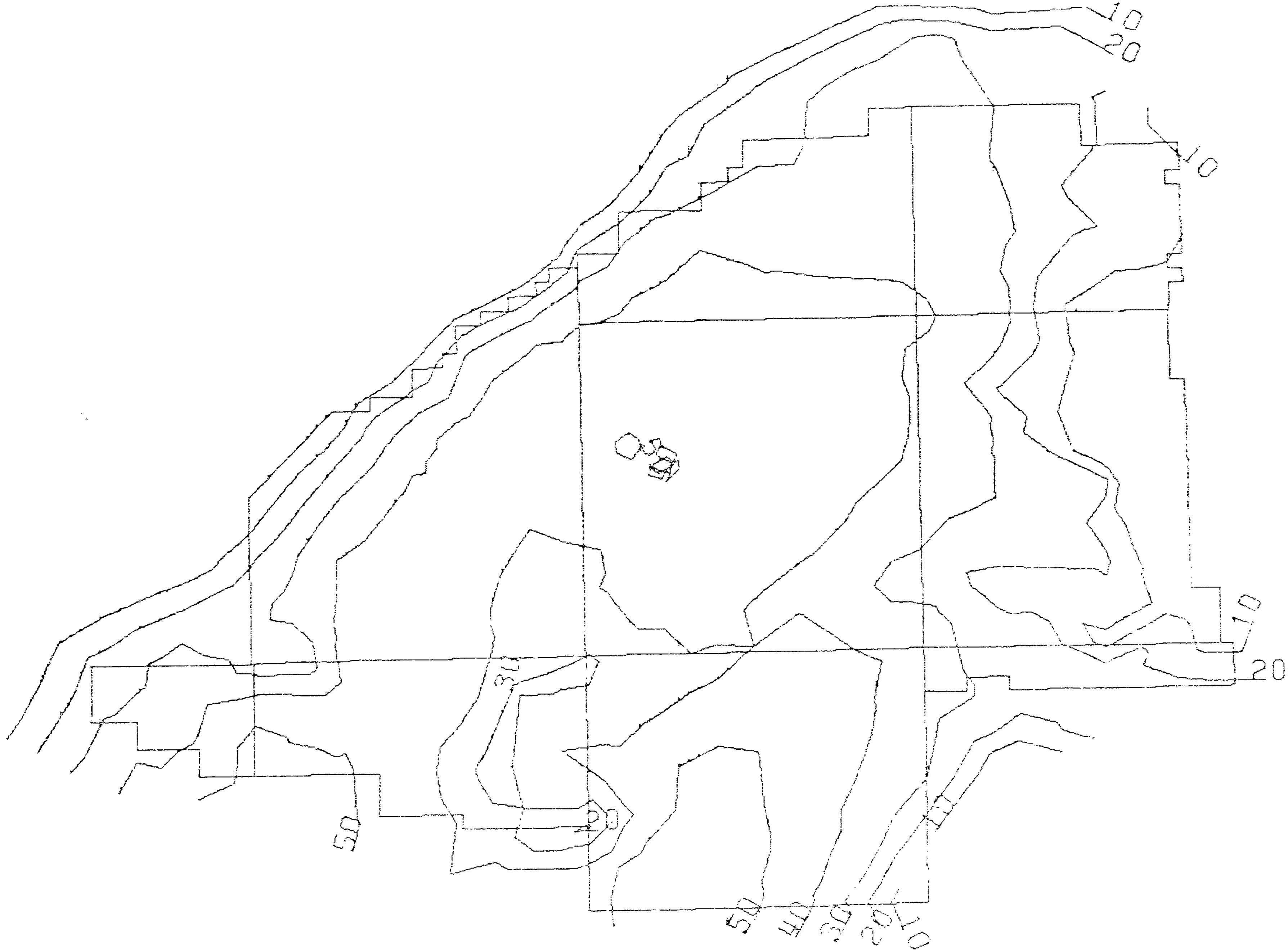



Figure 16g.--Simulated water-level decline from Spring 1975, in feet, under scheme 05 conditions, Spring 2000.

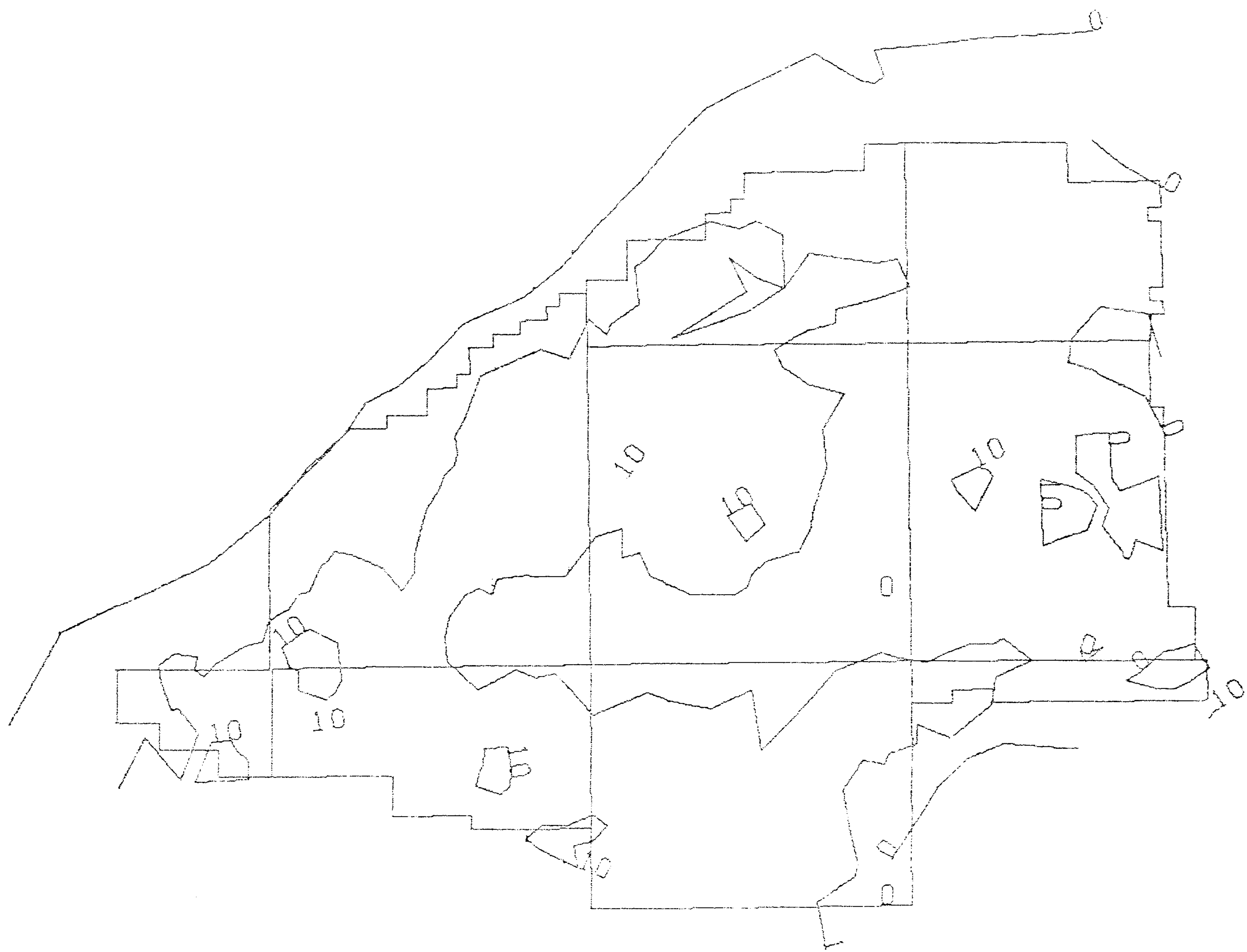



Figure 16h.--Simulated water-level decline from Spring 1975, in feet, under scheme 06 conditions, Spring 1990.

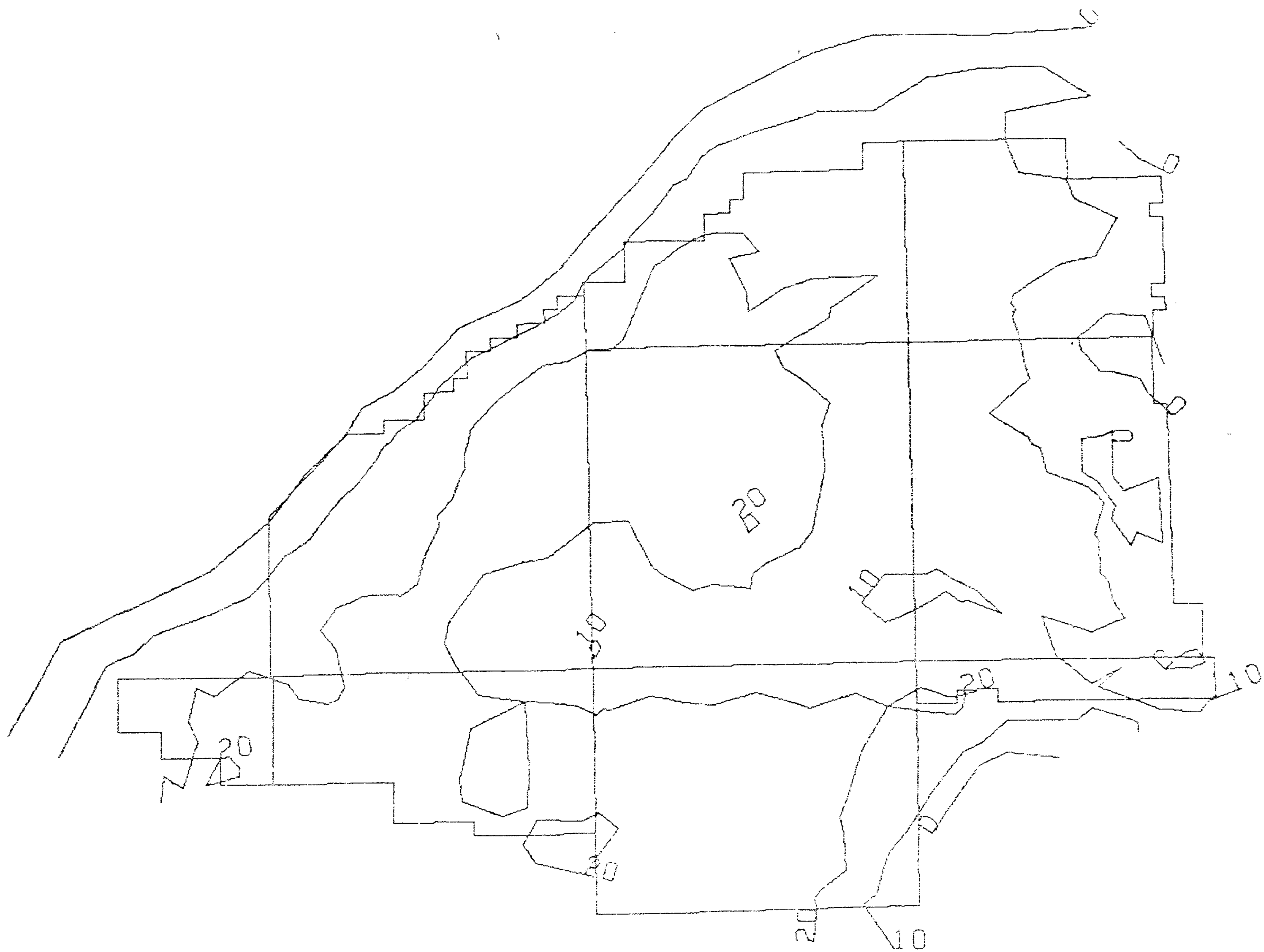



Figure 16i.--Simulated water-level decline from Spring 1975, in feet, under scheme 06 conditions, Spring 2000.

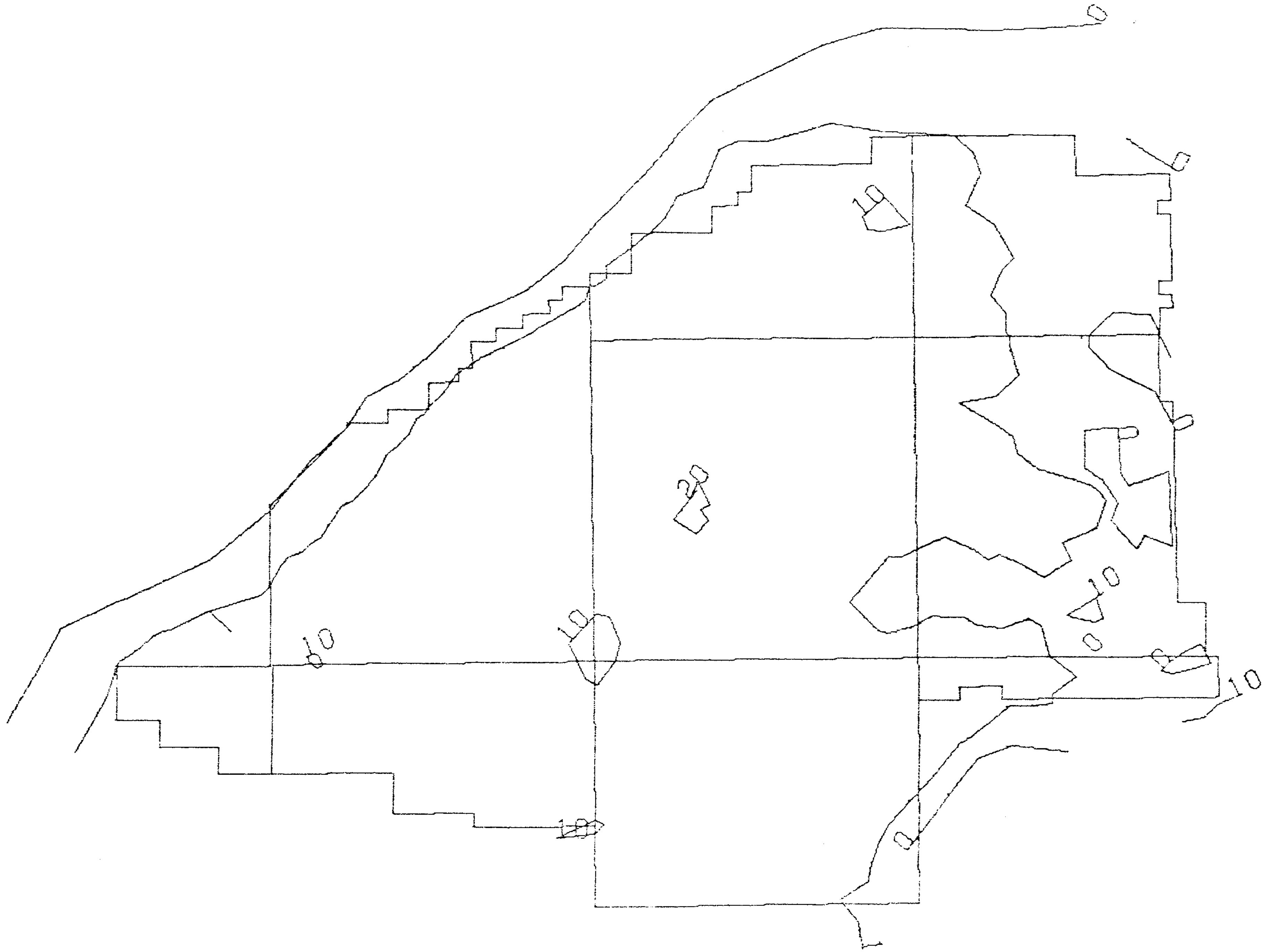


Figure 16j.--Simulated water-level decline from Spring 1975, in feet, under scheme 07 conditions, Spring 1990.

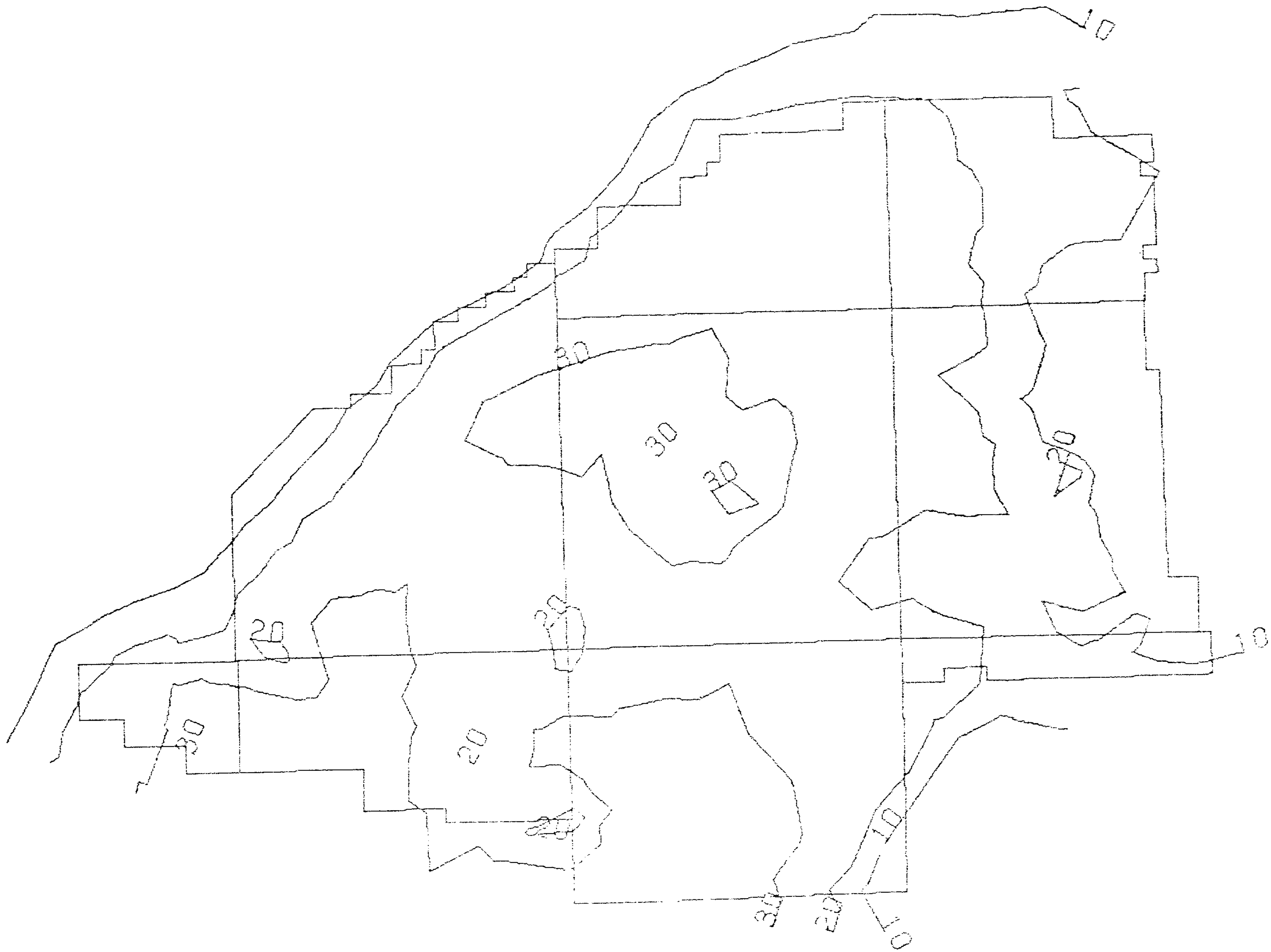



Figure 16k.--Simulated water-level decline from Spring 1975, in feet, under scheme 07 conditions, Spring 2000.

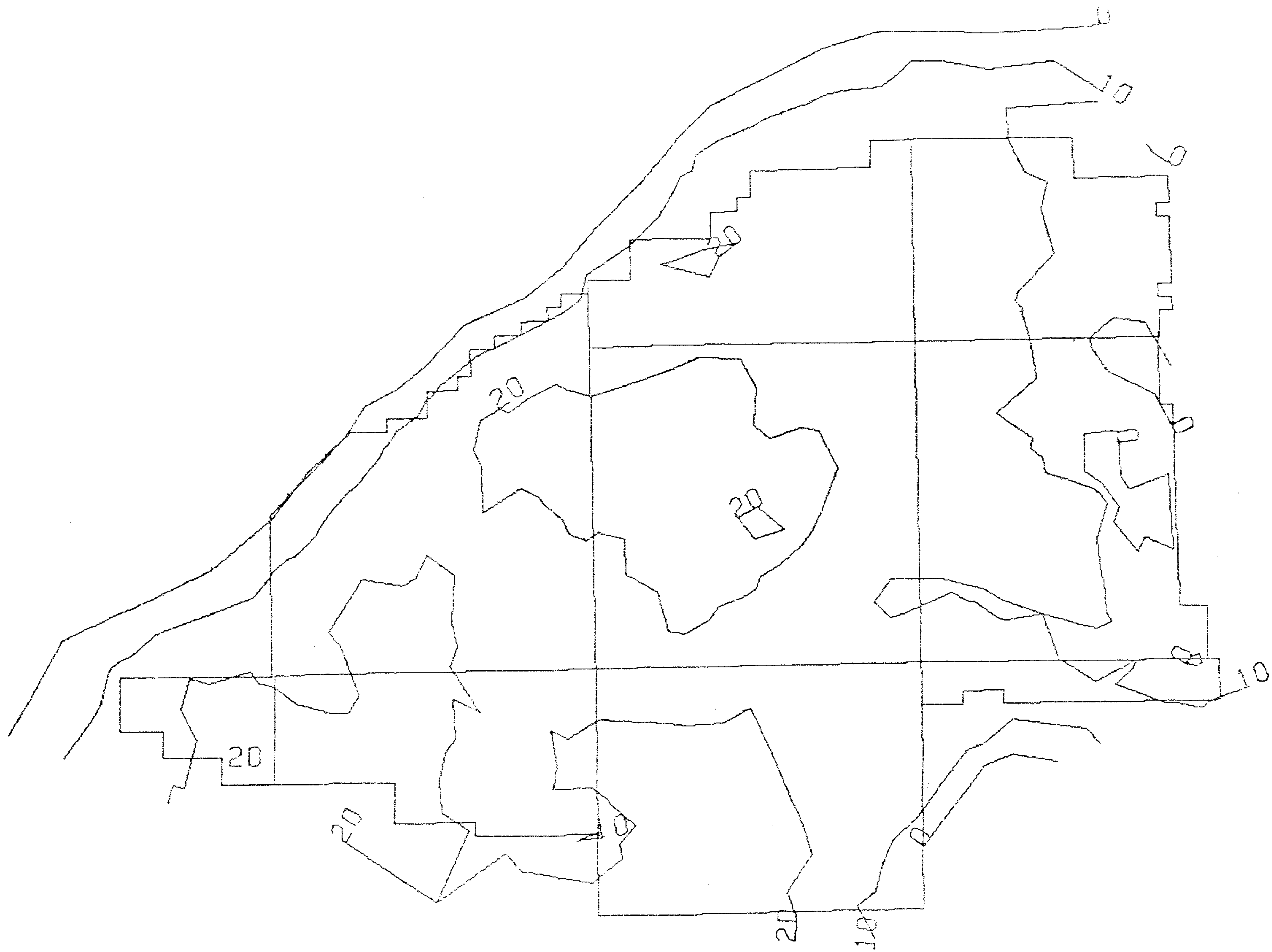



Figure 16ℓ.--Simulated water-level decline from Spring 1975, in feet, under scheme 08 conditions, Spring 1990.


Figure 16m.--Simulated water-level decline from Spring 1975, in feet, under scheme 08 conditions, Spring 2000.

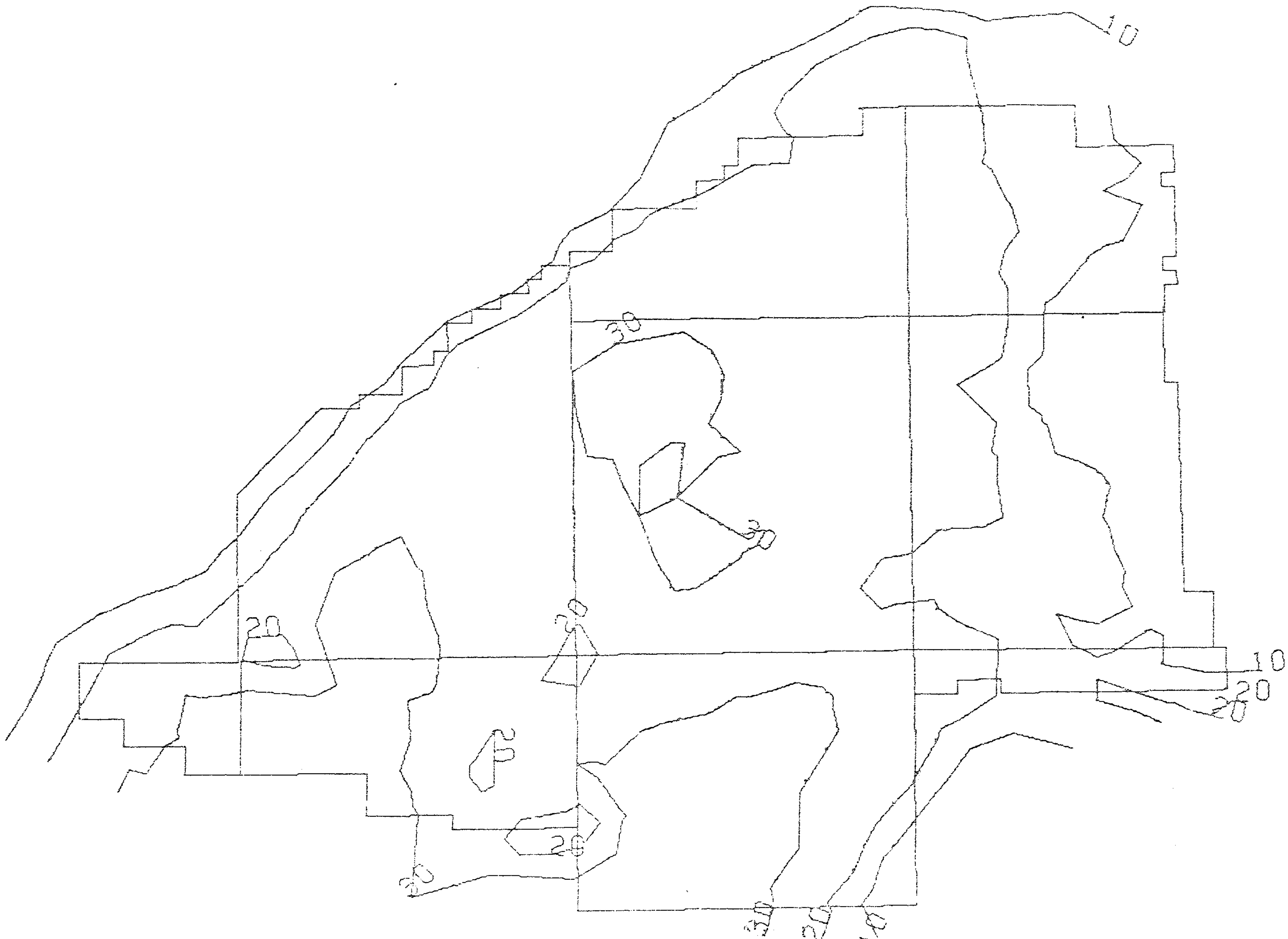



Figure 16n.--Simulated water-level decline from Spring 1975, in feet, under scheme 09 conditions, Spring 1990.

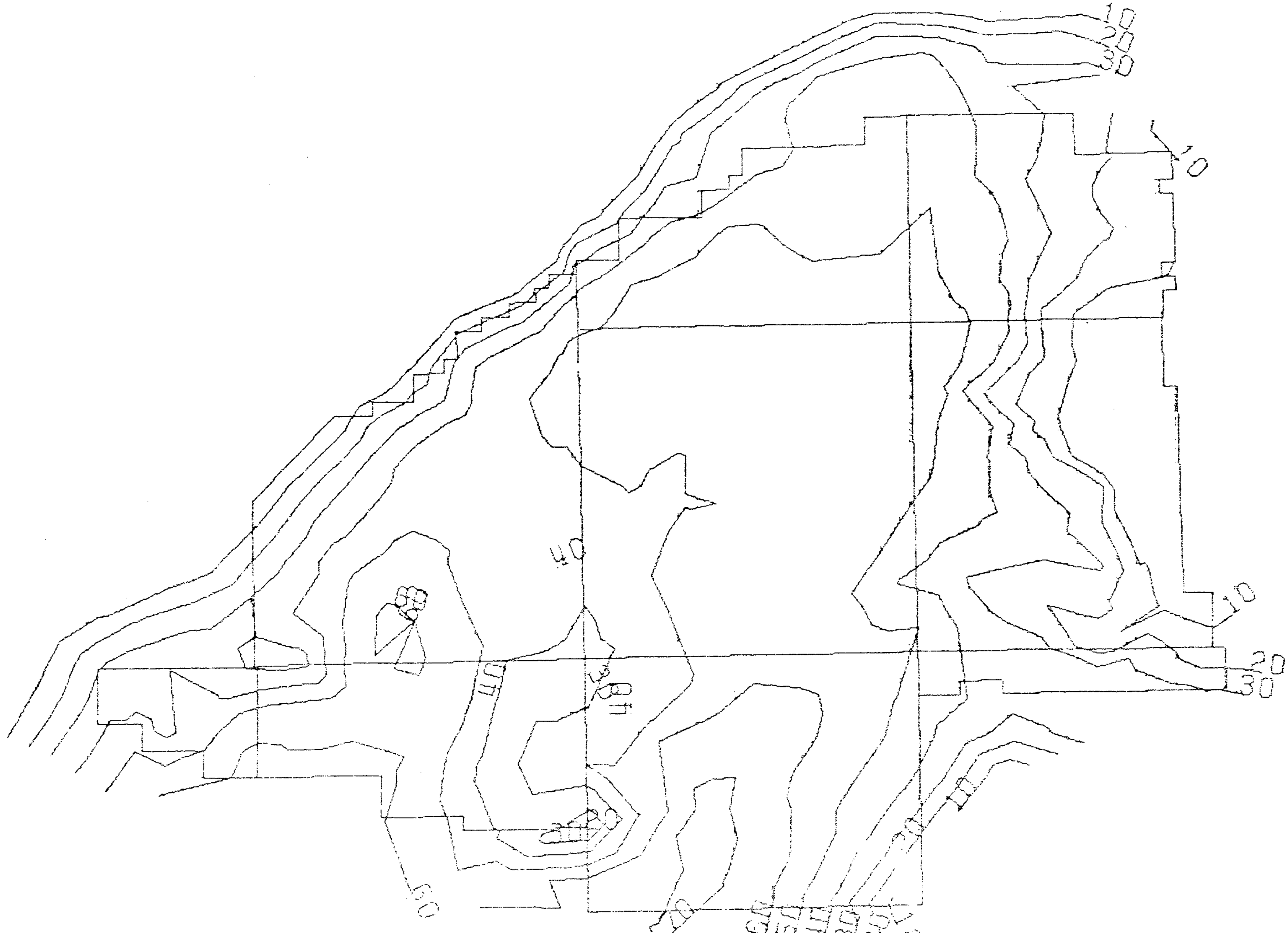



Figure 16o.--Simulated water-level decline from Spring 1975, in feet, under scheme 09 conditions, Spring 2000.

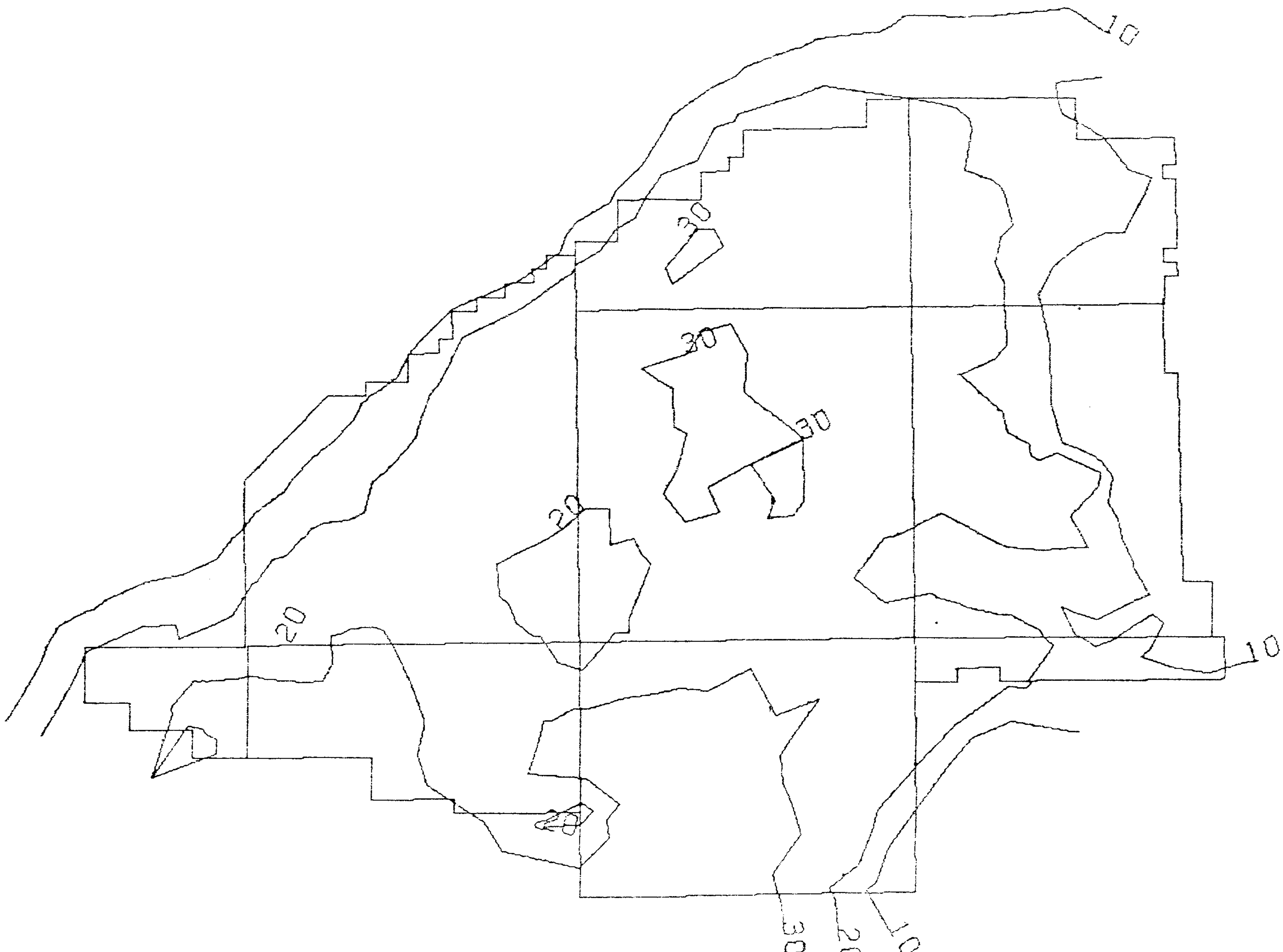


Figure 16p.--Simulated water-level decline from Spring 1975, in feet, under scheme 10 conditions, Spring 2000.

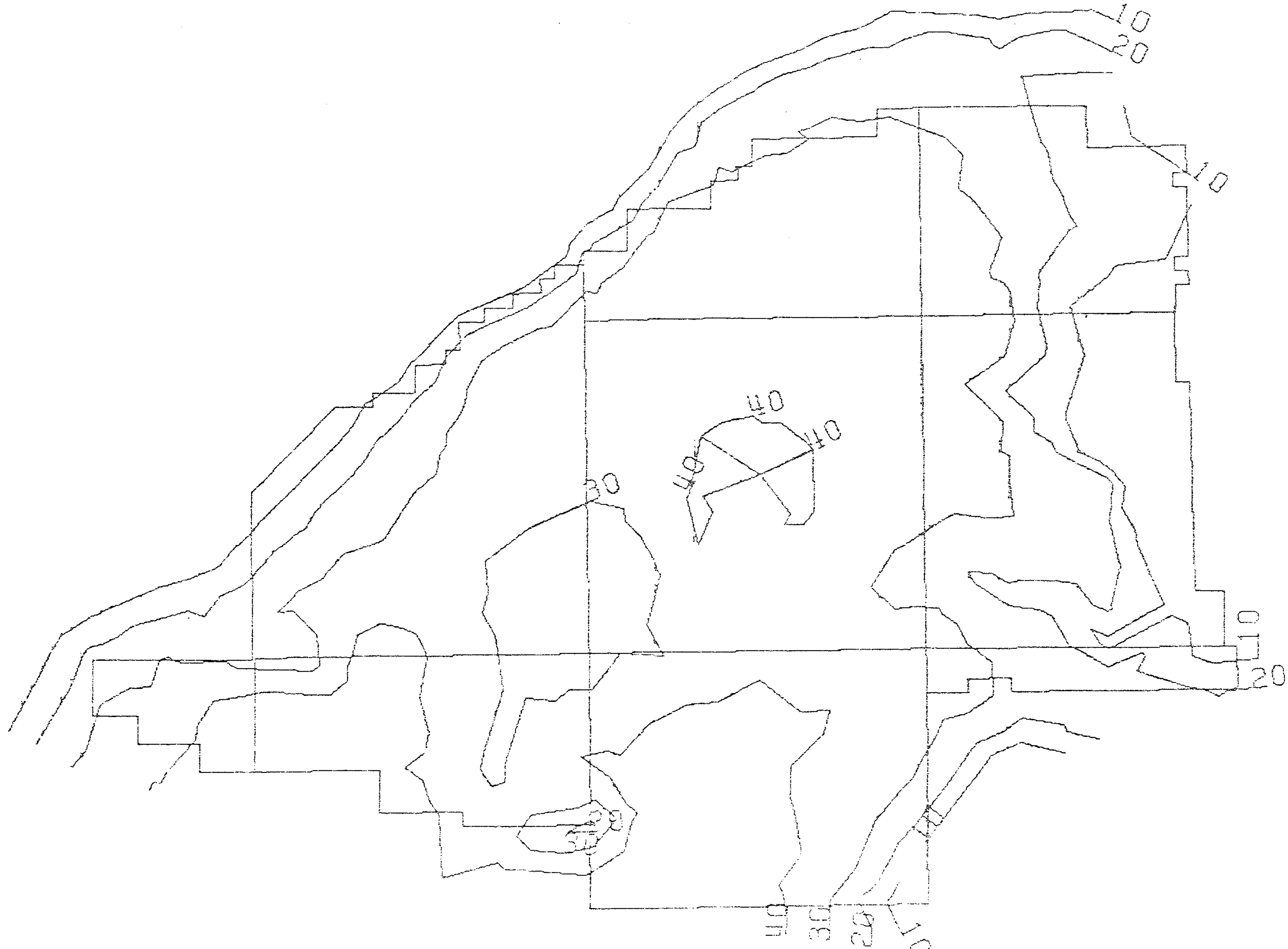


Figure 16q.--Simulated water-level decline from Spring 1975, in feet, under scheme 11 conditions, Spring 2000.


Figure 16r.--Simulated water-level decline from Spring 1975, in feet, under scheme 12 conditions, Spring 2000.

SIMULATED ELEVATION OF WATER LEVEL ABOVE BASE OF AQUIFER

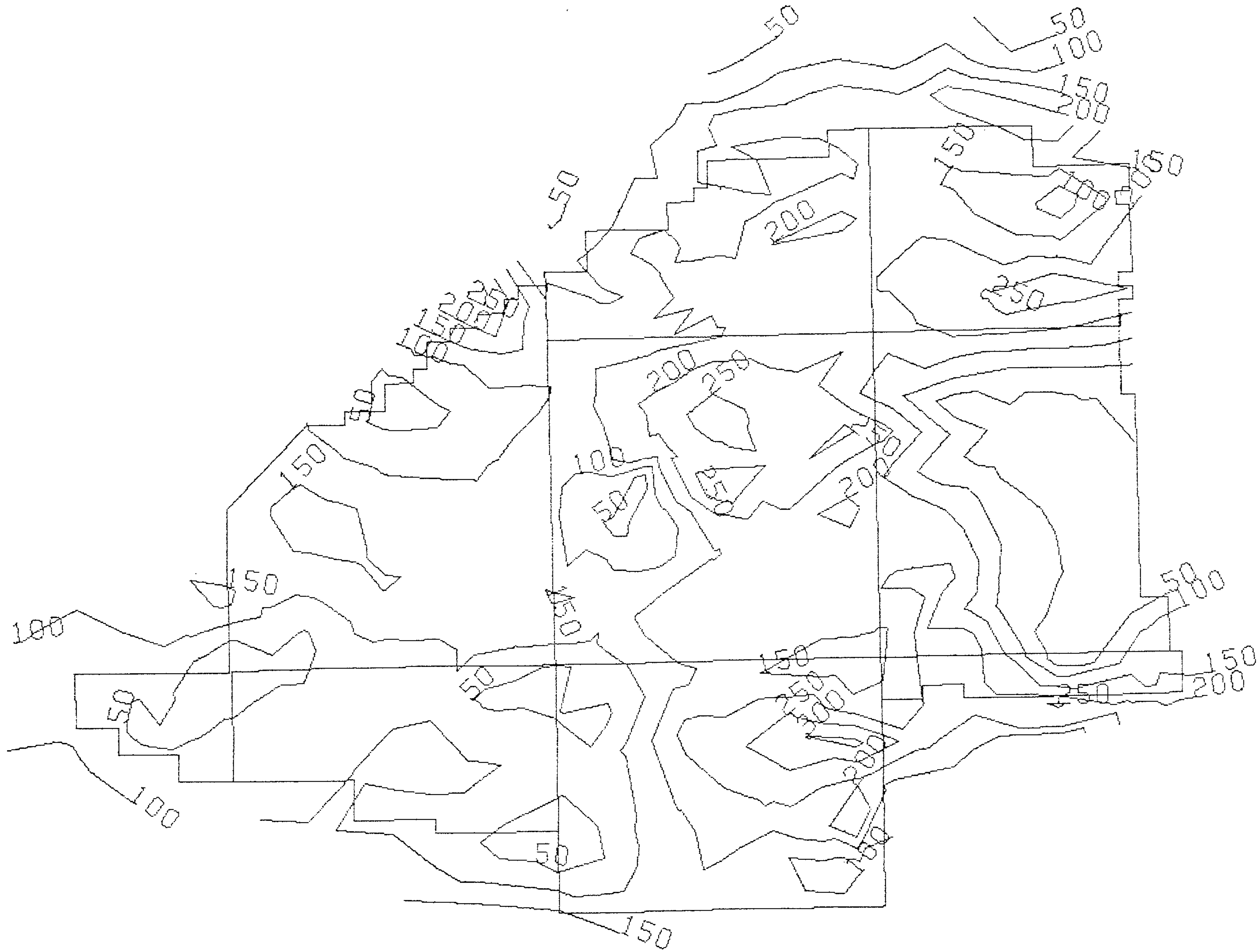



Figure 17a.--Simulated elevation of water level, in feet above base of aquifer, under scheme 01 conditions, Spring 1990.

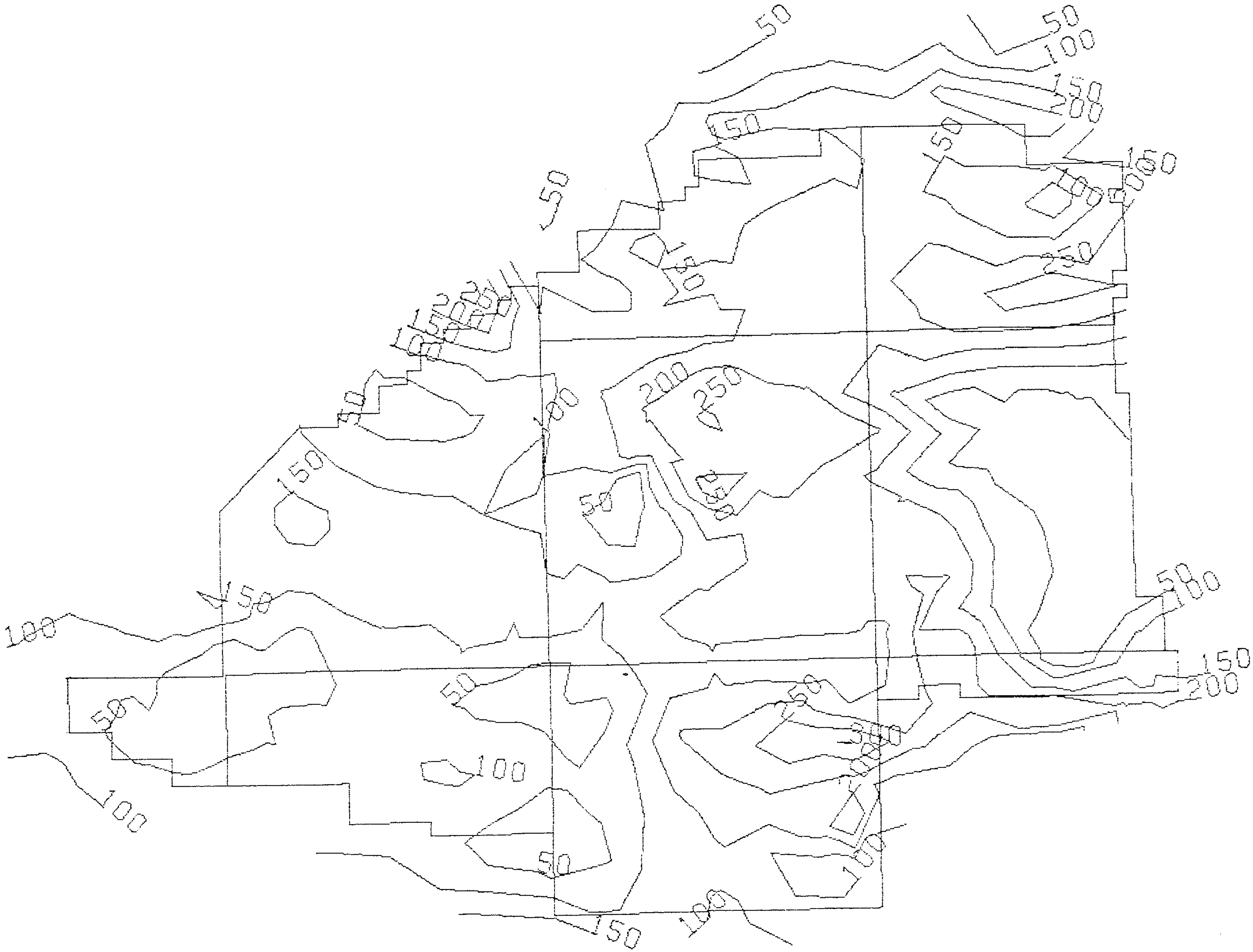


Figure 17b.--Simulated elevation of water level, in feet above base of aquifer, under scheme 01 conditions, Spring 2000.

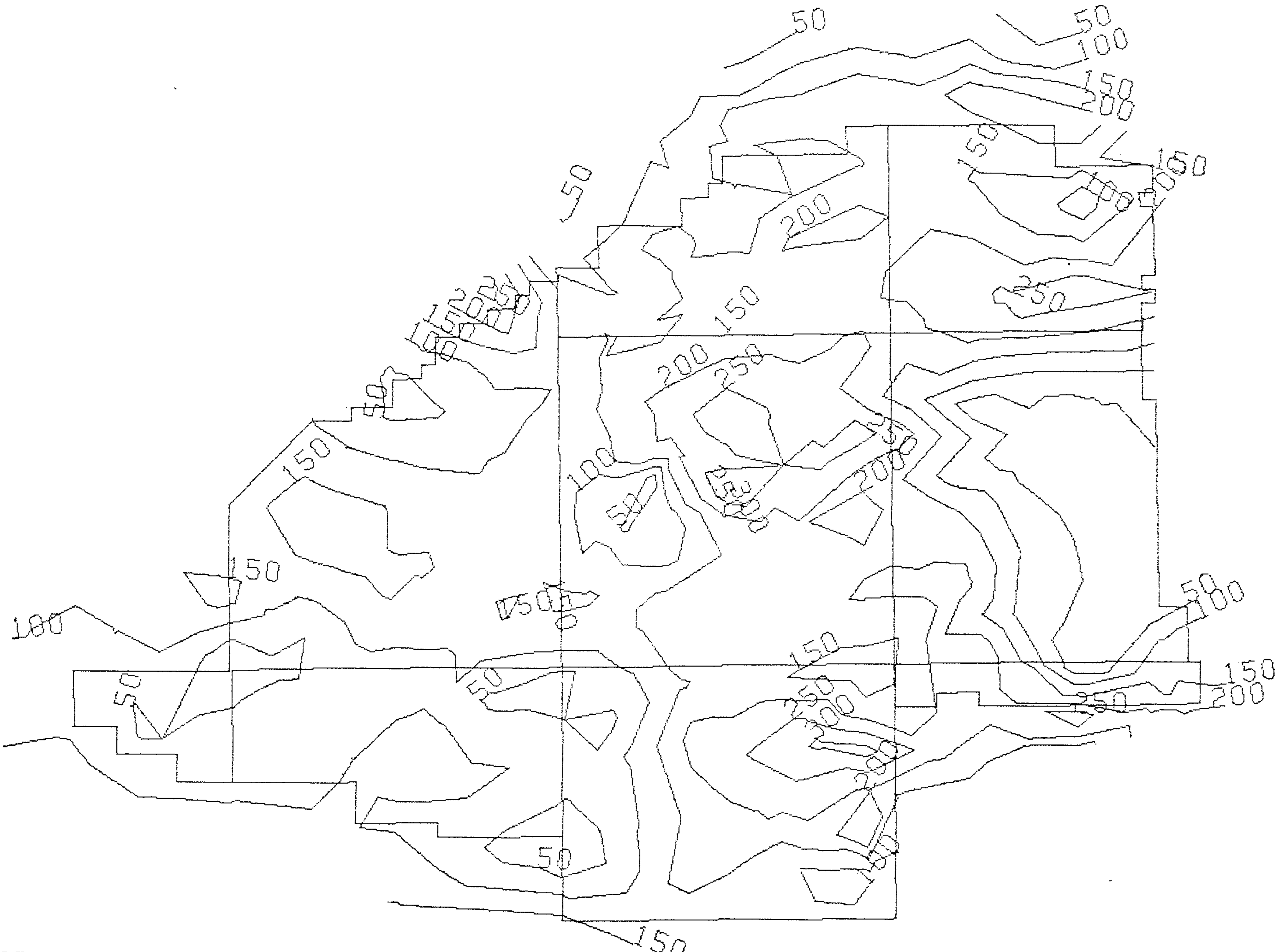



Figure 17c.--Simulated elevation of water level, in feet above base of aquifer, under scheme 02 conditions, Spring 2000.

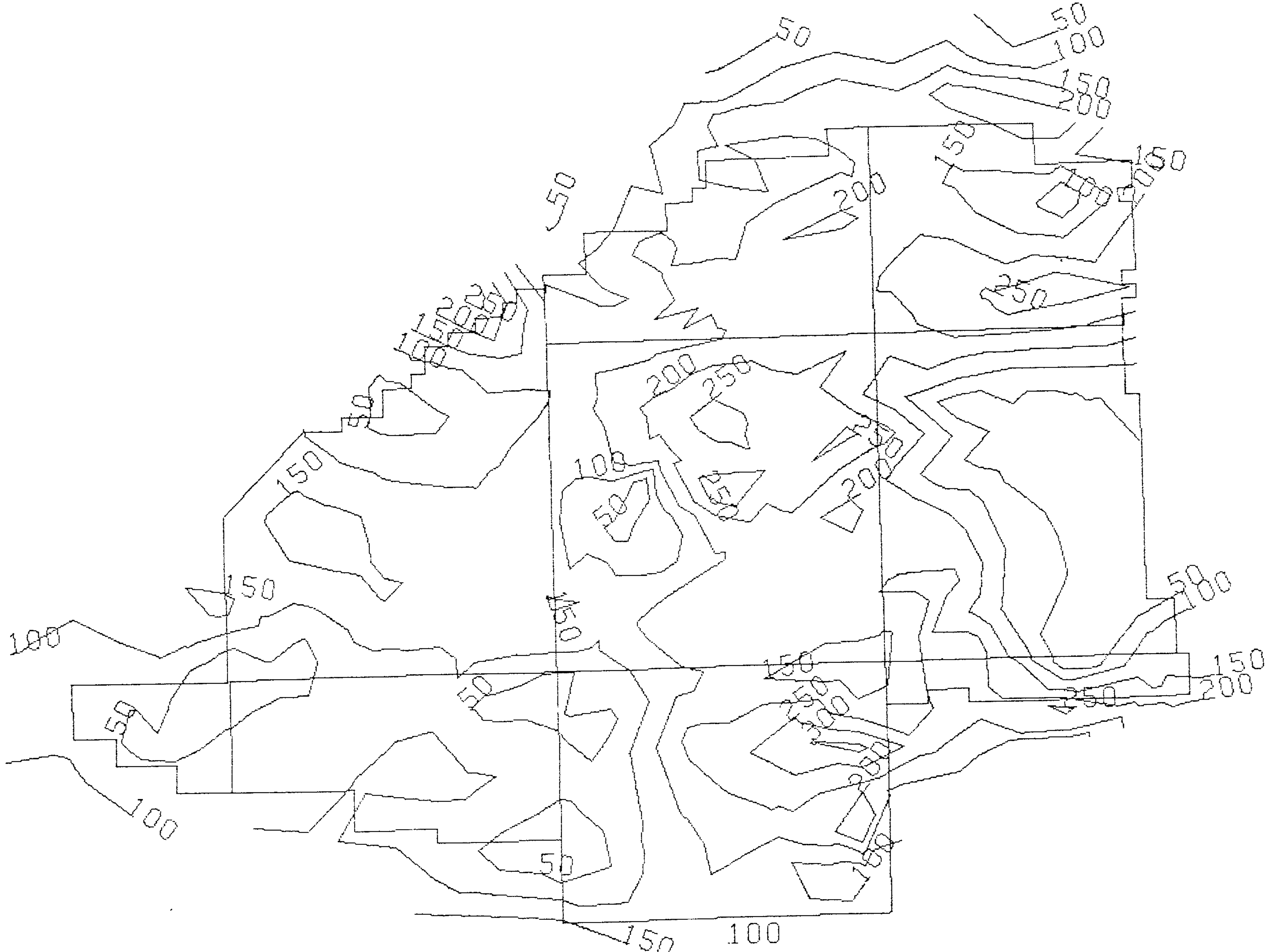


Figure 17d.--Simulated elevation of water level, in feet above base of aquifer, under scheme 03 conditions, Spring 2000.

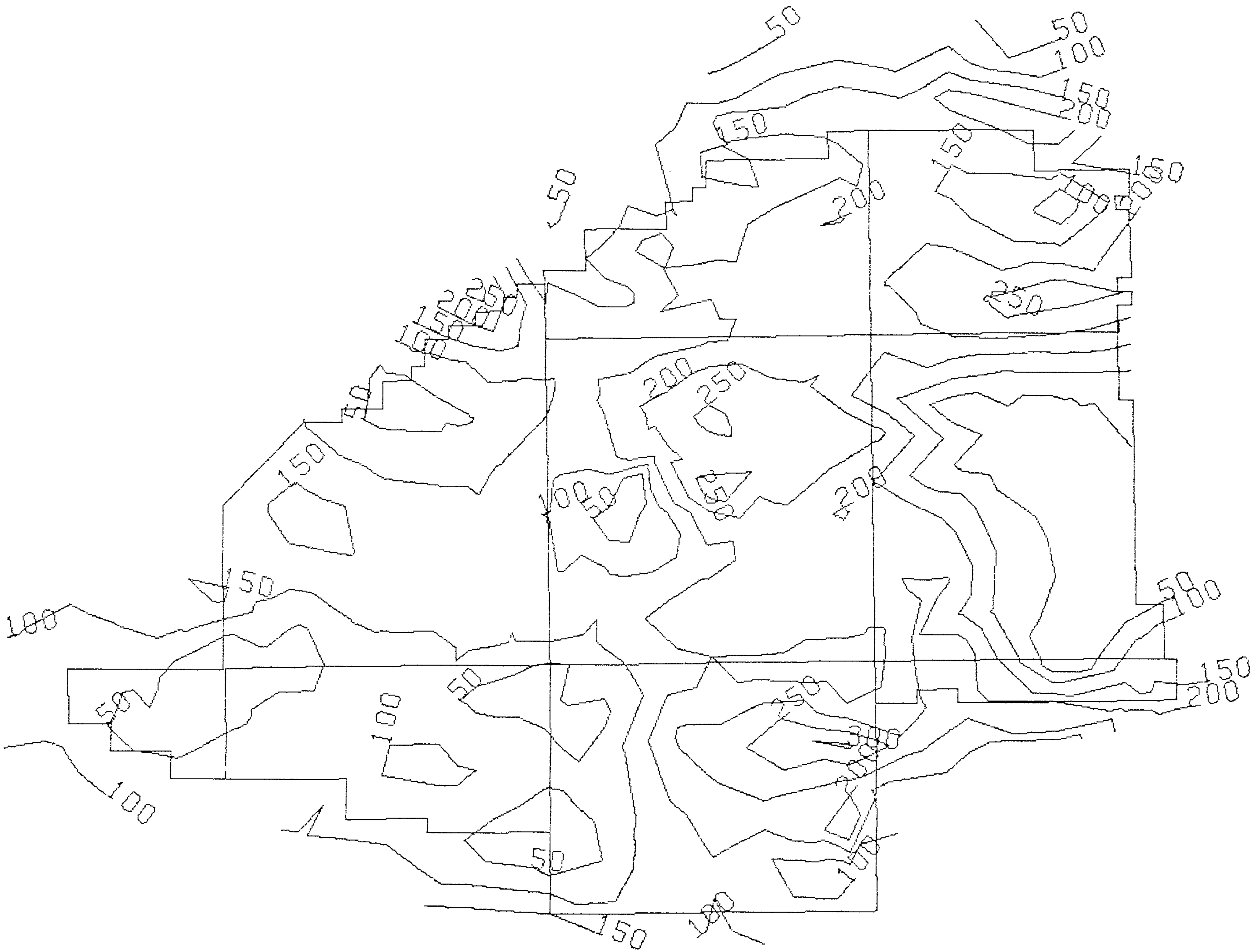



Figure 17e.--Simulated elevation of water level, in feet above base of aquifer, under scheme 04 conditions, Spring 2000.

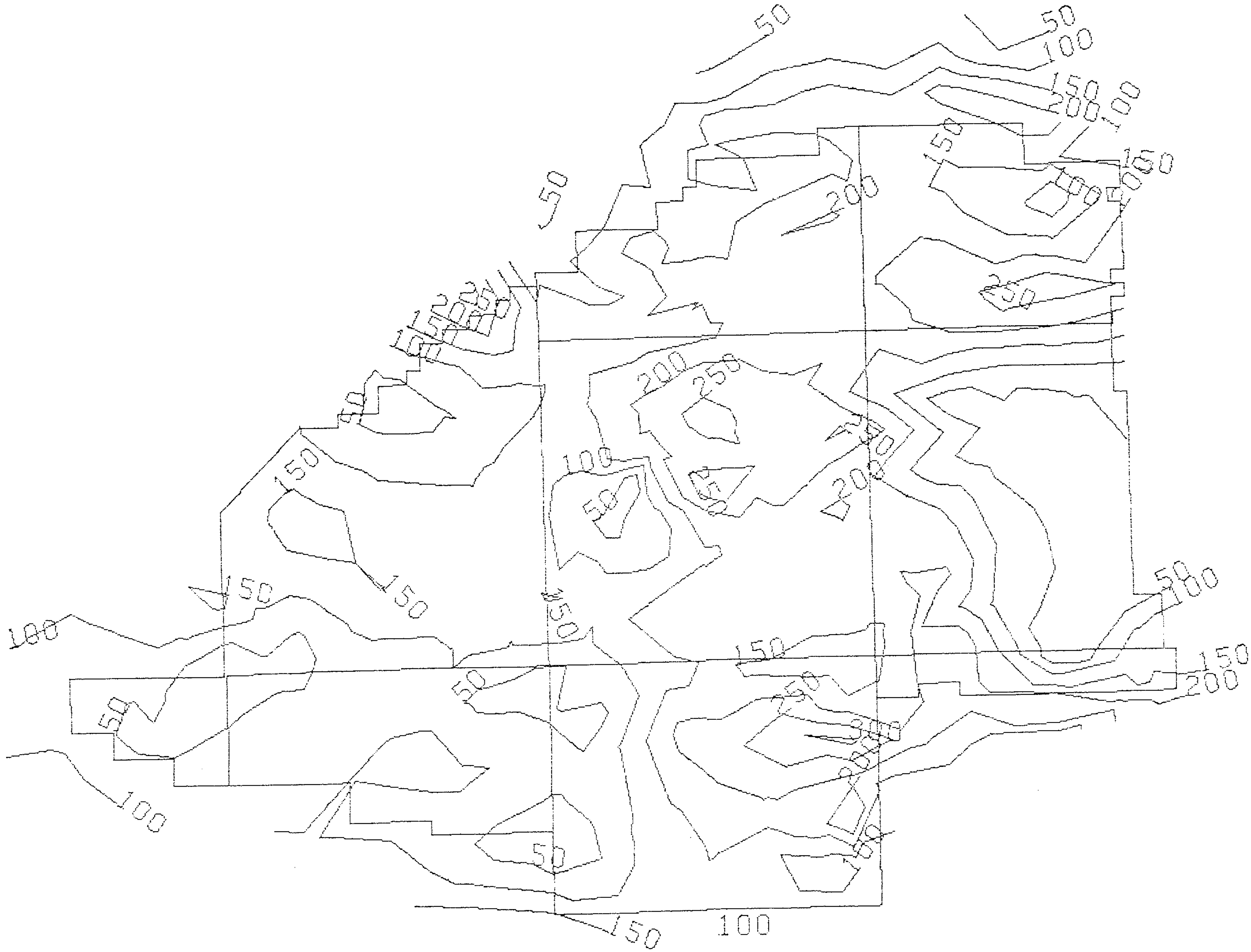


Figure 17f.--Simulated elevation of water level, in feet above base of aquifer, under scheme 05 conditions, Spring 1990.

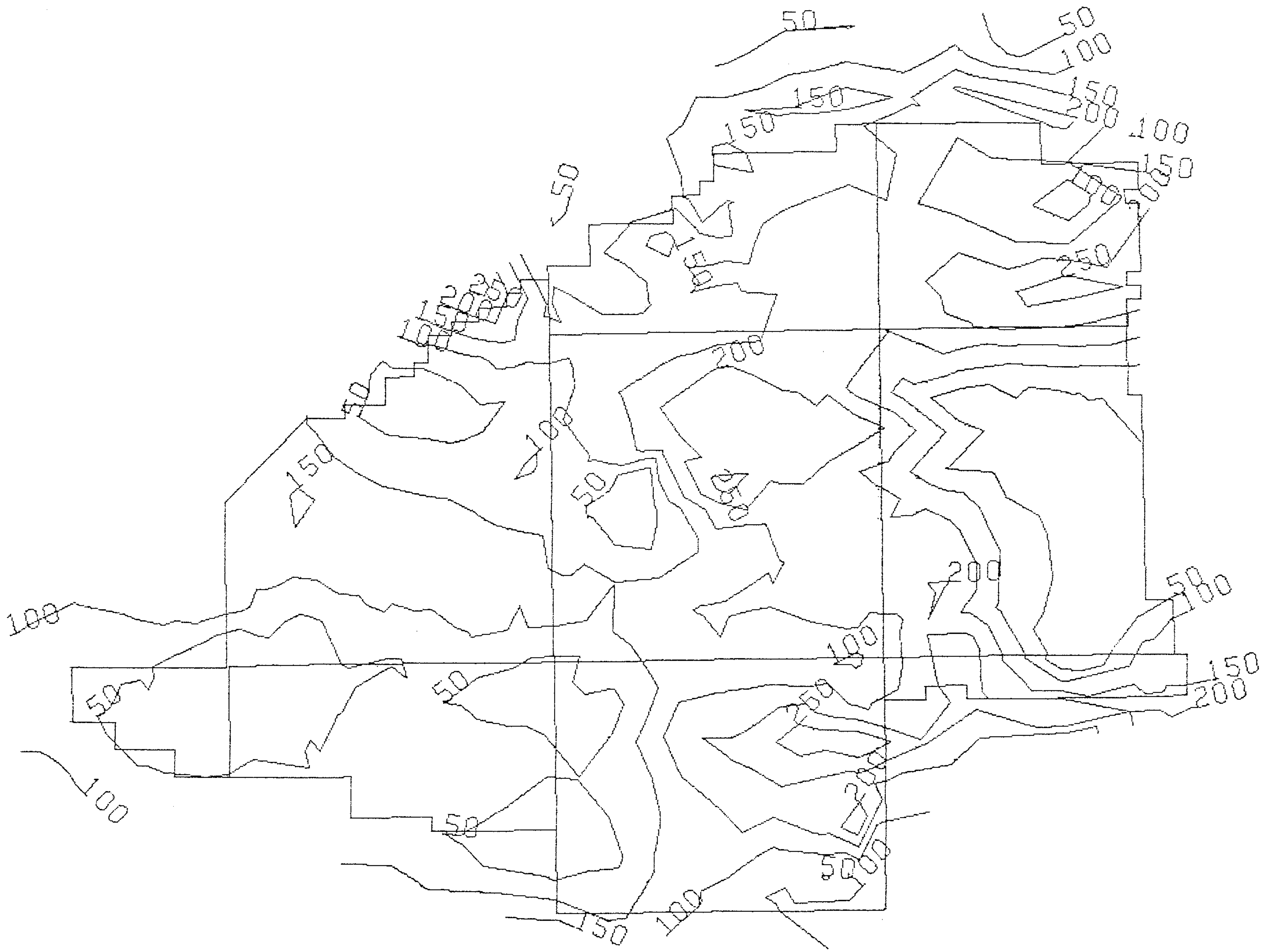


Figure 17g.--Simulated elevation of water level, in feet above base of aquifer, under scheme 05 conditions, Spring 2000.

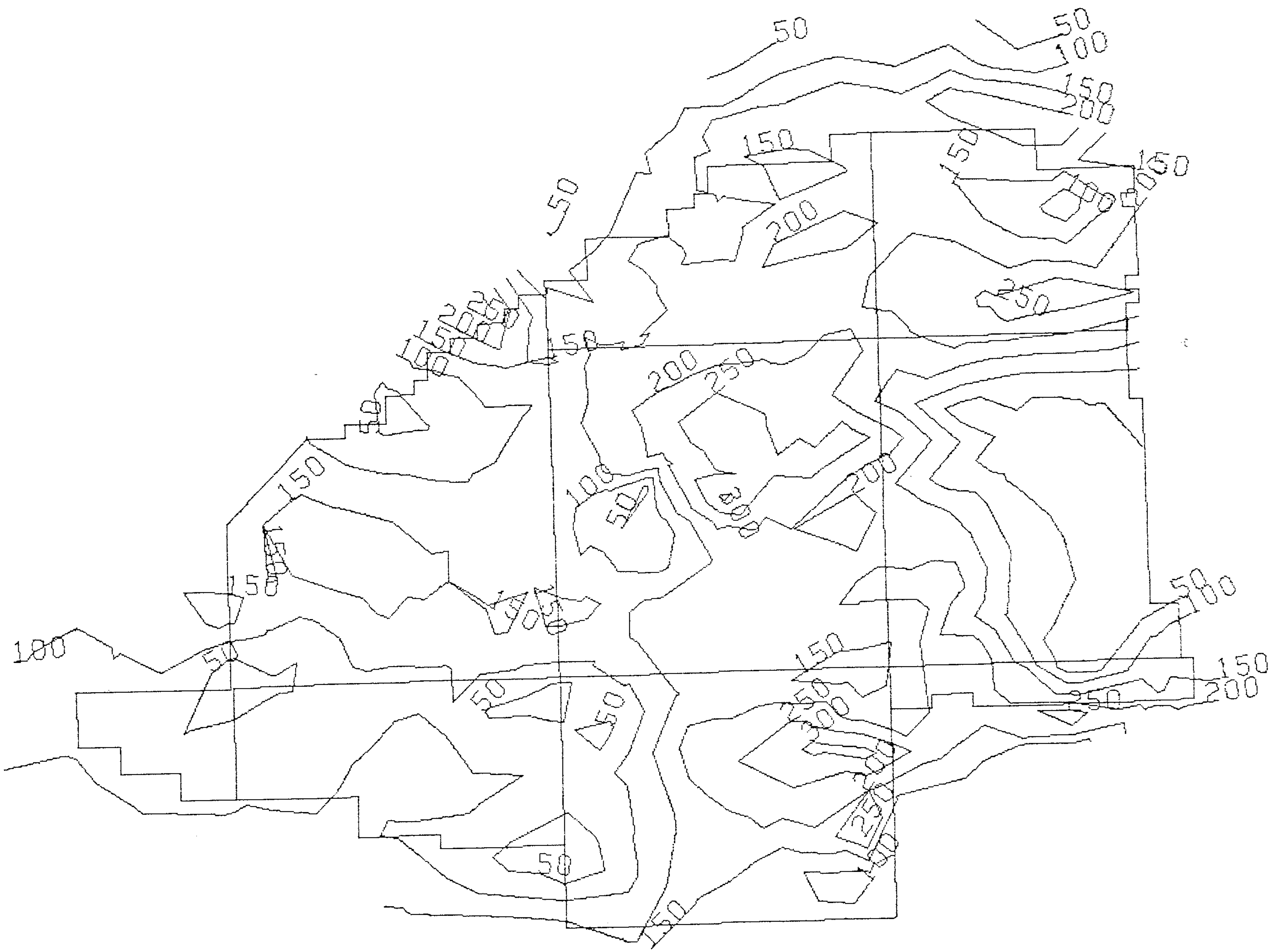



Figure 17h.--Simulated elevation of water level, in feet above base of aquifer, under scheme 06 conditions, Spring 1990.

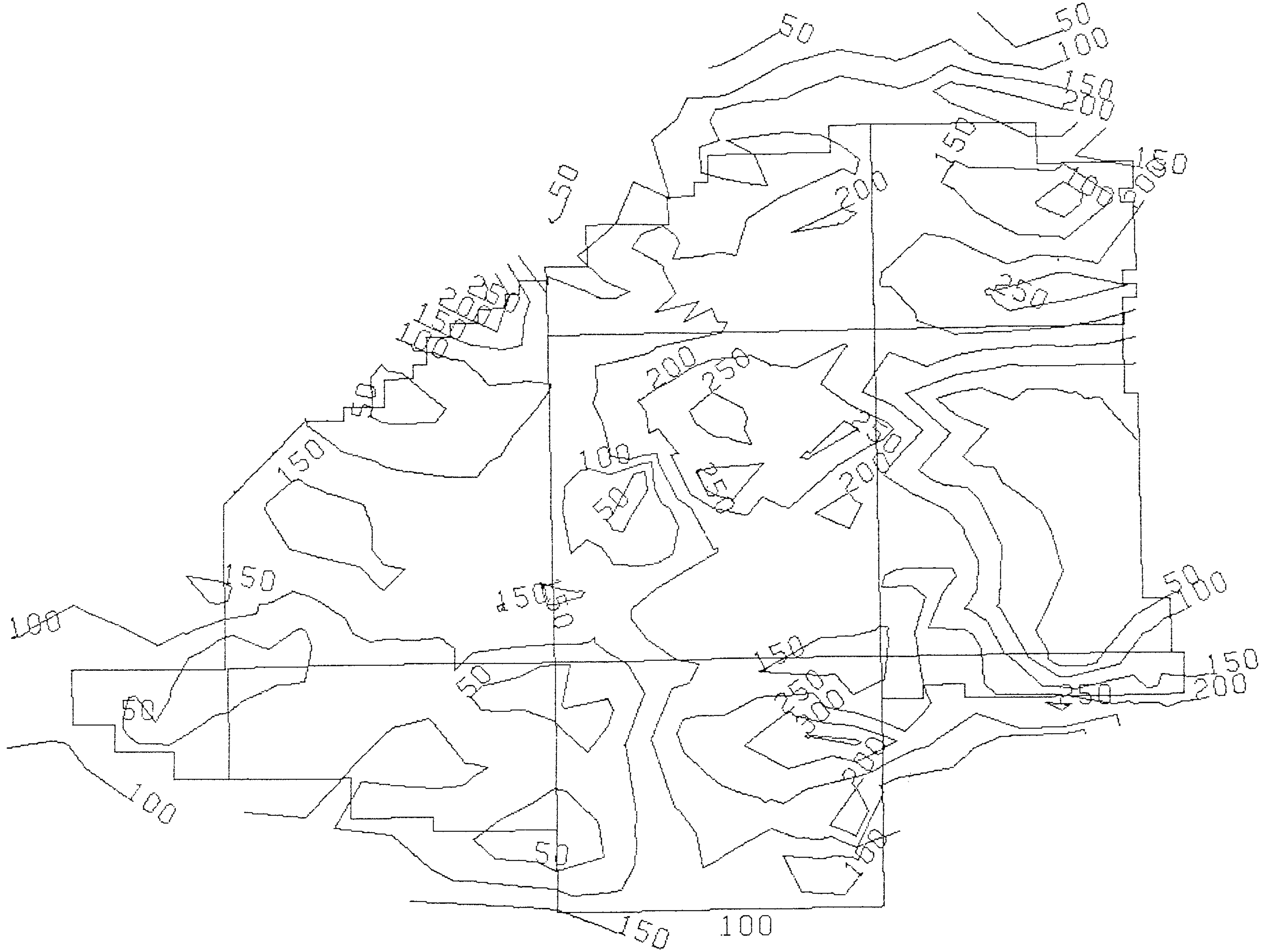



Figure 17i.--Simulated elevation of water level, in feet above base of aquifer, under scheme 06 conditions, Spring 2000.

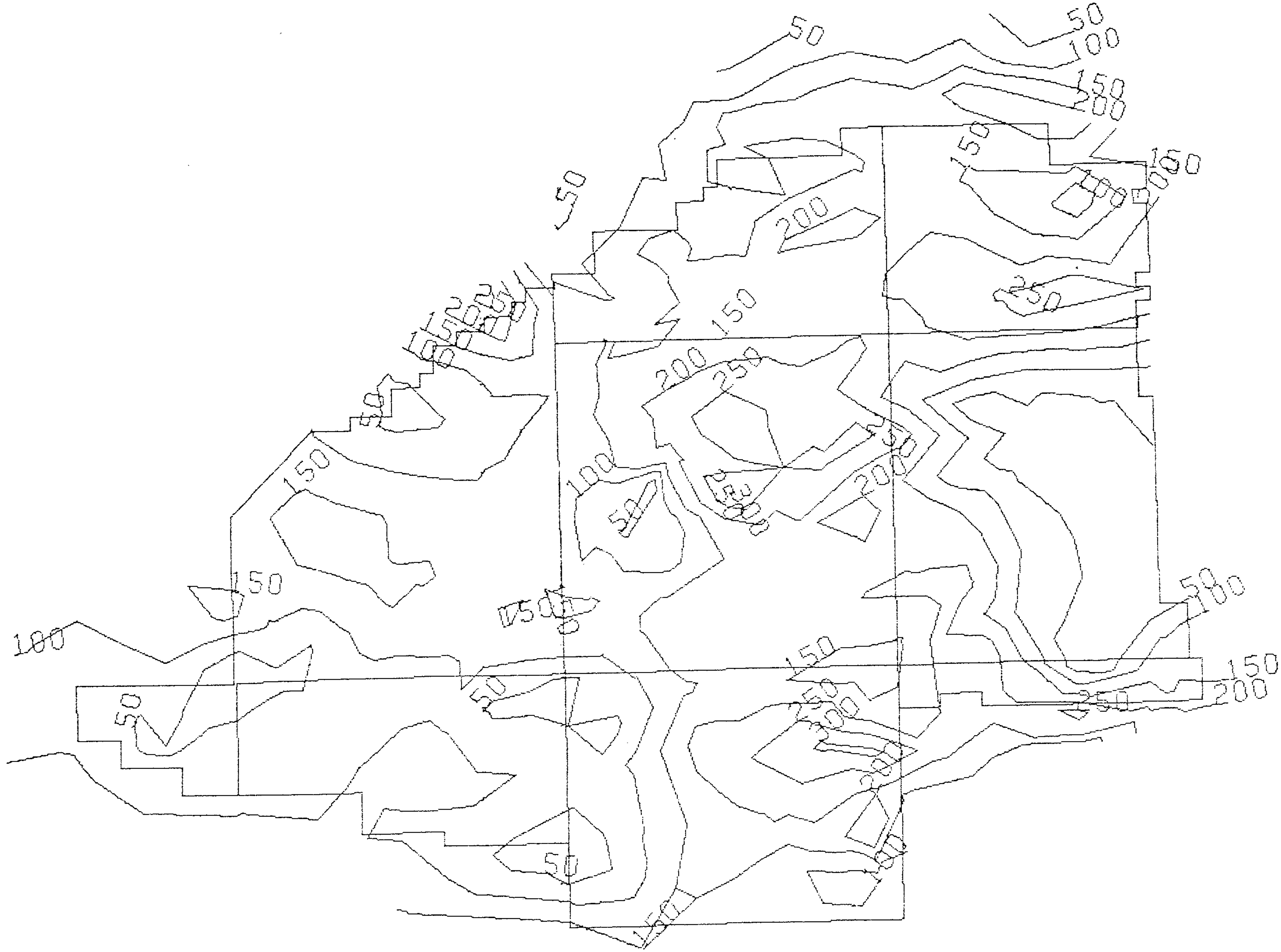



Figure 17j.--Simulated elevation of water level, in feet above base of aquifer, under scheme 07 conditions, Spring 1990.

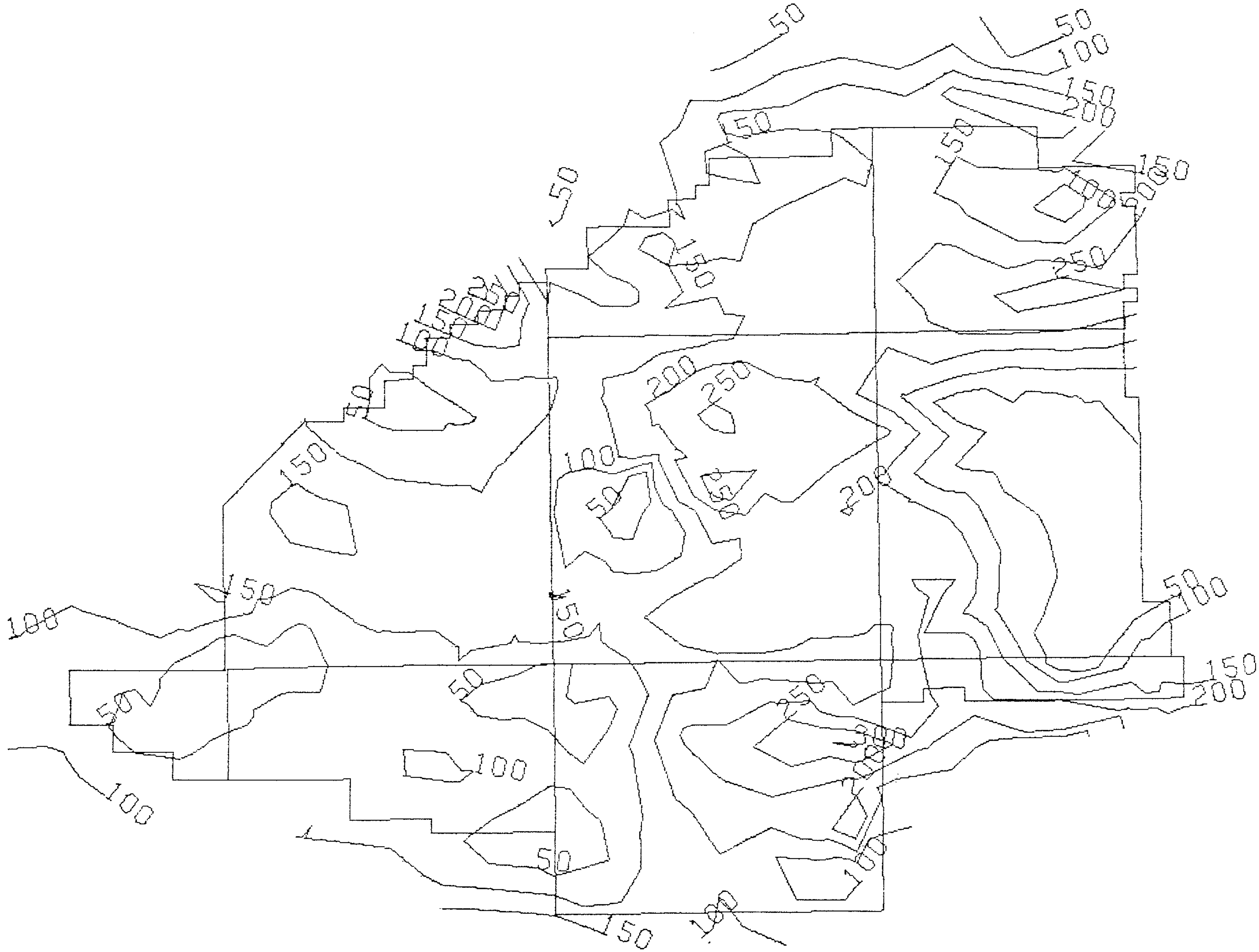



Figure 17k.--Simulated elevation of water level, in feet above base of aquifer, under scheme 07 conditions, Spring 2000.

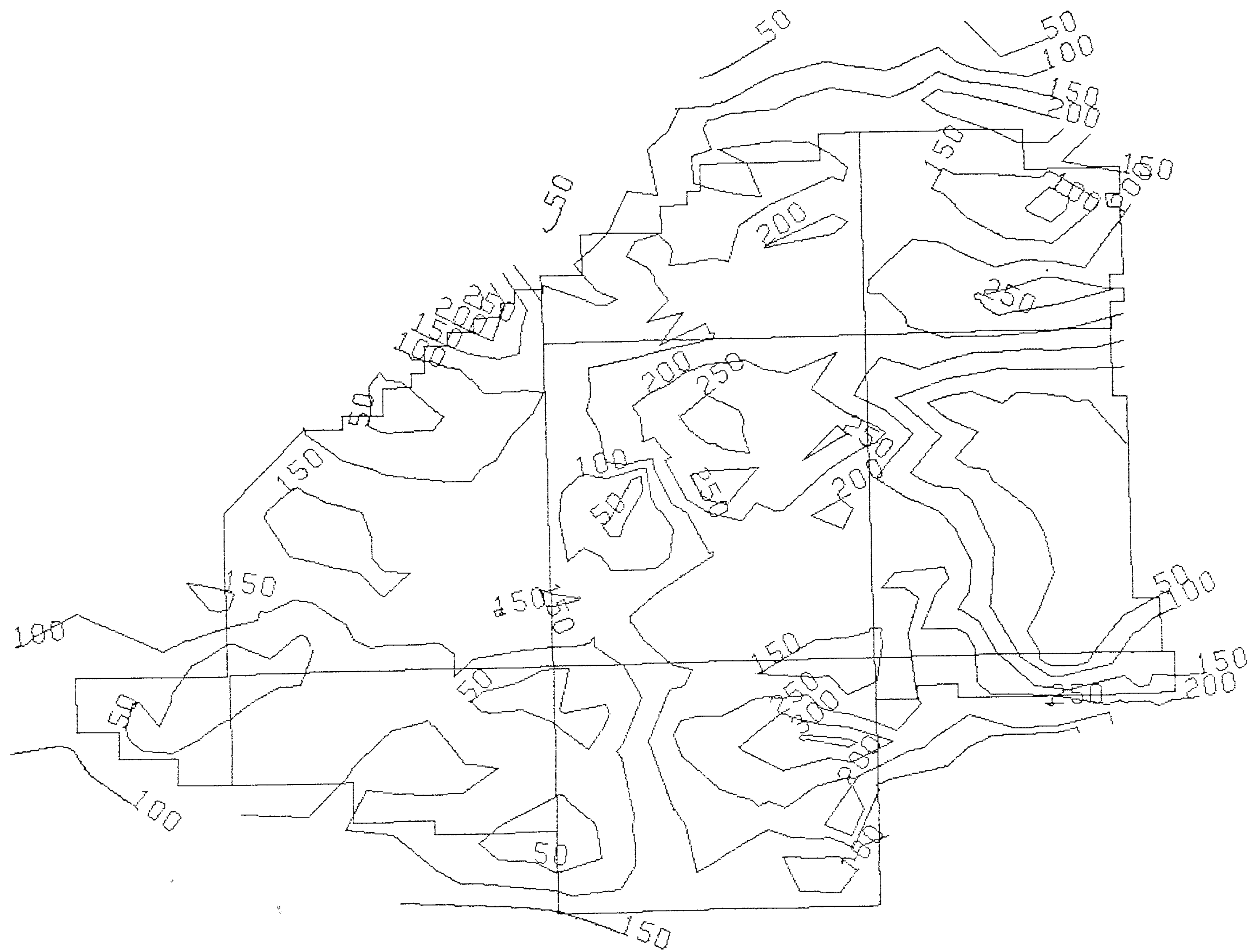



Figure 17ℓ.--Simulated elevation of water level, in feet above base of aquifer, under scheme 08 conditions, Spring 1990.

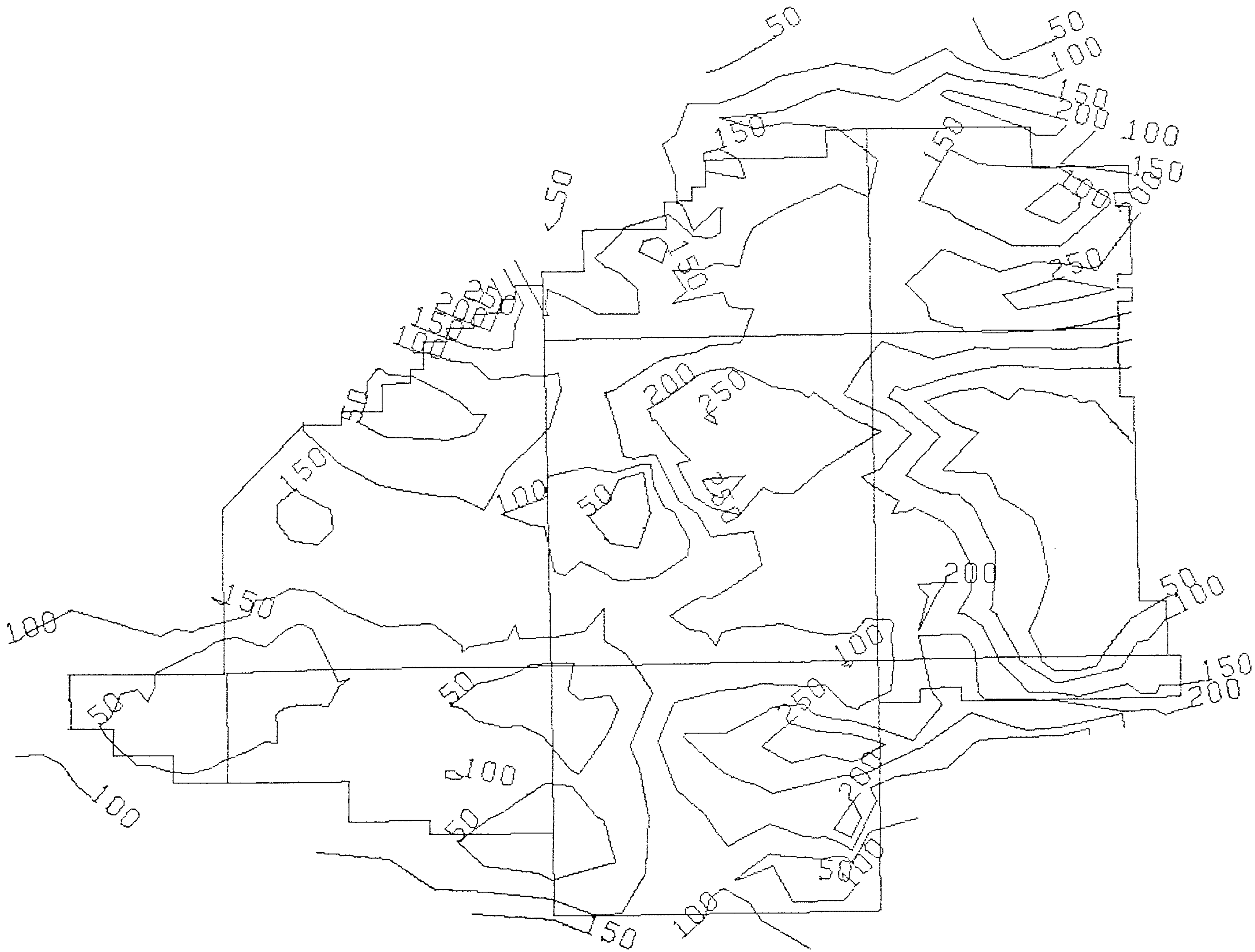



Figure 17m.--Simulated elevation of water level, in feet above base of aquifer, under scheme 08 conditions, Spring 2000.

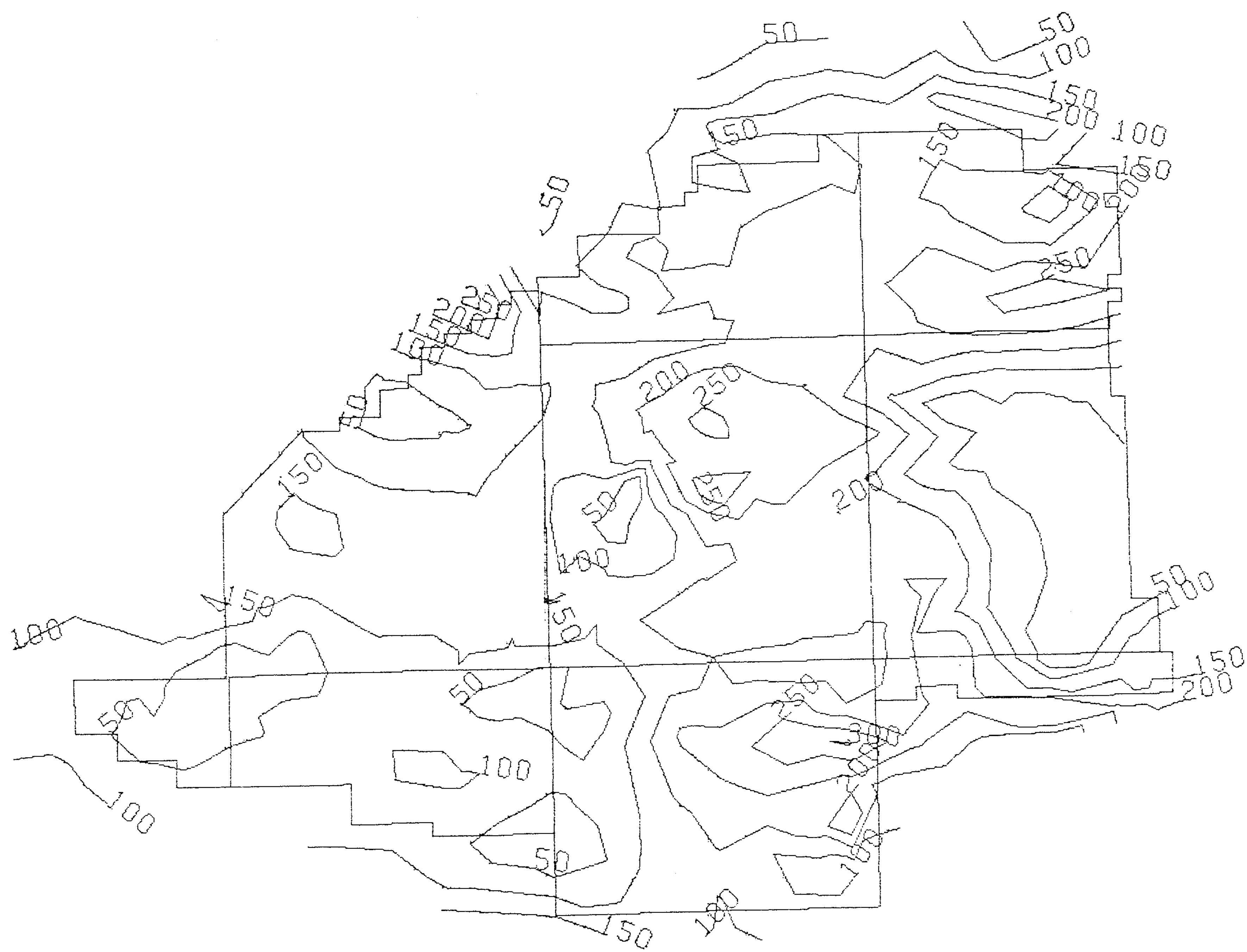


Figure 17n.--Simulated elevation of water level, in feet above base of aquifer, under scheme 09 conditions, Spring 1990.

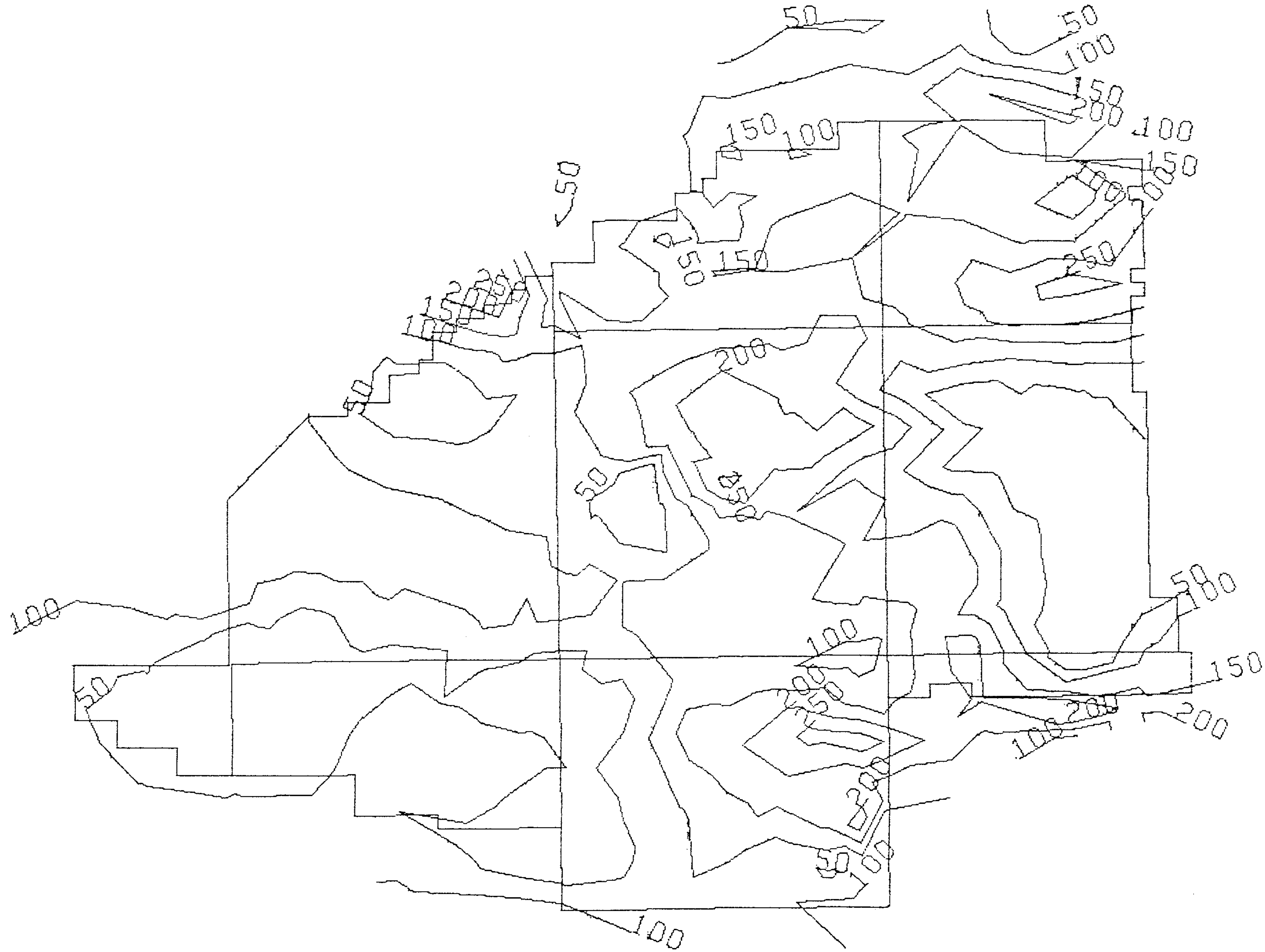



Figure 17o.--Simulated elevation of water level, in feet above base of aquifer, under scheme 09 conditions, Spring 2000.

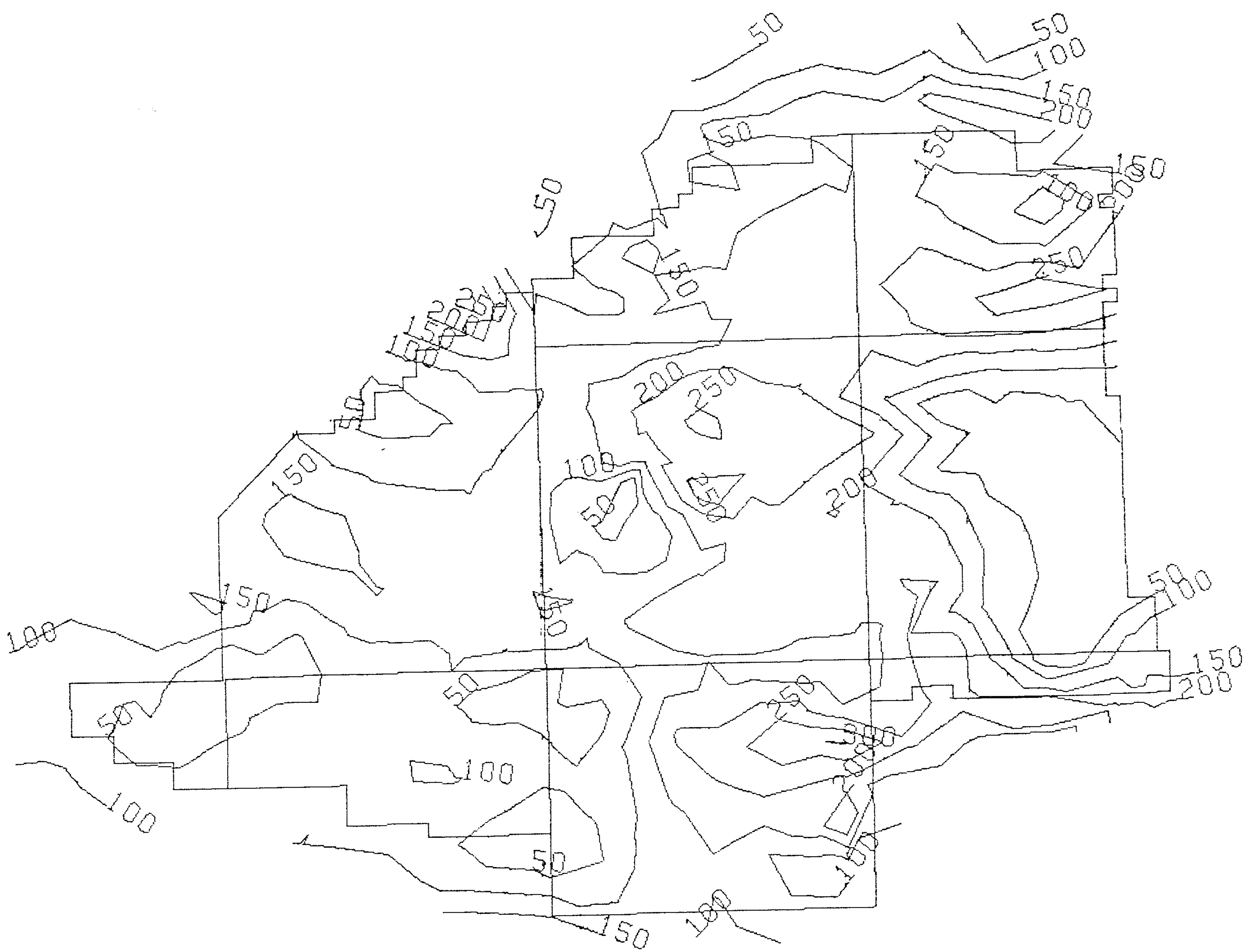



Figure 17p.--Simulated elevation of water level, in feet above base of aquifer, under scheme 10 conditions, Spring 2000.

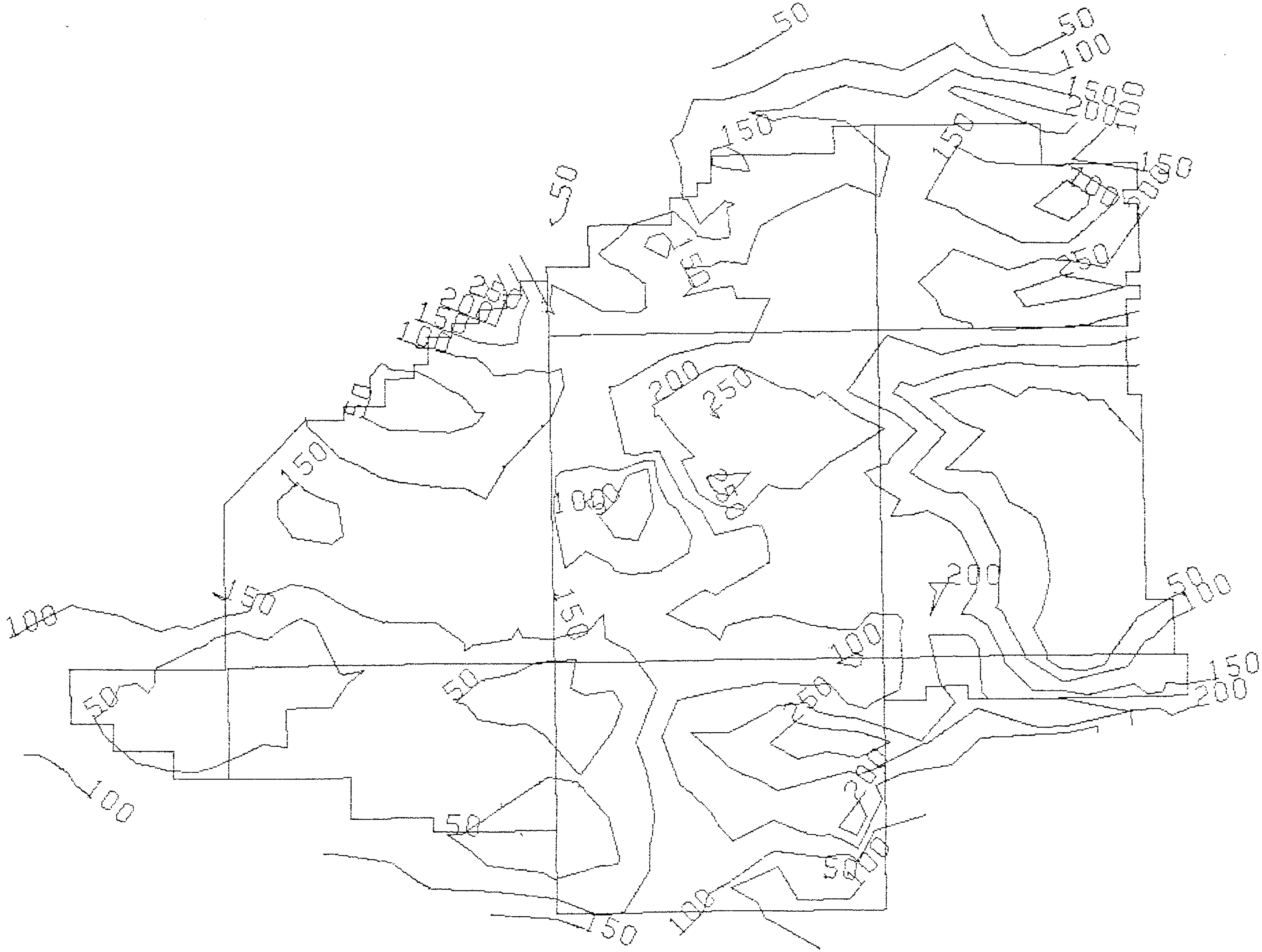


Figure 17q.--Simulated elevation of water level, in feet above base of aquifer, under scheme 11 conditions, Spring 2000.

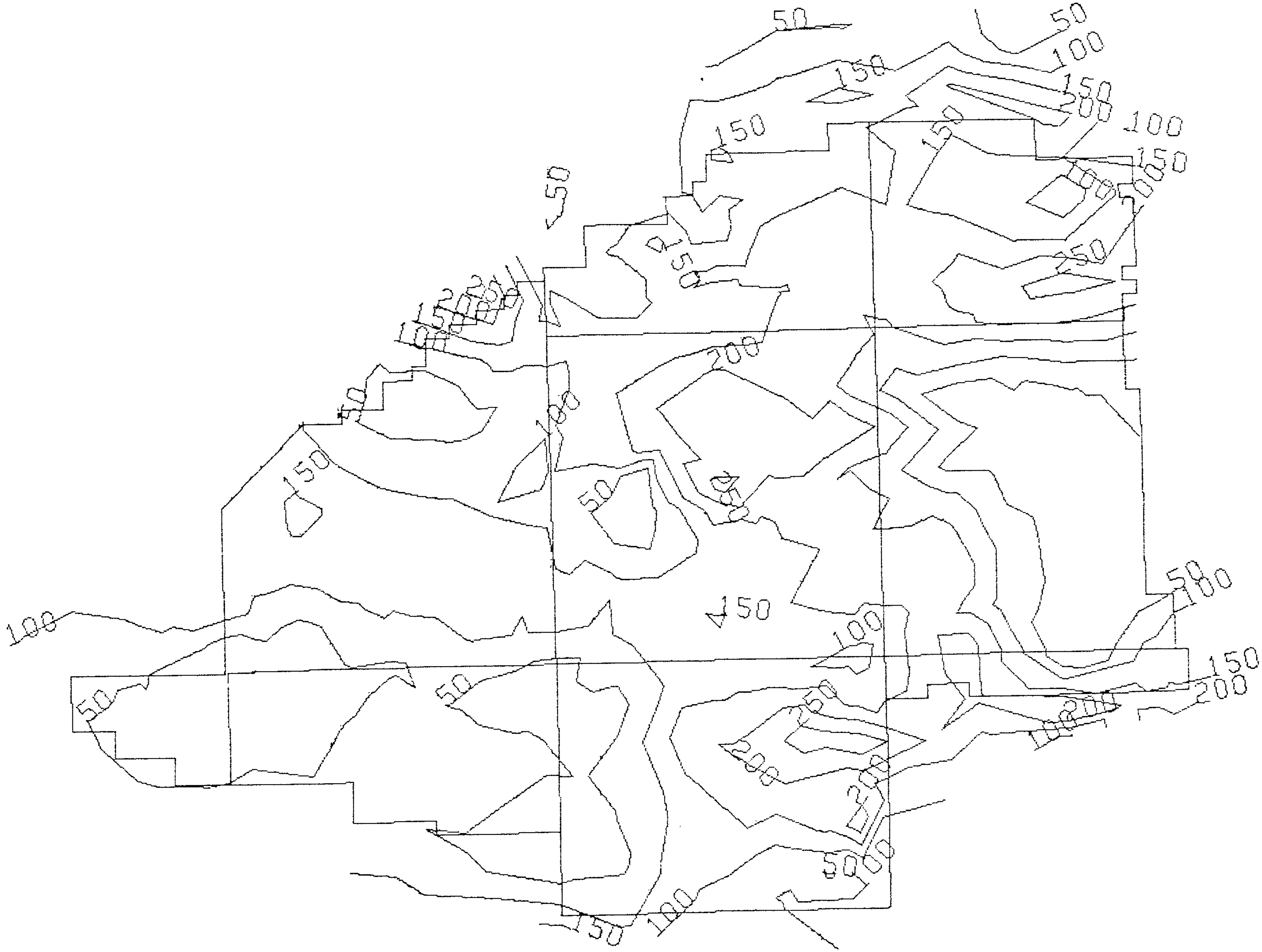


Figure 17r.--Simulated elevation of water level, in feet above base of aquifer, under scheme 12 conditions, Spring 2000.

ELEVATION OF WATER LEVEL ABOVE BASE OF AQUIFER, SPRING 1975

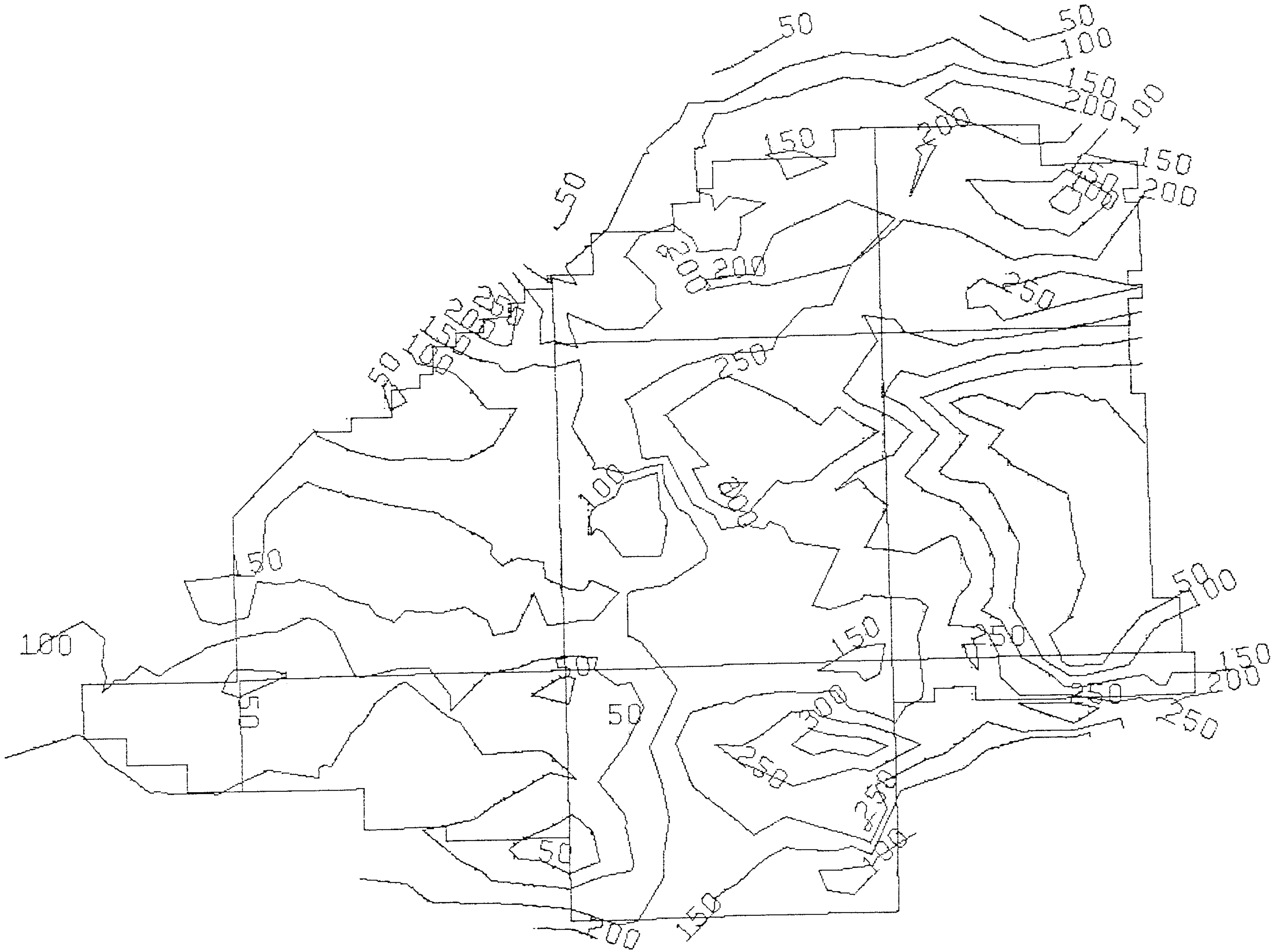


Figure 18.--Elevation of water level in feet above base of aquifer, Spring 1975.

## REFERENCES

Boulton, N. S., and Pontin, J. M. A. 1971. An extended theory of delayed yield from storage applied to pumping tests in unconfined anisotropic aquifers. Journal of Hydrology, v. 14, p. 53-65.

Gibbs, N. E., Poole, W. E., Jr., and Stockmeyer, P. K. 1976. An algorithm for reducing the bandwith and profile of a sparse matrix. Soc. Indust. and Applied Mathematics, Journal of Numerical Analysis, v. 13, p. 236-250.

Jess, J. M. 1970. Use of groundwater for irrigation, Seward County, Nebraska. University of Nebraska-Lincoln, Conservation and Survey Division, Nebraska Water Survey Paper 25, 24 pp.

Lappala, E. G. 1978. Quantitative hydrogeology of the Upper Republican Natural Resources District, southwest Nebraska. U.S. Geological Survey Water-Resources Investigations 78-38, 154 pp.

Logan, J. 1964. Estimating transmissibility from routine production tests of water wells. Groundwater, v. 2, (1) p. 35-37.

Lohman, S. W., et al. 1972. Definitions of selected groundwater terms--revisions and conceptual refinements. U.S. Geological Survey Water-Supply Paper 1988, 21 pp.

Pinder, G. F. 1974. A Galerkin-finite element model for aquifer evaluation. Princeton University, Water Resources Program, 29 pp.

Pinder, G. F., and Gray, W. G. 1977. Finite element simulation in surface and subsurface hydrology. Academic Press, 295 pp.

Reed, E. C. 1969. Underground water areas, University of Nebraska-Lincoln, Conservation and Survey Division map.

Steele, E. K., Jr. 1971a. Use of groundwater for irrigation in Hamilton and York counties, Nebraska. University of Nebraska-Lincoln, Conservation and Survey Division, Nebraska Water Survey Paper 27, 44 pp.

Steele, E. K., Jr. 1971b. Use of groundwater for irrigation in 1970, Hamilton and York counties, Nebraska. University of Nebraska-Lincoln, Conservation and Survey Division, Nebraska Water Survey Paper 30, 33 pp.

__________. 1972. Use of groundwater for irrigation in 1970, Clay County, Nebraska. University of Nebraska-Lincoln, Conservation and Survey Division, Nebraska Water Survey Paper 32, 40 pp.

Steele, E. K., Jr. 1973. Use of ground water for irrigation in Seward County, Nebraska in 1971. U.S. Geological Survey Open-file Report 7308, 28 pp.

## SELECTED BIBLIOGRAPHY

Ellis, M. J., and Pederson, D. T. 1977. Groundwater levels in Nebraska, 1976. University of Nebraska-Lincoln, Conservation and Survey Division, Nebraska Water Survey Paper 44, 96 pp.

Huntoon, P. W. 1974. Predicted water-level declines for alternative groundwater developments in the Upper Big Blue River basin, Nebraska. University of Nebraska-Lincoln, Conservation and Survey Division, Resource Rpt. 6, 102 pp.

Johnson, C. R., and Keech, C. F. 1959. Geology and ground-water resources of the Big Blue River basin above Crete, Nebraska. U.S. Geological Survey Water-Supply Paper 1474, 94 pp.

Keech, C. F. et al. 1967. Availability of ground water in York County, Nebraska. U.S. Geological Survey Water-Supply Paper 1839F, 17 pp.