1978/12

DEPARTMENT OF ~~NATIONAL RESOURCES~~ NATIONAL DEVELOPMENT

BUREAU OF MINERAL RESOURCES, GEOLOGY AND GEOPHYSICS

Record 1978/12

HYDRAULIC CALIBRATION OF THE GABHYD MODEL OF THE GREAT ARTESIAN BASIN

by

G.E. Seidel

Record 1978/12

HYDRAULIC CALIBRATION OF THE GABHYD MODEL OF THE GREAT ARTESIAN BASIN

by

G.E. Seidel

CONTENTS

TABLE

FIGURES

SUMMARY

Data for use by GABHYD, the finite difference digital model of the Great Artesian Basin, are subject to measurement errors and additionally to errors due to interpolation. Interpolation and extrapolation are necessary to specify data on all gridpoints of the model. A process of calibration had to be developed to remove the resulting data inconsistencies so that the model could function properly and produce predictions of acceptable accuracy. An 'inverse' method of calibration was developed by inverting the model equation applied to pairs of gridpoints rather than individual points. This method was used to determine aquifer transmissivities from aquifer potentials. The potentials themselves were subjected to smaller adjustments simultaneously with calculations of transmissivities to achieve an optimum overall consistency of data. Storage coefficients were approximated from the changes in water balances over the main development periods of the basin. To ensure the uniqueness of the calibration the boundary conditions were firstly the vertical permeabilities, secondly the discharge boundary transmissivities, and thirdly the well discharges. Although the first two could only be estimated, the well discharges are recorded data. As a result the calibration is sufficiently determined for the developed portion of the basin and is approximate only for the remainder. Sample model runs further verified the calibraiton by reproducing recorded data for the developed areas. Program descriptions, flowcharts, and printouts are presented for the computer programs used for the model calibration.

1. INTRODUCTION

1.1 Notes on the presentation

This Record is intended to provide a complete guide to the calibration of the GABHYD hydraulic model of the Great Artesian Basin. To achieve a satisfactory degree of self-sufficiency it was necessary to include chapters on matters, which are not exclusive to the model calibration, e.g. the problem of data availability and the derivation of the model equation. Some duplication of other parts of the model documentation became inevitable as a consequence.

Furthermore it was realised that interest in the model calibration may occur at different levels of detail. This documentation is correspondingly presented in three sections.

Section 1, the Introduction, is at the most general level. It provides the elementary definitions, structure of the model system, alternative calibration methods, and a summary of the results. It is aimed at the professional or administrator who is not specifically concerned with groundwater modelling but requires some knowledge of its application.

Section 2 is provided for the hydrogeologist or hydrologist concerned with only the use rather than the development of the model. It concentrates on the link between actual hydrogeological data and the model, problems of data uncertainty and unavailability, uniqueness of the model calibration, and provides a general description of the calibration programs and of the interactions between them.

Section 3 and the Appendix provide the detailed information necessary to actually carry out a model calibration with the programs or to modify the programs if found necessary. This level of detail is aimed at the modelling and computer programming specialist.

However, a deliberate attempt has been made to restrict the use of specialised jargon and of lengthy mathematical statements to the bare minimum compatible with logical precision. This should facilitate the exchange of ideas between the hydrogeologist and the modelling specialist even at this most detailed level.

1.2 Definitions

To define calibration it is necessary to define the concept of a model and its relation to reality first.

The original data on which models are constructed, represent reality as it is known and measured. The collection of these data in most cases, including this one, is based on practical convenience rather than the needs of some future analyst. For this reason data are clustered in areas and times of intense development of the Great Artesian Basin and almost totally absent in other areas and times. Such an irregular data base does not lend itself for use by a computer-based model without at least some restructuring. A data base suitable for analysis by such a model, obtained by restructuring and transforming the original data, is referred to as a model prototype.

For the Great Artesian Basin prototype the large number of individual aquifers are grouped into only two, and the whole area is subdivided into squares of uniform size with data defined for each of them.

In terms of systems analysis the prototype consists of system parameters - e.g. transmissivities and storativities, called hydraulic parameters in this case - and system variables - e.g. well flows and potentials (water-pressure heads). A numerical model is defined as any process, which when using the numerical equivalents of the system parameters together with the appropriate starting conditions and boundary conditions will reproduce numerical equivalents of the system variables with satisfactory accuracy. Applied to the Great Artesian Basin, the prototype values for the hydraulic parameters, the transmissivities, and storativities are inserted into the model. Then the model is set to the known starting conditions of potentials and discharges and is run. While running, the model produces values for potentials and discharges, which will match the known prototype values for the same time and place if the model is correctly calibrated. The model may be allowed to run beyond the time for which data (system variables) had been recorded. If the match between the model-produced data and the recorded data was good for the period of records, then the match between data produced by the model for the future and actual data not yet recorded is expected to be also good, i.e. the model can be used for predictions.

However, for predictions to be valid two conditions must be met. The model must be a valid representation of the real physical process, and the prototype must be a valid representation of the real physical quantities. The validity of the model is not the object of this Record, but the validity of the prototype as expressed by calibration is. We can do no more than assume that the data which are available represent the reality with sufficient accuracy, but we can insure that the prototype is in accordance with the available data. This is the objective of model calibration.

Part of the calibration process is to decide on the degree of simplification and the type of discretisation to be employed for the prototype. Often, however, these decisions are severely constrained by logistic considerations of computer storage and processing time for mathematical models, or availability of suitable materials and equipment for physical models. The greatest difficulty in the calibration process is in trying to ensure that the prototype parameters and variables do not contradict one another. The original data, within the limitations of their accuracy of course, do not contradict one another because they are the result of an actual physical process. However, the values averaged and extrapolated from them for a complete definition of the prototype are subject to all the errors introduced by the averaging and extrapolating procedures and are generally no longer consistent with one another. Let us assume that extrapolation of potentials from one area where records are available to another where there is none resulted in a potential which is much too high. Similarly, extrapolation of storativities for the same area resulted in an error of similar magnitude but opposite effect whereas other parameters might have been extrapolated accurately. Generally then, the potentials will no longer be consistent with either the storativity or the other hydraulic parameters. Neither will the hydraulic parameter of storativity be consistent with the other parameters. The process of calibration, then, involves alterations to the variables and parameters within the limits of their certainty until a maximum consistency is achieved between and among them. To make parameters and variables consistent with one another, however, is equivalent to ensuring that the prototype is in accordance with the available recorded data, because the recorded data are an integral part of the parameters and variables of the prototype.

1.3 The GABHYD model

The GABHYD model consists of groups of computer programs organized around a common set of data files (Fig. 1). The design of the GABHYD model has benefitted significantly from the experiences gained from the GABSIM model (Ungemach, 1975). The GABSIM model is no longer in use since it was found unsuitable for application to the Great Artesian Basin. However, there is substantial common ground for both models, both using a rather conventional and well proven finite-difference approach as their mathematical basis. The prototype of the Great Artesian Basin for use by the GABHYD model is largely based on, and in its structure very similar to, the prototype originally developed for the GABSIM model (Audibert, 1976).

The GABHYD model as shown in Figure 1 is organised into four major program groups:

The GABBRI group of programs provides the link between the GAB data bank (Ungemach & Habermehl, 1973; Seidel, 1973; Krebs, 1973) and the GABHYD system by producing the original data base in the form of the GABHYD model data files.

The CALSYS group of programs concerns itself with the model prototype calibration. It accesses the GABHYD model data files and rewrites them, incorporating calibration changes. Description of the CALSYS subsystem of the model is the specific objective of this Record.

The RUNMOD group of programs consists of the model proper and one model management program. The model program reads the GABHYD model data files and the running instructions produced as coded data by the model management program. For each time-step the model then produces a complete set of model system variables, i.e. potentials and discharges as model output.

The OUTSYS group of programs has the functions of presenting the model output in a variety of forms and of comparing different sections of the model output with each other (e.g. drawdowns as difference of potentials at different times), or with model input data, or with data items recorded in the GAB data bank. The advantage of separating the output analysis and presentation from the model itself is that no decisions on the form or quantity of final output presentation have to be made before running the model. Furthermore, it is possible to represent the output in different form or based on different comparisons long after the original model run.

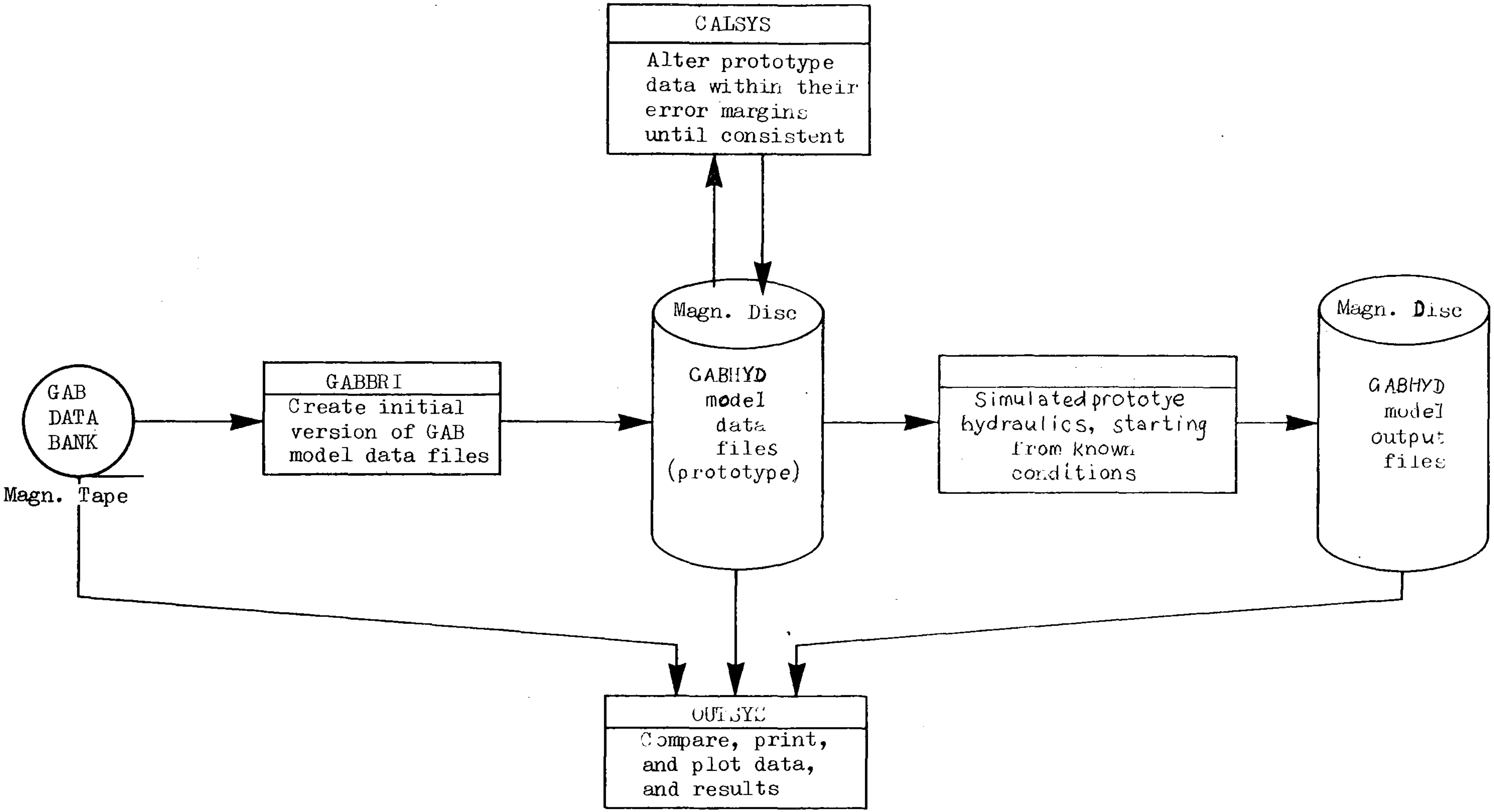


Fig.1 SCHEMATIC STRUCTURE OF THE GABHYD MODEL SYSTEM

The basic design philosophy for the program organisation was to confine each program unit to one logically independent step linked to the other units merely by a common data structure and by access to common data files. Such a system may be more time consuming to operate than the alternative of one complex program, but it reduces computer storage problems by minimising the size of the program code at any one time, maximizes development flexibility by allowing units to be altered or replaced without upsetting the others, and, most important, allows the hydrogeologist to separately monitor each logical step and check error magnitudes before they are swamped or concealed in subsequent steps.

1.4 Comparison of calibration methods

The most obvious and for small models often most efficient approach to calibration is by trial and error. Like all trial and error processes it is elementary in theory but becomes an art in application. The calibration of the Great Artesian Basin prototype for the original GABSIM model was also attempted by trial and error first. However, after an encouraging start, each further step towards an acceptable solution became progressively more difficult to find and less effective in result. During this phase it became obvious that pure trial and error would not provide a satisfactory answer. The number of possible permutations of different trials, even after restricting them to relatively coarse steps and grids coarser than the grid of model squares, is still much too high to even consider trying them out one by one. Clearly a more systematic search was required, and the different solutions to this problem, which have been published were studied.

Numerous papers describe the automated search or the optimisation approach (e.g. Kleinecke 1971, Emsellem & Marsily 1971, Hefez, Shamir, & Bear 1975). This method like that of trial and error involves running the model with a trial set of data and analysing the errors to determine corrections to the data. However, the data corrections are based on a mathematical algorithm rather than educated guesses, e.g. the use of linear programming to optimise the search. To achieve a unique solution these authors either rely on a large number of equations assumed to be independent (Hefez and others, 1975) or employ a criterion of 'uniformity' of the solution (Emsellam & Marsily 1971). Neither approach could be applied to the GABHYD model with satisfactory results, as is explained in 2.3 and Appendix B.

The 'direct approach' involves the solution of the model equation directly for the hydraulic parameters (Nelson 1968, Frind & Pinder 1973) of transmissivities and storativities. The major problem remains to secure a unique solution. Regardless of whether the model is used in its normal way to calculate potentials and flows as system variables from starting conditions and system parameters of transmissivities and storativities, or whether it is reversed for a direct calibration, a unique solution can be achieved only if all boundary conditions are specified. For the normal direction of calculations, i.e. calculation of potentials from the parameters, these boundary conditions may be the potentials at the boundaries of the aquifer, which usually are known as part of the model starting conditions. For the inverse problem, i.e. the calibration, the required boundary conditions to be specified are one transmissivity for every distinguishable flow path. These transmissivities are rarely all known, if at all. At this stage it should be remembered that the trial and error and the optimisation methods of calibration suffer from the same problem. The solution by model inversion merely makes it apparent. Generally then any valid calibration procedure with incomplete data will provide an infinite number of alternative solutions, each one of which is consistent with the available data. This leaves the hydrologist with the uncomfortable freedom of chosing from among them. The only rational solution is to progressively introduce more estimated, guessed, and assumed data in order of expected reliability until the solution is unique or at least adequate for the model to produce predictions of acceptable reliability.

The parameters which had to be defined for the GABHYD model in this manner include all vertical transmissivities and all discharge boundary horizontal transmissivities. Storage coefficients were approximated. The detailed calibration then was left to a determination of interior horizontal transmissivities and recharge boundary transmissivities consistent with the other data of the prototype.

For this purpose a new method was developed employing directly and only the GABHYD model equation. Because of this direct relation between model equation and its inversion for calibration it has become possible to achieve almost complete consistency between the transmissivities and the other data of the prototype. In many cases the differences between the model generated and recorded prototype potentials were too small to show in computer calculations carried out to five significant digits. Such an exact calibration may appear senseless in the light of the inaccuracies of the basic data. However, it is achieved without extra effort while reducing even the largest error to an acceptable level; and it is reassuring to know that no additional large errors are introduced into an already inaccurate data base.

1.5 Results of the calibration

Details of the calibration procedure and of the equations on which it is based are presented in the next two sections of this Record. The principle on which calibration is based can be summarised as adjustment of parameters and variables until the individual flow components for each model cell (aquifers are subdivided into cells of equal size, section 1.2, definition of prototype) and for the entire basin are sufficiently balanced.

The cell-by-cell and overall water-balance errors are calculated between steps as an indication of the progress of the calibration procedure. A convenient measure of the progress is the standard error, calculated from individual cell-by-cell errors like a statistical standard deviation. This standard error is a more sensitive measure of the overall quality of the calibration than the simple average cell error or overall total error.

The first standard error was calculated after several calibration steps, including scaling of hydraulic parameters and calculation of specific storage coefficients, and after the first iteration of adjustments to transmissivities. This first calculated standard error was 16 litres/second (l/s) whereas the overall total error was 9200 l/s. After 29 further iterations the standard error was reduced to 6.3 l/s and at the end of the fine calibration the standard error was 1.2 l/s and the overall error 253 l/s.

Although these results illustrate the progress during calibration quite well they do not by themselves indicate the error of prediction which may be expected when running the model with the calibrated prototype. A customary procedure for verifying the model calibration and for estimating the errors in model predictions is to run the model for a period for which data are available but with-holding these data from the model, reserving them for comparison with the model predictions after the run. This simulates the conditions when the model makes predictions for a future time for which data will become available eventually.

The period from 1960 to 1970 was selected for verifying the calibration of the GABHYD model prototype. Average errors of prediction for individual cells after 10 years, i.e. for 1970, were calculated as 11 percent for predictions of free-flowing discharge and as 0.5 percent for predictions of potentials. Cell-by-cell predictions have to be treated with caution, however, since the individual cell is the smallest unit of the model and hence constitutes its limit of resolution. A better measure of the model performance can be obtained by calculating predictions for areas consisting of several cells at least and measuring their errors.

This is illustrated in Figures 2 and 3. Figure 2 compares the predicted discharges with the recorded ones for an area around the Eulo Ridge in Queensland for the period 1960 to 1970. It was in this area that the largest balance errors were observed during calibration. Originally the model prediction error was 29 percent and it then settled down to 14 percent in 1970. Considering that the largest prediction errors must be expected in this area the result is quite acceptable. For the same period, a similar comparison is made for an area between Coonamble and Walgett, NSW, in Figure 3. Here the prediction error was very small throughout and never exceeded 2.5 percent. Both areas are among the most heavily developed areas in the basin, and data are abundant for them, so these results are far from trivial.

Overall it can be stated that the results of the calibration verification indicate that the model is suitable for predictions of free-flow discharges and of potentials over at least ten years. Much longer periods can be the subject of predictions, but only as long as the overall flow patterns in the basin do not change in a major way from those experienced during the model calibration period.

2. DESCRIPTION OF THE CALIBRATION

2.1 The available data

Only those data are considered here which have been included in the GAB data bank or are in any way part of the model prototype for either the original GABSIM or the currently used GABHYD models. Most data used originated from records of flowing artesian bores held by State water authorities in Queensland,New South Wales, and South Australia. Data were transcribed onto coding sheets and stored on the GAB data bank (Ungemach & Habermehl, 1973;

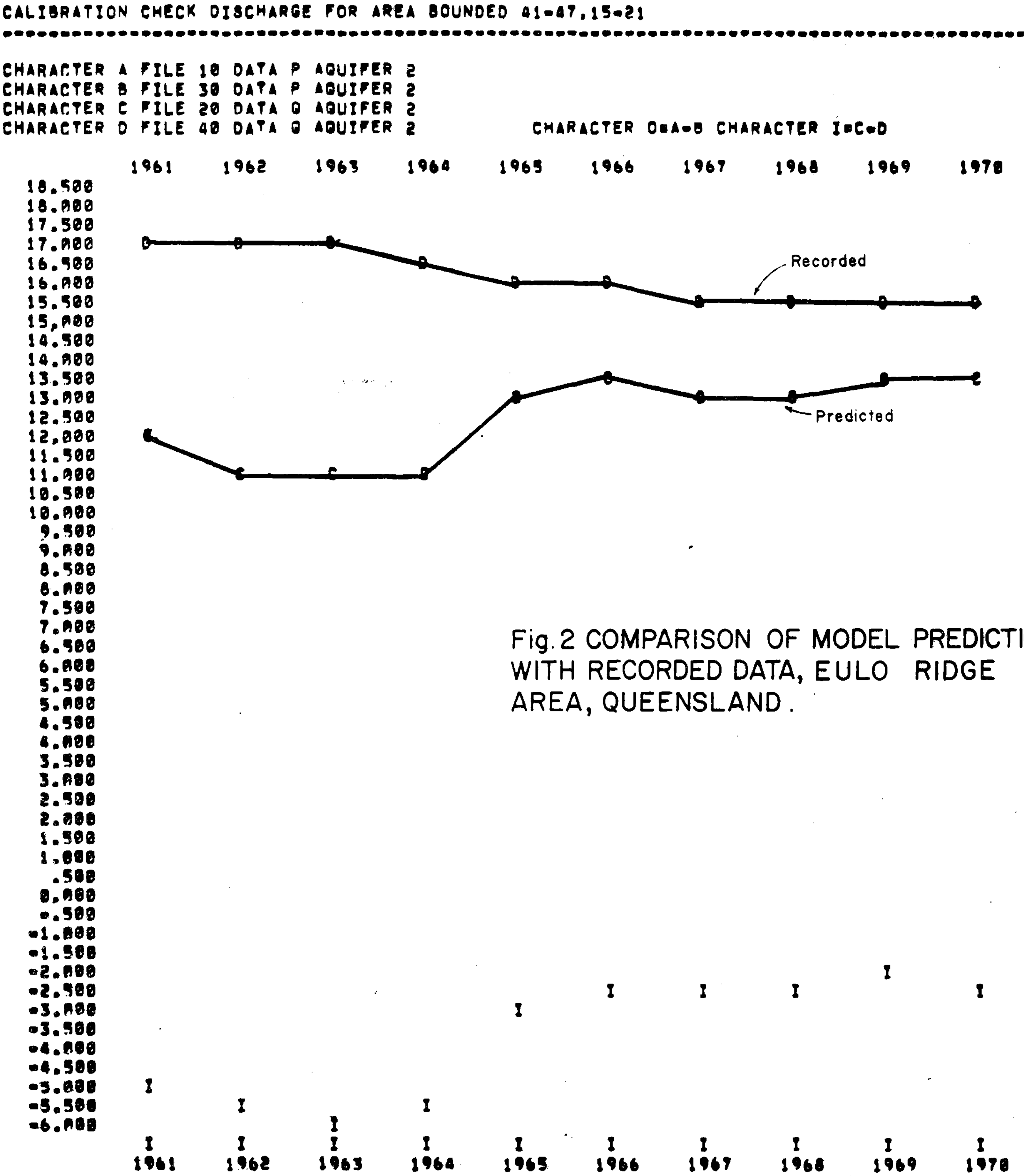


Fig.2 COMPARISON OF MODEL PREDICTIO
WITH RECORDED DATA, EULO RIDGE
AREA, QUEENSLAND.

CALIBRATION CHECK DISCHARGE FOR AREA BOUNDED 53-58,06-10

CHARACTER A FILE 10 DATA P AQUIFER 2
CHARACTER B FILE 30 DATA P AQUIFER 2
CHARACTER C FILE 20 DATA Q AQUIFER 2
CHARACTER D FILE 40 DATA Q AQUIFER 2 CHARACTER O=A-B CHARACTER I=C-D

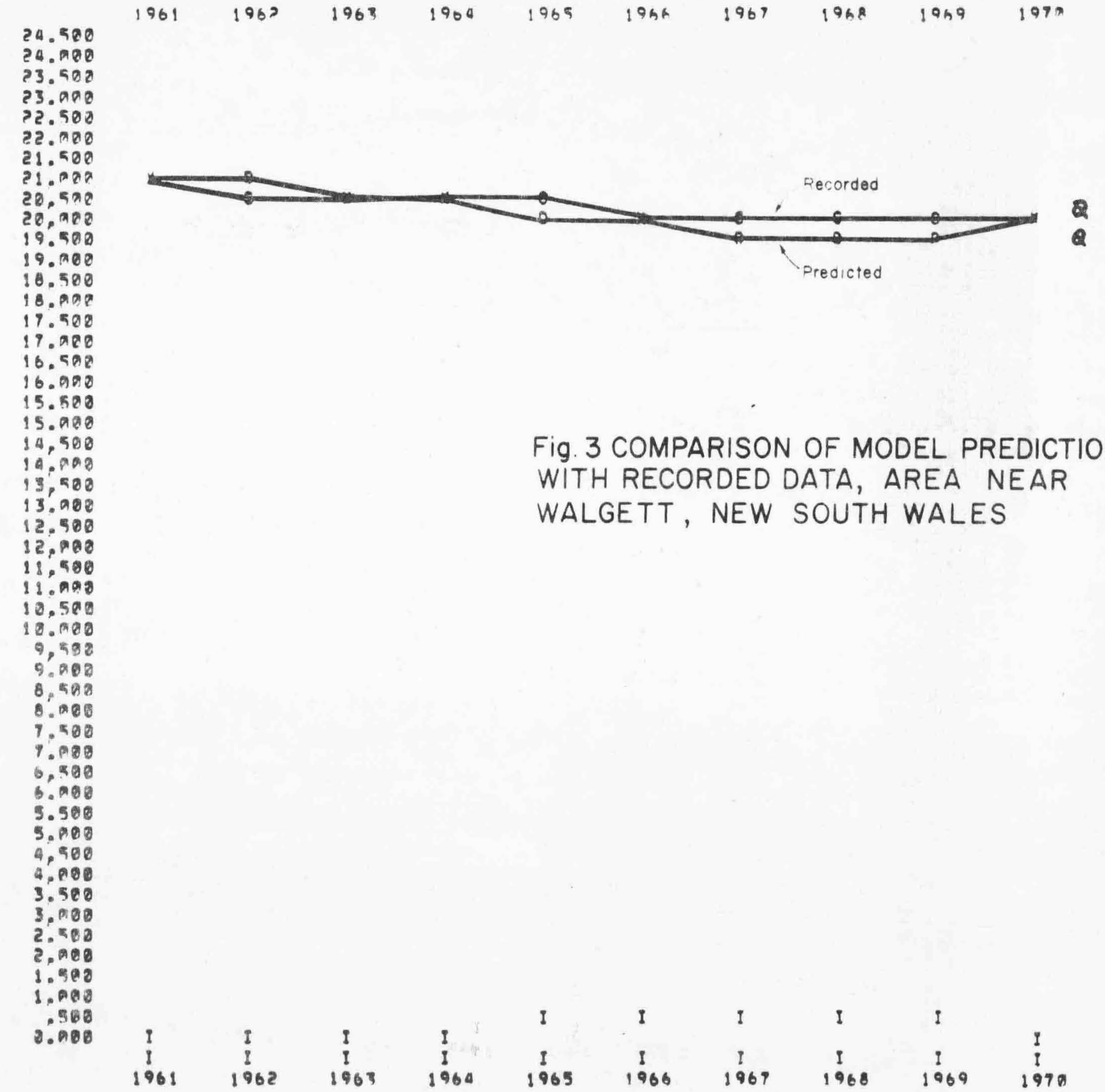


Fig. 3 COMPARISON OF MODEL PREDICTIONS WITH RECORDED DATA, AREA NEAR WALGETT, NEW SOUTH WALES

Krebs, 1973, 1974; Seidel, 1973). Other data were obtained from drillers' logs, geophysical logs of water bores or exploration and stratigraphic bores, and maps compiled by State organisations and BMR. Most of these data were used in profile or map form and were entered into the model prototype indirectly (Audibert, 1976).

Data of flowing artesian bores which are stored in the GAB data bank include:

1) Pressures obtained by shutting off flowing bores until the water pressure stabilised, usually after less than 12 hours. Readings were taken at irregular intervals, typically in the order of several years. The error of measurements is estimated as less than one meter except in areas of very high pressures, where errors can be expected to be higher.

2) Water temperatures measured at the discharge point on the surface. If the discharge rate is sufficiently fast this temperature approximates the water temperature in the source aquifer.

3) Discharge measurements obtained by various and often unrecorded means. Little is known of the accuracy of these data, but an error margin of $\pm$ 20 percent appears to be a reasonably safe estimate.

4) A few data were collected on the water chemistry, mostly the TDS. Early calculations indicated that the water chemistry was of little relevance to the current phase of the project.

5) Bore data, including location (elevation-surveyed or estimated from maps) and depth of bore. The accuracy of these data varies.

6) Hydraulic test data, mostly of the pressure-recovery type. The data accuracy is similar to the one for pressures and discharges. The accuracy of the results, e.g. transmissivities, is affected by further error sources, e.g. partial penetration effects.

7) Simplified drillers' logs including first water intersections, thickness and type of aquifer penetrated.

8) Calculated hydraulic parameters, in particular transmissivities available for many bores in Queensland (Hazel, 1973).

Data obtained mostly from maps and logs and entered indirectly into the prototype include:

9) The location of various types of springs.
10) The lateral extent of individual aquifers and aquicludes.
11) Depths and thicknesses of aquifers and aquicludes.
12) The relative proportion of various lithological units within aquifers and aquicludes.
13) The surface topography and drainage patterns.
14) Geological features, e.g. faults.

The distribution of data items is irregular both in time and space. This is illustrated in Figure 4, which shows for which cells of the Jurassic aquifer pressure records are available up to 1970. Each of the aquifer cells itself represents a variable number of bores so that in reality the data distribution is even more clustered than evident from this figure. Furthermore the density of data is not constant over the time but rather increases, usually in bursts starting from 1896 and corresponding to the development of new bores. Such development bursts occurred as recently as 1962 to 1966.

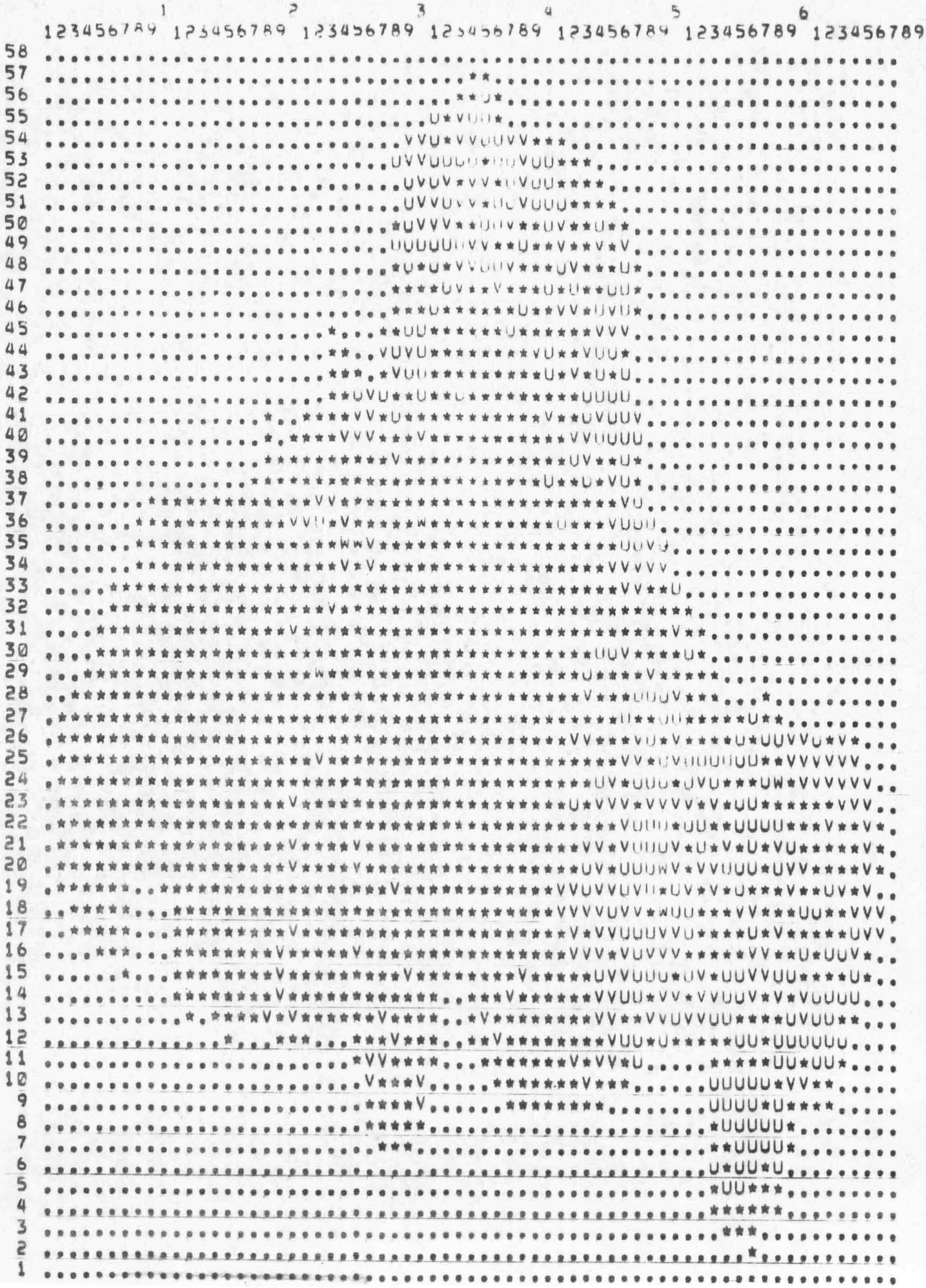
In many cases data are incomplete, e.g. pressures may have been measured accurately, but only a rough estimate is available of the ground elevation to convert the pressures into absolute potentials. This greatly reduces the practical value of many data items.

2.2 Brief description of the model prototype

As noted already in section 1.2 the prototype is defined as a representation of the real data set simplified for processing by the model. The structure of the prototype is hence determined by the requirements and characteristics of the model.

The GABHYD model is defined on a layered grid, where each gridpoint represents one cell of an aquifer and each layer represents one aquifer. In this quasi-three-dimensional representation, often referred to as the Hantush approach, flow within an aquifer is only along the layers and 'vertical' flow as leakage only normal to the layers. Aquicludes or aquitards are not repre-

Figure 4. Location of nodes with recorded data, Jurassic group aquifers.



sented by grid points at all but merely by the resistance they offer to flow between aquifers. This resistance is represented by the vertical permeability or leakage factor assigned to each vertical flowpath. Boundaries are defined surrounding each aquifer and above the top and below the bottom aquifer at any location.

These model properties specify the requirements for the prototype. The continuous aquifer geometry of the basin must be translated into a discrete specification of corresponding grid points. This may be a simple code indicating whether a particular aquifer is present on a gridpoint corresponding to a particular location, or it may be a more elaborate code indicating some further characteristics of an aquifer other than merely its occurrence.

On each gridpoint then there must be a specification of parameters and variables used by the model. This implies interpolation and averaging where more than one data item of a particular kind is available for a gridpoint, and extrapolation or even guesswork where there is none.

The parameters required by the model are:

- horizontal transmissivities
- vertical leakage factors
- storage coefficients

State variables, which are required as starting and boundary conditions for running the model and for verifying and calibrating the model are:

- potentials
- discharges

Decision variables, which are not part of the original prototype but must be allowed for in the data set for manipulating the model, are:

- ground elevation/temperature correction factors for potentials
- discharge coefficients for specification of discharge from pressures.

2.2.1 Prototype geometry

For the GABHYD prototype the aquifer geometry is specified by a string of ten integers for each grid point, each digit representing the corresponding number aquifer. The individual digits may assume the following values:

0 gridpoint is outside of aquifer
1 gridpoint is on impermeable boundary
2 gridpoint is on permeable boundary with prescribed potential
4 gridpoint is within aquifer, recorded potentials are available
5 gridpoint is within aquifer, no recorded potentials available

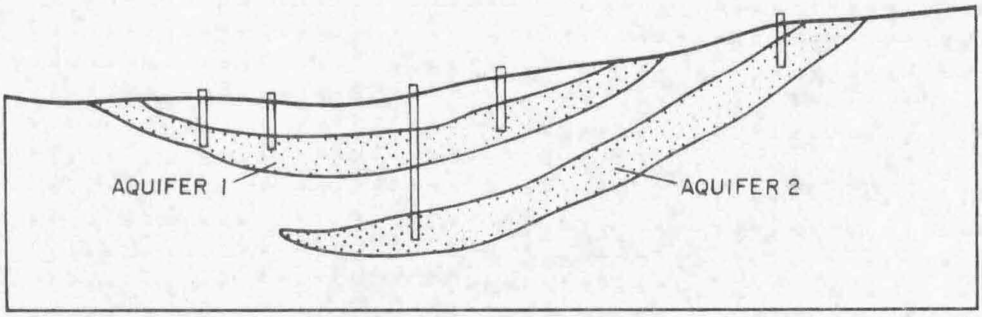
The correspondence between the prototype geometry and the geometry code is illustrated in Figure 5.

Only two aquifer groups are modelled in the current study; the Jurassic and the Cretaceous group of aquifers. In reality there is a number of aquifers in each of these groups; however, the restrictions imposed by the capacity and speed of available computers would not allow them all to be model led separately even if sufficient data were available. Furthermore, whenever potentials data were available for more than one aquifer of the same group for the same area, they were so close to each other in their respective values that it appeared unnecessary to distinguish between them hydraulically (Audibert, 1976). Aquifers are numbered starting from the top and hence the first integer of the aquifer code strings applies to the Cretaceous group, the second to the Jurassic, and digits 3 to 10 are not used for the current model.

The area of the GAB is subdivided into square cells of dimension 25 km x 25 km. The whole GAB prototype fits into a grid of 67 x 58 such cells. The Jurassic group has the largest areal extent as shown in Figure 4. The extent of the Cretaceous aquifer group is shown in Figure 6.

2.2.2. Prototype parameters

All prototype parameters are subject to adjustment or complete recalculation during calibration unless they form a prescribed boundary condition. However, a reasonable first estimate of these parameters is desired even where they are not prescribed. The calibration method used is basically iterative and will converge faster if better initial estimates for the parameters are provided. The initial estimates, prepared as described by Audibert (1976), are listed below.



Aquifer Geometry Code

0	2	4	4	5	5	4	5	2	0	0	0	0	Digit 1
0	0	0	1	5	4	5	5	5	5	4	2	0	Digit 2

Fig. 5 CORRESPONDENCE BETWEEN PROTOTYPE GEOMETRY AND GEOMETRY CODE

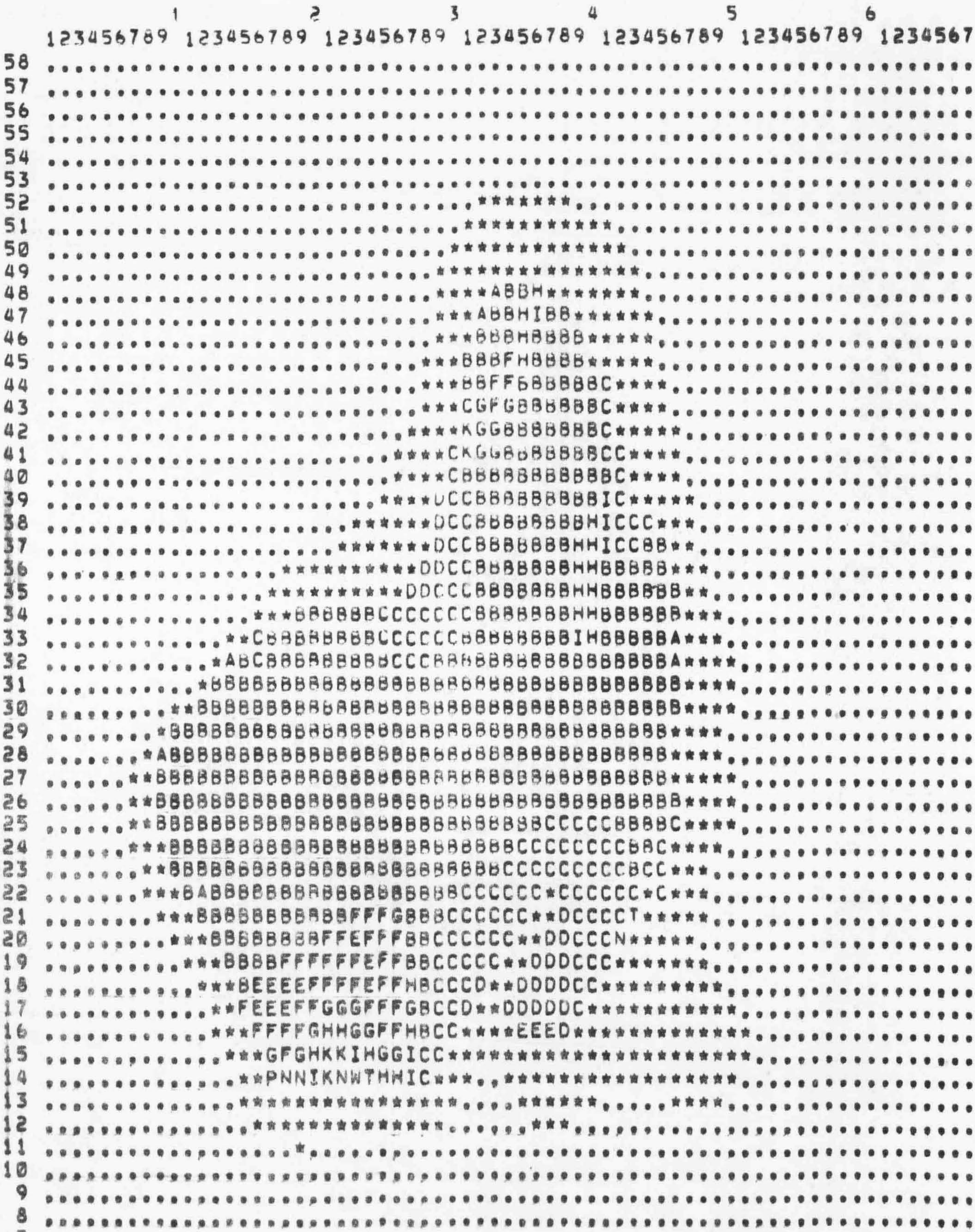


Figure 6. Area of Cretaceous group aquifers (* = outcropping).

1. Horizontal transmissivities - point transmissivities have been determined from recovery tests on many bores and are part of the data on the GAB data bank. These were converted into permeabilities through division by the aquifer thickness at each bore. Permeabilities, aquifer thicknesses and percentage permeability factors for the aquifer thickness were each interpolated and extrapolated over the whole aquifer and then multiplied to determine aquifer transmissivities.

2. Vertical leakage factors - no measured values of vertical permeabilities were available and indirect determinations based on the relative thickness of lithological units within the confining beds had to be used. In the GABHYD model equation the vertical permeability occurs only once and then it does so in a ratio with the thickness of the confining bed. So rather than storing in computer memory vertical permeabilities and confining bed thicknesses separately their ratio is stored and called the vertical leakage factor.

3. Storativities (storage coefficients) - no direct measurements were available. Estimates were prepared from acoustic logs for the GABSIM model. However, GABHYD determines storage coefficients as the first step of the calibration procedure and no longer requires these initial estimates.

2.2.3. Prototype state variables

State variables are calculated by the model as part of its predictions. Known state variables are used for comparison with corresponding model predictions to check the model's validity. However, during calibration the situation is mostly inverted with the state variables becoming the input or independent variables, and the parameters the output or dependent variables. A good data set of state variables is essential for calibrating a model.

Potentials are defined as the total hydraulic head at any one point. The measurement values available are the pressure heads at the surface. These are converted into reasonably close estimates of the true potentials by adding to them the ground elevation relative to the agreed datum (e.g. M.S.L.) and density corrections for water temperature and impurities. Corrections for impurities (dissolved solids) were found to be minor and were eventually ignored. The datum for the temperature corrections was 15^{o}C.

The prototype required one value of potential for each gridpoint of each aquifer. The approach adopted for the GABSIM model was based on contouring of the data and interpolation between contours to define potentials on each gridpoint. This approach was inaccurate, time consuming, and often produced potentials which for hydraulic continuity would have required negative transmissivities. The approach adopted originally for the GABHYD model was to directly accept potentials where they were recorded, to extrapolate current estimates of hydraulic parameters and then to use the model itself for interpolation of the potentials. Advantages are that parameters are easier to extrapolate than potentials and that using the model for interpolation of potentials produces potentials which are compatible with the model geometry and hence do not require negative transmissivities. This approach was further modified as part of the calibration procedure and is described in section 2.4.4.

The water-table potentials form the uppermost boundary condition for vertical leakage. Variation in water-table potentials of a few metres have little effect on the amount of vertical leakage, which itself is mostly a small component in the water balance for each cell. Because of this and because of the scarcity of data, the water-table potentials were treated as constant in the interior of the basin. On the permeable boundaries the water-table and the aquifer potentials coincide. Because it was found that overflow springs in the aquifer recharge area flow continuously even though at varying rates, it was concluded that here too variations of potentials with time were small in relation to other potential gradients and hence could be ignored. As a result the water-table forms a constant prescribed head condition throughout the basin. The water-table elevations were mostly obtained from drillers' logs.

Discharges. For purpose of this model neither vertical leakage flows nor boundary discharges are considered as usable predictions. They are calculated but the accuracy of these calculations is doubtful, because there is no data available to check them. Some moundspring discharges have recently become available, which when included in the model will improve the situation at least locally.

Excluding vertical leakage flows and boundary discharges, the aquifer discharges consist only of flows out of wells and bores. Most discharges from the Jurassic group aquifers are free flowing and recorded; most discharges from the Cretaceous aquifer group are pumped by windmills and not recorded. As a result most of the discharge from the Jurassic aquifers can be calculated easily from data on the GAB data bank, whereas the discharges from the Cretaceous aquifers can be estimated coarsely at best. A data set of discharges for the Cretaceous aquifers has been prepared, but model predictions for the Cretaceous group are not used and no detailed calibration was attempted for these aquifers.

2.2.4 Prototype decision and related variables

The decision variables of the GABHYD model are those variables which may be altered for the manipulation of the model. These are in particular the variables used to manipulate the discharges. They are mentioned here because they have to be provided for in the data base. They are not actually used during the calibration.

If we ignore the trivial case of a prescribed discharge, then the minimum requirement for calculating a free-flowing discharge from an artesian bore under local steady state conditions are the net available pressure and a flow coefficient. The net available pressure is obtained by inverting the procedure of the calculation of potentials; from the current potential are subtracted the ground-elevation and the temperature correction. Since both these can be considered constant they may be combined into one single correction value and determined during the original prototype generation. The flow coefficient is then the decision variable for calculating the discharge. This coefficient of course is not necessarily a constant but may itself be a function of the discharge (non-linear well loss).

2.3 Formulation of the calibration problem

The purpose of calibration is to adjust model parameters and data until the model performs satisfactorily, whatever the criterion for satisfactory performance may be. In practice that usually means to adjust the parameters until the model reproduces satisfactorily a set of known state variables. Often this led to the definition of calibration as the 'inverse problem', that of determining parameters from state variables, e.g. the transmissivities from potentials. Such a definition, however, is unduly restrictive as will be shown below. Instead a more general definition may be adopted: calibration is the process of adjusting model parameters and variables until they are hydraulically consistent with each other, with the largest adjustments applied to the data items of least certainty; or in other words: whichever data item is the more reliable is to be the independent variable, and the less reliable one the dependent variable in the process of calibration.

After the objective of calibration has been stated it has to be determined whether a solution exists, whether such a solution can be unique, and whether the data are adequate to carry out the calibration. Neither of these questions is trivial.

2.3.1 Existence of a solution

Let us assume that adequate data are indeed available. Whether a solution to the calibration problem exists then depends on what definition of calibration is adopted and what physical constraints are imposed on the solution. The real values of the state variables of a physical system by definition must be physically possible. However, we do not have these real values. Instead our data are measurement values, subject to measurement errors, and furthermore they had to be defined on regularly spaced discrete points which generally do not coincide with the points where the measurements were made. As a result the data of state variables in a model prototype need no longer be physically possible. Many examples were found on the maps of potentials prepared for the GAB where the potentials in the prototype could be reproduced by the model only by assuming extreme or even negative transmissivities, e.g. particular nodes were known to be well discharge points, yet had a prototype potential higher than each of the surrounding nodes, in fact requiring water to flow 'uphill' to reach the discharge point. Negative transmissivities are physically impossible and hence rightly are rejected. This would seem to imply that a solution is not possible then. However, this apparent conflict exists only as long as the restrictive definition of calibration as the inverse problem is accepted. The more general definition of calibration as stated above would prescribe for this condition to vary the physically impossible potentials first. This could be in the form of refining the model prototype discretisation or by allowing potentials to vary within their suspected error margins until they become physically sensible. If the general definition of calibration is adopted then subject to data availability a solution exists.

2.3.2 Uniqueness of the solution

The system which we attempt to model is a real physical system and hence we may stipulate that this system is described by a set of real physical parameters, and we may stipulate a physical process that will for any complete and, unique specification of boundary conditions using that real and hence unique set of parameters produce a unique set of system state variables. However, about all of these we possess only imprecise information and hence we cannot apply the concept of uniqueness in its strict mathematical sense to any model of that system. Instead we may adopt a 'soft' definition of uniqueness for the model of the system as: If firstly the values of the parameters are known with specified limits of uncertainty of errors, if secondly the model of the physical process is subject to a specified margin of error, if thirdly the boundary conditions are given to within a specified error, then the state variables produced by the model will also be within a specified margin of the true state variables of the real system.

The uniqueness of the inverse problem can be defined correspondingly as specification of any number of sets of parameters, each of which is within an acceptable margin of the corresponding true value of the physical system. It should be noted that there could be an infinite number of parameter sets meeting this requirement of 'uniqueness'. This 'soft' definition holds only for physical processes which in the real system are properly unique. An important consequence is that any inverse problem solution which does not calculate uniquely true parameters when applied to true state variables of the real system should not then be expected to calculate parameters close to their respective true values even where it may have been applied with state variables which themselves are close to the true values. That applies even where the method produces only one set of parameters and so on first glance appears to be properly unique.

The problem of uniqueness has been considered by many researchers dealing with the calibration problem and a variety of approaches was adopted, which can be classified into three groups:

1) introduce physical constraints on parameters until there is only one solution. This approach is represented by the work of Emsellem & Marsily (1971).

2) eliminate all model parameters by using a sufficient number of observations of state variables presumed to yield independent equations. Use optimisation techniques (linear programming) where there are more observations than there are unknowns (e.g. Kleinecke, 1971; Hefez, Shamir, & Bear, 1975)

3) solve the model equation for the parameters as unknowns and specify all boundary conditions required for this solution to be unique (Nelson, 1968).

Of these only method 3) satifies the criterion stated above that the method must work truly and uniquely when applied to true variables. Method 1) does obviously not satisfy this requirement, but it does introduce the useful restraint that parameters ought to be physically reasonable. This constraint should not be applied to ensure uniqueness within an otherwise arbitrary inverse process, but rather it should be used to decide whether the inverse process is applicable at all. If it is not applicable then the proper procedure is to adjust state variables instead or the model discretisation as discussed already in 2.3 and 2.3.1.

Method 2) would indeed be valid if it could be assumed that all observations of state variables necessarily lead to independent equations for the purpose of eliminating parameters. That this assumption is not valid becomes obvious by considering that the number of observations can quite easily exceed any finite number of parameters to be determined. The problems associated with this method are dealt with in greater detail in Appendix B.

As long as it is at all possible the uniqueness and trueness of the solution to the parameter identification problem should be ensured by using method 3) or an equivalent.

Nelson's approach can be summarised as a numerical equivalent of a flow-net analysis. A polynomial representation of the potentials yields the characteristics, or in the specific case of plane steady isotropic flow, the flow lines. Along each flowline only one flux needs to be determined to simultaneously specify all other fluxes along the same line. This flux may be determined by specifying one only parameter (transmissivity) value (Nelson, 1968) or by directly specifying the flux e.g. from well data (Frind & Pinder, 1973). The uniqueness requirement for steady state conditions may be stated as: a unique parameter identification from a complete set of state variables is obtained if and only if one parameter or flux is specified on each distinguishable flow path of the system. This requirement can be generalised for non-steady state conditions by considering that the solution is in fact a determination of the parameters from the continuity equation. The non-steady state version is obtained by including a change in storage term in that continuity equation. This will be further considered in 2.4.

An additional source of non uniqueness may be introduced by the discretisation. Rarely do gridlines coincide with natural flow paths and hence paths which are unique in the real system may be translated into arbitrary alternatives of branching and merging paths along the gridlines. Unless precautions for this source of non-uniqueness are taken it may easily lead to an artificial anisotropy of the solution through discretisation. A uniformity criterion like that used by Emsellem & Marsily (1971) may be used to artificially remove this artificial effect.

2.3.3 Adequacy of data

The basic prerequisite for using method 3) of 2.3.2 for ensuring a 'unique' solution which is reasonably close to the true physical parameters is of course adequate data.

Let us assume that our data on the state variables of the system, i.e. potentials and free-flow discharges, are indeed sufficient to define the states of the system over the historical period, in this particular case from 1880-1970. Our data on the hydraulic parameters of the system are insufficient to proceed with any form of calibration unless we accept this assumption. This leaves us with the task of determining or adjusting hydraulic parameters from the state variables after specifying the necessary boundary conditions. The hydraulic parameters to be considered are:

- storativities
- horizontal transmissivities
- vertical leakage factors.

As it will be shown in 2.4.2 an approximate determination of storativities is adequate for the present study. Such approximate values can easily be obtained from the changes in state variables over time after some initial estimates on the other hydraulic parameters have been made. The major remaining problem then is the accurate determination of horizontal transmissivities and vertical leakage factors. Using method 3) the data required in addition to a complete set of state variables are one parameter value for each distinguishable flow path, either by defining it directly as parameter or indirectly through specification of flux. Figure 7a illustrates typical branching of a flow path starting from the recharge boundary into individual vertical leakage paths and at last the discharge boundary. To adequately define such a system we require

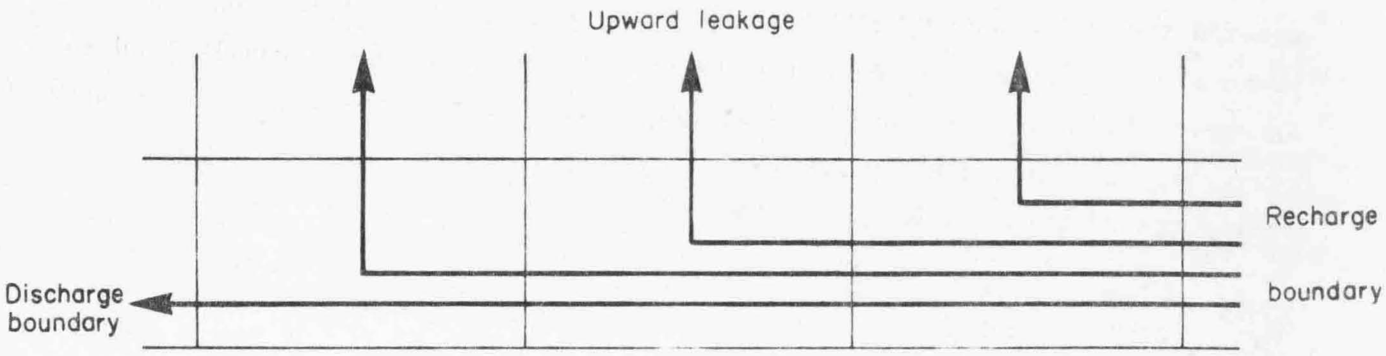


Figure 7a

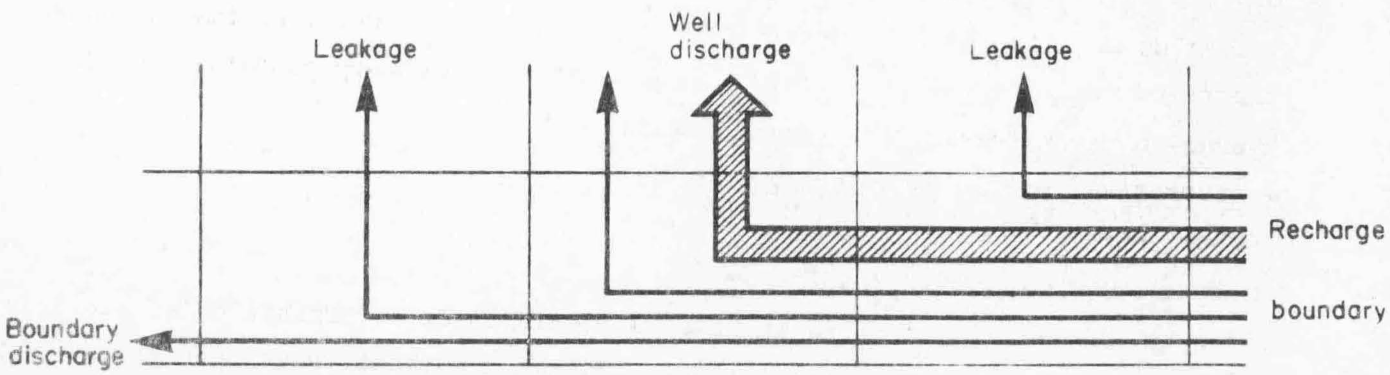


Figure 7b

Branching of distingnishable flow paths in undeveloped (a) and in developed (b) condition the direct or indirect specification of each discharge boundary transmissivity and of each separate vertical leakage factor. Method 3) then can be used only to define the interior horizontal transmissivities and the recharge boundary transmissivities. All the other parameters will have to be defined in some other way.

No measurements of vertical leakage factors were available for the present study. Some estimates of vertical leakage factors were prepared using geological data and the overall water balance of the basin. A model calibration could be based on these estimates only but any result would have to treated with extreme caution. It should be noted in this context, however, that this method involves a conscious scientific guess on these data items and as a result the likely errors can be appreciated. Whereas other methods of ensuring uniqueness of the parameter identification as for example methods of type 2) of 2.3.2 would leave the determination of vertical leakage factors almost entirely to statistical chance, with little or no control on the resulting accuracy.

However the risk of major errors is much reduced when one includes the observed values of artificial discharge in the system as in Figure 7b. Now the well discharge forms a major portion of the total flux along most of the shared flow paths starting from the recharge boundary. This well discharge is a measured data item and no longer merely a scientific guess. Available measured data define the fluxes reasonably well from the discharge boundaries to and through the developed areas of the basin. It is for these portions that most model predictions will be required. For the remaining portions of the basin the calibration will be less reliable, but still of a more controlled accuracy than with other methods.

2.4 Solution by model inversion

As stated in 2.3.1 a solution by pure model inversion is possible only for the ideal case where all state variables are available and accurate. In the real case, as for the Great Artesian Basin, calibration consists mostly of model inversion but with other processes complementing it. As mentioned already, prototype potentials are in need of occasional adjustments to ensure their physical reasonableness. This involves using the model in its normal operating direction rather than inverted. It could also be argued that the relatively simplistic approach for determining storativities is not a true model inversion (2.4.2). This simple approach was found satisfactory in this case because answers required from the GABHYD model are relatively insensitive to errors in storativities. A more direct model inversion for determination of storativities might be required if short-term model responses are to be studied which are more sensitive to errors in the storage terms.

2.4.1 Initial values and boundary conditions

What initial conditions and what boundary conditions are required depends on the processing direction of the model. For the normal operating direction which is used for adjusting potentials during calibration the required initial values are the initial potentials and discharges in the interior of the aquifer. The required boundary conditions depend on the type of aquifer boundary - for the Great Artesian Basin they are either zero transmissivities on impermeable boundaries, or prescribed potentials on permeable boundaries. Whether or not a boundary was permeable was determined from the aquifer geometry, i.e. was derived from geological profiles. Where a boundary was permeable it was outcropping and hence the aquifer potential was identical to the water-table potential. For the normal processing direction, boundary conditions are provided by the aquifer geometry and by the water-table potentials and the initial conditions by sets of state variables derived from recorded values of potentials and discharges. The hydraulic parameters are accepted as fixed and are then used to adjust the potentials through use of the model.

For the inverted operating direction used for adjusting the parameters the initial conditions are initial estimates of the hydraulic parameters in the interior of the aquifers. The boundary conditions are the estimated discharge boundary transmissivities, the vertical leakage factors, and the well discharges. Of these only the well discharges are considered to be accurate. Both discharge boundary transmissivities and vertical leakage factors were subjected to overall scaling to balance the overall water budget of the aquifer. As a result the major source of errors will be the relative variation between boundary transmissivities and vertical leakage factors. In this processing direction the state variables are assumed to be fixed.

2.4.2 Determination of approximate storativities

The approach for determining storativities is based on one strong idealisation. It is assumed that the storage coefficient throughout each aquifer is directly proportional to the total thickness of the aquifer. This in fact implies a uniform specific storage coefficient. The justification for this simple approach is:

1) Unlike permeability the specific storage coefficient does not depend on the relative pore-size distribution in the aquifer but only on the total volume of interconnected porosity. Total porosity varies less within the same aquifer than does the pore-size distribution, and hence the specific storage coefficient varies less than the permeability.

2) The model is required to predict slow processes mostly close to some equilibrium condition. In these conditions the storage change component in the water balance is relatively small.

3) The quality of the available data is adequate for determining a uniform specific storage coefficient for a relatively large area, e.g. for the whole basin. Any attempt to calculate specific storage coefficients for smaller areas results in a corresponding increase in the inaccuracy of the individual calculated value and hence defeats the purpose of the refinement.

In this simple approach the calculation of a specific storage coefficient is the result of an overall water-balance determination of the basin. During the actual application, water balances were calculated at 5-yearly intervals:

$$S_i = R_i - D_i + L_i - Q_i \qquad \text{for period } i$$

where S_i = change in storage

R_i = boundary recharge

D_i = boundary discharge

L_i = leakage flow balance (vertical leakage gain - loss)

Q_i = well discharge

All values on the right hand side of this equation were calculated from initial estimate data (R_i, D_i, L_i) or from measured data (Q_i).

Although the budged had previously been balanced by scaling the vertical leakage factors for 1880, when there was a steady state condition without storage changes prior to development, it was found that for later periods the change in storage S_i was calculated as positive instead of the negative value expected for a period when potentials were falling. Clearly when using the initial estimates of transmissivities together with the observed increases in hydraulic gradients at recharge boundaries, the resulting increase in recharge flow exceeded the incremental well discharge, i.e. the recharge boundary transmissivities were too high. To correct this anomaly it was decided to scale all hydraulic parameters by a factor f to achieve a balanced budget.

If such a scaling is carried out in a near equilibrium condition following developemnt then the change in storage S_i will be small in comparison to Q_i. Studying record data for the Great Artesian Basin it was estimated that for the period 1950-1960 yield from storage was about 25% of the well discharge.

$$S_i = 0.25\ Q_i \qquad \text{for 1950-1960}$$

The exact percentage is of little significance for the calculation to follow as long as it is relatively small. The scaled balance for 1950-1960 then is:

$$-0.25\ Q_i = f(R_i - D_i + L_i) - Q_i$$

or solved for f:

$$f = \frac{(1-0.25)\ Q_i}{R_i - D_i + L_i}$$

The scaling factor obtained this way for 1950-1960 was 0.68.

After application of this scaling factor to all hydraulic parameters the changes in storage for the five-year periods starting from the beginning of development unitl 1970 were indeed all negative, as they should be.

For each time interval the corresponding drop in potentials at each node was multiplied by the cell area and the aquifer thickness. These products were added up over the whole basin to yield what might be termed the 'drained aquifer volume' V_i for the period i.

$$V_i = \sum_{n=1}^{N} (h_{i+1} - h_i)\ Z\ A \qquad \text{for all N nodes}$$

where h_{i+1} potential at end of period i

h_i potential at beginning of period i

Z aquifer thickness at node

A area of aquifer cell surrounding the node

the specific storage coefficient C_i for period i is:

$$C_i = \frac{S_i}{V_i}$$

When applied to the Jurassic group aquifer an average specific storage coefficient of 2.75×10^{-6} (metres $^{-1}$) was determined for the period 1930 to 1960. All but one of the individual 5-year periods within this interval yielded similar values and as a result the total Jurassic aquifer storativities were determined as the aquifer thickness in metres multiplied by 2.75×10^{-6}. A typical value corresponding to an aquifer thickness of just under 400 m is 1.0×10^{-3}.

2.4.3 Determination of transmissivities

Normally transmissivities are defined as continuous properties of an aquifer which during discretisation on a finite difference grid are allocated to gridpoints representing the respective aquifer cells. During application of the finite difference form of the continuity equation, transmissivities are required to calculate the flow between adjoining pairs of aquifer cells, i.e. a directional transmissivity is required for the path connecting each pair. These are normally provided by calculating in effect the arithmetic, geometric, or harmonic mean of the respective point values. This is mathematically sound if one accepts the assumptions inherent in the original derivation of the differential equations. However one of these assumptions is that the transmissivity or permeability indeed is a property of the same point on which the potential is defined, and this assumption is not entirely valid. In fact the transmissivity always is the property of a path and never that of a point. This distinction may have been of little significance in previous applications considering the inaccuracy of available data, but it becomes of fundamental importance when considering the inversion of the continuity equation to solve for the hydraulic parameters.

In the present case the inversion is carried out using directly the finite difference form of the continuity equation rather than its partial differential form. The reason for this is that the purpose of the inversion is primarily to determine the optimum model parameters rather than parameters of the system to be modelled. All further considerations in this section apply to the model rather than to the real system.

The requirements for model parameters can be stated as:

1) the parameters must satisfy the continuity equation with all available potentials;

2) the parameters must be physically sensible.

There would be no need to state the second requirement if we would deal with ideal data and perfect discretisation (coinciding with natural co-ordinates of the system). However, as emphasised already, the available prototype potentials are not always physically sensible and our flow paths on a regular square grid do not coincide with natural flow paths. We will further deal with requirement 2) below and in 2.4.4.

The requirement that the transmissivities must satisfy the continuity equation can be rephrased as that zero imbalances must be calculated for all aquifer cells. If they are not and we accept the potentials as correct then the transmissivities are not correct and must be adjusted so that the imbalances are zero.

Each transmissivity is the property of the path between two cells of the aquifer and with prescribed potentials on each of the corresponding nodes it determines the flow between those two cells. Any change to it simultaneously affects the water balances of both cells, and hence both balances must be considered in making changes to the connecting transmissivity. In general the imbalances of the two cells will not be equal and of opposite sign and hence it will not be possible to adjust the connecting flow so that both will become zero. However, an adjustment can always be made to distribute the imbalances between the two cells equally, provided of course that the values which transmissivities may assume are not additionally constrained.

Let us consider a pair of adjoining cells a and b with potentials h_a and h_b, with a connecting transmissivity T_{ab}, and having the residuals in their water balances of B_a and B_b. The one item the two water balances share is the flow component from one cell into the other.

$$F_a = (h_b - h_a)\ T_{ab} \qquad \text{(flow from b into a)} \qquad (1)$$

$$F_b = -\ F_a \qquad \text{(flow from a into b)} \qquad (2)$$

To distribute the imbalances equally between the cells we seek the new balances B_a* and B_b* with

$$B_a* = B_b* \qquad (3)$$

this is achieved by replacing F_a with a new flow F_a* (or F_b with F_b*) corresponding to a new transmissivity $T_{ab}*$

$$B_a* = B_a - F_a + F_a* \qquad (4)$$

$$B_b* = B_b + F_a - F_a* \qquad (F_b = -F_a\text{, and } F_b* = -F_a*) \qquad (5)$$

$$F_a* = (h_b - h_a)\ \ T_{ab}* \qquad (6)$$

and because of equation (3)

$$B_b - B_a = 2F_a* - 2F_a$$
$$= 2\ (h_a - h_b)\ (T_{ab}* - T_{ab})$$

and solved for the new transmissivity $T_{ab}*$

$$T_{ab}* = T_{ab} + \frac{B_b - B_a}{2(h_a - h_b)} \qquad (7)$$

In words the difference in the balance of the two cells is halved, divided by the potential gradient between them, and this value is added to the old transmissivity. The new balances then will be the average of the two original ones.

$$B_a* = B_b* = \frac{B_a + B_b}{2} \qquad (8)$$

The cumulative effect of applying such balance equalising operations systematically and repeatedly to all pairs of aquifer cells is that all residual balances will approach a uniform value throughout the aquifer. If there is N aquifer cells and the boundary conditions are specified so that the overall water balance error for the aquifer is E then the residual balance for each cell will tend to become uniformly $\frac{E}{N}$. And if the overall error is zero as it should then the individual residual balances will tend to zero and hence the continuity equation will be satisfied everywhere.

So in addition to distributing the imbalances equally two more requirements have to be met:

1) boundaries must be adjusted to that the overall aquifer error is zero

2) physical constraints must be observed when calculating T_{ab}*.

Discharge boundary transmissivities and vertical leakage factors are fixed boundary conditions (2.4.1), but recharge boundary transmissivities remain to be adjusted for achieving an overall aquifer water-balance residual of zero. In our particular case the recharge boundaries are of prescribed head type. Unlike an interior aquifer cell a prescribed head-type boundary cell has an open-ended water balance. Any arbitrary water balance may be assigned to such a cell for purpose of calculating transmissivities. Let us assume that cell b is such a boundary cell whereas cell a is in the interior. We may then arbitrarily set $B_b = -B_a$. If a new connecting transmissivity is calculated with the above method the new balance B_a* then will be:

$$B_a{}^* = \frac{B_a - B_a}{2} = 0$$

If this method of adjusting recharge boundary transmissivities is adopted then the overall method has the result:

- all residual balances tend to a uniform value $\frac{E}{N}$
- all residual balances adjoining a recharge boundary tend to zero
- hence all residual balances tend to zero satisfying the continuity equation.

The process can be visualised as the imbalances being spread out until they reach a recharge boundary where they are absorbed.

Let us consider now the effect of introducing physical restraints.

$$T_{max} \leq T_{ab}{}^* \leq T_{min} \qquad 0$$

We may reasonably assume that the old transmissivities are already within these constraints. Then the worst possible result of the constraints is that the new transmissivity is the same as the old. If there is any adjustment possible at all it will be towards a more uniform pair B_a* and B_b*, although the constraints may frequently prevent them from becoming equal. Constraints will generally slow the convergence of the method. At worst they may stop it. A typical result as experienced during the practical application is a complete convergence over most of the aquifer's area with some islands of obstinate residuals. Such problems may result either from unreasonably strict constraints or more frequently from potentials which are physically unreasonable. Only unreasonable potentials can require unreasonable transmissivities to satisfy the continuity equation in a reasonable model. If all potentials and the constraints on the transmissivities are physically reasonable then the method converges completely.

A number of additions to the method can be used to handle specific conditions. One is the artificial 'anisotropy' caused by the fact that model flow paths along gridlines differ from the natural flow paths in a continuous medium. For each cell a decision must be made whether the first transmissivity correction should be along the x or along the y axis. Rather than making this choice arbitrary the alternatives can be precalculated and the one resulting in the lesser ultimate difference between the transmissivities can be applied.

To accelerate the convergence of the method it was found useful to calculate during each pass of all cells the overall balance error and to superimpose an opposite error on the region adjoining the adjustable recharge boundary. A reduction of around 40% was observed in the number of iterations required.

2.4.4 Adjustment of potentials

To obtain physically sensible parameters by model inversion from potentials, these potentials themselves have to be physically sensible. The physical reasonableness of interpolated values of potential is trivial if the model itselt has been used to calculate the interpolated values. But this does not affect recorded potentials. Because recorded potentials are subject to measurement and to discretisation errors they are inaccurate and no longer physically true. They may also be no longer physically sensible.

A test program applied to the potentials recorded for the Great Artesian Basin and designed specifically to detect such flaws in the recorded potentials discovered that a significant portion of the recorded potentials could be satisfied only by assuming extreme or even impossible values of transmissivities. However, further checks revealed that most of these potentials could be made physically sensible by allowing a variation of a few percent to the originally recorded pressure data, i.e. by allowing the potentials to vary within their error margins. There were some exceptions, however, where the apparent errors exceeded the likely measurement error significantly. Such exceptions occurred almost exclusively in areas where the real aquifer geometry was much more complex than what could be represented by a two-layer quasi-three-dimensional model. Here it became necessary to define some virtual prototype potentials which were physically possible yet were still as close as possible to the originally recorded values. The relation between the virtual and the recorded potentials might consist for example of an additive constant caused physically by the leakage headloss across a local aquitard between two aquifers which are modelled as only one. The alternative to this approach would be to use a more complex model.

The method originally was to use the model with the first estimates for the hydraulic parameters to calculate corrections for the recorded potentials. These corrections were restrained to be less or equal to the estimated error margins of the recorded values. The disadvantage of this method was that corrections to the potentials were applied to conform with first estimates of hydraulic parameters even where the original potentials were physically sensible. The information content of these potentials to determine corrections to parameters was lost. What was required was a method which allowed the prototype potentials to remain as close as possible to the original recorded potentials whilst ensuring that these potentials were compatible with physically sensible parameters. This could be achieved only by a method which adjusts potentials and hydraulic parameters simultaneously.

A numerical simulation approach was adopted which consists of alternating steps of calculating potentials from transmissivities, and then transmissivities from potentials. This otherwise pointless exercise obtains its purpose by introducting a bias into the calculation of transmissivities proportional to the difference between the current prototype and the recorded potential. This bias is designed to cause the new potentials then calculated from the new transmissivities to be closer to the recorded values than the old ones. The bias is introduced by an additional term in the water balance for each cell:

$$R_a = C_r \, (h_a - p_a)$$

R_a = balance bias (m^3/s) for cell a

C_r = empirical bias factor

h_a = current prototype potential

p_a = recorded potential

A corresponding value R_b is calculated for cell b.

To calculate the new transmissivity as in 2.4.3 the terms B_b and B_a are replaced by the terms $B_b + R_b$ and $B_a + R_a$.

The modified equation for calculating $T_{ab}{}^*$ then is:

$$T_{ab}{}^* = T_{ab} + \frac{B_b + R_b - B_a - R_a}{2\ (h_a - h_b)}$$

The effect is best illustrated by example. Assume that for cell a $h_a = 251.5$ and $p_a = 265.3$, i.e. the current potential is below the recorded. With C_r = 0.005.

$$R_a = 0.005\ (251.5 - 265.3) = -0.019\ m^3/s$$

A negative bias is introduced into the water balance of cell a. As a result inflow transmissivities will be calculated higher and outflow transmissivities lower than they would have been without the bias. The potential calculated from these new transmissivities in the next step will be higher and hence closer to the recorded value.

The factor C_r is determined empirically and may be varied during the calibration sequence. It was chosen here so that the maximum bias was in the same order of magnitude as the other components in the water balance.

If all recorded potentials were physically sensible within the framework of the model discretisation, then this method would change any arbitrary starting potentials into the recorded ones whilst simultaneously determining the matching transmissivities. However, where recorded potentials are not physically sensible the physically sensible potential closest to the recorded one will result.

When applied to the Great Artesian Basin, isolated recorded potentials were approximated accurately. However, in areas where recorded values were frequent and an appreciable number of physically unreasonable potentials occurred, some large differences resulted. These were inevitable, as far as could be determined, with the current model discretisation. The maximum discrepancies were around 20 m although between 1 and 3 m is a more representative value for the observed differences in the problem areas.

The performance of the model in predicting artesian discharges was much better when using the above method than by using as reference potentials either a manually contoured map or directly the recorded values. In predicting potentials the model predicted gradients accurately, but absolute values often required correction with the difference between prototype and recorded potential observed during the calibration.

2.4.5 Adjustment of vertical leakage factors

Vertical leakage factors form part of the set of prescribed boundary conditions and hence should not require adjustments. For the purpose of calibration they could be chosen arbitrarily and still allow the water balances of all cells to be balanced by appropriate determination of transmissivities. However, exceptions from this rule may be allowed close to the boundaries where aquifers and confining beds thin out. In several of these areas the vertical leakage components are large, as indicated by the occurrence of mound springs. The method itself used to estimate the vertical leakage factors too is affected adversely by the thinning out of layers. In these circumstances the initial estimates for vertical leakage factors may lead to the determination of unlikely extreme values of transmissivity. In these cases it is considered preferable to allow for some variation to the vertical leakage factors in the interest of a more likely distribution of transmissivities.

The method of adjustment is simple. For each cell the water balance is calculated using the current parameters. The water-balance equation is solved for the vertical leakage factor and the new value is restrained to physically sensible values and optionally to be within a specified percentage margin around the old value. By carrying out all other calibration steps first it can be ensured that only minor adjustments will be required as a result of the circumstances described above.

2.5 Calibration sequence

The following logical sequence is used in application of the calibration method described in 2.4:

1. scale vertical leakage factors to balance initial steady state conditions without discharges
2. scale vertical leakage factors and transmissivities together in accordance with observed well discharges for later periods
3. calculate storativities
4. simultaneously calculate model transmissivities and adjust model potentials
5. marginally adjust vertical leakage factors.

The theoretical details of each step are described in 2.4. The practical application somewhat deviated from the straight path because of unexpected characteristics of the prototype. For example it was found that the prototype geometry was in need of adjustment, and errors were detected in data items not included in the routine adjustment by calibration, e.g. well discharges and ground elevation corrections. An additional complication during the calibration of the GABHYD model arose from the fact that the calibration programs were still under development while the calibration proceeded. Any second calibration of the model would follow a much smoother path than the one actually used for its first calibration and presented schematically in Figure 8. The steps of this calibration sequence are numbered, and details of each step are supplied below; a flow-chart is provided for each step.

2.5.1 Data base generation (Fig. 9)

The dashed line across the flow chart represents the boundary between the old GABSIM and the new GABHYD model. As a result of the makeshift bridging of the two systems this step appears rather complex. In any repetition of the data base generation it would be advantageous to prepare a new set of programs for it as was planned originally in the design of the GABHYD model. As presented the data base generation results in the following files:

CALTHI - a file containing aquifer thicknesses, required for the calculation of storage coefficients.

OLDHYD - containing the initial estimates of hydraulic parameters. Vertical leakage factors have been scaled to satisfy the initial steady state balance.

OLDPOT - starting potentials obtained from manually contoured maps and including water-table elevations.

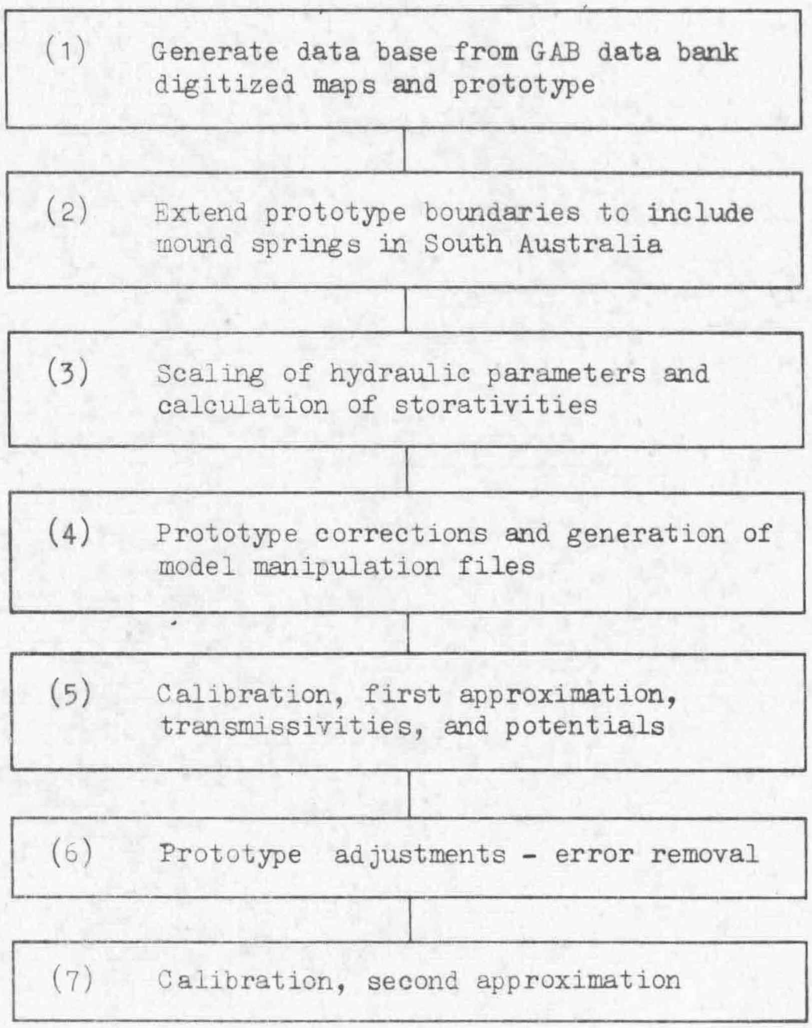


Fig. 8 ACTUAL CALIBRATION SEQUENCE

Record No.1978/12

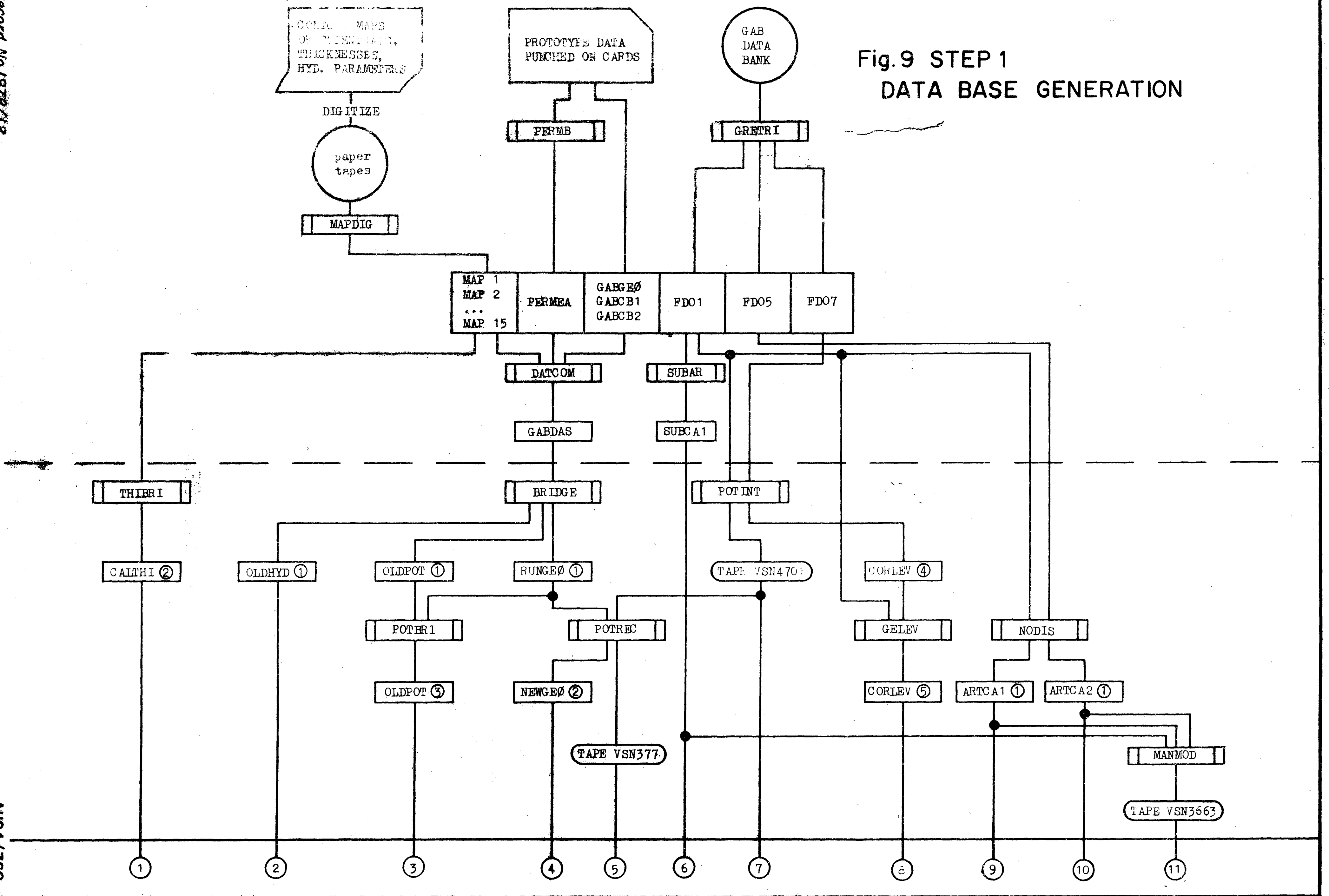


Fig. 9 STEP 1
DATA BASE GENERATION

AUS 1/760

NEWGEO - model geometry file.

Tape No 4701 - contains all recorded potentials assigned to the appropriate grid nodes year by year 1880 to 1970.

Tape No 377 - as above but first and last recorded potentials are extrapolated backwards and forwards in time respectively.

CORLEV - file of additive ground elevation and temperature density corrections.

ARTCA1, ARTCA 2 - condensed discharge files for artesian flows from aquifers 1 and 2.

Tape No 3663 - all recorded discharges assigned to the appropriate grid nodes year by year 1880 to 1970.

2.5.2 Boundary extension of prototype (Fig. 10)

It was discovered that the prototype designed for the GABSIM model had placed mound springs on the boundary so that they were effectively outside the aquifer. For logical consistency it was decided to extend aquifers so that they just included the mound springs within their area and to treat the mound spring discharge as localised high vertical leakage. The programs used in this step simply expand the model geometry and define hydraulic parameters and new boundary potentials by extrapolation over a small distance.

2.5.3 Scaling of hydraulic parameters and calculation of storativities (Fig. 11)

For this step, potentials are required at regular intervals and in accordance with hydraulic parameters. For this purpose, RUNSTE calculated the initial steady state potentials to match the original hydraulic parameters and boundary conditions. HISMOQ extended potentials, calculating year by year up to 1970, interpolating them when no recorded data were available while retaining recorded values without change. The interpolation tool was the model itself. Scaling and calculation of storativities then proceeded as described in 2.4.2.

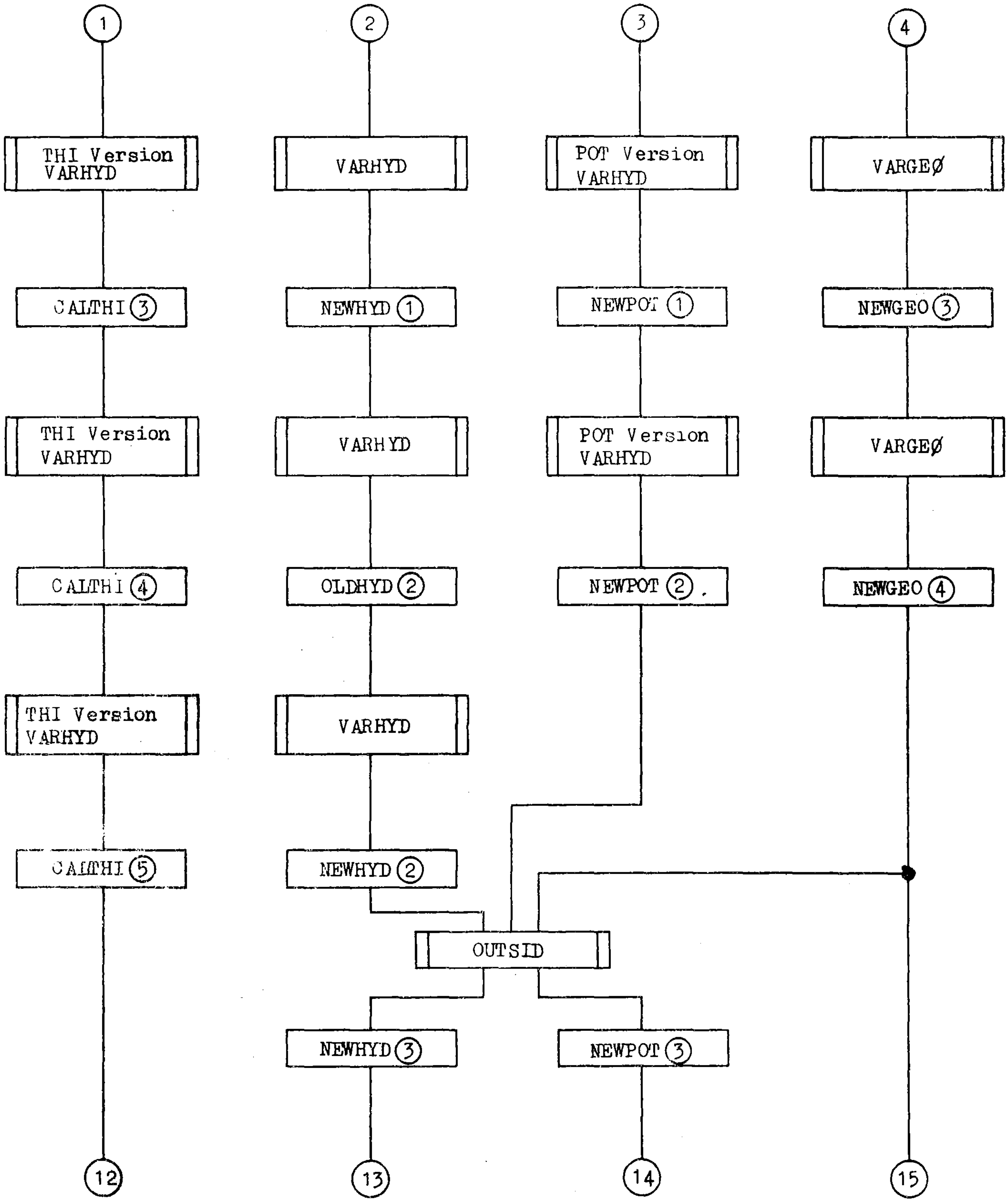


Fig.10 STEP 2 BOUNDARY EXTENSION OF PROTOTYPE

2.5.4 Prototype corrections and generation of model manipulation files (Fig. 12)

Some physically impossible (within the context of the prototype) potentials were discovered. The geometry codes were altered so that hydraulic parameters for these locations were not determined through model inversion using those potentials. A value for the storativity resulting from a fault in the aquifer thickness file was corrected, special versions of the discharge files for trial running of the model were prepared, and a subset of the recorded potentials was produced for 1970.

2.5.5 Calibration first approximation (Fig. 13)

COMCAL is the program combining calculation of potentials from parameters and transmissivities from potentials as described in 2.4.3 and 2.4.4. The first step was to generate a first approximation set of potentials for 1970 matching the original transmissivities. This was followed by simultaneous adjustment of transmissivities and potentials in two lots of iterations. A minor adjustment of vertical leakage factors through program VERTAD followed. The final steps consisted of residual adjustments to transmissivities leaving potentials unaltered and a residual adjustment of vertical leakage factors.

2.5.6 Prototype adjustment (Fig. 14)

A new version of interpolated potentials using the new parameters was produced for purpose of model verification. A correction was applied to a major error in a ground-elevation correction value. A separate test showed that the storativities determined in step 2.5.3 did not require adjustment.

2.5.7 Calibration second approximation (Fig. 15)

The second approximation was required to adjust for the effect of the correction to one ground-elevation value in 2.5.6. This step would not have been necessary otherwise. A simultaneous adjustment step for transmissivities and potentials is followed by further adjustments to transmissivities in two steps and a final adjustment to vertical leakage factors.

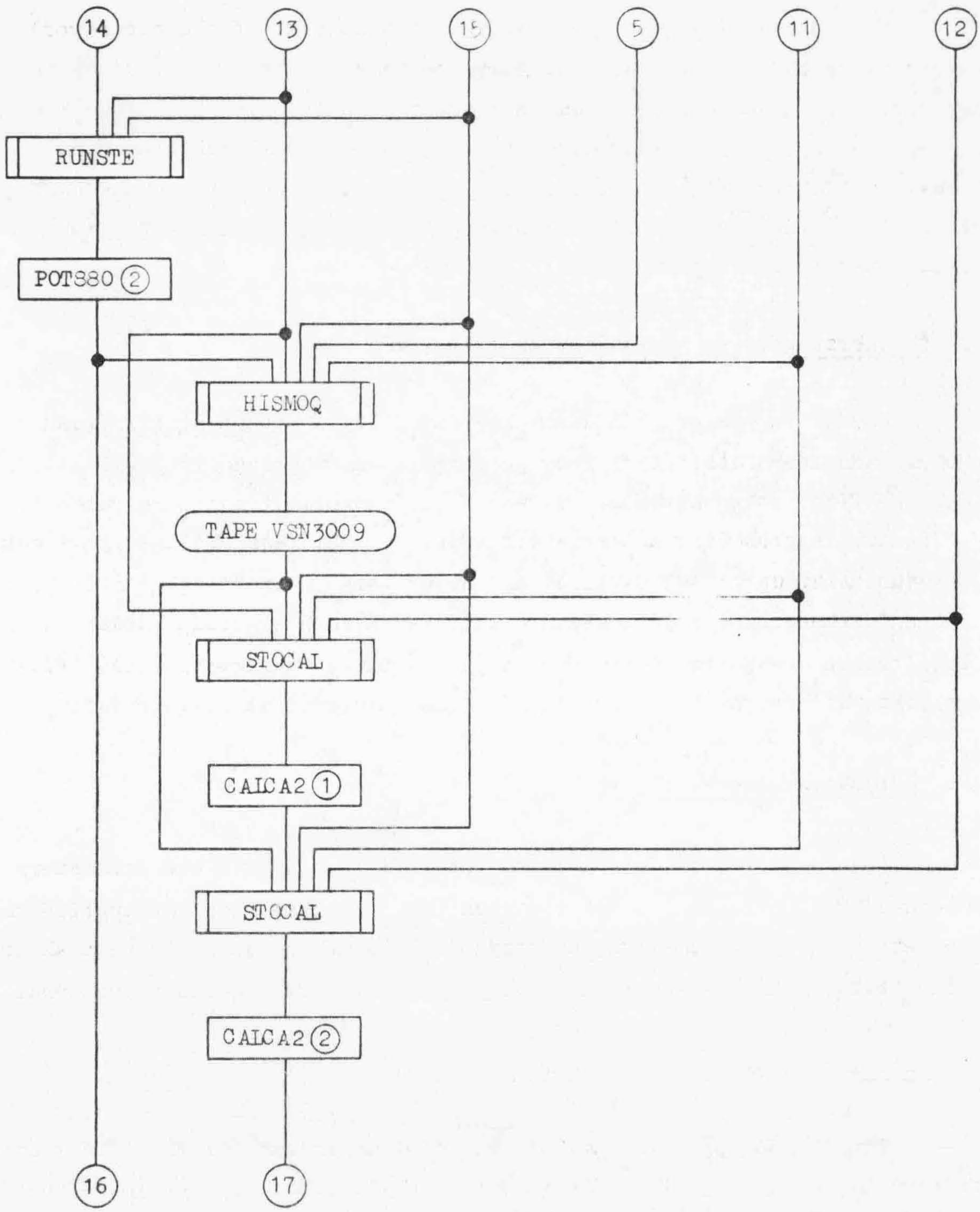


Fig.11 STEP 3
SCALING OF HYDR. PARAMETERS AND CALCULATION OF STORAGE COEFFICIENTS

Record No 1978/12

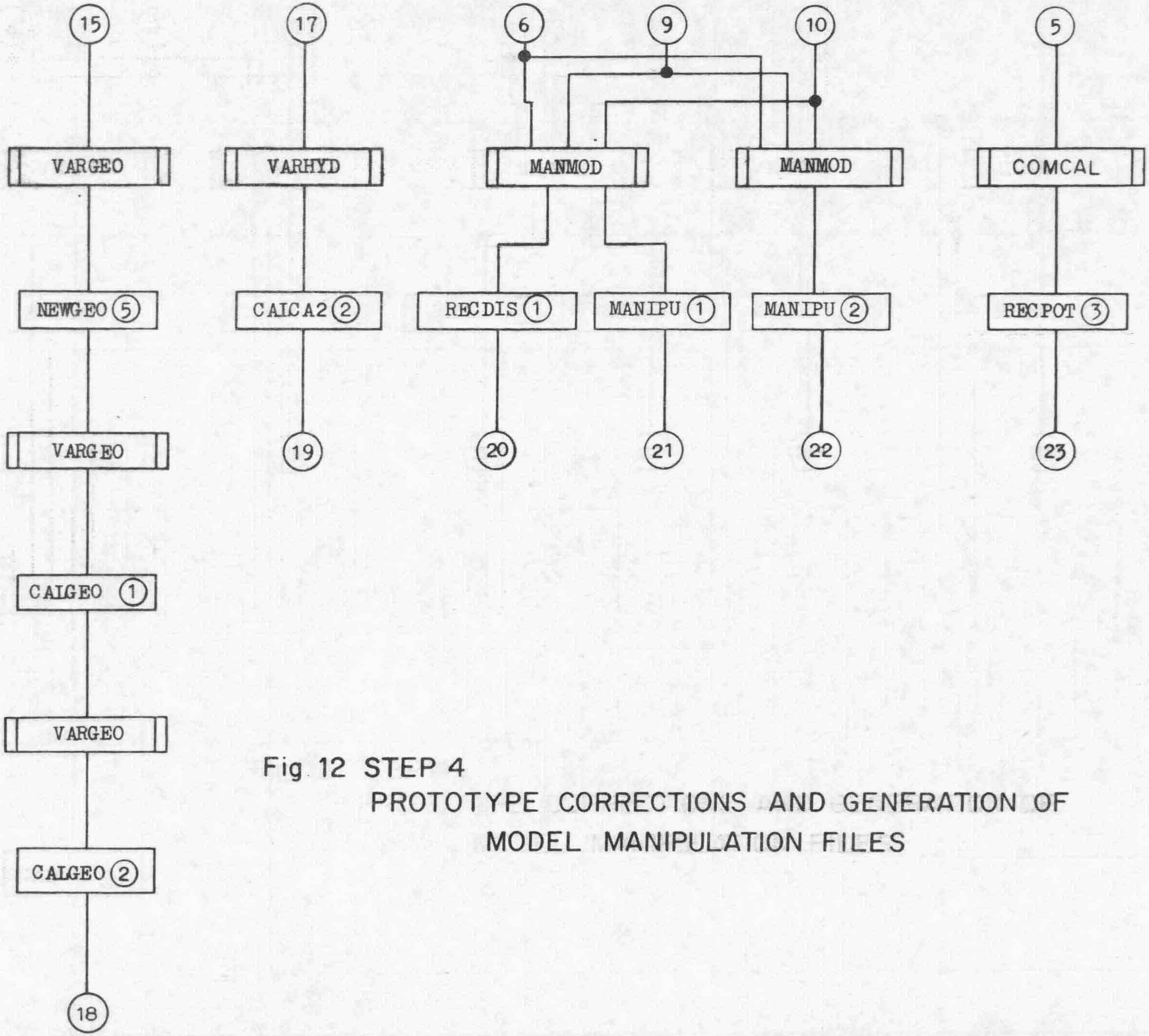


Fig 12 STEP 4

PROTOTYPE CORRECTIONS AND GENERATION OF MODEL MANIPULATION FILES

AUS 1/763

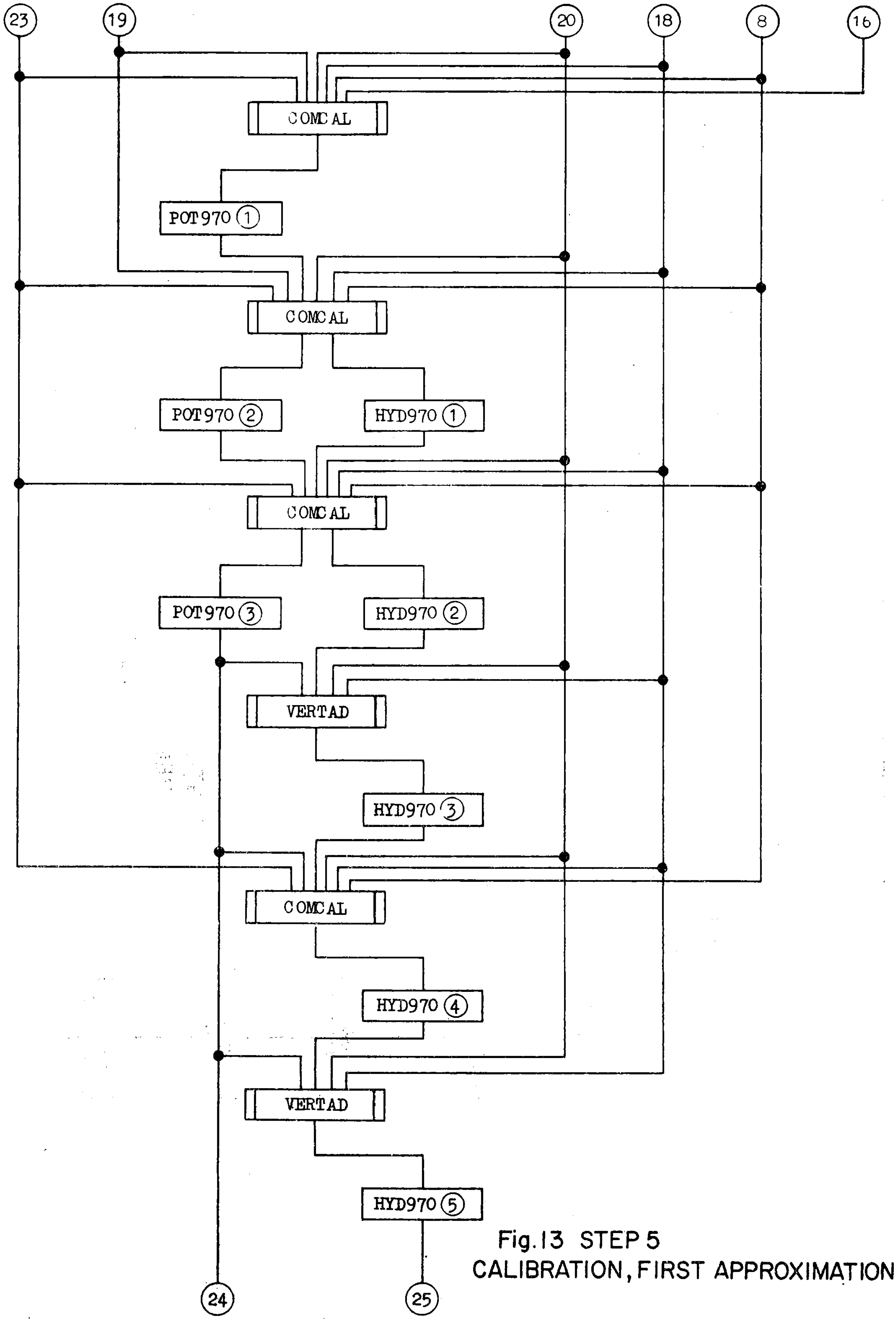


Fig. 13 STEP 5
CALIBRATION, FIRST APPROXIMATION

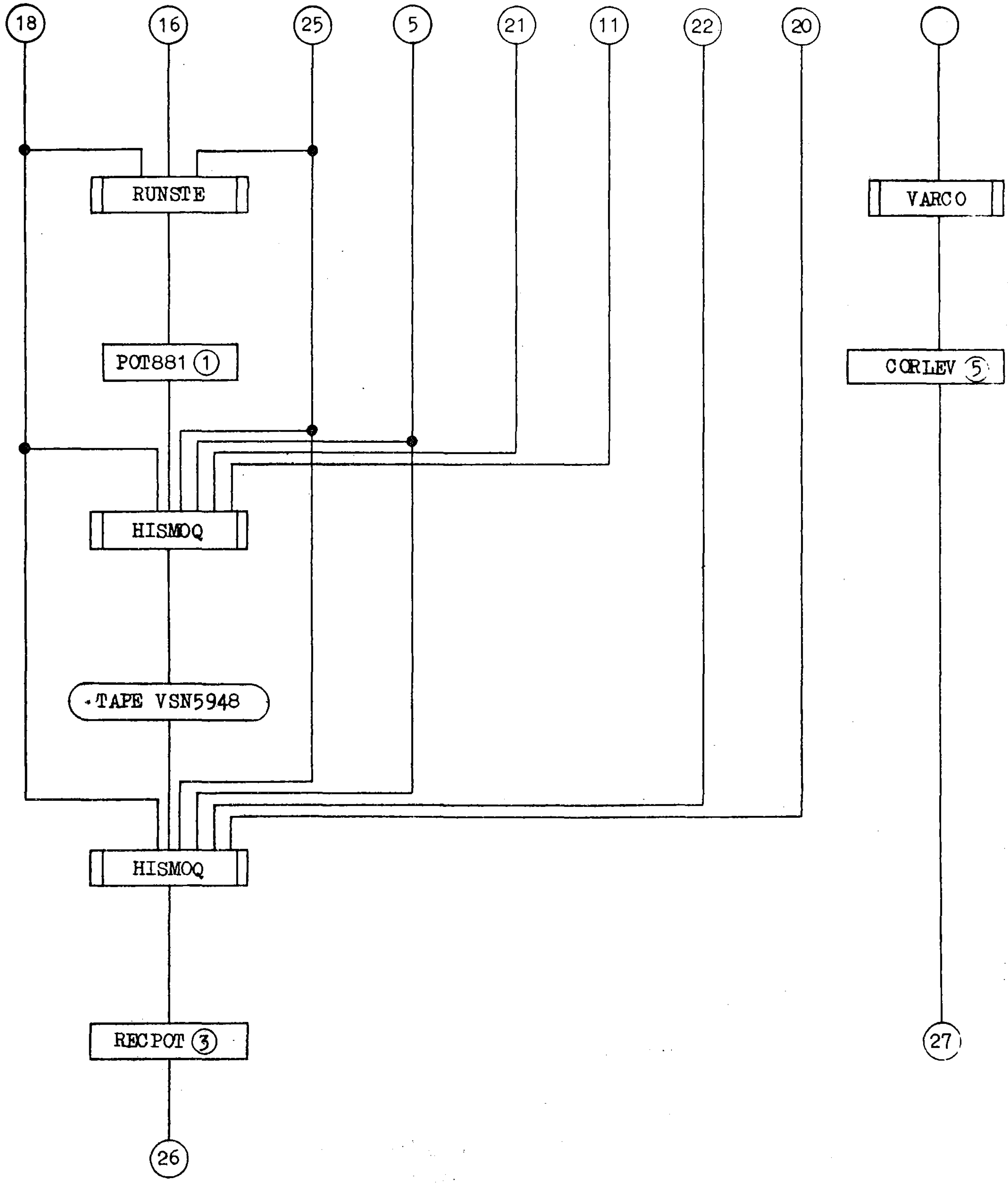


Fig. 14 PROTOTYPE ADJUSTMENT

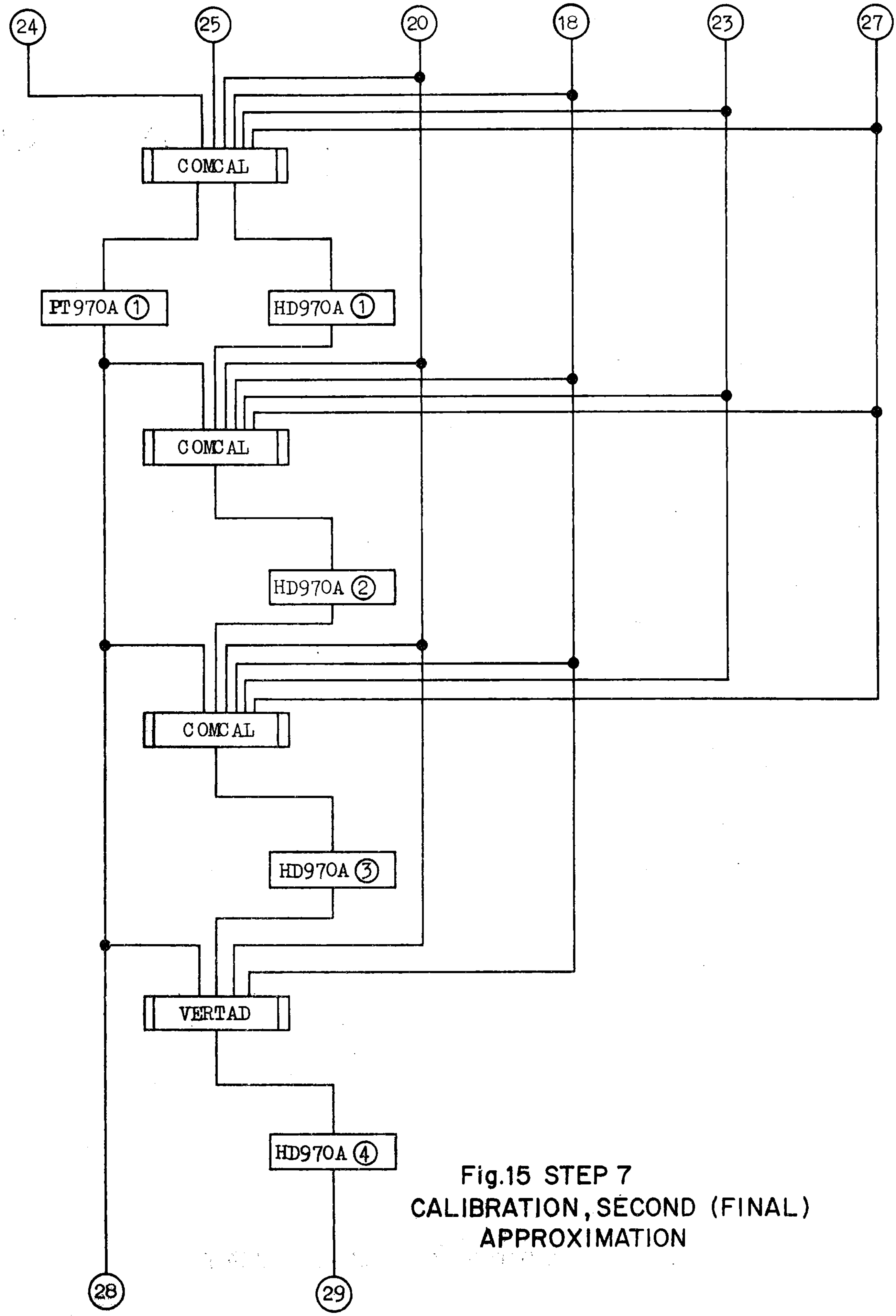


Fig.15 STEP 7
CALIBRATION, SECOND (FINAL) APPROXIMATION

2.6 Further work

The GABBRI system of programs to generate the initial data base is still incomplete because intermediate data processing results from the obsolete GABSIM model were available. GABBRI will not be required until a re-calibration of the model is attempted and it should be completed then.

The present data base is not complete. A significant number of flowing artesian bores in the New South Wales portion of the basin were missed during the data transcription phase of the study. Some discharge values have become available for South Australian bores, which were not available when the data base was generated and should be included at the first opportunity. The same applies to discharge measurements of mound springs. There appears to be no prospect of obtaining enough data for the Cretaceous aquifers to include them in the model more effectively than is done at present.

Some further work is recommended on the study of the aquifer geometry and hydraulics in the Eulo Ridge area and other areas where 'physically impossible' potentials were observed. If accurate predictions are required for these areas it might be necessary to establish a more detailed model specifically for them.

3. THE CALIBRATION PROGRAMS

The major programs developed for the calibration are STOCAL for the scaling of parameters and determination of storativities; COMCAL for calculation of transmissivities and simultaneous adjustment of potentials; and VERTAD for checking the cell-by-cell water balances and optionally adjusting vertical leakage factors. COMCAL and VERTAD are both based directly on the GABHYD model equation, which is described briefly in 3.1. Sections 3.4, 3.5, and 3.6, which describe the operation of the individual calibration programs, are each preceded by a summary of the theory and principles on which they operate; however it is recommended that the more detailed description of the theory presented in chapters 2.4.2 through 2.4.5 is referred to in each case. Each program description is accompanied by the corresponding flow chart. For full program printouts refer to Appendix A.

3.1 The GABHYD model equation

The GABHYD model equation is the finite difference form of the continuity equation for the model discretisation shown in Figure 16. Each model cell water balance consists of horizontal flow terms, vertical flow terms, artificial discharge terms, and change in storage terms. The sum of horizontal and vertical flow terms may be presented by the equation based on Darcy's Law;

$$\sum_i K_i \, (Q_i - Q_o) \, g_i$$

with i referring to each of the six sides of the model cells.

The variables and constants are:

K = hydraulic conductivity

Q = hydraulic head

g = geometrical constant (cross-section of flow/length of path)

Using the configuration as in Figure 16:

$$g_i = \frac{d\ a}{a} = D \text{ for horizontal flows}$$

$$g_i = \frac{a^2}{b} \quad \text{for vertical flows}$$

From this we may define directional transmissivities for both horizontal and vertical flows:

$$T_i = K_i \, d \qquad \text{horizontal}$$

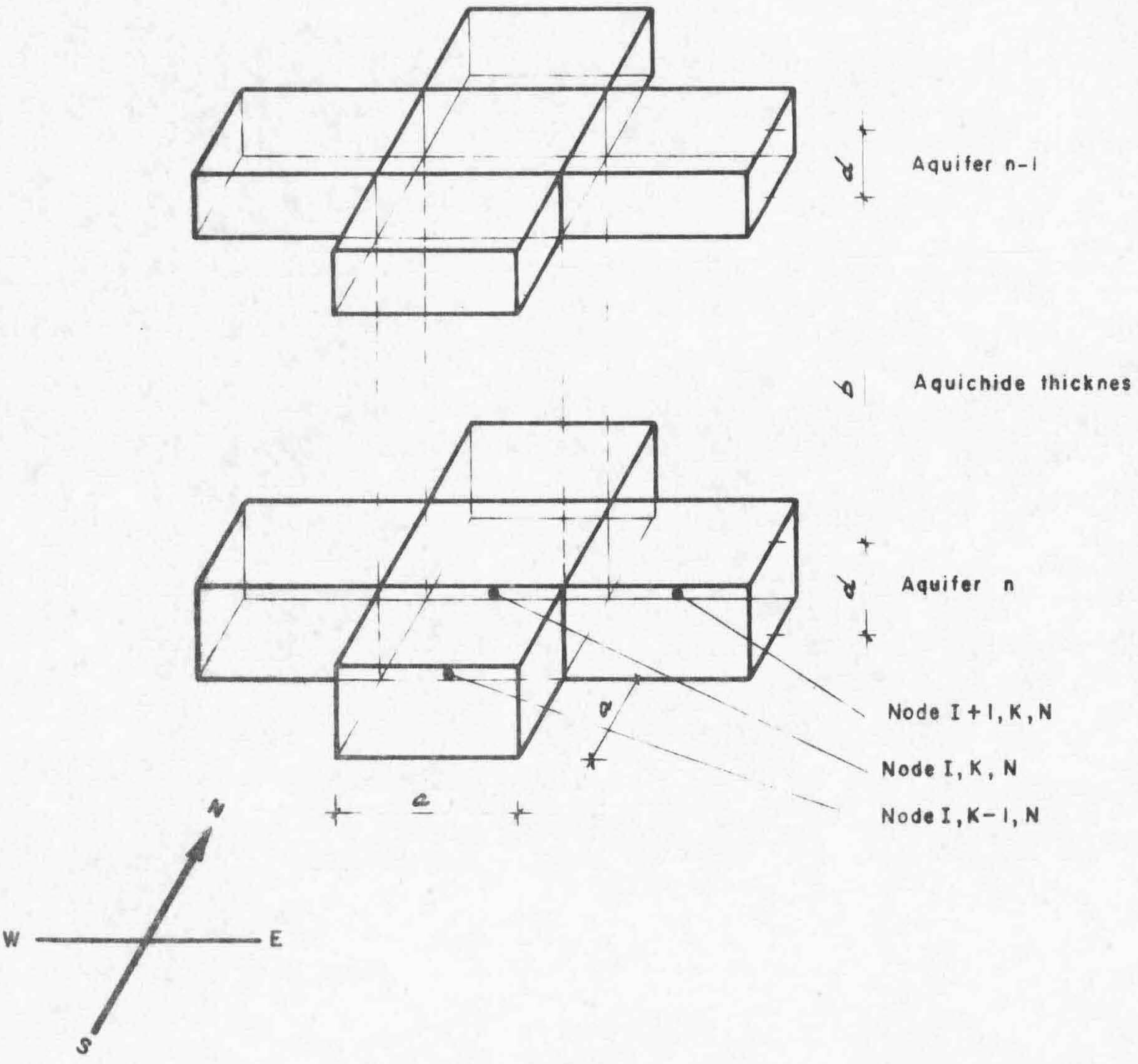
$$T_i = K_i \, \frac{a^2}{b} \qquad \text{vertical}$$

The generalised flow balance may then be written as:

$$\sum_i \; T_i \, (Q_i - Q_o)$$

If g is the well discharge and $\frac{Q\ a^2\ S}{t}$ is the change in storage, where S is the storativity, the complete equation is:

$$\sum_i \; T_i \, (Q_i - Q_o) + \frac{Q\ a^2\ S}{t} + g = 0$$



Fig, 16. CELL (NODE) STRUCTURE OF GREAT ARTESIAN BASIN PROTOTYPE

In the ultimate form of this model equation, the term $(Q_o^{t+\Delta t} - Q_o^{t})$ is subsituted for Q. Subscript 0 denotes the central nodes and the overscripts refer to the time at which the value is taken. If $Q_o^{t+\Delta t}$ is used as Q_o in the horizontal and vertical flow components the implicit form of the equation results. But this is of little relevance for the calibration. A more detailed derivation is presented in the documentation of the model itself.

3.2 The GABHYD model data files

A consistent internal data structure is maintained throughout the GABHYD program system. This allows standardisation of data read and write routines and of the data file structures. Table 1 illustrates this data structure with the dimensions of the GABHYD prototype. The same groups of variables may be referenced by different names. For example the east-west directional transmissivity may be referenced either separatelyas TE or as part of HY. Within the computer all these data are defined in one 'COMMON' block which is passed from one program routine to another as a complete unit. The array names of Table 1 occur throughout the programs. Other recurring variable names are:

MAQ or NAQ	number of aquifer(s)
MEW	number of east-west nodes
MNS	number of north-south nodes
A	separation between gridlines (km)

Other variable names or differing uses of variable names are defined on the flow-charts of the individual program units.

For data input and output the data block is subdivided into four logical groups corresponding to standard data files:

PA	standard potentials file POT
HY	standard hydraulic parameter file HYD
DQ	standard discharge file DIS, or manipulation file MAN
IG	geometry code file GEO

For compatability with card files each of these standard files has a logical record length of eighty characters.

Files POT, HYD, and DIS share the same record structure.

TABLE 1. INTERNAL DATA STRUCTURE OF GABHYD

Data layer	Contents	Array detailed	Names general
1	Water-table potentials (metres)	PA	PA
2	Potentials aquifer 1 at time t - Δ t		
3	Potentials aquifer 1 at time t		
4	Potentials aquifer 2 at time t - Δ t		
5	Potentials aquifer 2 at time t		
6	Vertical leakage factor up from aquifer 1	VP	HY
7	Vertical leakage factor up from aquifer 2		
8	Transmissivity east-west aquifer 1	TE	
9	Transmissivity east-west aquifer 2		
10	Transmissivity north-south aquifer 1	TS	
11	Transmissivity north-south aquifer 2		
12	Storativity aquifer 1	SC	
13	Storativity aquifer 2		
14	Well discharge year t aquifer 1	DQ	DQ
15	Well discharge year t aquifer 2		
16	Geometry codes combined	IG	IG

Each of the records consists of an identifier field 14 characters long followed by 6 numerical data fields of 11 characters each. The identifier field consists of:

YEAbAbNSbEWbTb

b	=	blank field
YEA	=	year number in three digits (e.g. 910 for 1910)
A	=	aquifer number 1 or 2 (0 for water table)
NS	=	node index in north-south direction
EW	=	node index in east-west direction for the node to which the first data value in the record applies.
T	=	data type code, which may be
		E = east-west transmissivity

N = north-south transmissivity

S = storativity

Z = vertical leakage factor

P = current potential (at time t)

I = previous potential (at time t $- \Delta t$)

W = water-table potential

Q = discharge

T = aquifer thickness

G = ground elevation correction

T and G refer to items which are not part of the standard GABHYD data block, but they are stored in files of identical structure. The numerical fields for the standard files are all in format E11.5 (FORTRAN).

File GEO contains the geometry codes. Each record is headed by a ten-character identifier field.

NSbEWbbbGb

The symbols are as above except that the letter G here stands for geometry. The identifier field is followed by seven numerical fields, each containing ten integer digits. Successive integers in the field correspond to the aquifer numbers, e.g. the first integer is for aquifer number one, etc. The codes themselves may be

0 node is outside of defined area of aquifer

1 node is on an impermeable boundary

2 node is on permeable boundary with prescribed potential

4 node is inside aquifer, recorded potentials are available

5 node is inside aquifer, recorded potentials are not available.

File MAN, the discharge manipulation file, has an identifier field of eight characters preceding each record.

YEAANSEW

codes YEA, A, NS, EW are defined as above. There are four numerical data fields each consisting of one blank and seventeen integers. Each group of seventeen integers is decoded into a prescribed discharge or into a set of instructions and parameters to calculate a free-flowing discharge. The decoding and discharge calculation is done by subroutine PLAYMO, which is described as part of the GABHYD model programs.

3.3 Common subroutines

The subroutines which were developed for a specific step of the calibration are described with that step in the following sections. Some of the subroutines which are used in the calibration have originally been developed for other program groups and are described in the documentation of these. An alphabetical list of programs and subroutines (Appendix A) provides the necessary references. Subroutines developed for the calibration program group which are used by more than one program are:

ERANG, a modification of subroutine RANGE of the RUNMOD group. Both subroutines determine the indices of the nodes at the aquifer boundaries. ERANG includes permeable boundary nodes with the aquifer; RANGE excludes them.

IGEO is a function returning one value only: for the specified node the compound geometry code is split and the geometry code for the specified aquifer is returned.

OUTHYD is a simple output routine which writes the data block containing the hydraulic parameters onto a standard hydraulic data file of the format described in 3.2.

3.4 Calculation of storativities with program STOCAL

STOCAL has been designed to carry out two functions, both based on balancing the overall water budget of an aquifer. Firstly it scales all transmissivities and permeabilities with a uniform factor so that the balance of all flows other than well discharge and without yield from storage provides a specified percentage portion of the recorded well discharge. The remaining portion of the well discharge is then provided from changes in storage. Secondly it was designed to calculate the overall volume drawdown of the potential surface for a specified time interval, multiply it by the aquifer thickness to provide the 'effective drawdown volume', and relate this to the cumulative imbalance of the water budget for the same period. From this is obtained a uniform specific storage coefficient for the aquifer. This specific storage coefficient when multiplied with the aquifer thickness yields the storativity to exactly balance the water budget of the aquifer for the time period considered.

3.4.1 Program STOCAL (Fig. 17)

STOCAL has four operating nodes which are invoked by specification of a three-character control word. These control words and the associated operating modes are listed below.

IMB: imbalanced mode - hydraulic parameters are scaled so that a specified percentage of well discharges is provided from surplus natural flows.

BAL: balanced mode - hydraulic parameters are assumed to be properly scaled. A specific storage coefficient is calculated and applied to balance the budget.

INF: information mode - both parameter scale factors and specific storage coefficient are calculated but not applied, i.e. no new parameter file is written.

SPE: specified mode - specified specific storage coefficients applied without balance calculations.

Figure 17 is a detailed flow chart for STOCAL. The program accesses 6 input and 2 output files:

INPUT	6 records as cards or card images containing control parameters (see 3.4.3).
GEO	geometry file
POT	potentials file
DIS	discharge file
OLD	hydraulic parameters file containing old data to be adjusted
THI	aquifer thickness file
OUTPUT	printed messages and summary of results
NEW	hydraulic parameter file containing new values

STOCAL calls the subroutines SLEAKA, SBOUND, STOTAQ, EFDRA, which are described in 3.4.2, and the external subroutines HISDIS, REDGEO, REDPOT, REDHYD, OUTHYD, and function IGEO.

As its first step the program initialises the arrays for water balances, drawdowns, and storativities, and reads from file INPUT the operating parameters. Time step parameters are adjusted to conform with the time loop logic of the program. Still from INPUT values for the specific storage coefficients for each aquifer are read. These will be of significance only in the 'SPE' mode of operation. Calling subroutines REDGEO and REDHYD the geometry file and the old hydraulic parameters are read. Then the program branches. If the operating mode selected is 'SPE' then control is transferred to a point further down on the flow chart, otherwise the loop counter is set to zero and the time step loop is entered. For each time step the first and the last year of the time step are transmitted to the subroutines SLEAKA, SBOUND, STOTAQ, EFDRA: these return average values for this period of leakage volumes, boundary discharge volumes, well discharge volumes, and drawdown volumes which are added into the relevant arrays and summation variables. After completion of the loop, specific storage coefficients are calculated and printed together with the summation variables. The parameter scaling factor is calculated from the summation variables of the last time step and from the percentage value, and is printed. If the operating mode is 'INF', execution terminates here. Else another branch is entered. If the operating mode is not 'BAL' then all transmissivities and vertical leakage factors are multiplied by the scaling factor, the message 'new hydraulic parameters' is printed, and a new hydraulic parameter file is written. If the operating mode is 'BAL' then the other branch is taken where the program flow is joined by the transfer for operating mode 'SPE'. The aquifer thicknesses are read and multiplied by the specified or calculated specific storage coefficients. The messáge 'new storage coefficients' is printed and a new hydraulic parameter file is written.

3.4.2 Subroutines SLEAKA, SBOUND, STOTAQ, EFDRA

These subroutines calculate individual subtotals of water balances or effective drawdowns. The flowcharts are Figures 18a-d.

SLEAKA adds vertical leakage flows from below and to above. Subroutine call parameters are:

NAQ number of aquifer
NYB start of time interval
NYE end of time interval
NFP logical unit number of file containing potentials.

By calling subroutine REDPOT the potentials for year NYB are obtained. Leakage arrays are set to initial values. Then the aquifer is traversed systematically, geometry codes are broken up, and if indicated by the code the leakage to or from the aquifer above or below is calculated. The same procedure is followed for year NYE. The individual leakage values are then calculated as the arithmetic mean of the values for years NYB and NYE. Leakage rates are converted into volumes by multiplication with the time interval NYE-NYB in seconds. The difference between up and down leakage, i.e. the net gain or loss, for each node is calculated and added up to yield a leakage balance subtotal for the time period.

SBOUND calculates the boundary inflows and outflows around the perimeter of each aquifer. Call parameters are the same as for SLEAKA. SBOUND calls subroutine REDPOT to read potentials for year NYB and initialises the boundary discharge array. The aquifer is then traversed node by node. For each one and its four surrounding nodes the geometry code is split by calling function IGEO. The node is skipped unless it is of type 2, i.e. is on a permeable boundary, otherwise the discharge to or from each interior node connected to the boundary node is calculated by multiplying the potential gradient with the connecting directional transmissivity. The same procedure is followed for the year NYE. An average boundary discharge volume for the time interval is determined by averaging discharges node by node and by multiplying them with the time interval NYE-NYB. All boundary discharges are added to a boundary discharge subtotal for the period.

STOTAQ sums the well discharge for the specified aquifer and time period. Call parameters are as above except that NFD is the logical file number of the discharge file. STOTAQ obtains discharge values for years NYB and NYE by calling subroutine HISDIS. Discharges are averaged and multiplied by the time interval to yield a discharge volume. Node-by-node values are added to a discharge subtotal for the whole aquifer.

EFDRA calculates the effective drawdown volume for the specified aquifer and time interval. Call parameters are:

NAQ, NYB, NYE as above	
NFP	logical file number potentials file
NFT	logical file number aquifer thickness file

EFDRA obtains the potentials for years NYB and NYE by calling subroutine REDPOT and calculates the difference between them, which is the 'pressure drawdown'. The aquifer thickness is read from the thickness file and multiplied with the pressure drawdown and the aquifer cell cross section (A x A) for each node. These values are added to yield the effective drawdown volume for the specified aquifer and time period.

3.4.3 Specification of operating parameters

There is one set of semipermanent operating parameters which recurs throughout the programs and is specified at the beginning of each program by assignment rather than by reading data. These parameters are related to the prototype geometry and do not require further consideration after the prototype has been defined. These parameters are:

A spacing between gridlines (25 000 m)
MAQ number of aquifers in prototype (2)
MEW number of nodes in east-west direction (67)
MNS number of nodes in north-south direction (58)

Operating parameters subject to change from one program run to the next are defined by reading them as data from INPUT. These are:

NYST, NYEN, NSTP start and end of time interval and time step length. The specification 1900-1940, 10 would result in a time loop with the steps 1900-1910, 1910-1920, ..., 1930 - 1940. The interval should span a period during which significant changes in pressures occurred and the time step should be short enough for the averaging of flows to be meaningful.

NA the aquifer number. If specified either as 1 or 2 values for the corresponding aquifer will be calculated. Any other specification results in calculation for both aquifers.

FBA specifies the operating mode as discussed in 3.4.1

PCT specifies the percentage of well discharge derived from a surplus in natural flows rather than from a change in storage. This percentage is high when the aquifer is near equilibrium and is used for determination of the parameter scaling factor.

TSPEC (1),TSPE (2) prescribe specific storage coefficients for aquifers 1 and 2. They must be specified but are used only in the 'SPE' type operating mode.

3.5 Calculation of transmissivities and adjustment of potentials with program COMCAL

Program COMCAL is designed to calculate transmissivities according to current model potentials, while at the same time adjusting the current potentials towards the recorded potentials. In practice these two operations are only virtually simultaneous; in detail there is a rapid alteration between the two operations.

The first operation is the recalculation of transmissivities so that the flow balance between pairs of adjoining model cells, referred to as primary and secondary cell, is more uniform. This adjustment includes a simultaneous penalty in the form of an imposed extra component in the water balance proportional to the difference between the current and the recorded potential. If aquifer cells adjoin permeable recharge boundary, an attempt is made to adjust the recharge flow to achieve a completely neutral balance. The boundary node balance may in addition be loaded by an imposed extra component designed to accelerate overall convergence by over compensation. During the traverse of the aquifer cells, each interior cell becomes a primary node once with secondary cells to the north and to the east. Both directions will be taken but the choice of which is to be taken first is made based on the precalculated uniformity of the transmissivity distribution which would result from either decision.

After all interior nodes have been traversed once, subroutine CALMOD is called to calculate new current potentials based on the modified transmissivities. CALMOD is one of the major subroutines of the RUNMOD group of programs and is described in the documentation of the model itself.

3.5.1 Program COMCAL (Fig 19)

COMCAL was developed originally as a program to calculate transmissivities from potentials only and it can still be used in this mode by specifying a zero number of iterations for recalculating potentials through program control variable MAXI (3.5.3). In its current form it is applicable to near steady state conditions with negligible change in storage components. However, it could be extended to include change in storage provided that the requirement for accurate specification of the changes in potential for each node can be met.

Program COMCAL accesses 7 input and 3 output files:

INPUT	8 records in card image form containing program control parameters (3.5.3)
GEO	geometry data file
OLD	hydraulic parameter file containing old values
POT	potentials file containing current potentials
MAN	discharge data file
REC	potentials file containing recorded potentials
GEL	ground-elevation corrections file
OUTPUT	printer file containing messages and error log for each iteration
NEW	hydraulic parameter file containing new values
CAP	potentials file containing new current values.

Program COMCAL calls subroutine REXBAL (described in 3.5.2) and the external subroutines CALMOD, ERANG, REDGEO, REDHYD, REDPOT, SERDAT, OUTHYD, OUTPOT.

After reading from INPUT the control parameters COMCAL calls the input subroutines to read the model geometry, to calculate and geometrical loop ranges, and to read initial potentials, hydraulic parameters, and ground-elevation corrections for potentials. In a loop the discharge variables are initialised and the ground-elevation corrections are stored in an otherwise unused portion of the potentials data array. A scaling factor is applied to the vertical leakage factors. Subroutines are then called to read recorded discharges and recorded potentials. The control parameters read at the beginning are printed.

Loop 100, the iterative loop to adjust hydraulic parameters, is entered, summation variables are initialised to zero, and the model nodes are traversed (loop 10) within the geometrical range calculated by subroutine ERANG previously. For each node, two secondary nodes are defined: one to the north, the other to the east of the primary node. A provision for reversing these directions is not used in the current version of the program. Subroutine REXBAL is then called for both combinations of primary with secondary node. For each case the variation coefficients are calculated for both the original water-balance residuals and the new residuals. The ratios between the respective coefficients, which indicate the improvement in the water balances are stored. Similar ratios indicating the degree of uniformity are then calculated from the variations of old and new transmissivities. The respective ratios are then multiplied as criterion to decide which combination of primary and secondary node adjustment results in the more homogeneous combination of balance residuals and transmissivities. The transmissivities from the more uniform combination are adopted as new transmissivities and then used when calling subroutine REXBAL for the other combination of primary and secondary node. The purpose of this sequence is to use a rational rather than a chance criterion for choosing the direction for the first transmissivity adjustments. The residual water balances and hydraulic gradients at boundaries, if applicable, are added to the statistical summation variables. This completes loop 10.

After all modes have been traversed the absolute and standard water-balance errors are calculated and printed.

Then subroutine CALMOD is called to recalculate potentials with the new transmissivities provided that the control parameter MAXI had been specified as 1 or greater and provided that the standard error is not already less than the convergence criterion. This completes iteration loop 100. By subroutine call the new hydraulic parameters and the new potentials are written out onto their respective files.

3.5.2 Subroutine REXBAL (Fig. 20)

REXBAL calculates the current water balances for two cells called primary and secondary, and adjusts the directional transmissivity connecting them. Subroutine call parameters for REXBAL are:

RCF	potentials bias factor
IP, KP, NP	node (cell) indices for primary
IS, KS, NS	node indices for secondary
LFL	boundary treatment code
SBC, SBN	old and new water-balances primary node
TC, TN	old and new connecting transmissivities
SBS, SBT	old and new water-balances secondary node
NSTP	time step (only required for non-steady state)

Subroutine REXBAL first initialises balances and splits up the geometry code string for each node into individual codes for each aquifer. The current potentials in array PA are redefined as simple variables, and the difference between them is stored as the hydraulic gradient. In loops 11, 13, 21, and 23 is determined which number aquifer is in hydraulic connection with the aquifer considered at the primary and secondary node respectively. Then the individual flow components are calculated for the primary node. A target potential is defined in accordance with the recorded potential or if that is not available the current potential is restrained by any data of ground-elevation correction which might be available. The overall flow balance is calculated and is loaded additionally with an imbalance proportional to the difference between current and target potential. The same calculations are then carried out for the secondary node. The connecting transmissivity is defined as TC.

If any of the two nodes is on a boundary which is not subject to adjustment a transfer is made past the transmissivity adjustment branch. If one of the two nodes is on a boundary to be adjusted (recharge or discharge) its balance is set to a nominal value opposite to the residual balance of the interior node connected to it. This interior node, if so specified by control parameter ORF, has already been loaded with a portion of the previous total imbalance of all interior nodes. Then the connecting flow is calculated, which if applied would equalise the balances of primary and secondary nodes. The corresponding value for the connecting transmissivity is calculated and restrained by maxima and minima. The restrained value multiplied with the hydraulic gradient then yields the actual new connecting flow and the water balances for each node are adjusted accordingly. The water balance of the primary node is reset for zero if it is on any boundary.

3.5.3 Specification of operating parameters

The parameters to control the operation of COMCAL are:

IYST, IYEN start and end of time interval in years, e.g. 1960-1970. The initial potentials will be read for IYST. The recorded potentials and discharges for IYEN, IYST and IYEN need not to be different.

NA number of aquifer.

NBN boundary treatment code defined as:

0 no boundary transmissivities are calculated

+1 only recharge boundary transmissivities are calculated

-1 only discharge boundary transmissivities are calculated

9 all boundary (permeable) transmissivities are calculated.

MIT maximum number of iterations in loop 100 to recalculate transmissivities.

CONV convergence criterion (m^3/sec). Iterations continue only as long as the standard water-balance error exceeds CONV.

ORF overcompensation factor, determines the degree to which nodes at boundaries are loaded with the overall imbalance from the previous iteration. Normal range for ORF : 0.0 to 1.0.

VERSC scaling factor applied to vertical leakage factors. The file of new hydraulic parameters written out after execution of the program will retain this scaling factor.

BIAS bias factor (m^3/sec) to specify on overall water balance bias for the whole aquifer. The actual bias applied will be BIAS x ORF.

RCF potentials balance bias factor specifies a balance penalty for each node proportional to the difference between the current and the target (recorded) potential at that node. In the case of GABHYD it was specified in the range 0.003 to 0.005 m^2/sec.

MAXI number of iteration in each adjustment step for potentials from hydraulic parameters.

CON convergence limit for iterations to recalculate potentials.

3.6 Adjustment of vertical leakage factors

The purpose of this final step in the calibration sequence is to check the node-by-node balances and allow for some residual corrections by altering vertical leakage factors. To achieve this the program can be operated in the following modes:

Mode 0 vertical leakage factors are not altered except where a downward leakage is observed. Downward pressure gradients are assumed to be negligible in the current version of the model prototype and in this case the vertical leakage is set to a minimum. All imbalances are calculated and printed out symbolically on a small map.

Mode 1 as mode 0 but in addition all vertical leakage factors are adjusted to minimise the node water balance errors. The new values are restrained by the limits of 0.5 and 2.0 times the original value

Mode 2 as mode 1 but no restraints on new vertical leakage factor other than to be positive.

3.6.1 Program VERTAD (Fig. 21)

VERTAD uses 5 input and 2 output files:

INPUT	3 card image records specifying program control parameters
GEO	geometrical data file
POT	current potentials file
DIS	discharge data file
OLD	hydraulic parameter file containing old values
OUTPUT	printer file for messages, aquifer balances, standard error, and one page symbolic map of residual balance for each node
NEW	hydraulic parameter file containing new values.

The main subroutine called by VERTAD is VERCAL, which is described in 3.6.2. Other subroutines called are ERANG, OUTHYD, HISDIS, REDGEO, REDHYD, REDPOT, SERDAT, SMAMAP.

At first VERTAD reads the control parameters from file INPUT and assigns title headings. By calling the appropriate subroutines it then reads geometrical, potential, and hydraulic data. This data reading and assignment procedure is as described for COMCAL in 3.5.1.

Following data input and initialisation the statistical summation parameters are initialised. The aquifer is then traversed systematically. For each node, subroutine VERCAL is called and the corresponding element of the balance error array AR is set to the corrected node balance. Error balances are added to the statistical summation parameters and the new vertical leakage factors are added to a subtotal.

After all nodes within the aquifer have been traversed the standard deviations and the ratios of average permeabilities are calculated and printed together with the error sums. A one-page map showing the error on each node is printed and the new hydraulic parameters are written onto file NEW by calling subroutine OUTHYD.

3.6.2 Subroutine VERCAL (Fig. 22)

VERCAL calculates the water balance for the specified node with the current hydraulic parameters and determines a modified vertical leakage factor for the upward leakage, which when replacing the existing vertical leakage factor would eliminate or reduce any water imbalance.

The GABHYD data set is transmitted to VERCAL through the COMMON block 'LEV'. The subroutine call parameters are:

IN, KN, NN indices of node to be adjusted

NSTP time step in years (non-steady state only)

OLB old node water balance (m^3/sec)

WEB new node water balance

MODE operating mode (as described in 3.5)

NSTP is converted into the time in seconds STP; EPSI the lower limit for leakage factors is set to a small positive value, e.g. 10^{-14}. Node water balances OLB and WEB are initialised to zero. The aquifer numbers are determined for the aquifers to which the node is connected by vertical leakage. The directional flow components of the water balance - FN, FS, FE, FW, FU, and FD - are calculated using the current hydraulic parameters and added to the components due to the change in storage ST and discharge FQ to yield the current (old) water balance OLB. FN is defined as the difference between the current upward leakage (FC - FU) and the residual balance OLB. The node balance would become zero if FN were to replace FC in the balance calculation. COR = FN/FC is the ratio by which the current vertical leakage factor would have to be multiplied to implement this corrected leakage. TN, the new leakage factor, however, is restrained to a positive value of at least EPSI and additionally if the operating mode is 1 it is restrained to ½ to 2 times the original value. If the operating mode is 0 the leakage factor is altered only (set to minimum) if the leakage is downwards.

3.6.3 Specification of operating parameters

The parameters read for INPUT are:

IYST, IYEN start and end of line interval in years, steady state is implied if IYST = IYEN.

NA number of aquifer to be adjusted

MODE operating mode 0, 1, or 2 as defined in 3.6.

4. REFERENCES

AUDIBERT M., 1976 - Progress report on the Great Artesian Basin hydrogeological study 1972-1974. Bureau of Mineral Resources, Australia, Record 1976/5 (unpublished).

EMSELLEM, Y., & MARSILY, G., 1971 - An automatic solution for the inverse problem. Water Resources Research, 7(5), 1264-83.

FRIND, E.O., & PINDER, G.F., 1973 - Galerkin solution of the inverse problem for aquifer transmissivity. Water Resources Research, 9(5), 1397-410.

HAZEL, C.P., 1973 - Lecture notes on groundwater hydraulics. Australian Water Resources, 1973, Groundwater School, Adelaide.

HEFEZ, E., SHAMIR, U., & BEAR, J., 1975 - Identifying the parameters of an aquifer cell model. Water Resources Research, 11(6), 993-1004.

KLEINECKE, D., 1971 - Use of linear programming for estimating geohydrological parameters of groundwater basins. Water Resources Research, 7(2), 367-74.

KREBS, G., 1973 - Data check programs for the Great Artesian Basin ADP system. Bureau of Mineral Resources, Australia, Record 1973/203 (unpublished).

KREBS, G., 1974 - Processing programs for the Great Artesian Basin ADP system. Bureau of Mineral Resources, Australia, Record 1974/154 (unpublished).

NELSON, R.N., 1968 - In place determination of permeability distribution for heterogeneous porous media through analysis of energy dissipation. Society of Petroleum Engineers Journal 8(1), 33-42.

SEIDEL, G., 1973 - The storage and retrieval system for hydrogeological data from the Great Artesian Basin. Bureau of Mineral Resources, Australia, Record 1973/211 (unpublished).

UNGEMACH, P., & HABERMEHL, M.A., 1973 - Great Artesian Basin groundwater project; automatic data processing storage and retrieval system - data transcription manual. Bureau of Mineral Resources, Australia, Record 1973/25 (unpublished).

UNGEMACH, P., 1975 - Great Artesian Basin groundwater project - explanatory note on digital model package, Great Artesian Basin simulation model (GABSIM). Bureau of Mineral Resources, Australia, Record 1975/169 (unpublished).

APPENDIX A: ALPHABETICAL LIST OF PROGRAMS AND SUBROUTINES

Name	Type	System	Use	Ref.
BRIDGE	M	GABBRI	Rewrite GABSIM data file for use by GABHYD	Tem
CALMOD	S	RUNMOD	Calculate potentials	Ext
COMCAL	M	CALSYS	Main calibration program	3.5.1
EFDRA	S	CALSYS	Calculate effective drawdown	3.4.2
ERANG	S	CALSYS	Determine indices of boundaries	3.3
GELEV	M	GABBRI	Write ground-elevation correction file	Ext
GRETRI	M	DATA	Retrieve data from GAB data bank	Ext
HISDIS	S	RUNMOD	Read discharge data file	Ext
HISMOQ	M	RUNMOD	Reproduce historic record with interpolations	Ext
IGEO	F	CALSYS	Split compound geometry code	3.3
MANMOD	M	RUNMOD	Write discharge model manipulation file	Ext
MAPDIG	M	GABBRI	Assign grid values from digitised contours	Ext
NODIS	M	GABBRI	Calculate discharges by node from well data	Ext
OUTFIL	S	GABBRI	Write specified data array on file	Ext
OUTHYD	S	CALSYS	Write block of hydraulic parameters on file	3.3
OUTPOT	S	GABBRI	Write block of potentials on file	Ext
PERMB	M	GABBRI	Assign original permeability values	Tem
PLAYMO	S	RUNMOD	Decode manipulation file	Ext
POTBRI	M	GABBRI	Rewrite initial potentials file	Tem
POTINT	M	GABBRI	Calculates potentials by node from well data	Ext
POTREC	M	GABBRI	Extrapolates recorded data and geometry file	Ext
RANGE	S	RUNMOD	Determine indices of boundaries	Ext
REDGEO	S	RUNMOD	Read geometry code file	Ext
REDHYD	S	RUNMOD	Read hydraulic parameter file	Ext
REDPOT	S	RUNMOD	Read potentials file	Ext
REXBAL	S	CALSYS	Adjust transmissivity through inversion	3.5.2
RUNSTE	M	RUNMOD	Steady state version of model	Ext
SBOUND	S	CALSYS	Calculate flow across boundaries	3.4.2
SERDAT	S	OUTSYS	Search file for specified data	Ext
SLEAKA	S	CALSYS	Calculate leakage between aquifers	3.4.2
SMAMAP	S	OUTSYS	Print symbolic map of data array	Ext
STOCAL	M	CALSYS	Scale parameters and calculate storativities	3.4.1

Name	Type	System	Use	Ref.
THIBRI	M	GABBRI	Write aquifer thickness data file	Tem
VARGEO	M	GABBRI	Modify geometry file	Ext
VARHYD	M	GABBRI	Modify hydraulic parameter file	Ext
VARPOT	M	GABBRI	Modify potentials file	Ext
VERCAL	S	CALSYS	Calculate balances and optimal vertical flows	3.6.2
VERTAD	M	CALSYS	Modify vertical leakage factors	3.6.1

Type: M= Main program S = subroutine F = function

Ref: 3.4.2: described in 3.4.2 of this record

Ext : described in other record

Tem : temporary program, handwritten notes only

APPENDIX B: NOTES ON THE CALIBRATION BY PARAMETER ELIMINATION

Calibration by parameter elimination is based on the principle that if there are M unknown parameters and an equal number of independent equations about them, then the system of equations can be solved to determine each of the parameters uniquely. Applied to the determination of hydraulic parameters it is assumed that sets of observations of state variables provide such independent equations, and if there is more observations than there is unknown parameters then it is assumed that an optimisation technique can be employed to find parameters, which are more accurate than if only M equations had been used.

The critical assumption is that of independence of the observations. The number of observations which may be made of state variables is infinite and so is the number of equations about the parameters. This applies even to a model described by a finite number of parameters. In this case if all equations were to be considered truly independent then the system were overdetermined. Hence not all equations can be independent. So which equations may be considered independent, or in other words where and when should observations be taken to properly define all parameters?

The problem can be illustrated in a one-dimensional form by considering a flood hydrograph. To properly define the hydrograph, one must have observations from all its independent sections. That may necessitate observations from the pre-rise recession, the initial rise, the flood peak, flood recession, interflow recession, baseflow recession, and possibly more. If observations are available from only one section, e.g. the interflow section, then even an infinite number of such observations will define only that one section, leaving the other sections undefined. To define a system's parameters from observations of state variables alone it is necessary to know all independent states the system is capable of assuming and to obtain representative observations of each. The mere number of observations is meaningless as a criterion for the definition of the system.

If applied to an arbitrary system S_1, described by a set of parameters $\{P_1\}$ and capable of assuming M independent states (conditions) C_{11}, C_{12}, ... , C_{1M} we may say that none of these states can be derived from the others without knowing the parameter set $\{P_1\}$ (definition of independence).

Assume that we have observations of the first K states only: $C_{11}, C_{12}, \ldots, C_{1K}$. We may then construct an infinite number of other systems S_x described by sets $\}P_x\{$, which share with S_1 the first K states but differ on the remaining M-K states.

param. set	system	shared states	differing states
$\}P_1\{$	S_1	$C_{11}, C_{12}, \ldots, C_{1K},$	$C_{1K+1}, \ldots C_{1M}$
$\}P_2\{$	S_2	$C_{11}, C_{12}, \ldots, C_{1K},$	$C_{2K+1}, \ldots C_{2M}$
.	.		
.	.		
$\}P_x\{$	S_x	$C_{11}, C_{12}, \ldots, C_{1K},$	$C_{xK+1}, \ldots C_{xM}$
.	.		
.	.		

Only if all M states are known is any of the systems defined uniquely from observations of its state variables. Anything less than a unique identification may result in a chance selection of another system arbitrarily different in the M-K remaining states.

Hyraulic systems of course are not arbitrary and their different states are rarely entirely independent. For this reason the identification of parameters by less than complete observations of all representative states will not be as inaccurate as it would be for an arbitrary system. But by the same logic the model verification becomes less reliable too. It is normally assumed that if some observations of the state variables are not used during a model calibration but subsequently reproduced by the model that this verifies the model. This would be true for an arbitrary system consisting only of independent states. If however the observations reproduced by the model are not independent from the data used for calibration, e.g. are on the same recession limb of a hydrograph, then such a verification is meaningless.

In summary it can be said that a proper calibration by parameter elimination requires many more data than is apparent on first glance, more than is required for other calibration methods. In fact such a complete knowledge of the system to be modelled is required that it becomes doubtful whether the model could add any new information to this knowledge. If on the other hand those stringent data requirements are not met then the accuracy of the model becomes unpredictable.

LIST OF VARIABLES

TBAL(*)	overall water balance
TDRA(*)	overall effective drawdown
TSPE(*)	overall specific storage coefficient
BAL(*,*)	time step water balance
DRA(*,*)	time step effective drawdown
SPE(*,*)	time step specific storage coefficient
NYST	time interval start
NYEN	time interval end
NSTP	time step
NA	aquifer number
FBA	operating code
PCT	percent of well discharge to be provided by natural inflow
NLT,TL	number of time step
FAC	scale factor for hydraulic parameters
IA,IB	range of aquifer numbers
HY(*,*,*)	hydraulic parameters (excluding storage coefficient)
SC(*,*,*)	storage coefficients
DQ(*,*,*)	aquifer thickness

SUBROUTINES

REDGEO
REDHYD
SLEAKA
SBOUND
STOTAQ
EFDRAW
OUTHYD

Fig. 17 FLOWCHART OF PROGRAM STOCAL

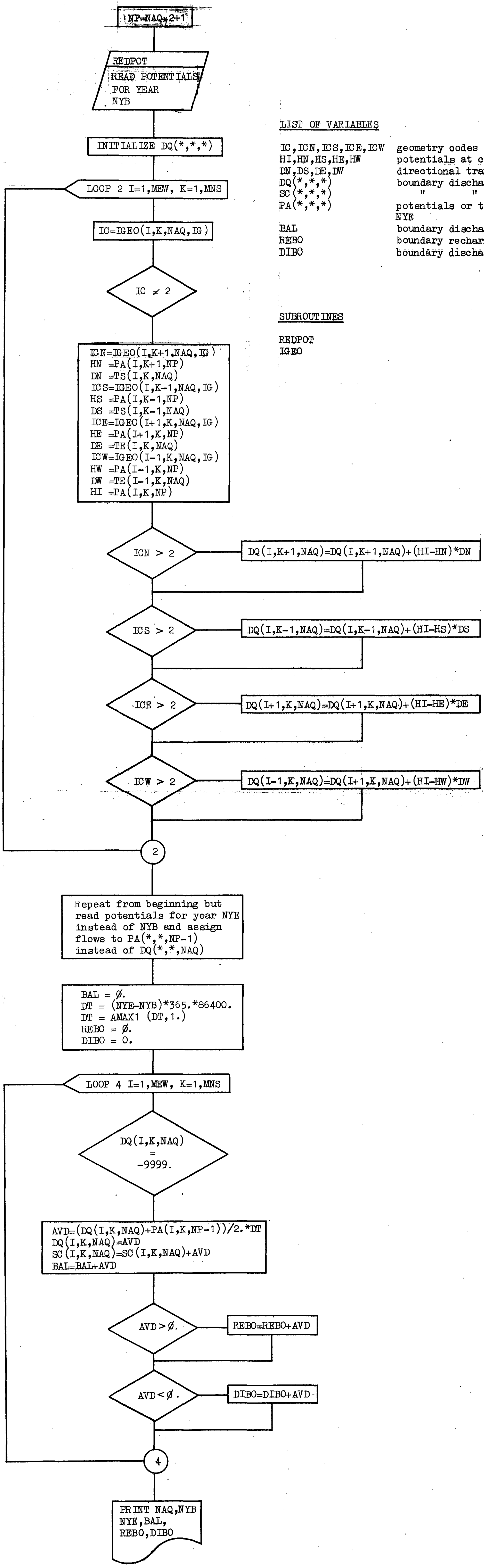
LIST OF VARIABLES

DQ(*,*,*)	upwards leakage flows
SC(*,*,*)	downwards or difference between upwards and downwards leakage flows
PA(*,*,*)	potentials or downwards leakage flows
UPBA	total upwards leakage
DOBA	total downwards leakage
BAL	balance of leakage flows

SUBROUTINES

REDPOT

Fig. 18a FLOWCHART OF SUBROUTINE SLEAKA



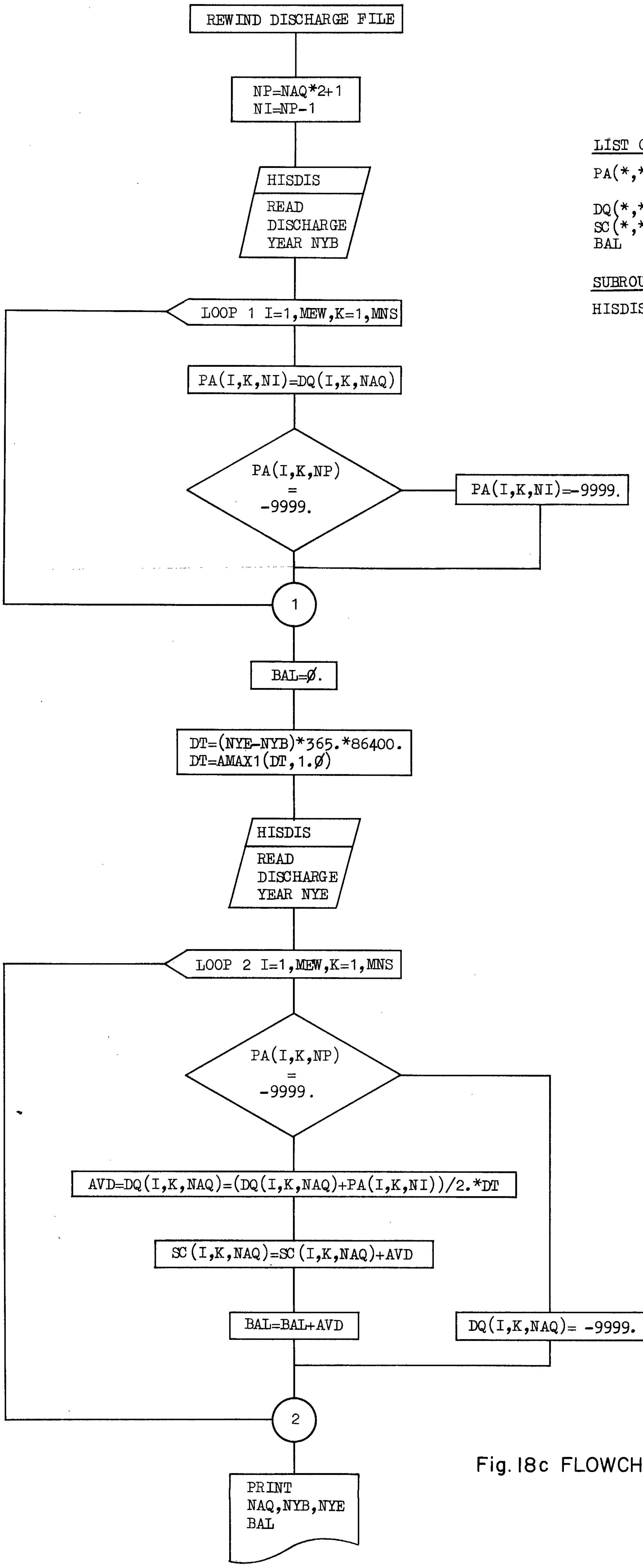
LIST OF VARIABLES

IC,ICN,ICS,ICE,ICW	geometry codes at centre, north, south, east, west
HI,HN,HS,HE,HW	potentials at centre, north, south, east, west
DN,DS,DE,DW	directional transmissivities north, south, east, west
DQ(*,*,*)	boundary discharges for year NYB or average
SC(*,*,*)	" " cummulative
PA(*,*,*)	potentials or temporarily boundary discharges for year NYE
BAL	boundary discharge balance
REBO	boundary recharge
DIBO	boundary discharge

SUBROUTINES

REDPOT
IGEO

Fig. 18b FLOWCHART OF SUBROUTINE SBOUND



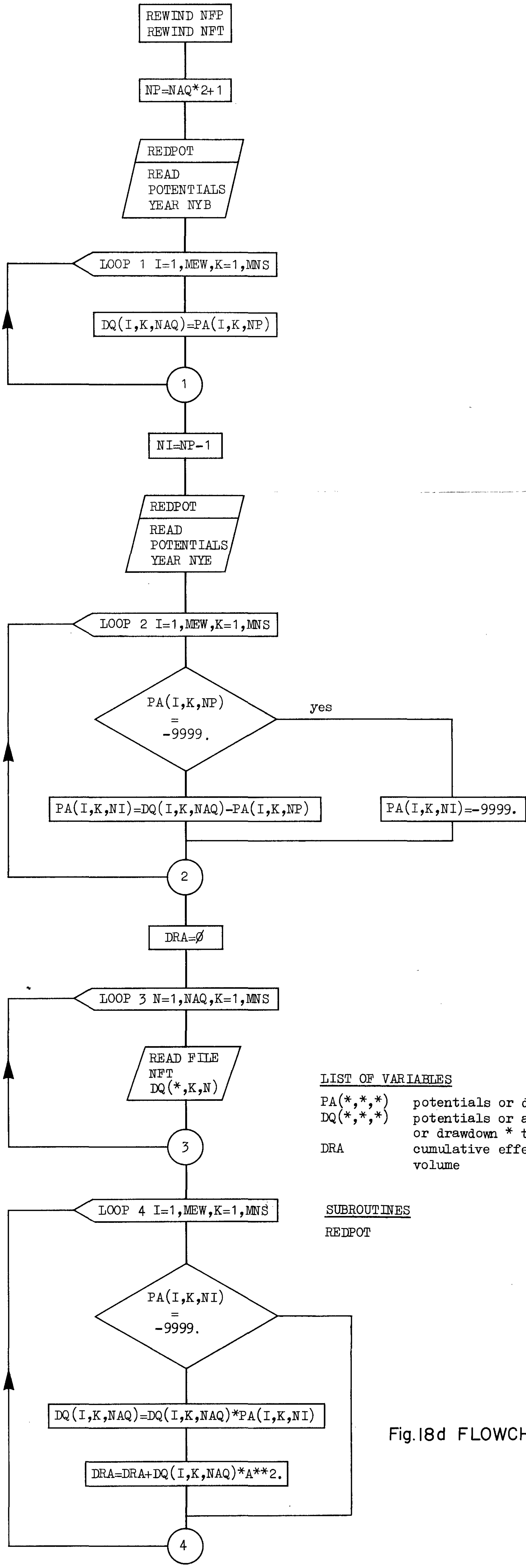
LIST OF VARIABLES

PA(*,*,*)	potentials or discharges for year NYB
DQ(*,*,*)	discharges for year NYE or averages
SC(*,*,*)	cumulative discharges
BAL	total discharge

SUBROUTINES

HISDIS

Fig. 18c FLOWCHART OF SUBROUTINE TOTAQ



<u>LIST OF VARIABLES</u>

PA(*,*,*)	potentials or drawdowns
DQ(*,*,*)	potentials or aquifer thickness or drawdown * thickness
DRA	cumulative effective drawdown volume

<u>SUBROUTINES</u>

REDPOT

Fig. 18d FLOWCHART OF SUBROUTINE EFDRA

LIST OF VARIABLES

AR(*,*)	groundelevation corrections or recorded potentials
PA(*,*,*)	current potentials or ground-elevation corrections
IYST	year, start of time interval
IYEN	year, end of time interval
NA	number of aquifer
NBN	boundary treatment code
MIT	maximum number of iterations parameter loop
CONV	convergence criterion parameter loop (metres)
ORF	over compensation factor
VERSC	vertical scaling factor
BIAS	overall balance bias (m^3/s)
RCF	recorded potentials control factor
MAXI	maximum number of iterations potentials loop
CON	convergence criterion potentials loop
SIG	boundary hydraulic gradient
SBAL	sum of boundary hydraulic gradients
SBC(*)	old water balance primary node
SBN(*)	new " " " "
SBS(*)	old " " secondary node
SBT(*)	new " " " "
TC(*)	old connecting transmissivity
TN(*)	new connecting transmissivity
TBAL	sum of residual water balances
SBALQ	sum of residual water balances squared
ERAQ	standard balance error around average balance
ERNO	standard balance error around zero
AVER	boundary balance bias

SUBROUTINES

REDGEO
REDPOT
REDHYD
SERDAT
ERANG
HISDIS
REXBAL
CALMOD
OUTHYD
OUTPOT

Fig.19 FLOWCHART OF PROGRAM COMCAL

LIST OF VARIABLES

PA(*,*,*)	current potentials or groundelevation corrections
AR(*,*)	recorded potentials
IGP(*)	geometry code split into individual codes for each aquifer
STP	time step (seconds)
DC	potential gradient between secondary and primary node
NPD	number of aquifer below (down) at primary node
NPU	number of aquifer above (up) at primary node
NSD,NSU	as above but for secondary node
FN,FS,FE,FW,FU,FD,FQ	flow balance components
TARGET	potential to be approximated
SBC,SBN,	current and new water balance for primary node
SBS,SBT	current and new water balance for secondary node
TC,TN	current and new transmissivity connecting primary and secondary node
SIG	potential gradient at permeable boundary
AVER	boundary balance bias
ORF	overcompensation factor
FC,FN	current and new flow connecting primary and secondary node
COR	ratio of new to old flow
LFL	boundary treatment code
RCF	potentials balance bias factor

Fig. 20 FLOWCHART OF SUBROUTINE REXBAL

LIST OF VARIABLES

IYST,IYEN	year numbers start and end
NA	aquifer number
MODE	operating mode
OAVP	sum of or average vertical permeability old
RAVP	" " " " " " new
NAVP	counter
AR(*,*)	residual water balances
OLB	old residual balance
WEB	new residual balance
SIGO	old standard error of water balances
SIGN	new standard error of water balances

SUBROUTINES

REDGEO
ERANG
REDPOT
REDHYD
HISDIS
SERDAT
VERCAL
SMAMAP
OUTHYD

Fig. 21 FLOWCHART OF PROGRAM VERTAD

LIST OF VARIABLES

STP time step (seconds)
OLB old residual water balance
WEB new " " "
IN,KN,NN node indices
NPD aquifer connected down direction
NPU aquifer connected up direction
IGA(*) geometry code split into aquifers
FN,FS,FE,FW,FU,FD directional flows
ST charge in storage component
DC potential gradient upward leakage
FC old vertical leakage (up)
FN new " "
TC old vertical leakage factor
TN new " " "
MODE operating mode

Fig.22 FLOWCHART OF SUBROUTINE VERCAL

ALPHABETICAL INDEX OF LIBRARY GABLIB
(PAGE REFERENCES INDEX AT END OF FICHE)
ENTRIES 53

NAME	SIZE	T	CREATION	
ASSIGN	1	D	791114	1101
BOUNDQ	1	D	791114	1106
CALMOD	3	D	791115	1105
COMCAL	4	D	791114	1058
DISCOP	1	D	791114	1126
DISRED	1	D	791115	1124
EFDRAW	1	D	791114	1109
ERANG	1	D	791114	1134
GELEV	1	D	791115	1126
GRICNO	2	D	791115	1127
HISDIS	1	D	791114	1125
HISMOQ	2	D	791114	1128
HISPOT	1	D	791114	1123
IGEO	1	D	791114	1113
LARMAP	2	D	791114	1104
LEAKAG	1	D	791114	1107
MANCHE	1	D	791114	1120
MANMOD	10	D	791114	1055
MANWRI	1	D	791115	1123
NODBAL	3	D	791115	1104
NODIS	4	D	791115	1117
OUTFIL	1	D	791115	1125
OUTHYD	1	D	791114	1100
OUTPOT	1	D	791114	1127
PLAYCO	2	D	791115	1122
PLAYMO	2	D	791114	1124
PROCCO	2	D	791109	1126
QUICK	3	D	791109	1130
RANGE	1	D	791114	1133
REDGEO	1	D	791114	1131
REDHYD	1	D	791114	1130
REDPOT	1	D	791114	1122
REXBAL	4	D	791115	1109
RUNMOD	2	D	791114	1129
RUNSTE	1	D	791114	1120
SBOUND	2	D	791109	1133
SERDAT	2	D	791114	1105
SIMPRI	1	D	791109	1136
SLEAKA	2	D	791114	1108
SMAMAP	2	D	791109	1138
STEADY	2	D	791114	1129
STOCAL	4	D	791115	1057
STOTAQ	1	D	791114	1111
TABLE	1	D	791109	1129
THREED	1	D	791109	1137
TIMESE	1	D	791114	1103
TOTALQ	2	D	791109	1132
TRANSI	2	D	791114	1132
TWOFUN	3	D	791109	1134
VARGEO	2	D	791115	1118
VARHYD	2	D	791115	1120
VERCAL	2	D	791115	1102
VERTAD	3	D	791115	1101

```
      PROGRAM PROCCO(INPUT,OUTPUT,TEN,TWE,THI,FOU,FIF,TAPE60=INPUT,
     1TAPE61=OUTPUT,TAPE10=TEN,TAPE20=TWE,TAPE30=THI,TAPE40=FOU,
     2TAPE50=FIF,GEO,TAPE1=GEO,PLO,TAPE2=PLO)
      DIMENSION IF(4),TD(4),NY(4),NQ(4),TP(10)
      COMMON/LEV/PA(67,58,15),IG(67,58)
      COMMON Z,MNS,MEW,MAQ
      LEVEL 2,PA,IG
      MAQ=2
      MEW=67
      MNS=58
      MF=NF=61
      NCP=7
      CALL REDGEO(1)
C READ INPUT SPECIFICATIONS
    1 READ(60,90) PO
   90 FORMAT(A1)
      IF(EOF(60)) 100,2
    2 READ(60,91) IF(1),IF(3),IF(2),IF(4),IST,TD(1),TD(3),TD(2),TD(4),
     1NY(1),NY(2),NQ(1),NQ(3),NQ(2),NQ(4),LOG,OO,NCP,FC,NRW,NRE,NRS,NRN
   91 FORMAT (4I2,I2,4A1,1X,2(I3,1X),4I1,1X,A1,1X,A1,1X,I2,1X,A1,4(1X,
     1I3))
      READ(60,92) (TP(I),I=1,NCP)
   92 FORMAT(10A10)
      DO 10 I=1,4
      NY(I)=NY(I)+1000
      IF(NY(I).LT.1800) NY(I)=NY(I)+1000
   10 CONTINUE
C LIST OUTPUT SPECIFICATION
      WRITE(61,900) PO
  900 FORMAT(1H1,*SPECIFICATION*,//1X,A1)
      WRITE(61,901) IF(1),IF(3),IF(2),IF(4),IST,TD(1),TD(3),TD(2),TD(4),
     1NY(1),NY(2),NQ(1),NQ(3),NQ(2),NQ(4),LOG,OO,NCP,FC,NRW,NRE,NRS,NRN
  901 FORMAT(1X,4I2,I2,4A1,1X,2(I3,1X),4I1,1X,A1,1X,A1,1X,I2,1X,A1,
     14(1X,I3))
      WRITE(61,902) (TP(I),I=1,10)
  902 FORMAT(1X,10A10)
C SELECT OUTPUT ROUTINE
      IF(PO.EQ.1HO) CALL SIMPRI(IF,NY,NQ,TD,OO,LOG,TP,NCP,MF,NF)
      IF(PO.EQ.1HF) CALL TWOFUN(IF,NY,NQ,TD,FC,OO,LOG,MF,NF,TP,NCP)
      IF(PO.EQ.1HL) CALL LEAKAG(IF,NY,NQ,FC,OO,LOG,MF,NF,TP,NCP)
      IF(PO.EQ.1HB) CALL BOUNDQ(IF,NY,NQ,OO,LOG,MF,NF,TP,NCP)
      IF(PO.EQ.1HS) CALL TOTALQ(IF,IST,NY,NQ,OO,LOG,TP,NCP,FC,NRW,NRE,
     1NRS,NRN,MF,NF)
      IF(PO.EQ.1HT) CALL TIMESE(IF,TD,IST,NY,NQ,OO,TP,NCP,FC,NRW,NRE,
     1NRS,NRN,MF,NF)
      GO TO 1
  100 CONTINUE
      END
```

```
      SUBROUTINE TABLE(NF,TD,NY,NQ,TP,FC,VA,VB,VC,VD,VE,VF,NST)
      DIMENSION NF(4),TD(4),NY(2),NQ(4),TP(10),VA(100),VB(100),
     1VC(100),VD(100),VE(100),VF(100)
      WRITE(61,900) (TP(N),N=1,9)
  900 FORMAT(1H1,9A10,/1X,90(1H-))
      WRITE(61,901) (NF(N),N=1,4)
  901 FORMAT(1H0,*FILE NUMBER*,4(8X,I2))
      WRITE(61,902) (TD(N),N=1,4)
  902 FORMAT(1X,*DATA TYPE  *,4(9X,A1))
      WRITE(61,903) (NQ(N),N=1,4)
  903 FORMAT(1X,*AQUIFER NO.*,4(8X,I2),* COL1-COL2*,* COL3-COL4*,
     1/1X,*YEAR*)
      MA=1
      MZ=NY(2)-NY(1)+1
      DO 1 M=MA,MZ,NST
      MY=1000+M+NY(1)-1
      WRITE(61,904) MY,VA(M),VB(M),VC(M),VD(M),VE(M),VF(M)
  904 FORMAT(1X,I4,7X,6F10.3)
    1 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE QUICK(NF,TD,NY,NO,TP,FC,VA,VB,VC,VD,VE,VF,NST)
      DIMENSION NF(4),TD(4),NY(2),NO(4),TP(10),VA(100),VB(100),
     1VC(100),VD(100),VE(100),VF(100)
      COMMON/LEV/CH(100,50)
      LEVEL 2,CH
C ESTABLISH RANGE OF VARIABLES
      RAMIN=1000000.
      RAMAX=-1000000.
      MA=1
      MZ=NY(2)-NY(1)+1
      DO 1 N=MA,MZ,NST
      IF(FC.EQ.1H1.OR.FC.EQ.1H3) GO TO 3
      IF(FC.EQ.1H7) GO TO 2
      RAMIN=AMIN1(RAMIN,VA(N))
      RAMAX=AMAX1(RAMAX,VA(N))
      RAMIN=AMIN1(RAMIN,VB(N))
      RAMAX=AMAX1(RAMAX,VB(N))
    2 IF(FC.EQ.1H5) GO TO 3
      RAMIN=AMIN1(RAMIN,VC(N))
      RAMAX=AMAX1(RAMAX,VC(N))
      RAMIN=AMIN1(RAMIN,VD(N))
      RAMAX=AMAX1(RAMAX,VD(N))
    3 IF(FC.EQ.1H0) GO TO 1
      IF(FC.EQ.1H1.OR.FC.EQ.1H5) GO TO 4
      RAMIN=AMIN1(RAMIN,VF(N))
      RAMAX=AMAX1(RAMAX,VF(N))
    4 IF(FC.EQ.1H3.OR.FC.EQ.1H7) GO TO 1
      RAMIN=AMIN1(RAMIN,VE(N))
      RAMAX=AMAX1(RAMAX,VE(N))
    1 CONTINUE
C INITIALIZE PLOT ARRAY
      DO 10 I=1,100
      DO 10 K=1,50
      CH(I,K)=1H
   10 CONTINUE
C SET VERTICAL SCALE
      RANGE=ABS(RAMAX-RAMIN)
      RAF=RANGE/52
      ENCODE(10,90,DUM) RAF
   90 FORMAT(E10.3)
      DECODE(10,91,DUM) MAN,NEX
   91 FORMAT(3X,I3,1X,I3)
      IF(MAN.GT.100) SCF=200.
      IF(MAN.GT.200) SCF=250.
      IF(MAN.GT.250) SCF=400.
      IF(MAN.GT.400) SCF=500.
      IF(MAN.GT.500) SCF=1000.
      SCF=(SCF/1000.)*10.0**NEX
      RAMIN=RAMIN-SCF
      NRAM=RAMIN/SCF
      IF(NRAM.LE.0) NRAM=NRAM-1
      RAMIN=NRAM*SCF
C SET HORIZONTAL SPACER
      LHOR=MZ-MA
      NS=8
      IF(LHOR.LE.40)NS=4
      IF(LHOR.LE.20) NS=2
```

```
      IF(LHOR.LE.10) NS=1
      MS=8/NS
C SET PLOT ARRAY
      DO 100 N=MA,MZ,NST
      NN=N*MS-MS+1
      IF(FC.EQ.1H1.OR.FC.EQ.1H3) GO TO 23
      IF(FC.EQ.1H7) GO TO 22
      CALL ASSIGN(VA(N),1HA,SCF,RAMIN,NN)
      CALL ASSIGN(VB(N),1HB,SCF,RAMIN,NN)
   22 IF(FC.EQ.1H5) GO TO 23
      CALL ASSIGN(VC(N),1HC,SCF,RAMIN,NN)
      CALL ASSIGN(VD(N),1HD,SCF,RAMIN,NN)
   23 IF(FC.EQ.1H0) GO TO 100
      IF(FC.EQ.1H1.OR.FC.EQ.1H5) GO TO 24
      CALL ASSIGN(VF(N),1HI,SCF,RAMIN,NN)
   24 IF(FC.EQ.1H3.OR.FC.EQ.1H7) GO TO 100
      CALL ASSIGN(VE(N),1HO,SCF,RAM'N,NN)
  100 CONTINUE
C PRINT PLOT
      WRITE(61,900) (TP(N),N=1,9)
  900 FORMAT(1H1,9A10,/1X,100(1H-))
      WRITE(61,901) (NF(N),TD(N),NO(N),N=1,4)
  901 FORMAT(1H0,*CHARACTER A*,* FILE*,I3,* DATA*,1X,A1,* AQUIFER*,
     1I2,/1X,*CHARACTER B*,* FILE*,I3,* DATA*,1X,A1,* AQUIFER*,I2,
     2/1X,*CHARACTER C*,* FILE*,I3,* DATA*,1X,A1,* AQUIFER*,I2,
     3/1X,*CHARACTER D*,* FILE*,I3,* DATA*,1X,A1,* AQUIFER*,I2,
     410X,*CHARACTER O=A-B*,* CHARACTER I=C-D*)
      NA=NY(1)
      NZ=NY(2)
      WRITE(61,902) (NN,NN=NA,NZ,NS)
  902 FORMAT(1H0,11X,12(1H1,I3,4X),/1X,9X,12(3X,1HI,4X))
      DO 200 K=1,50
      KK=50-K+1
      VAL=KK*SCF+RAMIN
      WRITE(61,903) (VAL,(CH(I,KK),I=1,100))
  903 FORMAT(1X,F8.3,4X,100A1)
  200 CONTINUE
      WRITE(61,904) (NN,NN=NA,NZ,NS)
  904 FORMAT(1X,9X,12(3X,1HI,4X),/1X,11X,12(1H1,I3,4X))
      RETURN
      END
```

```
      SUBROUTINE TOTALQ(IF,NST,NY,NQ,OO,LOG,TP,NCP,FC,NRW,NRE,NRS,NRN,
     1MF,NF)
      DIMENSION IF(4),NY(4),NQ(4),TP(10)
      COMMON/LEV/PA(67,58,5),A(67,58),B(67,58),C(67,58),X(67,58,5),
     1DQ(67,58,2),IG(67,58)
      COMMON Z,MNS,MEW,MAQ
      LEVEL 2,PA,A,B,C,X,DQ,IG
      SUM=0.
      DT=NST*365.*86400.
      DT=AMAX1(DT,1.0)
      NA=NQ(1)
      NFM=IF(1)
      NFD=IF(2)
C INITIALIZE
      DO 10 I=1,MEW
      DO 10 K=1,MNS
      A(I,K)=B(I,K)=C(I,K)=0.
   10 CONTINUE
      NYE=NY(1)
    1 NYS=NYE
      NYE=NYS+NST
C TIME LOOP
      IF(NYE.GT.NY(2)) GO TO 100
      IF(NFM.EQ.0) GO TO 2
      REWIND NFM
      READ(NFM,90) XX
      READ(NFM,90) XX
   90 FORMAT(A1)
      CALL PLAYMO(NYS,IY,NFM)
      DO 3 I=1,MEW
      DO 3 K=1,MNS
      A(I,K)=DQ(I,K,NA)
    3 CONTINUE
      CALL PLAYMO(NYE,IY,NFM)
      DO 4 I=1,MEW
      DO 4 K=1,MNS
      B(I,K)=DQ(I,K,NA)
    4 CONTINUE
      GO TO 5
    2 REWIND NFD
      CALL SERDAT(MNS,MEW,NFD,NYS,NA,1HQ,A,MF)
      CALL SERDAT(MNS,MEW,NFD,NYE,NA,1HQ,B,MF)
    5 DO 6 I=1,MEW
      DO 6 K=1,MNS
      IF(A(I,K).EQ.-9999) GO TO 8
      IF(FC.NE.1HR) GO TO 7
      IF(I.LT.NRW.OR.I.GT.NRE) GO TO 6
      IF(K.LT.NRS.OR.K.GT.NRN) GO TO 6
    7 BAL=(A(I,K)+B(I,K))/2.*DT
      C(I,K)=C(I,K)+BAL
      SUM=SUM+BAL
    8 IF(A(I,K).EQ.-9999.) C(I,K)=-9999.
    6 CONTINUE
      IF(NYE.GE.NY(2)) GO TO 100
      GO TO 1
  100 IF(OO.NE.1HS) CALL LARMAP(MEW,MNS,NF,C,TP,NCP)
      IF(OO.NE.1HL) CALL SMAMAP(MEW,MNS,NF,C,LOG,TP,NCP)
```

```
      WRITE(NF,900) NY(1),NY(2),NRW,NRE,NRS,NRN,SUM
  900 FORMAT(1H0,*FOR PERIOD 1*,I3,* TO 1*,I3,* IN AREA BOUNDED W*,I3,
     1*, E*,I3,*, S*,I3,*, N*,I3,/1X,*TOTAL ARTIFICIAL DISCHARGE:*,
     2E13.6,* CUBIC METERS*)
      RETURN
      END
```

```
      SUBROUTINE SBOUND(NAQ,NYB,NYE,NFP,BAL)
      COMMON/LEV/PA(67,58,5),VP(67,58,2),TE(67,58,2),TS(67,58,2),SC(67,
     158,2),DQ(67,58,2),IG(67,58)
      COMMON A,MNS,MEW,MAQ
      LEVEL 2,PA,VP,TE,TS,SC,DQ,IG
      CALL REDPOT(NFP,NYB)
C INITIZLIZE DQ ARRAY
      NP=NAQ*2+1
      DO 1 I=1,MEW
      DO 1 K=1,MNS
      DQ(I,K,NAQ)=0.
      IF(PA(I,K,NP).EQ.-9999.) DQ(I,K,NAQ)=-9999.
    1 CONTINUE
C CALCULATE BOUNDARY DISCHARGE RATE YEAR NYB
      DO 2 I=1,MEW
      DO 2 K=1,MNS
      IC=IGEO(I,K,NAQ,IG)
      IF(IC.NE.2) GO TO 2
      ICN=IGEO(I,K+1,NAQ,IG)
      HN=PA(I,K+1,NP)
      DN=TS(I,K,NAQ)
      ICS=IGEO(I,K-1,NAQ,IG)
      HS=PA(I,K-1,NP)
      DS=TS(I,K-1,NAQ)
      ICE=IGEO(I+1,K,NAQ,IG)
      HE=PA(I+1,K,NP)
      DE=TE(I,K,NAQ)
      ICW=IGEO(I-1,K,NAQ,IG)
      HW=PA(I-1,K,NP)
      DW=TE(I-1,K,NAQ)
      HI=PA(I,K,NP)
      IF(ICN.GT.2) DQ(I,K+1,NAQ)=DQ(I,K+1,NAQ)+(HN-HI)*DN*(-1)
      IF(ICS.GT.2) DQ(I,K-1,NAQ)=DQ(I,K-1,NAQ)+(HS-HI)*DS*(-1)
      IF(ICE.GT.2) DQ(I+1,K,NAQ)=DQ(I+1,K,NAQ)+(HE-HI)*DE*(-1)
      IF(ICW.GT.2) DQ(I-1,K,NAQ)=DQ(I-1,K,NAQ)+(HW-HI)*DW*(-1)
    2 CONTINUE
C CALCULATE BOUNDARY DISCHARGE YEAR NYE
      BAL=0.
      DT=(NYE-NYB)*365.*86400.
      DT=AMAX1(DT,1.0)
      REBO=0.
      DIBO=0.
      CALL REDPOT(NFP,NYE)
      NI=NP-1
      DO 9 I=1,MEW
      DO 9 K=1,MNS
      IF(PA(I,K,NP).EQ.-9999.) GO TO 9
      PA(I,K,NI)=0.
    9 CONTINUE
      DO 3 I=1,MEW
      DO 3 K=1,MNS
      IC=IGEO(I,K,NAQ,IG)
      IF(IC.NE.2) GO TO 3
      ICN=IGEO(I,K+1,NAQ,IG)
      HN=PA(I,K+1,NP)
      DN=TS(I,K,NAQ)
      ICS=IGEO(I,K-1,NAQ,IG)
```

```
      HS=PA(I,K-1,NP)
      DS=TS(I,K-1,NAQ)
      ICE=IGEO(I+1,K,NAQ,IG)
      HE=PA(I+1,K,NP)
      DE=TE(I,K,NAQ)
      ICW=IGEO(I-1,K,NAQ,IG)
      HW=PA(I-1,K,NP)
      DW=TE(I-1,K,NAQ)
      HI=PA(I,K,NP)
      IF(ICN.GT.2) PA(I,K+1,NI)=PA(I,K+1,NI)+(HN-HI)*DN*(-1)
      IF(ICS.GT.2) PA(I,K-1,NI)=PA(I,K-1,NI)+(HS-HI)*DS*(-1)
      IF(ICE.GT.2) PA(I+1,K,NI)=PA(I+1,K,NI)+(HE-HI)*DE*(-1)
      IF(ICW.GT.2) PA(I-1,K,NI)=PA(I-1,K,NI)+(HW-HI)*DW*(-1)
    3 CONTINUE
C AVERAGE DISCHARGE RATES, CONVERT TO VOLUMES,TOTAL
      DO 4 I=1,MEW
      DO 4 K=1,MNS
      IF(DQ(I,K,NAQ).EQ.-9999.) GO TO 4
      AVD=(DQ(I,K,NAQ)+PA(I,K,NI))/2.*DT
      DQ(I,K,NAQ)=AVD
      SC(I,K,NAQ)=SC(I,K,NAQ)+AVD
      BAL=BAL+AVD
      IF(AVD.GT.0.) REBO=REBO+AVD
      IF(AVD.LT.0.) DIBO=DIBO+AVD
    4 CONTINUE
      WRITE(61,900) NAQ,NYB,NYE,BAL
  900 FORMAT(1H1,*BOUNDARY DISCHARGE TOTAL FOR AQUIFER*,I3,* PERIOD 1*,
     1I3,* TO 1*,I3,*  *,E12.6,* CUBIC METERS*)
      WRITE(61,901) REBO,DIBO
  901 FORMAT(1H0,*AS*,/1X,*RECHARGE  *,E12.6,/1X,*DISCHARGE *,E12.6)
      RETURN
      END
```

```
      SUBROUTINE TWOFUN(IF,NY,NQ,TD,FC,OO,LOG,MF,NF,TP,NCP)
      DIMENSION IF(4),NY(4),NQ(4),TD(4),TP(10)
      COMMON/LEV/A(67,58),B(67,58),C(67,58)
      COMMON Z,MNS,MEW,MAQ
      LEVEL 2,A,B,C
      DO 13 I=1,MEW
      DO 13 k=1,MNS
      C(I,K)=-9999.
   13 CONTINUE
C OBTAIN DATA
      CALL SERDAT(MNS,MEW,IF(1),NY(1),NQ(1),TD(1),A,MF)
      CALL SERDAT(MNS,MEW,IF(2),NY(2),NQ(2),TD(2),B,MF)
      IF(TD(1).EQ.1H0.OR.TD(2).EQ.1H0) RETURN
C DETERME REQUESTED FUNCTION TYPE
      IF(FC.EQ.1H+) GO TO 1
      IF(FC.EQ.1H-) GO TO 3
      IF(FC.EQ.1HX.OR.FC.EQ.1H*) GO TO 5
      IF(FC.EQ.1H/) GO TO 7
      IF(FC.EQ.1HA) GO TO 9
      IF(FC.EQ.1HH) GO TO 11
      IF(FC.EQ.1HP.OR.FC.EQ.1HE) GO TO 16
      WRITE(MF,900) FC
  900 FORMAT(1X,17H***FUNCTION CODE ,A1,* NOT VALID*)
      RETURN
    1 DO 2 K=1,MNS
      DO 2 I=1,MEW
      C(I,K)=A(I,K)+B(I,K)
    2 CONTINUE
      GO TO 20
    3 DO 4 K=1,MNS
      DO 4 I=1,MEW
      C(I,K)=A(I,K)-B(I,K)
    4 CONTINUE
      GO TO 20
    5 DO 6 K=1,MNS
      DO 6 I=1,MEW
      C(I,K)=A(I,K)*B(I,K)
    6 CONTINUE
      GO TO 20
    7 DO 8 K=1,MNS
      DO 8 I=1,MEW
      IF(B(I,K).EQ.0) GO TO 8
      C(I,K)=A(I,K)/B(I,K)
    8 CONTINUE
      GO TO 20
    9 DO 10 K=1,MNS
      DO 10 I=1,MEW
      C(I,K)=(A(I,K)+B(I,K))/2.
   10 CONTINUE
      GO TO 20
   11 DO 12 K=1,MNS
      DO 12 I=1,MEW
      IF(A(I,K)+B(I,K).EQ.0) GO TO 12
      C(I,K)=2*A(I,K)*B(I,K)/(A(I,K)+B(I,K))
   12 CONTINUE
      GO TO 20
C ERROR ANALYSIS FUNCTION APPLIED TO NON-ZERO VALUES
```

```
   16 ERMAX=ERSUM=ERSQU=0.
      NSUM=0
      DO 17 I=1,MEW
      DO 17 K=1,MNS
      IF(A(I,K).EQ.0..AND.B(I,K).EQ.0.) GO TO 17
      IF(A(I,K).EQ.-9999..OR.B(I,K).EQ.-9999.) GO TO 17
      C(I,K)=ABS(A(I,K)-B(I,K))
      IF(FC.EQ.1HE) GO TO 18
      DIVA=ABS(A(I,K))
      DIVB=ABS(B(I,K))
      IF(DIVA.GE.DIVB) C(I,K)=C(I,K)/DIVA*100.
      IF(DIVB.GT.DIVA) C(I,K)=C(I,K)/DIVB*100.
   18 ERMAX=AMAX1(ERMAX,C(I,K))
      ERSUM=ERSUM+C(I,K)
      ERSQU=ERSQU+C(I,K)**2
      NSUM=NSUM+1
   17 CONTINUE
      AVER=ERSUM/NSUM
      STAND=SQRT(ERSQU/NSUM)
      WRITE(NF,909) NSUM,ERMAX,AVER,STAND
  909 FORMAT(1H1,*ERROR ANALYSIS*,1X,*OF*,I6,* POINTS*,/1X,
     1*MAX.ERROR:*,F10.4,/1X,*AVER.ERROR:*,F10.4,/1X,
     2*STAND.ERROR:*,F10.4)
      GO TO 20
C CALLL SPECIFIED PRINTING ROUTINE
   20 DO 15 K=1,MNS
      DO 15 I=1,MEW
      IF(A(I,K).EQ.-9999.) C(I,K)=-9999.
      IF(B(I,K).EQ.-9999.) C(I,K)=-9999.
   15 CONTINUE
      IF(OO.EQ.1HS.OR.OO.EQ.1HB) CALL SMAMAP(MEW,MNS,NF,C,LOG,TP,NCP)
      IF(OO.EQ.1HL.OR.OO.EQ.1HB) CALL LARMAP(MEW,MNS,NF,C,TP,NCP)
      IF(OO.EQ.1HD) CALL THREED(MEW,MNS,C,TP)
      RETURN
      END
```

```
      SUBROUTINE SIMPRI(IF,NY,NO,TD,OO,LOG,TP,NCP,MF,NF)
      DIMENSION TP(10),IF(4),NY(4),NO(4),TD(4)
      COMMON/LEV/A(67,58)
      COMMON Z,MNS,MEW,MAO
      LEVEL 2,A
      CALL SERDAT(MNS,MEW,IF(1),NY(1),NO(1),TD(1),A,MF)
      IF(OO.EQ.1HS.OR.OO.EQ.1HB) CALL SMAMAP(MEW,MNS,NF,A,LOG,TP,NCP)
      IF(OO.EQ.1HL.OR.OO.EQ.1HB) CALL LARMAP(MEW,MNS,NF,A,TP,NCP)
      IF(OO.EQ.1HD) CALL THREED(MEW,MNS,A,TP)
      RETURN
      END
```

```
      SUBROUTINE THREED(MEW,MNS,C,TP)
      DIMENSION C(MEW,MNS),TP(10)
      LEVEL 2,C
      COMMON A
      CALL DATE(TIC)
      TP(9)=10HPRINTED ON
      TP(10)=TIC
      WRITE(2,900) A,MNS,MEW
  900 FORMAT(E11.5,2I3)
      WRITE(2,901) (TP(I),I=1,10)
  901 FORMAT(10A10)
      DO 1 I=1,MEW
      WRITE(2,902) (C(I,K),K=1,MNS)
  902 FORMAT(10E11.5)
    1 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE SMAMAP(MEW,MNS,NF,VA,LOG,TP,NCP)
      DIMENSION VA(MEW,MNS),AL(3),ALPH(30),CL(24),COD(100),TP(10)
     1,FMA(4)
      LEVEL 2,VA
      DATA (AL(I),I=1,3)/10HABCDEFGHIK,10HLMNOPRSTUV,
     110HWXYZ       /
C ESTABLISH RANGE OF VALUES
      XMAX=10000000.*(-1)
      XMIN=10000000.
      DO 1 K=1,MNS
      DO 1 I=1,MEW
      IF(VA(I,K).EQ.-9999.) GO TO 1
      IF(LOG.EQ.1HL.AND.VA(I,K).EQ.0.) GO TO 1
      IF(VA(I,K).GT.XMAX) XMAX=VA(I,K)
      IF(VA(I,K).LT.XMIN) XMIN=VA(I,K)
    1 CONTINUE
C PRINT MAP HEADING
      MC=NCP+1
      DO 2 N=MC,8
      TP(N)=10H
    2 CONTINUE
      TP(9)=10HPRINTED ON
      CALL DATE(TIC)
      TP(10)=TIC
      WRITE(NF,90) (TP(I),I=1,10)
   90 FORMAT(1H1,10A10,/1X,10(*----------*))
C SPLIT ALPHABET
      DO 3 N=1,3
      ENCODE(10,900,AA) AL(N)
      NA=(N-1)*10+1
      NZ=NA+9
      DECODE(10,901,AA) (ALPH(K),K=NA,NZ)
  900 FORMAT(A10)
  901 FORMAT(10A1)
    3 CONTINUE
C DETERMINE CLASS INTERVALS
      IF(XMAX.EQ.XMIN) RETURN
      DEL=(XMAX-XMIN)/20.
      IF(LOG.NE.1HL) GO TO 4
      IF(XMAX.GT.0..AND.XMIN.GT.0.) GO TO 10
      WRITE(NF,910)
  910 FORMAT(1X,3H***,*LOGARITHM. CLASS. NOT POSSIBLE*)
      LOG=1HN
      GO TO 4
   10 DEL=(ALOG(XMAX)-ALOG(XMIN))/20.
      FDE=EXP(DEL)
    4 DO 5 N=1,23
      CL(N)=XMIN+(N-1)*DEL
      IF(LOG.EQ.1HL) CL(N)=XMIN*FDE**(N-1)
    5 CONTINUE
C WRITE TOP INDICES
      KI=MEW/10+1
      WRITE(NF,91) (I,I=1,KI)
   91 FORMAT(1H0,12X,I1,9(9X,I1))
      WRITE(NF,92) ((I,I=1,9),K=1,KI)
   92 FORMAT(4X,10(9I1,1X))
C ASSIGN CODES AND PRINT LINE BY LINE
```

```
      FMA(1)=10H(1X,I2,1X,
      FMA(3)=10H,* > *,E12
      FMA(4)=10H.6)
      ENCODE(10,93,FM) MEW
   93 FORMAT(I2,8HA1,9X,A1)
      DECODE(10,94,FM) FMA(2)
   94 FORMAT(A10)
      DO 6 K=1,MNS
      J=MNS-K+1
      JJ=K-(MNS-23)
      DO 7 I=1,MEW
      A=VA(I,J)
      COD(I)=1H
      DO 8 N=1,22
      IF(A.GE.CL(N).AND.A.LT.CL(N+1)) COD(I)=ALPH(N)
      IF(A.EQ.-9999.) COD(I)=1H.
      IF(LOG.EQ.1HL.AND.A.EQ.0.) COD(I)=1H*
      IF(A.EQ.0) COD(I)=1H*
    8 CONTINUE
    7 CONTINUE
      IF(JJ.GE.1) WRITE(NF,FMA) (J,(COD(I),I=1,MEW),ALPH(JJ),CL(JJ))
      IF(JJ.LT.1) WRITE(NF,FMA) (J,(COD(I),I=1,MEW))
    6 CONTINUE
      RETURN
      END
```

```
      PROGRAM MANMOD(INPUT,DUM,OUTPUT,TAPE60=INPUT,TAPE61=
     1OUTPUT,TAPE50=DUM,ARTCA1,ARTCA2,SUBCA1,POT,COL,OMAN,ODIS,MAN,DIS,
     2TAPE1=SUBCA1,TAPE2=ARTCA1,TAPE4=ARTCA2,TAPE5=POT,TAPE6=COL,
     3TAPE7=ODIS,TAPE8=OMAN,TAPE10=DIS,TAPE11=MAN)
      DIMENSION DIS(67,58,2),COR(67,58,2),COF(67,58,2),IDIS(67,58,2),
     1POT(67,58,2),MA(2),MS(2),VAR(3),XX(10)
      COMMON/LEV/AR(67,58)
      COMMON A,MNS,MEW,MAQ
      LEVEL 2,AR
      INTEGER FROM,TO,STEP,WESTFR,EASTTO,SOUTFR,NORTTO,EAST,WEST,SOUTH,
     1AQUIFR,AT,FILE
      REAL MINDIS,MAXDIS
      NAMELIST/TIME/FROM,TO,STEP,FILE/FLOW/AT,COMAX
      NAMELIST/CODE/DISCH,THRES,RISPIP,MAXDIS,MINDIS,FLOWCO,GELEV
      NAMELIST/MODEL/FROM,TO,STEP,TMF,YMIN,YMAX,MAXI,ORFI,CON,NCAR
      NAMELIST/COORDI/WESTFR,EASTTO,SOUTFR,NORTTO,EAST,WEST,NORTH,
     1SOUTH,AQUIFR,CONTI
      A=25000.
      MNS=58
      MEW=67
      MAQ=2
C SPECIFICATION OF DEFAULT VALUES
      DFROM=1880
      DTO=1881
      DSTEP=1
      TMF=1.5
      YMIN=365.
      YMAX=1.
      MAXI=20
      ORFI=1.0
      CON=0.01
      NCAR=0
      FLAGCT=0
      FLAGFR=0
C READ OPTION KEYWORD FOR CENTRAL SWITCHING
      NOPT=0
      WRITE(61,899)
  899 FORMAT(1H1,*PROGRAM MANMOD*,/1X,14(1H-))
      GO TO 5
    1 READ(60,89) KEY
   89 FORMAT(A10)
      IF(EOF(60)) 10,2
    2 NOPT=NOPT+1
      IF(KEY.EQ."MODELCON") GO TO 100
      IF(KEY.EQ."FREECOEF") GO TO 200
      IF(KEY.EQ."HISTORIC") GO TO 300
      IF(KEY.EQ."COPY OLD") GO TO 400
      IF(KEY.EQ."CONTROLLED") GO TO 500
      IF(KEY.EQ."END") GO TO 10
      WRITE(61,900) NOPT,KEY
  900 FORMAT(1H1,17H****OPTION NUMBER,I4,8H READ AS,1X,A10,* NOT *,
     1*RECOGNIZED AND IGNORED*)
      GO TO 5
  555 WRITE(61,913) NOPT
  913 FORMAT(1H0,17H****OPTION NUMBER,I4,*DISCONTINUED*)
    5 READ(60,90) XX
C SEARCH FOR NEXT KEYWORD
```

```
   90 FORMAT(10A1)
      IF(EOF(60)) 10,6
    6 DO 7 L=1,10
      IF(XX(L).EQ.1H$) GO TO 5
    7 CONTINUE
    8 BACKSPACE 60
      CALL ERRSET(KOUNT,2)
      GO TO 1
   10 CONTINUE
      CALL MANCHE(11)
C LOGICAL END OF PROGRAM
      STOP
C READ MODEL CONTROL PARAMETERS
  100 CONTINUE
      IF(FLAGCT.EQ.0) GO TO 101
      WRITE(61,901) NOPT,KEY
  901 FORMAT(1H1,16H***OPTION NUMBER,I4,1X,A10,*WAS SPECIFIED BEFORE*,
     1/1X,* NEW SPECIFICATION HAS BEEN IGNORED*)
      GO TO 5
  101 FROM=DFROM
      TO=DTO
      STEP=DSTEP
      READ(60,MODEL)
      IF(KOUNT.GT.0) GO TO 555
      WRITE(61,902) NOPT,KEY
  902 FORMAT(1H1,13HOPTION NUMBER,I4,1X,A10,* THE VALUES BELOW HAVE*,
     1* BEEN USED*)
      FLAGCT=1
  110 CONTINUE
      WRITE(61,921) FROM,TO,STEP,TMF,YMAX,YMIN,MAXI,CON,NCAR,ORFI
  921 FORMAT(1H0,28HMODEL TIME RANGE        FROM=,I5,7H  TO  =,I5,2X,
     1*STEP=*,I5,/1X,28HTIME DISCRETIZATION    TMF =,F5.2,7H  YMAX=,
     2F5.1,7H  YMIN=,F5.1,/1X,28HITERATION CONTROL      MAXI=,I5,2X,
     3*CON =*,F6.3,/1X,28HOVERRELAXATION CONTROL NCAR=,I5,7H  ORFI=,
     4F5.2)
      FROM=FROM-1000
      IF(FROM.GE.1000) FROM=FROM-1000
      TO=TO-1000
      IF(TO.GE.1000) TO=TO-1000
      WRITE(11,903) FROM,TO,STEP,MAXI,NCAR
  903 FORMAT(1X,5I3)
      WRITE(11,904) TMF,YMAX,YMIN,CON,ORFI
  904 FORMAT(3F7.2,F5.4,F4.2)
      IF(FLAGCT.EQ.1) GO TO 5
      FLAGCT=1
      GO TO 510
C DETERMINE FREE FLOW COEFFICIENTS
  200 AT=1970
      COMAX=0.1
      READ(60,FLOW)
      IF(KOUNT.GT.0) GO TO 555
  201 WRITE(61,905) NOPT,KEY,AT,COMAX
  905 FORMAT(1H1,13HOPTION NUMBER,I4,1X,A10,* COEFFICIENTS DETERMINED*,
     1* FOR YEAR*,I5,* MAX.COEF.*,F7.3)
      FLAGFR=1
  210 CONTINUE
      COMAX=COMAX*1000.
```

```
      IY=AT
      REWIND 5
      REWIND 6
      REWIND 7
      MS(1)=MA(1)=MA(2)=1
      MS(2)=0
C OBTAIN HISTORICAL DISCHARGE IN LITRES/SEC
      DO 206 N=1,2
      CALL SERDAT(MNS,MEW,7,IY,N,1HQ,AR,61)
      DO 206 I=1,MEW
      DO 206 K=1,MNS
      DIS(I,K,N)=AR(I,K)
  206 CONTINUE
C AQUIFER LOOP REMOVED (NO CA1 COR)
      N=2
C OBTAIN POTENTIALS
      CC=1HP
      CALL SERDAT(MNS,MEW,5,IY,N,CC,AR,61)
      DO 203 I=1,MEW
      DO 203 K=1,MNS
      POT(I,K,N)=AR(I,K)
  203 CONTINUE
C OBTAIN GROUND ELEVATION CORRECTIONS
      CC=1HG
      CALL SERDAT(MNS,MEW,6,IY,N,CC,AR,61)
C CALCULATE COEFFICIENTS
      DO 204 I=1,MEW
      DO 204 K=1,MNS
      COR(I,K,N)=AR(I,K)
      COF(I,K,N)=0.
      IF(POT(I,K,N).EQ.-9999.) GO TO 204
      IF(DIS(I,K,N).LE.0.) GO TO 204
      PMIN=DIS(I,K,N)/COMAX
      IF(POT(I,K,N)-COR(I,K,N).GE.PMIN) GO TO 205
      CORNEW=POT(I,K,N)-PMIN
      WRITE(61,931) I,K,COR(I,K,N),CORNEW
  931 FORMAT(1X,*NODE*,I3,1X,I3,* GROUNDELEV.CORR. REDUCED*,/1X,
     1*FROM*,F7.2,* TO*,F7.2,* METRES*)
      COR(I,K,N)=CORNEW
  205 CONTINUE
      COF(I,K,N)=0.
      PRES=POT(I,K,N)-COR(I,K,N)
      IF(PRES.GT.0.) COF(I,K,N)=DIS(I,K,N)/PRES
  204 CONTINUE
  202 CONTINUE
      IF(FLAGFR.EQ.1) GO TO 5
      FLAGFR=1
      GO TO 520
C PRODUCE HISTORIC DISCHARGE FILE
C INITIALIZE
  300 CONTINUE
      FROM=DFROM
      TO=DTO
      STEP=DSTEP
      FILE=1
      MS(1)=MA(1)=MA(2)=1
      MS(2)=0
```

```
      READ(60,TIME)
      IF(KOUNT.GT.0) GO TO 555
      WRITE(61,902) NOPT,KEY
      WRITE(61,922) FROM,TO,STEP,FILE
  922 FORMAT(1H-,24HOPTION TIME RANGE  FROM=,I5,5H  TO=,I5,7H  STEP=,
     1I4,/1X,*FILE=*,I2)
      REWIND 1
      REWIND 2
      REWIND 4
      NYB=FROM
      NYE=TO
      NF=9+FILE
      DO 301 NY=NYB,NYE,STEP
C READ HISTORIC DATA
      CALL DISRED(MAQ,MNS,MEW,DIS,NY,MS,MA)
      DO 302 N=1,MAQ
      DO 302 I=1,MEW
      DO 302 K=1,MNS
  302 CONTINUE
C OUTPUT ON STANDARD FILE
      IF(FILE.GT.1) GO TO 303
      CALL OUTFIL(NY,10,1HQ,DIS)
      GO TO 301
C OUTPUT ON MANIPULATION FILE
  303 CONTINUE
      DO 304 N=1,MAQ
      DO 304 I=1,MEW
      DO 304 K=1,MNS
      VAR(1)=DIS(I,K,N)
      VAR(2)=VAR(3)=0.
      NCC=1
      IF(VAR(1).EQ.0) NCC=0
      CALL PLAYCO(NCC,VAR(1),VAR(2),VAR(3),IDIS(I,K,N),61)
  304 CONTINUE
      CALL MANWRI(MAQ,MNS,MEW,IDIS,NY,NF)
  301 CONTINUE
      GO TO 5
C COPY FROM OLD DISCHARGE FILES
  400 CONTINUE
      IF(FLAGCT.EQ.1) GO TO 402
      WRITE(61,906) NOPT,KEY
  906 FORMAT(1H1,16H***OPTION NUMBER,I4,1X,A10,* MODEL CONTROL VALUES*,
     1* NOT SPECIFIED.*,10X,/1X,*OLD ONES COPIED*)
      READ(8,903) FROM,TO,STEP,MAXI,NCAR
      WRITE(11,903) FROM,TO,STEP,MAXI,NCAR
      READ(8,904) TMF,YMAX,YMIN,CON,ORFI
      WRITE(11,904) TMF,YMAX,YMIN,CON,ORFI
      FROM=FROM+1000
      IF(FROM.LE.1800) FROM=FROM+1000
      TO =TO+1000
      IF(TO.LE.1800) TO=TO+1000
      FLAGCT=1
      WRITE(61,921) FROM,TO,STEP,TMF,YMAX,YMIN,MAXI,CON,NCAR,ORFI
  402 FROM=DFROM
      TO=DTO
      STEP=DSTEP
      FILE=1
```

```
      READ(60,TIME)
      IF(KOUNT.GT.0) GO TO 555
      WRITE(61,902) NOPT,KEY
      WRITE(61,922) FROM,TO,STEP,FILE
      NF=FILE+6
      NYB=FROM
      NYE=TO
      DO 403 NY=NYB,NYE,STEP
      CALL DISCOP(MAQ,MNS,MEW,IDIS,NY,NF)
      CALL MANWRI(MAQ,MNS,MEW,IDIS,NY,11)
  403 CONTINUE
      GO TO 5
C SECTION FOR CONTROLLED DISCHARGE MODIFICATIONS
  500 CONTINUE
C CHECK WHETHER CONTROL AND FREEFLOW VALUES HAVE BEEN DEFINED
      IF(FLAGCT.EQ.1) GO TO 510
      WRITE(61,908) NOPT,KEY
  908 FORMAT(1H1,16H***OPTION NUMBER,I4,1X,A10,* MODEL CONTROL VALUES*,
     1* REQUIRED BUT NOT SPECIFIED*,/1X,*DEFAULT VALUES TAKEN INSTEAD*,
     1* AS LISTED*)
      GO TO 110
  510 IF(FLAGFR.EQ.1) GO TO 520
      AT=1970
      COMAX=0.1
      WRITE(61,909) NOPT,KEY
  909 FORMAT(1H1,16H***OPTION NUMBER,I4,1X,A10,/1X,*FREE FLOW COEFFIC*,
     1*IENTS DETERMINED FOR YEAR 1970 AS DEFAULT*)
      GO TO 210
C READ TIME SPECIFICATIONS
  520 READ(60,89) KEY
      IF(KOUNT.GT.0) GO TO 555
      IF(KEY.EQ."NEW TIME") GO TO 521
      WRITE(61,910) NOPT,KEY
  910 FORMAT(1H0,15H**OPTION NUMBER,I4,1X,A10,/1X,*KEYWORD "NEW TIME"*,
     1* WAS EXPECTED. SEARCH FOR VALID KEYWORD INITIATED*)
      GO TO 2
  521 FROM=DFROM
      TO=DTO
      STEP=DSTEP
      FILE=0
      READ(60,TIME)
      IF(KOUNT.GT.0) GO TO 555
      WRITE(61,911) NOPT
  911 FORMAT(1H1,13HOPTION NUMBER,I4,* CONTROLLED APPLIED TO*)
      WRITE(61,922) FROM,TO,STEP,FILE
C READ DETAIL SPECIFICATIONS AND WRITE SCRATCH FILE
      NYB=FROM
      NYE=TO
      REWIND 50
  522 READ(60,89) KEY
      IF(KOUNT.GT.0) GO TO 555
      WRITE(50,89) KEY
      IF(KEY.NE."NEW CODE") GO TO 540
      READ(60,89) KODE
      IF(KOUNT.GT.0) GO TO 555
      WRITE(50,89) KODE
      WRITE(61,912) KODE
```

```
  912 FORMAT(1H0,*DISCHARGE TYPE: *,A10)
      DISCH=THRES=RISPIP=MINDIS=0.
      MAXDIS=1.
      FLOWCO=0.
      GELEV=0.0
      READ(60,CODE)
      IF(KOUNT.GT.0) GO TO 555
      DISCH=DISCH*1000.
      MAXDIS=MAXDIS*1000.
      MINDIS=MINDIS*1000.
      FLOWCO=FLOWCO*1000.
      WRITE(50,CODE)
      WRITE(61,923) DISCH,THRES,RISPIP,MAXDIS,MINDIS,FLOWCO,GELEV
  923 FORMAT(1X,33HDISCHARGE PARAMETERS      DISCH =,F7.2,9H  THRES =,
     1F7.2,9H  RISPIP=,F7.2,/1X,10X,7HMAXDIS=,F7.2,9H  MINDIS=,F7.2,
     29H  FLOWCO=,F7.3,9H  GELEV =,F7.2,* FLOWS IN LITERS/SEC.*)
  523 CONTI=0
      AQUIFR=2
      WESTFR=EASTTO=SOUTFR=NORTTO=EAST=WEST=NORTH=SOUTH=1
      READ(60,COORDI)
      IF(KOUNT.GT.0) GO TO 555
      WRITE(50,COORDI)
      IF(EAST.EQ.1) EAST=WEST
      IF(NORTH.EQ.1) NORTH=SOUTH
      WRITE(61,924) WESTFR,EASTTO,SOUTFR,NORTTO,EAST,NORTH,AQUIFR
  924 FORMAT(1X,30HGRID COORDINATE RANGE  WESTFR=,I3,9H  EASTTO=,I3,
     19H  SOUTFR=,I3,9H  NORTTO=,I3,7H  EAST=,I3,8H  NORTH=,I3,
     29H  AQUIFR=,I3)
      IF(CONTI.EQ.1) GO TO 523
      GO TO 522
  540 ENDFILE 50
C APPLY DETAILED SPECS. BY TIME STEPS
      NF=FILE+6
      DO 550 NY=NYB,NYE,STEP
C INITIALIZE DISCHARGE ARRAY
      IF(NF.GT.6) GO TO 542
C INITIALIZE TO ZERO
      DO 541 N=1,MAQ
      DO 541 I=1,MEW
      DO 541 K=1,MNS
      IDIS(I,K,N)=0
  541 CONTINUE
      GO TO 539
C INITIALIZE FROM OLD DISCHARGE FILE
  542 CALL DISCOP(MAQ,MNS,MEW,IDIS,NY,NF)
C READ DETAILS FROM SCRATCH FILE
  539 REWIND 50
  543 READ(50,89) KEY
      IF(KEY.NE."NEW CODE") GO TO 549
      READ(50,89) KODE
      READ(50,CODE)
  548 READ(50,COORDI)
C DETERMINE LIMITS OF AREA
      IF(EAST.EQ.1) EAST=WEST
      IF(WEST.EQ.1) WEST=EAST
      IF(SOUTH.EQ.1) SOUTH=NORTH
      IF(NORTH.EQ.1) NORTH=SOUTH
```

```
      IF(WESTFR.LT.EASTTO) GO TO 544
      EASTTO=WESTFR=EAST
  544 IF(SOUTFR.LT.NORTTO) GO TO 545
      SOUTFR=NORTTO=NORTH
  545 IF(EASTTO.EG.1.OR.NORTTO.EQ.1) GO TO 547
      N=AQUIFR
C APPLY SPECIFIED MODS. TO EACH NODE OF THIS AREA
      DO 546 I=WESTFR,EASTTO
      DO 546 K=SOUTFR,NORTTO
      IF(KODE.NE."CONT THRES") GO TO 561
      VAR(1)=DISCH
      VAR(2)=THRES
      VAR(3)=COR(I,K,N)
      IF(VAR(3).EQ.0..AND.GELEV.NE.0.) VAR(3)=GELEV
      NCOD=2
      GO TO 560
  561 IF(KODE.NE."FREE FLOW") GO TO 562
      VAR(1)=COF(I,K,N)
      IF(FLOWCO.GT.0) VAR(1)=FLOWCO
      VAR(2)=0
      VAR(3)=COR(I,K,N)
      IF(VAR(3).EQ.0..AND.GELEV.NE.0.) VAR(3)=GELEV
      NCOD=3
      GO TO 560
  562 IF(KODE.NE."FREE THRES") GO TO 563
      VAR(1)=COF(I,K,N)
      IF(FLOWCO.GT.0.) VAR(1)=FLOWCO
      VAR(2)=THRES
      VAR(3)=COR(I,K,N)
      IF(VAR(3).EQ.0..AND.GELEV.NE.0.) VAR(3)=GELEV
      NCOD=4
      GO TO 560
  563 IF(KODE.NE."ELEVATED") GO TO 564
      VAR(1)=COF(I,K,N)
      IF(FLOWCO.GT.0) VAR(1)=FLOWCO
      VAR(2)=RISPIP
      VAR(3)=COR(I,K,N)
      IF(VAR(3).EQ.0..AND.GELEV.NE.0.) VAR(3)=GELEV
      NCOD=5
      GO TO 560
  564 IF(KODE.NE."MAXIMUM") GO TO 565
      VAR(1)=COF(I,K,N)
      IF(FLOWCO.GT.0.) VAR(1)=FLOWCO
      VAR(2)=MAXDIS
      VAR(3)=COR(I,K,N)
      IF(VAR(3).EQ.0..AND.GELEV.NE.0) VAR(3)=GELEV
      NCOD=6
      GO TO 560
  565 IF(KODE.NE."MINIMUM") GO TO 566
      VAR(1)=COF(I,K,N)
      IF(FLOWCO.GT.0.) VAR(1)=FLOWCO
      VAR(2)=MINDIS
      VAR(3)=COR(I,K,N)
      IF(VAR(3).EQ.0..AND.GELEV.NE.0) VAR(3)=GELEV
      NCOD=7
      GO TO 560
  566 VAR(1)=DISCH
```

```
                  VAR(2)=VAR(3)=0
                  NCOD=1
              560 CONTINUE
                  CALL PLAYCO(NCOD,VAR(1),VAR(2),VAR(3),IDIS(I,K,N),61)
              546 CONTINUE
              547 CONTINUE
            C READ MORE COORDINATES OR ANOTHER MODS. SPEC. FROM SCRATCH FILE
                  IF(CONTI.EQ.1) GO TO 548
                  GO TO 543
              549 CONTINUE
            C WRITE OUT COMPLETE DISCHARGE ARRAY
                  CALL MANWRI(MAQ,MNS,MEW,IDIS,NY,11)
              550 CONTINUE
            C READ ANOTHER TIME STEP SPEC. OR A DIFFERENT OPTION
                  IF(EOF(60)) 10,3
                3 CONTINUE
                  IF(KEY.NE."NEW TIME") GO TO 2
                  NOPT=NOPT+1
                  GO TO 521
                  END
```

CARD NR.	SEVERITY	DETAILS	DIAGNOSIS OF PROBLEM
100	I	23CD 100	FIELD WIDTH OF A CONVERSION DESCRIPTOR SHOULD BE AS LARGE AS THE MINIMUM SPECIFIED FOR THAT DESCRIPTOR.

```
      PROGRAM COMCAL(INPUT,OUTPUT,GEO,OLD,POT,MAN,NEW,TAPE60=INPUT,
     1TAPE61=OUTPUT,TAPE1=GEO,TAPE2=OLD,TAPE3=POT,TAPE4=MAN,TAPE5=NEW,
     2REC,TAPE7=REC,GEL,TAPE8=GEL,CAP,TAPE9=CAP)
      DIMENSION IRS(2),IRE(2),KRS(67,2),KRE(67,2),IN(2),KN(2),TC(2),
     1TN(2),SBN(2),SBC(2),SBS(2),SBT(2),VTN(2),BR(2),VR(2),SI(2)
      COMMON/LEV/PA(67,58,5),VP(67,58,2),TE(67,58,2),TS(67,58,2),
     1SC(67,58,2),DQ(67,58,2),IG(67,58),AR(67,58)
      COMMON A,MNS,MEW,MAQ,ORF,AVER,SIG
      LEVEL 2,PA,VP,TE,TS,SC,DQ,IG,AR
      EPSI=0.00000001**2
      A=25000.
      MNS=58
      MEW=67
      MAQ=2
C READ CONTROL DATA
      READ(60,90) IYST,IYEN
   90 FORMAT(I4,1X,I4)
      READ(60,91) NA,NBN,MIT
   91 FORMAT(I1,1X,I2,1X,I3)
      READ(60,92) CONV
   92 FORMAT(F15.5)
      READ(60,92) ORF
      READ(60,92) VERSC
      READ(60,92) BIAS
      READ(60,92) RCF
      READ(60,93) MAXI,CON
   93 FORMAT(I3,F15.5)
   99 FORMAT(A1)
C READ GEOMETRY AND ESTABLISH RANGE,STARTING POTENTIALS,HYDRAULIC DATA
      CALL REDGEO(1)
      CALL ERANG(IRS,IRE,KRS,KRE)
      CALL REDPOT(3,IYST)
      CALL REDHYD(2)
      CALL SERDAT(MNS,MEW,8,-90,NA,1HG,AR,61)
C INITIALIZE POTENTIALS,DISCHARGES
      AVER=0.
      DO 1 N=1,MAQ
      DO 1 I=1,MEW
      DO 1 K=1,MNS
      NQ=N*2
      NP=NQ+1
      DQ(I,K,N)=0.
      PA(I,K,NQ)=AR(I,K)
      IF(N.NE.NA) GO TO 1
      VP(I,K,NA)=VP(I,K,NA)*VERSC
    1 CONTINUE
      NSTP=1
      NSTP=MAX0(1,IYEN-IYST)
      CALL HISDIS(4,IYEN)
      CALL SERDAT(MNS,MEW,7,IYEN,NA,1HP,AR,61)
      WRITE(61,901) NA
  901 FORMAT(1H1,15X,*AQUIFER *,I2,/1X,15X,10(1H-))
      WRITE(61,902) IYST,IYEN
  902 FORMAT(1H0,*NEW TRANSMISSIVITIES FOR*,I5,* TO*,I5)
      WRITE(61,903) NBN,MIT
  903 FORMAT(1H0,*CONTROL DATA SPEC.*,  /1X,*BOUNDARY CODE *,I4,/1X,
     1*MAX. ITERATION*,I4)
```

```
      WRITE(61,904) CONV,ORF,VERSC
  904 FORMAT(1X,*CONVERGENCE CRITERION*,2X,E13.6,/1X,
     1*OVERRELAXATION FACTOR *,1X,F13.2,/1X,*VERTICAL SCALING FACT.*,
     21X,E13.6)
      WRITE(61,905) BIAS
  905 FORMAT(1X,*BIAS*,19X,E13.6)
      WRITE(61,906) RCF
  906 FORMAT(1X,*RCF FACTOR*,13X,E13.6)
C ITERATIONS LOOP, INITIALIZE STATISTICAL SUMS
      DO 100 IT=1,MIT
      NSUM=NBOU=0
      SBAL=SBALQ=TBAL=0.
C NODAL LOOP
      IST=IRS(NA)
      IEN=IRE(NA)
      DO 10 II=IST,IEN
      I=II
      KST=KRS(I,NA)
      KEN=KRE(I,NA)
      DO 10 KK=KST,KEN
      K=KK
      NAD=1
      IN(2)=I+NAD
      KN(1)=K+NAD
      IN(1)=I
      KN(2)=K
      CALL REXBAL(RCF,I,K,NA,IN(1),KN(1),NA,NBN,SBC(1),SBN(1),TC(1),
     1TN(1),SBS(1),SBT(1) ,NSTP)
      SI(1)=SIG
      CALL REXBAL(RCF,I,K,NA,IN(2),KN(2),NA,NBN,SBC(2),SBN(2),TC(2),
     1TN(2),SBS(2),SBT(2) ,NSTP)
      SI(2)=SIG
C CALCULATE IMPROVEMENT PARAMETERS
      VBC=(SBC(1)**2+SBS(1)**2)/2.-((SBC(1)+SBS(1))/2.)**2
      VBN=(SBN(1)**2+SBT(1)**2)/2.-((SBN(1)+SBT(1))/2.)**2
      BR(1)=100.
      IF(VBN.GT.EPSI) BR(1)=VBC/VBN
      VBC=(SBC(2)**2+SBS(2)**2)/2.-((SBC(2)+SBS(2))/2.)**2
      VBN=(SBN(2)**2+SBT(2)**2)/2.-((SBN(2)+SBT(2))/2.)**2
      BR(2)=100.
      IF(VBN.GT.EPSI) BR(2)=VBC/VBN
      VTC=(TC(1)**2+TC(2)**2)/2.-((TC(1)+TC(2))/2.)**2
      VTC=AMAX1(VTC,EPSI)
      VTN(1)=(TN(1)**2+TC(2)**2)/2.-((TN(1)+TC(2))/2.)**2
      VTN(2)=(TC(1)**2+TN(2)**2)/2.-((TC(1)+TN(2))/2.)**2
      VR(1)=VR(2)=10
      IF(VTN(1).GT.EPSI) VR(1)=VTC/VTN(1)
      IF(VTN(2).GT.EPSI) VR(2)=VTC/VTN(2)
 CHOSE ALTERNATIVE WITH HIGHER RATIO
      IF(TC(1).EQ.0..AND.TC(2).EQ.0.) GO TO 15
      IF(TC(1).EQ.0.) GO TO 14
      IF(TC(2).EQ.0.) GO TO 13
      IF(BR(1)*VR(1)-BR(2)*VR(2)) 11,12,12
   11 CONTINUE
      SBAL=SBAL+SI(2)
      IF(NAD.GT.0) TE(I,K,NA)=TN(2)
      IF(NAD.LT.0) TE(I-1,K,NA)=TN(2)
```

```
      CALL REXBAL(RCF,I,K,NA,IN(1),KN(1),NA,NBN,SBC(1),SBN(1),TC(1),
     1TN(1),SBS(1),SBT(1) ,NSTP)
   13 TBAL=TBAL+SBN(1)
      SBAL=SBAL+SIG
      SBALQ=SBALQ+SBN(1)**2
      NSUM=NSUM+1
      IF(NAD.GT.0) TS(I,K,NA)=TN(1)
      IF(NAD.LT.0) TS(I,K-1,NA)=TN(1)
      GO TO 10
   12 CONTINUE
      SBAL=SBAL+SI(1)
      IF(NAD.GT.0) TS(I,K,NA)=TN(1)
      IF(NAD.LT.0) TS(I,K-1,NA)=TN(1)
      CALL REXBAL(RCF,I,K,NA,IN(2),KN(2),NA,NBN,SBC(2),SBN(2),TC(2),
     1TN(2),SBS(2),SBT(2) ,NSTP)
   14 CONTINUE
      IF(NAD.GT.0) TE(I,K,NA)=TN(2)
      IF(NAD.LT.0) TE(I-1,K,NA)=TN(2)
   15 CONTINUE
      TBAL=TBAL+SBN(2)
      SBAL=SBAL+SIG
      SBALQ=SBALQ+SBN(2)**2
      NSUM=NSUM+1
   10 CONTINUE
 CALCULATE CONVERGENCE CHECKS
      ERAQ=SBALQ/NSUM
      ERNO=ERAQ-(TBAL/NSUM)**2
      ERAQ=SQRT(ERAQ)
      ERNO=SQRT(ERNO)
      ERAM=TBAL
      AVER=0.
      IF(SBAL.GT.0) AVER=(TBAL-BIAS)/SBAL
      WRITE(61,900) IT,ERAM,ERAQ,ERNO
  900 FORMAT(1H0,*ITERATION*,I4,* ABSOLUTE IMBALANCE*, 6X,E13.6,* M3*,
     1/1X,14X,*STANDARD DEV. REL. ZERO*,1X,E13.6,/1X,14X,
     2*STAND. DEV. REL. AVER.*,2X,E13.6)
      IF(IT.EQ.1) BALIN=ABS(TBAL)
      IF(IT.GT.3.AND.ABS(TBAL).GT.BALIN) GO TO 101
      IF(ERNO.LE.CONV) GO TO 101
      IF(MAXI.LT.1) GO TO 100
      CALL CALMOD(MAXI,1.0,CON,0,IRS,IRE,KRS,KRE)
  100 CONTINUE
  101 CONTINUE
      CALL OUTHYD(5,IYEN)
      CALL OUTPOT(IYEN,9)
      END
```

```
      SUBROUTINE OUTHYD(NFH,NYY)
      COMMON/LEV/PA(67,58,5),HY(67,58,8)
      COMMON A,MNS,MEW,MAQ
      DIMENSION CH(8)
      LEVEL 2,PA,HY
      CH(1)=CH(2)=10HZ
      CH(3)=CH(4)=10HE
      CH(5)=CH(6)=10HN
      CH(7)=CH(8)=10HS
      NY=NYY-1000
      IF(NY.GE.1000) NY=NY-1000
      MM=MEW/6+1
      DO 1 NT=1,4
      DO 1 NA=1,MAQ
      N=(NT-1)*MAQ+NA
      CC=CH(N)
      DO 2 K=1,MNS
      DO 2 M=1,MM
      IA=(M-1)*6+1
      IZ=MINO(IA+5,MEW)
      WRITE(NFH,90) (NY,NA,K,IA,CC,(HY(I,K,N),I=IA,IZ))
   90 FORMAT(I3,1X,I1,1X,I2,1X,I2,1X,A1,1X,6E11.5)
    2 CONTINUE
    1 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE ASSIGN(V,C,F,RAM,N)
      COMMON/LEV/CH(100,50)
      LEVEL 2,CH
      VI=(V-RAM)/F
      I=VI
      IF(VI-I.GT.0.5) I=I+1
      IF(CH(N,I).NE.1H ) CH(N,I)=1HX
      IF(CH(N,I).NE.1HX) CH(N,I)=C
      RETURN
      END
```

```
      SUBROUTINE TIMESE(IF,TD,NST,NY,NQ,OO,TP,NCP,FC,NRW,NRE,NRS,
     1NRN,MF,NF)
      DIMENSION IF(4),TD(4),NY(2),NQ(4),TP(10),VA(100),VB(100),VC(100),
     1VD(100),VE(100),VF(100)
      COMMON/LEV/A(67,58),B(67,58),C(67,58),D(67,58)
      COMMON Z,MNS,MEW,MAQ
      LEVEL 2,A,B,C,D
      MA=1
      MZ=NY(2)-NY(1)+1
      DO 100 M=MA,MZ,NST
      MY=M+NY(1)-1
      IF(IF(1).EQ.0) GO TO 1
      CALL SERDAT(MNS,MEW,IF(1),MY,NQ(1),TD(1),A,MF)
    1 IF(IF(2).EQ.0) GO TO 2
      CALL SERDAT(MNS,MEW,IF(2),MY,NQ(2),TD(2),B,MF)
    2 IF(IF(3).EQ.0) GO TO 3
      CALL SERDAT(MNS,MEW,IF(3),MY,NQ(3),TD(3),C,MF)
    3 IF(IF(4).EQ.0) GO TO 4
      CALL SERDAT(MNS,MEW,IF(4),MY,NQ(4),TD(4),D,MF)
    4 VA(M)=VB(M)=VC(M)=VD(M)=VE(M)=VF(M)=0.
      NSUM=0
      DO 5 I=NRW,NRE
      DO 5 K=NRS,NRN
      NSUM=NSUM+1
      VA(M)=VA(M)+A(I,K)
      VB(M)=VB(M)+B(I,K)
      VC(M)=VC(M)+C(I,K)
      VD(M)=VD(M)+D(I,K)
    5 CONTINUE
      VA(M)=VA(M)/NSUM
      VB(M)=VB(M)/NSUM
      VC(M)=VC(M)/NSUM
      VD(M)=VD(M)/NSUM
      VE(M)=VA(M)-VB(M)
      VF(M)=VC(M)-VD(M)
  100 CONTINUE
      IF(OO.EQ.1HT) GO TO 11
      CALL QUICK(IF,TD,NY,NQ,TP,FC,VA,VB,VC,VD,VE,VF,NST)
   11 IF(OO.EQ.1HQ) GO TO 12
      CALL TABLE(IF,TD,NY,NQ,TP,FC,VA,VB,VC,VD,VE,VF,NST)
   12 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE LARMAP(MEW,MNS,NF,VA,TP,NCP)
      DIMENSION VA(MEW,MNS),TP(10),PL(20,2),CH(4)
      LEVEL 2,VA
      CH(1)=5H.....
      CH(2)=5H  .
      CH(3)=5H
      CH(4)=5H-----
      M=NCP+1
      DO 5 N=M,8
      TP(N)=10H
    5 CONTINUE
      CALL DATE(TIC)
      TP(9)=10HPRINTED ON
      TP(10)=TIC
      MM=MEW/20+1
      DO 1 M=1,MM
      WRITE(NF,90)
   90 FORMAT(1H0)
      WRITE(NF,91) ((TP(I),I=1,10),(CH(4),I=1,20))
   91 FORMAT(1H1,10A10,/1H ,20A5)
      IA=(M-1)*20+1
      IZ=MIN0(IA+19,MEW)
      WRITE(NF,92) (I,I=IA,IZ)
   92 FORMAT(1H0,6X,20(1X,I3.3,1X))
      WRITE(NF,93) (CH(2),I=1,20)
   93 FORMAT(7X 20A5)
      WRITE(NF,94) (CH(4),I=1,20)
   94 FORMAT(6X,*I*,20A5,*--I*)
C PAGE TITLE COMPLETED
C GENERATE MAP LINES AS CHARACTER STRINGS
      DO 2 K=1,MNS
      J=MNS-K+1
C WRITE SPACER LINE
      WRITE(NF,96) (CH(2),I=1,20)
C FORM CHARACTER STRINGS
      DO 3 I=1,20
      PL(I,1)=CH(1)
      PL(I,2)=CH(2)
      IB=(M-1)*20+I
      IF(IB.GT.IZ) GO TO 3
      IF(VA(IB,J).EQ.-9999.) GO TO 3
      ENCODE(10,900,DM) VA(IB,J)
  900 FORMAT(E10.4)
      DECODE(10,901,DM) VN,IE
  901 FORMAT(F6.4,1X,I3)
      MAN=VN*10000
      ENCODE(5,902,DM) MAN
  902 FORMAT(I5)
      ENCODE(3,903,DE) IE
  903 FORMAT(I3)
      DECODE(5,904,DM) PL(I,1)
  904 FORMAT(A5)
      DECODE(4,905,DE) PL(I,2)
  905 FORMAT(A3)
    3 CONTINUE
C PRINT TWO CODED LINES
      WRITE(NF,95) (J,(PL(I,1),I=1,20))
```

```
 95 FORMAT(2X,I3.3,2H.I,20A5,3H..I)
    WRITE(NF,96) (PL(I,2),I=1,20)
 96 FORMAT(6X,*I*,20A5,*  I*)
  2 CONTINUE
  1 CONTINUE
    WRITE(NF,94) (CH(4),I=1,20)
    RETURN
    END
```

```
      SUBROUTINE SERDAT(MNS,MEW,IF,NYS,NQ,TD,AR,MF)
      DIMENSION AR(MEW,MNS)
      LEVEL 2,AR
C VERSION 21 JULY 1978
C REWIND DATA FILE AND SEARCH FOR CORRECT YEAR
      DO 50 N=1,750
      BACKSPACE IF
   50 CONTINUE
      NY=NYS-1000
      IF(TD.EQ.1HG.OR.TD.EQ.1HT) NY=-1
      IF(TD.EQ.1HS.OR.TD.EQ.1HZ.OR.TD.EQ.1HE.OR.TD.EQ.1HN)NY=-1
      IFL=0
   10 READ(IF,91) IY
      IF(EOF(IF)) 100,1
   91 FORMAT(I3,1X,I1,7X,A1)
    1 IF(IY.LT.800) IY=IY+1000
      IF(NY.LT.0) GO TO 20
      IF(IY-NY) 10,20,30
   30 IF(IFL) 100,31,100
   31 REWIND IF
      IFL=1
      GO TO 10
C CORRECT YEAR FOUND
   20 BACKSPACE IF
   22 READ(IF,91) IY,IQ,T
      IF(EOF(IF)) 100,21
   21 IF(IY.LT.800) IY=IY+1000
      IF(NY.LT.0) NY=IY
      IF(NY.NE.IY) GO TO 100
      IF(NQ.NE.IQ.OR.TD.NE.T) GO TO 22
C START OF DATA SET FOUND
      WRITE(MF,901) T,IQ,IY+1000,IF
  901 FORMAT(1X,*DATA TYPE *,A1,* AQUIFER*,I2,* FOR YEAR*,I5,
     1* LOCATED ON FILE*,I3)
      BACKSPACE IF
      MR=MEW/6+1
      DO 40 K=1,MNS
      DO 41 N=1,MR
      IA=(N-1)*6+1
      IZ=MIN0(IA+5,MEW)
      READ(IF,92) (AR(I,K),I=IA,IZ)
   92 FORMAT(14X,6E11.5)
   41 CONTINUE
   40 CONTINUE
      RETURN
  100 NY=NY+1000
      WRITE(MF,900) TD,NQ,NY,IF
  900 FORMAT(1X,13H***DATA TYPE ,A1,* AQUIFER*,I2,* FOR YEAR *,I4,
     1* NOT FOUND ON FILE NO.*,I3)
      RETURN
      END
```

```
      SUBROUTINE BOUNDQ(IF,NY,NQ,OO,LOG,MF,NF,TP,NCP)
      DIMENSION IF(4),NY(4),NQ(4),TP(10)
      COMMON/LEV/PA(67,58,5),HY(67,58,8),DQ1(67,58),DQ2(67,58)
     1,IG(67,58)
      COMMON A,MNS,MEW,MAQ
      LEVEL 2,PA,HY,DQ1,DQ2,IG
      NHF=IF(1)
      NY(1)=NY(2)
      REWIND NHF
      CALL REDHYD(IF(1))
      CALL SBOUND(NQ(1),NY(1),NY(2),IF(2),BAL)
      IF(NQ(1).EQ.2) GO TO 2
      IF(OO.EQ.1HL) GO TO 3
      CALL SMAMAP(MEW,MNS,NF,DQ1,LOG,TP,NCP)
    3 IF(OO.EQ.1HS) GO TO 5
      CALL LARMAP(MEW,MNS,NF,DQ1,TP,NCP)
      GO TO 5
    2 IF(OO.EQ.1HL) GO TO 4
      CALL SMAMAP(MEW,MNS,NF,DQ2,LOG,TP,NCP)
    4 IF(OO.EQ.1HS) GO TO 5
      CALL LARMAP(MEW,MNS,NF,DQ2,TP,NCP)
    5 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE LEAKAG(IF,NY,NQ,FC,OO,LOG,MF,NF,TP,NCP)
      DIMENSION IF(4),NY(4),NQ(4),TP(10)
      COMMON/LEV/W(67,58),PI1(67,58),PC1(67,58),PI2(67,58),PC2(67,58),
     1HY(67,58,6),SC1(67,58),SC2(67,58),DQ1(67,58),DQ2(67,58)
     1,IG(67,58)
      COMMON A,MNS,MEW,MAQ
      LEVEL 2,W,PI1,PI2,PC1,PC2,HY,SC1,SC2,DQ1,DQ2,IG
      NFH=IF(1)
      NY(1)=NY(2)
      REWIND NFH
      CALL REDHYD(IF(1))
      CALL SLEAKA(NQ(1),NY(1),NY(2),IF(2),BAL)
      IF(NQ(1).EQ.2) GO TO 2
      IF(OO.EQ.1HL) GO TO 3
      IF(FC.EQ.1HU.OR.FC.EQ.1HA) CALL SMAMAP(MEW,MNS,NF,DQ1,LOG,TP,NCP)
      IF(FC.EQ.1HD.OR.FC.EQ.1HA) CALL SMAMAP(MEW,MNS,NF,PI1,LOG,TP,NCP)
      IF(FC.EQ.1H+.OR.FC.EQ.1HA) CALL SMAMAP(MEW,MNS,NF,SC1,LOG,TP,NCP)
      IF(OO.EQ.1HS) GO TO 4
    3 IF(FC.EQ.1HU.OR.FC.EQ.1HA) CALL LARMAP(MEW,MNS,NF,DQ1,TP,NCP)
      IF(FC.EQ.1HD.OR.FC.EQ.1HA) CALL LARMAP(MEW,MNS,NF,PI1,TP,NCP)
      IF(FC.EQ.1H+.OR.FC.EQ.1HA) CALL LARMAP(MEW,MNS,NF,SC1,TP,NCP)
      GO TO 4
    2 IF(OO.EQ.1HL) GO TO 5
      IF(FC.EQ.1HU.OR.FC.EQ.1HA) CALL SMAMAP(MEW,MNS,NF,DQ2,LOG,TP,NCP)
      IF(FC.EQ.1HD.OR.FC.EQ.1HA) CALL SMAMAP(MEW,MNS,NF,PI2,LOG,TP,NCP)
      IF(FC.EQ.1H+.OR.FC.EQ.1HA) CALL SMAMAP(MEW,MNS,NF,SC2,LOG,TP,NCP)
      IF(OO.EQ.1HS) GO TO 4
    5 IF(FC.EQ.1HU.OR.FC.EQ.1HA) CALL LARMAP(MEW,MNS,NF,DQ2,TP,NCP)
      IF(FC.EQ.1HD.OR.FC.EQ.1HA) CALL LARMAP(MEW,MNS,NF,PI2,TP,NCP)
      IF(FC.EQ.1H+.OR.FC.EQ.1HA) CALL LARMAP(MEW,MNS,NF,SC2,TP,NCP)
    4 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE SLEAKA(NAQ,NYB,NYE,NFP,BAL)
      COMMON/LEV/PA(67,58,5),VP(67,58,2),TE(67,58,2),TS(67,58,2),SC(67,
     158,2),DQ(67,58,2),IG(67,58)
      COMMON A,MNS,MEW,MAQ
      DIMENSION IGA(10)
      LEVEL 2,PA,VP,TE,TS,SC,DQ,IG
      CALL REDPOT(NFP,NYB)
C INITIALIZE LEAKAGE ARRAYS
      NP=NAQ*2+1
      DO 1 I=1,MEW
      DO 1 K=1,MNS
      SC(I,K,NAQ)=DQ(I,K,NAQ)=0.
      IF(PA(I,K,NP).EQ.-9999.) SC(I,K,NAQ)=DQ(I,K,NAQ)=-9999.
    1 CONTINUE
C LEAKAGE FOR YEAR NYB  (RATE)
      DO 2 I=1,MEW
      DO 2 K=1,MNS
      ENCODE(10,91,IGC) IG(I,K)
   90 FORMAT(10I1)
      DECODE(10,90,IGC) (IGA(L),L=1,10)
   91 FORMAT(I10)
      IF(IGA(NAQ).LE.2) GO TO 2
      NPD=NAQ
      NN=NAQ-1
      NPU=0
      IF(NN.LT.1) GO TO 9
      DO 3 L=1,NN
      IF(IGA(L).GT.1) NPU=L
    3 CONTINUE
    9 CONTINUE
      DO 4 L=NAQ,10
      IF(IGA(L).GT.1.AND.NPD.EQ.NAQ) NPD=L
    4 CONTINUE
      DQ(I,K,NAQ)=(PA(I,K,NPU*2+1)-PA(I,K,NP))*VP(I,K,NAQ)
      SC(I,K,NAQ)=(PA(I,K,NPD*2+1)-PA(I,K,NP))*VP(I,K,NPD)
    2 CONTINUE
C LEAKAGE FOR YEAR NYE - AVERAGE RATE
      CALL REDPOT(NFP,NYE)
      DO 5 I=1,MEW
      DO 5 K=1,MNS
      ENCODE(10,91,IGC) IG(I,K)
      DECODE(10,90,IGC) (IGA(L),L=1,10)
      IF(IGA(NAQ).LE.2) GO TO 5
      NPD=NAQ
      NN=NAQ-1
      NPU=0
      IF(NN.LT.1) GO TO 10
      DO 6 L=1,NN
      IF(IGA(L).GT.1) NPU=L
    6 CONTINUE
   10 CONTINUE
      DO 7 L=NAQ,10
      IF(IGA(L).GT.1.AND.NPD.EQ.NAQ) NPD=L
    7 CONTINUE
      DQ(I,K,NAQ)=(DQ(I,K,NAQ)+(PA(I,K,NPU*2+1)-PA(I,K,NP))*VP(I,K,NAQ)
     1)/2.
      SC(I,K,NAQ)=(SC(I,K,NAQ)+(PA(I,K,NPD*2+1)-PA(I,K,NP))*VP(I,K,NPD)
```

```
               1)/2.
              5 CONTINUE
          C CONVERT RATES TO VOLUMES  DQ-UP  PI-DOWN  SC-BALANCE
                NI=NP-1
                DT=(NYE-NYB)*365.*86400.
                DT=AMAX1(DT,1.0)
                BAL=0.
                UPBA=0.
                DOBA=0.
                DO 8 I=1,MEW
                DO 8 k=1,MNS
                IF(DQ(I,K,NAQ).EQ.-9999.) GO TO 8
                IF(SC(I,K,NAQ).EQ.-9999.) GO TO 8
                DQ(I,K,NAQ)=UP=DQ(I,K,NAQ)*DT
                PA(I,K,NI)=DOWN=SC(I,K,NAQ)*DT
                SC(I,K,NAQ)=UP+DOWN
                UPBA=UPBA+UP
                DOBA=DOBA+DOWN
                BAL=BAL+UP+DOWN
              8 CONTINUE
                WRITE(61,900) NAQ,NYB,NYE,BAL
            900 FORMAT(1H1,*LEAKAGE TOTAL BALANCE FOR AQUIFER*,I3,* PERIOD 1*,I3
               1,*, TO 1*,I3,* *,E12.6,* CUBIC METERS*)
                WRITE(61,901) UPBA,DOBA
            901 FORMAT(1X,*CONNECTING WITH AQUIFER ABOVE*,E12.6,/1X,
               1*CONNECTING WITH AQUIFER BELOW*,E12.6)
                RETURN
                END
```

```
      SUBROUTINE EFDRAW(NAQ,NYB,NYE,NFP,NFT,DRA)
      COMMON/LEV/PA(67,58,5),HY(67,58,6),SC(67,58,2),DQ(67,58,2),
     1IG(67,58)
      COMMON A,MNS,MEW,MAQ
      LEVEL 2,PA,HY,SC,DQ,IG
      REWIND NFT
      NP=NAQ*2+1
C READ AND STORE POTENTIAL YEAR NYB
      CALL REDPOT(NFP,NYB)
      DO 1 I=1,MEW
      DO 1 K=1,MNS
      DQ(I,K,NAQ)=PA(I,K,NP)
    1 CONTINUE
C READ POTENTIAL YEAR NYE, CALCULATE AND STORE DRAWDOWN
      NI=NP-1
      CALL REDPOT(NFP,NYE)
      DO 2 I=1,MEW
      DO 2 K=1,MNS
      PA(I,K,NI)=DQ(I,K,NAQ)-PA(I,K,NP)
      IF(PA(I,K,NP).EQ.-9999.) PA(I,K,NI)=-9999.
    2 CONTINUE
C READ AQUIFER THICKNESS, CALC.EFFECTIVE DRAWDOWN, ACCUMULATE
      DRA=0.
      DO 3 N=1,NAQ
      DO 3 K=1,MNS
      READ(NFT,90) (DQ(I,K,N),I=1,MEW)
   90 FORMAT(14X,6E11.5)
    3 CONTINUE
      DO 4 I=1,MEW
      DO 4 K=1,MNS
      IF(PA(I,K,NI).EQ.-9999.) GO TO 4
      DQ(I,K,NAQ)=DQ(I,K,NAQ)*PA(I,K,NI)
      DRA=DRA+DQ(I,K,NAQ)*A**2.
    4 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE STOTAQ(NAQ,NYB,NYE,NFD,BAL)
      COMMON/LEV/PA(67,58,5),VP(67,58,6),SC(67,58,2),DQ(67,58,2),IG(67,
     158)
      COMMON A,MNS,MEW,MAQ
      LEVEL 2,PA,VP,SC,DQ,IG
C SKIP CONTROL VARIABLES
   90 FORMAT(A1)
C READ DISCHARGE YEAR NYB, SET OUTSIDE TO -9999.
      NP=NAQ*2+1
      NI=NP-1
      CALL HISDIS(NFD,NYB)
      DO 1 I=1,MEW
      DO 1 K=1,MNS
      PA(I,K,NI)=DQ(I,K,NAQ)
      IF(PA(I,K,NP).EQ.-9999.) PA(I,K,NI)=-9999.
    1 CONTINUE
C READ Q FOR YEAR NYE, AVERAGE, TO VOLUME, ACCUMULATE
      BAL=0.
      DT=(NYE-NYB)*365.*86400.
      DT=AMAX1(DT,1.0)
      CALL HISDIS(NFD,NYE)
      DO 2 I=1,MEW
      DO 2 K=1,MNS
      IF(PA(I,K,NP).EQ.-9999.) GO TO 3
      AVD=DQ(I,K,NAQ)=(DQ(I,K,NAQ)+PA(I,K,NI))/2.*DT
      SC(I,K,NAQ)=SC(I,K,NAQ)+AVD
      BAL=BAL+AVD
      GO TO 2
    3 DQ(I,K,NAQ)=-9999.
    2 CONTINUE
      WRITE(61,900) NAQ,NYB,NYE,BAL
  900 FORMAT(1X,*ARTIFICIAL DISCHARGE AQUIFER*,I3,* PERIOD 1*,I3,* TO
     11*,I3,2X,E12.6,* CUBIC METERS*)
      RETURN
      END
```

```
      FUNCTION IGEO(I,K,NA,IG)
      DIMENSION IG(67,58),IGA(10)
      LEVEL 2,IG
      ENCODE(10,90,IGC) IG(I,K)
      DECODE(10,91,IGC) (IGA(L),L=1,10)
   90 FORMAT(I10)
   91 FORMAT(10I1)
      IGEO=IGA(NA)
      RETURN
      END
```

```
  1            PROGRAM RUNSTE(INPUT,OUTPUT,GEO,HYD,POT,TAPE60=INPUT,TAPE61=
              1OUTPUT,TAPE10=GEO,TAPE20=HYD,TAPE30=POT,CAP,TAPE2=CAP)
               DIMENSION IRS(2),IRE(2),KRS(67,2),KRE(67,2),IGA(10)
               COMMON/LEV/PA(67,58,5),HY(67,58,8),DQ(67,58,2),IG(67,58)
  5            COMMON A,MNS,MEW,MAQ
               LEVEL 2,PA,HY,DQ,IG
               MNS=58
               MEW=67
               MAQ=2
 10            A=25000.
         C READ CONTROL VARIAPLES FROM INPUT
               READ(60,90) IYB,IYE,IYS,MAXI,NCAR
            90 FORMAT(I4,1X,I4,1X,I2,2I3)
               READ(60,91) TMF,YMAX,YMIN,CON,ORFI
 15         91 FORMAT(3F7.2,F5.4,F4.2)
         C CALL INPUT ROUTINES
               CALL REDGEO(10)
               CALL RANGE(IRS,IRE,KRS,KRE)
               CALL REDHYD(20)
 20            CALL REDPOT(30,IYB)
         C CALCULATE
               DO 1 M=1,MAQ
               NA=M*2
               NB=NA+1
 25            DO 1 K=1,MNS
               DO 1 I=1,MEW
               PA(I,K,NA)=PA(I,K,NB)
             1 CONTINUE
               CALL STEADY(MAXI,IT,ORFI,CON,NCAR,IRS,IRE,KRS,KRE)
 30      C OUTPUT POTENTIALS
               CALL OUTPOT(IYB,2)
               END
```

CARD NR.	SEVERITY	DETAILS	DIAGNOSIS OF PROBLEM
15	I	23 CD 15	FIELD WIDTH OF A CONVERSION DESCRIPTOR SHOULD BE AS LARGE AS THE MINIMUM SPECIFIED FOR THAT DESCRIPTOR.

```
      SUBROUTINE MANCHE(NF)
      REWIND NF
      WRITE(61,899)
  899 FORMAT(1H1,*CONTINUITY CHECK OF FILE MANIPU*)
      FLAG=0
      READ(NF,90) IYBB,IYEE,IYS
      IYB=IYBB+1000
      IF(IYB.LT.1800) IYB=IYB+1000
      IYE=IYEE+1000
      IF(IYE.LE.1800) IYE=IYE+1000
   90 FORMAT(1X,3I3)
      READ(NF,91) XX
   91 FORMAT(A1)
      DO 1 N=IYB,IYE,IYS
    2 READ(NF,92) NY
   92 FORMAT(I3)
      IF(EOF(NF)) 3,4
    3 FLAG=1
    5 NNY=N
      WRITE(61,900) NNY
  900 FORMAT(1H0,*DISCHARGE DATA MISSING FOR YEAR*,I5)
      IF(FLAG.EQ.0) GO TO 1
      WRITE(61,901) NNY
  901 FORMAT(1H0,*NO FURTHER DATA FOR ANY YEAR AFTER*,I5)
      RETURN
    4 NY=NY+1000
      IF(NY.LT.1800) NY=NY+1000
      IF(NY-N) 2,1,5
    1 CONTINUE
      WRITE(61,902)
  902 FORMAT(1H0,*CONTINUITY CHECK COMPLETED*)
      RETURN
      END
```

```
      SUBROUTINE REDPOT(NF,IBY)
      COMMON/LEV/PA(67,58,5),HY(67,58,10),IG(67,58)
      COMMON A,MNS,MEW,MAQ
      LEVEL 2,PA,HY,IG
C SEARCH FOR DATA OF YEAR IYB
      BACKSPACE NF
      IYB=IBY
      IFL=0
      IYB=IYB-1000
      IY=IYB
   77 MUF=IY-IYB+1
      IF(MUF.LT.0) MUF=0
      MUF=MUF*(MAQ+1)*MNS*(MEW/6+1)
      DO 78 N=1,MUF
      BACKSPACE NF
   78 CONTINUE
    1 READ(NF,90) IY
   90 FORMAT(I3)
      IF(EOF(NF)) 100,2
    2 IF(IY.LT.800) IY=IY+1000
      IF(IY.LT.IYB) GO TO 1
      IF(IY.GT.IYB.AND.IFL.EQ.1) GO TO 100
      IF(IY.GT.IYB) IFL=1
      IF(IY.EQ.IYB) GO TO 3
      GO TO 77
C READ POTENTIALS
    3 BACKSPACE NF
      NA=MAQ+1
      DO 4 N=1,NA
      DO 5 K=1,MNS
      READ(NF,91) (PA(I,K,N),I=1,MEW)
   91 FORMAT(14X,6E11.5)
    5 CONTINUE
    4 CONTINUE
      DO 6 I=1,MEW
      DO 6 K=1,MNS
      PA(I,K,5)=PA(I,K,4)=PA(I,K,3)
      PA(I,K,3)=PA(I,K,2)
    6 CONTINUE
      RETURN
  100 IYB=IYB+1000
      WRITE(61,900) IYB
  900 FORMAT(1H1,5H*****,*POTENTIALS FOR YEAR*,I5,* NOT FOUND*)
      STOP
      END
```

```
      SUBROUTINE HISPOT(NF,IYY)
      COMMON/LEV/PA(67,58,5)
      COMMON A,MNS,MEW,MAQ
      LEVEL 2,PA
C SEARCH FOR YEAR IY
      BACKSPACE NF
      IY=IYY
      IFL=0
      IY=IY-1000
      NY=IY
   77 MUF=NY-IY+1
      IF(MUF.LT.0) MUF=0
      MUF=MUF*(MAQ+1)*MNS*(MEW/6+1)
      DO 78 N=1,MUF
      BACKSPACE NF
   78 CONTINUE
    1 READ(NF,90) NY
   90 FORMAT(I3)
      IF(EOF(NF)) 100,2
    2 IF(NY.LT.800) NY=NY+1000
      IF(NY.LT.IY) GO TO 1
      IF(NY.GT.IY.AND.IFL.EQ.1) GO TO 100
      IF(NY.GT.IY) IFL=1
      IF(NY.EQ.IY) GO TO 3
      GO TO 77
C READ POTENTIALS
    3 BACKSPACE NF
      MAQA=MAQ+1
      DO 4 N=1,MAQA
      DO 4 K=1,MNS
      M=(N+1)/2
      M=M*2
      READ(NF,91) (PA(I,K,M),I=1,MEW)
   91 FORMAT(14X,6E11.5)
    4 CONTINUE
      RETURN
  100 WRITE(61,900) IY,NF
  900 FORMAT(1H1,4H****,*POTENTIALS FOR YEAR 1*,I3,* NOT FOUND ON*,
     2* FILE NO*,I3)
      STOP
      END
```

```
      SUBROUTINE PLAYMO(NYEAR,IY,NF)
      DIMENSION ISC(3),IVA(3),VA(3),MQ(4),IMQ(2)
      COMMON/LEV/PA(67,58,5),VP(67,58,2),TE(67,58,2),TS(67,58,2),
     1SC(67,58,2),DQ(67,58,2),IG(67,58)
      COMMON/PCBL/PNC(67,58,2),PCO(67,58,2)
      COMMON A,MNS,MEW,MAQ
      LEVEL 2,PA,VP,TE,TS,SC,DQ,IG
      DO 7 N=1,MAQ
      DO 7 K=1,MNS
      DO 7 I=1,MEW
      DQ(I,K,N)=0.
    7 CONTINUE
C READ ONE RECORD FROM DISCHARGE FILE
      FLAG=0
    1 READ(NF,90) (IY,NQ,K,I,(MQ(L),L=1,4))
   90 FORMAT(I3,I1,2I2,4(1X,I17))
      IF(EOF(NF)) 100,10
   10 IY=IY+1000
      IF(IY.LT.1800) IY=IY+1000
      IF(IY.GT.NYEAR) GO TO 100
      IF(IY.LT.NYEAR) GO TO 1
C FOR EACH MQ DETERMINE GRID INDEX
      FLAG=1
      IF(I.EQ.0) GO TO 1
      I=I-1
      DO 22 M=1,4
      I=I+1
      IF(I.LE.MEW) GO TO 3
      K=K+1
      IF(K.GT.MNS) GO TO 22
      I=I-MEW
    3 DQ(I,K,NQ)=0.
C SPLIT MQ INTO COMPONENTS
      MQ(M)=MQ(M)+0
      ENCODE(18,91,IMQ) MQ(M)
   91 FORMAT(I18)
      DECODE(18,92,IMQ) (MCO,(ISC(L),IVA(L),L=1,3))
   92 FORMAT(1X,I1,I1,I4,I1,I4,I1,I4,1X)
C BYPASS IF TREATMENT CODE=0
      IF(MCO.EQ.0) GO TO 2
      NN=NQ*2+1
C SCALE VARIABLES
      DO 4 L=1,3
      IS=1
      IF(ISC(L).LE.4) GO TO 5
      ISC(L)=ISC(L)-5
      IS=-1
    5 VAR=IVA(L)*IS
      EX=10**ISC(L)
      VA(L)=VAR/EX
    4 CONTINUE
C ASSIGN PRESCRIBED DISCHARGE
      VAP=VA(3)+VA(2)
      IF(MCO.GT.2) GO TO 6
      IF(MCO.EQ.1) DQ(I,K,NQ)=VA(1)
      IF(MCO.EQ.2.AND.PA(I,K,NN).GT.VAP) DQ(I,K,NQ)=VA(1)
      GO TO 2
```

```
      C ASSIGNMENT OF FREE FLOWING DISCHARGE
          6 IF(MCO.EQ.4.AND.PA(I,K,NN).LT.VAP) GO TO 2
            GRAD=PA(I,K,NQ*2+1)-VA(3)
            IF(MCO.EQ.5) GRAD=GRAD-VA(2)
            DQ(I,K,NQ)=GRAD*VA(1)
            IF(MCO.EQ.6) DQ(I,K,NQ)=AMIN1(DQ(I,K,NQ),VA(2))
            IF(MCO.EQ.7) DQ(I,K,NQ)=AMAX1(DQ(I,K,NQ),VA(2))
          2 CONTINUE
            DQ(I,K,NQ)=DQ(I,K,NQ)/1000.
            IF(GRAD) 11,12,13
         11 PNC(I,K,NQ)=0.
            PCO(I,K,NQ)=0.
            GO TO 21
         12 PNC(I,K,NQ)=0.
            PCO(I,K,NQ)=DQ(I,K,NQ)/PA(I,K,NQ*2+1)
            GO TO 21
         13 PNC(I,K,NQ)=PA(I,K,NQ*2+1)-GRAD
            PCO(I,K,NQ)=DQ(I,K,NQ)/GRAD
         21 IF(MCO.LE.2) PCO(I,K,NQ)=-1.
         22 CONTINUE
            GO TO 1
        100 CONTINUE
            IF(FLAG.EQ.1) RETURN
            WRITE(61,902) NYEAR
        902 FORMAT(1H1,24H****DISCHARGE FOR YEAR* ,I5,* NOT FOUND*)
            AX=0
            TERM=AX/AX
            RETURN
            END
```

```
      SUBROUTINE HISDIS(NF,IYY)
      COMMON/LEV/PA(67,58,5),HY(67,58,8),DQ(67,58,2)
      COMMON A,MNS,MEW,MAQ
      LEVEL 2,PA,HY,DQ
C SEARCH FOR YEAR IY
      BACKSPACE NF
      IY=IYY
      IFL=0
      IY=IY-1000
      NY=IY
   77 MUF=NY-IY+1
      IF(MUF.LT.0) MUF=0
      MUF=MUF*MAQ*MNS*(MEW/6+1)
      DO 78 N=1,MUF
      BACKSPACE NF
   78 CONTINUE
    1 READ(NF,90) NY
   90 FORMAT(I3)
      IF(EOF(NF)) 100,2
    2 IF(NY.LT.800) NY=NY+1000
      IF(NY.LT.IY) GO TO 1
      IF(NY.GT.IY.AND.IFL.EQ.1) GO TO 100
      IF(NY.EQ.IY) GO TO 3
      IFL=1
      GO TO 77
C READ DISCHARGE
    3 BACKSPACE NF
      DO 4 N=1,MAQ
      DO 4 K=1,MNS
      READ(NF,91) (DQ(I,K,N),I=1,MEW)
   91 FORMAT(14X,6E11.5)
      DO 5 I=1,MEW
      IF(DQ(I,K,N).EQ.-9999.) GO TO 5
      DQ(I,K,N)=DQ(I,K,N)/1000.
    5 CONTINUE
    4 CONTINUE
      RETURN
  100 WRITE(61,900) IY,NF
  900 FORMAT(1H1,4H****,*DISCHARGE FOR YEAR 1*,I3,* NOT FOUND ON*,
     1* FILE NO*,I3)
      STOP
      END
```

```
      SUBROUTINE DISCOP(MAQ,MNS,MEW,IDIS,NY,NF)
      DIMENSION IDIS(67,58,2),VAR(3)
      COMMON/LEV/AR(67,58)
      LEVEL 2,AR
      IF(NF.GT.7) GO TO 10
      DO 1 N=1,MAQ
      CALL SERDAT(MNS,MEW,7,NY,N,1HQ,AR,61)
      DO 1 I=1,MEW
      DO 1 K=1,MNS
      VAR(1)=AR(I,K)
      VAR(2)=VAR(3)=0.
      NCC=1
      IF(VAR(1).EQ.0) NCC=0
      CALL PLAYCO(NCC,VAR(1),VAR(2),VAR(3),IDIS(I,K,N),61)
    1 CONTINUE
      IDIS(1,1,1)=1
      RETURN
   10 READ(8,91) IY
   91 FORMAT(I3)
      IF(EOF(8)) 11,12
   11 NNY=NY
      WRITE(61,900) NNY
  900 FORMAT(1H0,29H***NO DISCHARGE DATA FOR YEAR,I5,* FOUND ON *,
     1*MANIPULATION FILE*,/1X,*RUN TERMINATED*)
      STOP
   12 IY=IY+1000
      IF(IY.LT.1800) IY=IY+1000
      IF(IY-NY) 10,13,11
   13 MM=MEW/4+1
      ICH=14000140000400000
      DO 14 NQ=1,MAQ
      DO 14 K=1,MNS
      DO 14 L=1,MM
      IA=(L-1)*4+1
      IZ=MIN0(MEW,IA+3)
      READ(8,90) (IDIS(I,K,NQ),I=IA,IZ)
   90 FORMAT(8X,4I18)
      DO 15 I=IA,IZ
      IF(IDIS(I,K,NQ).LT.ICH) IDIS(I,K,NQ)=0
   15 CONTINUE
   14 CONTINUE
      IDIS(1,1,1)=1
      RETURN
      END
```

```
      SUBROUTINE OUTPOT(NYY,NF)
      COMMON/LEV/PA(67,58,5)
      COMMON A,MNS,MEW,MAQ
      LEVEL 2,PA
      MQ=MAQ+1
      NY=NYY
      NY=NY-1000
      IF(NY.GE.1000) NY=NY-1000
      DO 1 NQ=1,MQ
      NA=NQ-1
      NAQ=(NA*2)+1
      CC=1HP
      IF(NQ.EQ.1) CC=1HW
      DO 2 K=1,MNS
      MM=MEW/6+1
      DO 3 N=1,MM
      IA=(N-1)*6+1
      IZ=MIN0(IA+5,MEW)
      WRITE(NF,90)(NY,NA,K,IA,CC,(PA(I,K,NAQ),I=IA,IZ))
   90 FORMAT(I3,1X,I1,1X,I2,1X,I2,1X,A1,1X,6E11.5)
    3 CONTINUE
    2 CONTINUE
    1 CONTINUE
      RETURN
      END
```

```
      PROGRAM HISMOQ(INPUT,OUTPUT,GEO,HYD,POT,TAPE60=INPUT,TAPE61=
     1OUTPUT,TAPE10=GEO,TAPE20=HYD,TAPE30=POT,CAP,POTHIS,MAN,DIS,
     2TAPE2=CAP,TAPE40=MAN,TAPE3=POTHIS,TAPE4=DIS,REC,TAPE5=REC)

      DIMENSION IRS(2),IRE(2),KRS(67,2),KRE(67,2),IGA(10)
      COMMON/LEV/PA(67,58,5),HY(67,58,8),DQ(67,58,2),IG(67,58)
     1,RE(67,58)
      COMMON A,MNS,MEW,MAQ
      LEVEL 2,PA,HY,DQ,IG,RE

      MNS=58
      MEW=67
      MAQ=2
      A=25000.
C READ CONTROL VARIABLES FROM FILE MANIPU
      READ(40,90) IYB,IYE,IYS,MAXI,NCAR
   90 FORMAT(1X,5I3)
      IYB=IYB+1000
      IF(IYB.LT.1800) IYB=IYB+1000
      IYE=IYE+1000
      IF(IYE.LT.1800) IYE=IYE+1000
      READ(40,91) TMF,YMAX,YMIN,CON,ORFI
   91 FORMAT(3F7.2,F5.4,F4.2)
C READ GEOMETRICAL DATA FROM FILE GEO
      CALL REDGEO(10)
      CALL RANGE(IRS,IRE,KRS,KRE)
C READ HYDRAULIC DATA FROM HYD
      CALL REDHYD(20)
      CALL REDPOT(30,IYB)
      CALL SERDAT(MNS,MEW,5,1970,2,1HR,RE,61)
C INITIALIZE TIME INTERVAL
      DT=1./YMIN
      IT=MAXI/2
C START OF YEAR BY YEAR LOOP
      TIM=IYB
      DO 1 NY=IYB,IYE,IYS
C DETERMINE DISCHARGE FOR CURRENT YEAR AND PRODUCE YEARLY OUTPUT
      CALL HISPOT(3,NY)
      CALL HISDIS(4,NY)
      IYT=NY+IYS
C HISTORIC VERSION ENFORCEMENT OF RECORDED POTENTIALS
      DO 6 M=1,MAQ
      NA=M*2
      NB=NA+1
      DO 6 K=1,MNS
      DO 6 I=1,MEW
      ENCODE(10,92,IGC) IG(I,K)
   92 FORMAT(I10)
      DECODE(10,93,IGC) (IGA(L),L=1,10)
   93 FORMAT(10I1)
      IF(IGA(M).LT.3.OR.IGA(M).GT.4) GO TO 6
      IGA(M)=3
      IF(PA(I,K,NA).LE.0.) IGA(M)=4
      IF(PA(I,K,NA).GT.0.) PA(I,K,NB)=PA(I,K,NA)-RE(I,K)
      IG(I,K)=(10*IGA(1)+IGA(2))*100000000
    6 PA(I,K,NA)=PA(I,K,NB)
C START TIME INTERVAL LOOP
```

```
    2 IF(IT.LT.MAXI/4) DT=DT*TMF
      IF(IT.GT.MAXI/4*3) DT=DT/TMF
      DT=AMAX1(1./YMIN,DT)
      DT=AMIN1(1./YMAX,DT)
      DTN=AMIN1(DT,IYT-TIM)
      DTN=ABS(DTN)
      WRITE(61,909) TIM,DTN
  909 FORMAT(1H0,*ELAPSED TIME*,F8.3,* TIME INTERVAL*,F8.4)
      DTS=DTN*365.*86400.
C RE-INITIALIZE STARTING POTENTIALS
      DO 3 M=1,MAQ
      NA=M*2
      NB=NA+1
      DO 4 K=1,MNS
      DO 5 I=1,MEW
      PA(I,K,NA)=PA(I,K,NB)
    5 CONTINUE
    4 CONTINUE
    3 CONTINUE
C ADVANCE MODEL BY TIME STEP DT
      CALL TRANSI(MAXI,IT,DTS,ORFI,CON,NCAR,IRS,IRE,KRS,KRE)
      TIM=TIM+DTN
      IF(TIM.LT.IYT) GO TO 2
C END TIME INTERVAL LOOP
      CALL OUTPOT(IYT,2)
    1 CONTINUE
      END
```

CARD NR.	SEVERITY	DETAILS	DIAGNOSIS OF PROBLEM
23	I	23 CD 23	FIELD WIDTH OF A CONVERSION DESCRIPTOR SHOULD BE AS LARGE AS THE MINIMUM SPECIFIED FOR THAT DESCRIPTOR.

```
      PROGRAM RUNMOD(INPUT,OUTPUT,GEO,HYD,POT,MAN,TAPE60=INPUT,TAPE61=
     1OUTPUT,TAPE10=GEO,TAPE20=HYD,TAPE30=POT,TAPE40=MAN,DIS,CAP,
     2TAPE1=DIS,TAPE2=CAP,DEBUG=OUTPUT)
      DIMENSION IRS(2),IRE(2),KRS(67,2),KRE(67,2)
      COMMON/LEV/PA(67,58,5),HY(67,58,8),DQ(67,58,2),IG(67,58)
      COMMON/PCBL/PNC(67,58,2),PCO(67,58,2)
      COMMON A,MNS,MEW,MAQ
      LEVEL 2,PA,HY,DQ,IG
      MNS=58
      MEW=67
      MAQ=2
      A=25000.
C READ CONTROL VARIABLES FROM FILE MANIPU
      READ(40,90) IYB,IYE,IYS,MAXI,NCAR
   90 FORMAT(1X,5I3)
      IYB=IYB+1000
      IF(IYB.LT.1800) IYB=IYB+1000
      IYE=IYE+1000
      IF(IYE.LT.1800) IYE=IYE+1000
      READ(40,91) TMF,YMAX,YMIN,CON,ORFI
   91 FORMAT(3F7.2,F5.4,F4.2)
C READ GEOMETRICAL DATA FROM FILE GEO
      CALL REDGEO(10)
      CALL RANGE(IRS,IRE,KRS,KRE)
C READ HYDRAULIC DATA FROM HYD
      CALL REDHYD(20)
C SEARCH FOR POTENTIALS OF YEAR IYB AND READ
      CALL REDPOT(30,IYB)
C INITIALIZE TIME INTERVAL
      DT=1./YMIN
      IT=MAXI/2
C START OF YEAR BY YEAR LOOP
      TIM=IYB
      DO 1 NY=IYB,IYE,IYS
C DETERMINE DISCHARGE FOR CURRENT YEAR AND PRODUCE YEARLY OUTPUT
      CALL PLAYMO(NY,IY,40)
      IYT=NY+IYS
C START TIME INTERVAL LOOP
    2 IF(IT.LT.MAXI/4) DT=DT*TMF
      IF(IT.GT.MAXI/4*3) DT=DT/TMF
      DT=AMAX1(1./YMIN,DT)
      DT=AMIN1(1./YMAX,DT)
      DTN=AMIN1(DT,IYT-TIM)
      DTN=ABS(DTN)
      WRITE(61,909) TIM,DTN
  909 FORMAT(1H0,*ELAPSED TIME*,F9.3,* TIME INTERVAL*,F8.4)
      DTS=DTN*365.*86400.
C RE-INITIALIZE STARTING POTENTIALS
      DO 3 M=1,MAQ
      NA=M*2
      NB=NA+1
      DO 4 K=1,MNS
      DO 5 I=1,MEW
      PA(I,K,NA)=PA(I,K,NB)
      IF(PCO(I,K,M).EQ.-1.) GO TO 5
      DQ(I,K,M)=(PA(I,K,NB)-PNC(I,K,M))*PCO(I,K,M)
      DQ(I,K,M)=AMAX1(DQ(I,K,M),0.)
```

```
    5 CONTINUE
    4 CONTINUE
    3 CONTINUE
C ADVANCE MODEL BY TIME STEP DT
      CALL TRANSI(MAXI,IT,DTS,ORFI,CON,NCAR,IRS,IRE,KRS,KRE)
      TIM=TIM+DTN
      IF(TIM.LT.IYT) GO TO 2
C END TIME INTERVAL LOOP
      CALL OUTPOT(IYT,2)
      MM=MEW/6+1
      CC=1HQ
      DO 6 NA=1,MAQ
      DO 6 K=1,MNS
      DO 6 N=1,MM
      IA=(N-1)*6+1
      IZ=MIN0(IA+5,MEW)
      DO 7 I=IA,IZ
      DQ(I,K,NA)=DQ(I,K,NA)*1000.
    7 CONTINUE
      NYJ=NY-1000
      IF(NYJ.GE.1000.) NYJ=NYJ-1000
      WRITE(1,900) (NYJ,NA,K,IA,CC,(DQ(I,K,NA),I=IA,IZ))
  900 FORMAT(I3,1X,I1,2(1X,I2),1X,A1,1X,6E11.5)
    6 CONTINUE
    1 CONTINUE
      END
```

CARD NR.	SEVERITY	DETAILS	DIAGNOSIS OF PROBLEM
21	I	23 CD 21	FIELD WIDTH OF A CONVERSION DESCRIPTOR SHOULD BE AS LARGE AS THE MINIMUM SPECIFIED FOR THAT DESCRIPTOR.

```
      SUBROUTINE STEADY(MAXI,IT,ORFI,CON,NCAR,IRS,IRE,KRS,KRE)
      DIMENSION IGA(10),IRS(2),IRE(2),KRS(67,2),KRE(67,2)
      COMMON/LEV/PA(67,58,5),VP(67,58,2),TE(67,58,2),TS(67,58,2),
     1SC(67,58,2),DQ(67,58,2),IG(67,58)
      COMMON A,MNS,MEW,MAQ
      LEVEL 2,PA,VP,TE,TS,SC,DQ,IG
C GENERAL INITIALIZATIONS
      ORF=ORFI
      VMIN=0.000001**3
C MAIN LOOP
      DO 1 IT=1,MAXI
C INITIALIZE ERROR SUMS AND EXTREMES
      SUM=EMAX=0.
      NSU=0
C START OF INNER LOOPS
      DO 2 N=1,MAQ
      NA=N
      IST=IRS(N)
      IEN=IRE(N)
      DO 3 I=IST,IEN
      KST=KRS(I,N)
      KEN=KRE(I,N)
      DO 4 K=KST,KEN
C DECODE GEOMETRICAL DATA AND DETERMINE ADJOINING AQUIFERS
      ENCODE(10,90,IGC) IG(I,K)
   90 FORMAT(I10)
   91 FORMAT(10I1)
      DECODE(10,91,IGC) (IGA(L),L=1,10)
      IF(IGA(N).LE.3) GO TO 4
      NPD=NA
      NN=NA-1
      NPU=0
      IF(NN.LT.1) GO TO 10
      DO 5 L=1,NN
      IF(IGA(L).GT.1) NPU=L
    5 CONTINUE
   10 CONTINUE
      DO 6 L=N,10
      IF(IGA(L).GT.1.AND.NPD.EQ.NA) NPD=L
    6 CONTINUE
C ASSIGN POINT POTENTIALS
    7 NP=NA*2+1
      H=PA(I,K,NP)
      HI=PA(I,K,NP-1)
      HN=PA(I,K+1,NP)
      HS=PA(I,K-1,NP)
      HE=PA(I+1,K,NP)
      HW=PA(I-1,K,NP)
      HU=PA(I,K,NPU*2+1)
      HD=PA(I,K,NPD*2+1)
C ASSIGN HYDRAULIC PARAMETERS
      DN=TS(I,K,NA)
      DS=TS(I,K-1,NA)
      DE=TE(I,K,NA)
      DW=TE(I-1,K,NA)
      DU=VP(I,K,NA)
      DD=VP(I,K,NPD)
```

```
C APPLY FINITE DIFFERENCE EQUATION
C APPLY FINITE DIFFERENCE EQUATION
      SH=HN*DN+HS*DS+HE*DE+HW*DW+HU*DU+HD*DD
      ST=DN+DS+DE+DW+DU+DD
      SD=0.
      IF(ST.LE.0.) GO TO 4
      DEL=SH/ST-H
      PA(I,K,NP)=H+ORF*DEL
C CALCULATE CONVERGENCE CHECKS
      ER=ABS(DEL)
      SUM=SUM+ER
      EMAX=AMAX1(EMAX,ER)
      NSU=NSU+1
    4 CONTINUE
    3 CONTINUE
    2 CONTINUE
C END OF INNER LOOPS
C BYPASS CALL CARRE IF NCAR=0
      IF(NCAR.LE.0) GO TO 8
      CALL CARRE(ORF,SUM,IT,NCAR,MAXI)
C CHECK FOR CONVERGENCE
    8 CHE=SUM/NSU
      IF(CHE.LT.CON) GO TO 9
    1 CONTINUE
    9 WRITE(61,900) IT,CHE,EMAX,ORF,ORFI
  900 FORMAT(1H0,*CONVERGED AFTER*,I4,* ITERATIONS*,* ERROR AVER.*,
     1F7.3,* MAX.*,F8.3,/1X,*OVER RELAXED BY*,F7.3,* RESET TO*,F7.3)
      ORF=ORFI
      RETURN
      END
```

```
      SUBROUTINE REDHYD(NF)
      COMMON/LEV/PA(67,58,5),HY(67,58,8),DQ(67,58,2),IG(67,58)
      COMMON A,MNS,MEW,MAQ
      LEVEL 2,PA,HY,DQ,IG
C READ HYDRAULIC DATA
      NA=MAQ*4
      DO 1 N=1,NA
      DO 2 K=1,MNS
      READ(NF,90) (HY(I,K,N),I=1,MEW)
   90 FORMAT(14X,6E11.5)
    2 CONTINUE
    1 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE REDGEO(NF)
      COMMON/LEV/PA(67,58,5),VP(67,58,2),TE(67,58,2),TS(67,58,2),
     1SC(67,58,2),DQ(67,58,2),IG(67,58)
      COMMON A,MNS,MEW,MAQ
      LEVEL 2,PA,VP,TE,TS,SC,DQ,IG
C READ GEOMETRICAL DATA
      DO 1 K=1,MNS
      READ(NF,90) (IG(I,K),I=1,MEW)
   90 FORMAT(10X,7I10)
    1 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE TRANSI(MAXI,IT,DT,ORFI,CON,NCAR,IRS,IRE,KRS,KRE)
      DIMENSION IGA(10),IRS(2),IRE(2),KRS(67,2),KRE(67,2)
      COMMON/LEV/PA(67,58,5),VP(67,58,2),TE(67,58,2),TS(67,58,2)
     1,SC(67,58,2),DQ(67,58,2),IG(67,58)
      COMMON A,MNS,MEW,MAQ
      LEVEL 2,PA,VP,TE,TS,SC,DQ,IG
C GENERAL INITIALIZATIONS
      ORF=ORFI
      VMIN=0.000001**3
C MAIN LOOP
      DO 1 IT=1,MAXI
C INITIALIZE ERROR SUMS AND EXTREMES
      SUM=EMAX=0.
      NSU=0
C START OF INNER LOOPS
      DO 2 N=1,MAQ
      NA=N
      IST=IRS(N)
      IEN=IRE(N)
      DO 3 I=IST,IEN
      KST=KRS(I,N)
      KEN=KRE(I,N)
      DO 4 K=KST,KEN
C DECODE GEOMETRICAL DATA AND DETERMINE ADJOINING AQUIFERS
      ENCODE(10,90,IGC) IG(I,K)
   90 FORMAT(I10)
   91 FORMAT(10I1)
      DECODE(10,91,IGC) (IGA(L),L=1,10)
      IF(IGA(N).LE.3) GO TO 4
      NPD=NA
      NN=NA-1
      NPU=0
      IF(NN.LT.1) GO TO 10
      DO 5 L=1,NN
      IF(IGA(L).GT.1) NPU=L
    5 CONTINUE
   10 CONTINUE
      DO 6 L=N,10
      IF(IGA(L).GT.1.AND.NPD.EQ.NA) NPD=L
    6 CONTINUE
C ASSIGN POINT POTENTIALS
    7 NP=NA*2+1
      H=PA(I,K,NP)
      HI=PA(I,K,NP-1)
      HN=PA(I,K+1,NP)
      HS=PA(I,K-1,NP)
      HE=PA(I+1,K,NP)
      HW=PA(I-1,K,NP)
      HU=PA(I,K,NPU*2+1)
      HD=PA(I,K,NPD*2+1)
C ASSIGN HYDRAULIC PARAMETERS
      DN=TS(I,K,NA)
      DS=TS(I,K-1,NA)
      DE=TE(I,K,NA)
      DW=TE(I-1,K,NA)
      DU=VP(I,K,NA)
      DD=VP(I,K,NPD)
```

```
C APPLY FINITE DIFFERENCE EQUATION
C APPLY FINITE DIFFERENCE EQUATION
      SH=HN*DN+HS*DS+HE*DE+HW*DW+HU*DU+HD*DD
      ST=DN+DS+DE+DW+DU+DD
      SD=SC(I,K,NA)*A*A/DT
      DEL=(SH+HI*SD-DQ(I,K,NA))/(SD+ST)-H
      PA(I,K,NP)=H+ORF*DEL
C CALCULATE CONVERGENCE CHECKS
      ER=ABS(DEL)
      SUM=SUM+ER
      EMAX=AMAX1(EMAX,ER)
      NSU=NSU+1
    4 CONTINUE
    3 CONTINUE
    2 CONTINUE
C END OF INNER LOOPS
C BYPASS CALL CARRE IF NCAR=0
      IF(NCAR.LE.0) GO TO 8
      CALL CARRE(ORF,SUM,IT,NCAR,MAXI)
C CHECK FOR CONVERGENCE
    8 CHE=SUM/NSU
      IF(CHE.LT.CON) GO TO 9
    1 CONTINUE
    9 WRITE(61,900) IT,CHE,EMAX,ORF,ORFI
  900 FORMAT(1H0,*CONVERGED AFTER*,I4,* ITERATIONS*,* ERROR AVER.*,
     1F7.3,* MAX.*,F8.3,/1X,*OVER RELAXED BY*,F7.3,* RESET TO*,F7.3)
      ORF=ORFI
      RETURN
      END
```

```
      SUBROUTINE RANGE(IRS,IRE,KRS,KRE)
      DIMENSION IRS(2),IRE(2),KRS(67,2),KRE(67,2),IGA(10)
      COMMON/LEV/PA(67,58,5),HY(67,58,8),DQ(67,58,2),IG(67,58)
      COMMON A,MNS,MEW,MAQ
      LEVEL 2,PA,HY,DQ,IG
C INITIALIZE
      DO 1 N=1,MAQ
      IRS(N)=1
      IRE(N)=MEW
      DO 1 I=1,MEW
      KRS(I,N)=1
      KRE(I,N)=MNS
    1 CONTINUE
C DETERMINE RANGE K FOR EACH I
      DO 2 N=1,MAQ
      DO 2 I=1,MEW
      DO 2 K=1,MNS
      ENCODE(10,90,IGC) IG(I,K)
      DECODE(10,91,IGC) (IGA(L),L=1,10)
   90 FORMAT(I10)
   91 FORMAT(10I1)
      IF(IGA(N).LE.2) GO TO 2
      IF(KRE(I,N).EQ.MNS) KRS(I,N)=K
      KRE(I,N)=K
    2 CONTINUE
C DETERMINE RANGE I
      DO 3 N=1,MAQ
      DO 3 I=1,MEW
      IF(KRS(I,N).EQ.1) GO TO 3
      IF(IRE(N).EQ.MEW) IRS(N)=I
      IRE(N)=I
    3 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE ERANG(IRS,IRE,KRS,KRE)
      DIMENSION IRS(2),IRE(2),KRS(67,2),KRE(67,2),IGA(10)
      COMMON/LEV/PA(67,58,5),HY(67,58,8),DQ(67,58,2),IG(67,58)
      COMMON A,MNS,MEW,MAQ
      LEVEL 2,PA,HY,DQ,IG
C INITIALIZE
      DO 1 N=1,MAQ
      IRS(N)=1
      IRE(N)=MEW
      DO 1 I=1,MEW
      KRS(I,N)=1
      KRE(I,N)=MNS
    1 CONTINUE
C DETERMINE RANGE K FOR EACH I
      DO 2 N=1,MAQ
      DO 2 I=1,MEW
      DO 2 K=1,MNS
      ENCODE(10,90,IGC) IG(I,K)
      DECODE(10,91,IGC) (IGA(L),L=1,10)
   90 FORMAT(I10)
   91 FORMAT(10I1)
      IF(IGA(N).LE.1) GO TO 2
      IF(KRE(I,N).EQ.MNS) KRS(I,N)=K
      KRE(I,N)=K
    2 CONTINUE
C DETERMINE RANGE I
      DO 3 N=1,MAQ
      DO 3 I=1,MEW
      IF(KRS(I,N).EQ.1) GO TO 3
      IF(IRE(N).EQ.MEW) IRS(N)=I
      IRE(N)=I
    3 CONTINUE
      RETURN
      END
```

```
      PROGRAM STOCAL(INPUT,OUTPUT,GEO,OLD,POT,MAN,NEW,TAPE60=INPUT,
     1TAPE61=OUTPUT,TAPE1=GEO,TAPE2=OLD,TAPE3=POT,TAPE4=MAN,TAPE5=NEW,
     2THI,TAPE6=THI)
      DIMENSION TBAL(2),TDRA(2),TSPE(2),BAL(2,100),DRA(2,100),
     1SPE(2,100)
      COMMON/LEV/PA(67,58,5),HY(67,58,6),SC(67,58,2),DQ(67,58,2),
     1IG(67,58)
      COMMON A,MNS,MEW,MAQ
      LEVEL 2,PA,HY,SC,DQ,IG
      A=25000
      MNS=58
      MEW=67
      MAQ=2
C INITIALIZE TOTALS
      DO 1 N=1,MAQ
      TBAL(N)=TDRA(N)=TSPE(N)=0.
      DO 1 L=1,100
      BAL(N,L)=DRA(N,L)=SPE(N,L)=0.
    1 CONTINUE
C READ TIME STEPS AND MODEL PARAMETERS
      READ(60,90) NYST,NYEN,NSTP
      NSTEP=NSTP
      IF(NYST.EQ.NYEN) NSTEP=0
      NSTP=MAX0(NSTP,1)
   90 FORMAT(1X,I3,2X,I3,1X,I2)
      READ(60,94) NA
   94 FORMAT(I1)
      READ(60,92) FBA
   92 FORMAT(A3)
      READ(60,93) PCT
   93 FORMAT(F5.2)
      NYEN=NYEN-NSTP
      READ(60,95) TSPE(1),TSPE(2)
   95 FORMAT(F15.5)
      CALL REDGEO(1)
      CALL REDHYD(2)
      IF(FBA.EQ.3HSPE) GO TO 202
C START OF TIME STEP LOOP
      NTL=0
      DO 100 NY=NYST,NYEN,NSTP
      NYE=NY+NSTEP
      NTL=NTL+1
C DETERMINE LEAKAGE BY AQUIFER
      IF(NA.EQ.2) GO TO 101
      CALL SLEAKA(1,NY,NYE,3,BALA)
      TBAL(1)=TBAL(1)+BALA
      BAL(1,NTL)=BAL(1,NTL)+BALA
  101 CONTINUE
      IF(NA.EQ.1) GO TO 102
      CALL SLEAKA(2,NY,NYE,3,BALA)
      TBAL(2)=TBAL(2)+BALA
      BAL(2,NTL)=BAL(2,NTL)+BALA
C DETERMINE BOUNDARY FLOWS BY AQUIFER
  102 CONTINUE
      IF(NA.EQ.2) GO TO 103
      CALL SBOUND(1,NY,NYE,3,BALA)
      BAL(1,NTL)=BAL(1,NTL)+BALA
```

```
      TBAL(1)=TBAL(1)+BALA
  103 CONTINUE
      IF(NA.EQ.1) GO TO 104
      CALL SBOUND(2,NY,NYE,3,BALA)
      BAL(2,NTL)=BAL(2,NTL)+BALA
      TBAL(2)=TBAL(2)+BALA
C DETERMINE ARTIFICIAL DISCHARGE BY AQUIFER
  104 CONTINUE
      IF(NA.EQ.2) GO TO 105
      CALL STOTAQ(1,NY,NYE,4,BALA)
      BAL(1,NTL)=BAL(1,NTL)-BALA
      TBAL(1)=TBAL(1)-BALA
  105 CONTINUE
      IF(NA.EQ.1) GO TO 106
      CALL STOTAQ(2,NY,NYE,4,BALA)
      BAL(2,NTL)=BAL(2,NTL)-BALA
      TBAL(2)=TBAL(2)-BALA
C DETERMINE EFFECTIVE DRAWDOWN VOLUME, SPECIFIC STORAGE FOR TIME STEP
  106 CONTINUE
      IF(NA.EQ.2) GO TO 107
      CALL EFDRAW(1,NY,NYE,3,6,DRAW)
      DRA(1,NTL)=DRAW
      TDRA(1)=TDRA(1)+DRAW
      SPE(1,NTL)=BAL(1,NTL)/DRAW*(-1)
  107 CONTINUE
      IF(NA.EQ.1) GO TO 108
      CALL EFDRAW(2,NY,NYE,3,6,DRAW)
      DRA(2,NTL)=DRAW
      TDRA(2)=TDRA(2)+DRAW
      SPE(2,NTL)=BAL(2,NTL)/DRAW*(-1)
  108 CONTINUE
  100 CONTINUE
      IF(TDRA(1).NE.0.) TSPE(1)=TBAL(1)/TDRA(1)*(-1)
      IF(TDRA(2).NE.0.) TSPE(2)=TBAL(2)/TDRA(2)*(-1)
C TABULATED OUTPUT
      NYEND=NYEN+NSTP
      WRITE(61,901) NYST,NYEND
  901 FORMAT(1H1,20X,*CALCULATION OF SPECIFIC STORAGE COEF*,* FOR 1*,
     1I3,* TO 1*,I3,/21X,10(7H-------))
      WRITE(61,902)
  902 FORMAT(1H0,24X,*A Q U I F E R  1*,33X,*A Q U I F E R  2*)
      WRITE(61,903)
  903 FORMAT(1X,* INTERVAL*,5X,*FLOW-BAL (M3)*,5X,*DRAWDOWN (M4)*,5X,
     1*SPEC.ST.*,5X,*FLOW-BAL (M3)*,5X,*DRAWDOWN (M4)*,5X,*SPEC.ST.*,
     2/1X,108(1H.))
      NLT=0
      DO 200 NY=NYST,NYEN,NSTP
      NYE=NY+NSTP
      NLT=NLT+1
      WRITE(61,904) NY,NYE,BAL(1,NLT),DRA(1,NLT),SPE(1,NLT),BAL(2,NLT),
     1DRA(2,NLT),SPE(2,NLT)
  904 FORMAT(1X,2H 1,I3,2H-1,I3,2(5X,E13.6,5X,E13.6,3X,F10.8))
  200 CONTINUE
      WRITE(61,905)
  905 FORMAT(1X,108(1H-))
      NYEN=NYEN+NSTP
      WRITE(61,904) NYST,NYEN,TBAL(1),TDRA(1),TSPE(1),TBAL(2),TDRA(2),
```

```
     1TSPE(2)
      IA=1
      IB=2
      IF(NA.EQ.1) IB=1
      IF(NA.EQ.2) IA=2
      DO 203 IC=IA,IB
      FAC=PCT*BALA/100./(BAL(IB,NLT)+BALA)
      WRITE(61,906) IC,FAC,PCT
  906 FORMAT(1H0,*FINAL PERIOD SCALE FACTOR FOR AQ.*,I2,* IS*,F7.3,
     1* FOR*,F7.2,*  PERC.BAL.*)
  203 CONTINUE
      IF(FBA.EQ.3HINF) GO TO 400
      IF(FBA.EQ.3HBAL) GO TO 202
      DO 201 NN=1,5,2
      DO 201 IC=IA,IB
      N=NN-1+IC
      DO 201 I=1,MEW
      DO 201 K=1,MNS
      IF(HY(I,K,N).EQ.-9999.) GO TO 201
      HY(I,K,N)=HY(I,K,N)*FAC
  201 CONTINUE
      WRITE(61,909)
  909 FORMAT(1H0,*NEW HYDR. PARAMETERS*)
      GO TO 302
  202 CONTINUE
      IA=1
      IB=2
      IF(NA.EQ.1) IB=1
      IF(NA.EQ.2) IA=2
C CALCULATE STORAGE COEFFICIENT
      REWIND 6
      DO 300 N=1,MAQ
      DO 300 K=1,MNS
      READ(6,91) (DQ(I,K,N),I=1,MEW)
   91 FORMAT(14X,6E11.5)
  300 CONTINUE
      DO 301 N=IA,IB
      DO 301 K=1,MNS
      DO 301 I=1,MEW
      SC(I,K,N)=DQ(I,K,N)*TSPE(N)
      IF(PA(I,K,N*2+1).EQ.-9999.) SC(I,K,N)=-9999.
  301 CONTINUE
      WRITE(61,908)
  908 FORMAT(1H0,*NEW STORAGE COEFFICIENT*)
  302 CONTINUE
      CALL OUTHYD(5,880)
  400 CONTINUE
      END
```

```
      PROGRAM VERTAD(INPUT,OUTPUT,GEO,OLD,POT,MAN,NEW,TAPE60=INPUT,
     1TAPE61=OUTPUT,TAPE1=GEO,TAPE2=OLD,TAPE3=POT,TAPE4=MAN,TAPE5=NEW,
     2NFY,TAPE6=NFY)
      DIMENSION IRS(2),IRE(2),KRS(67,2),KRE(67,2)
      DIMENSION TP(10)
      COMMON/LEV/PA(67,58,5),HY(67,58,8),DQ(67,58,2),IG(67,58),
     1AR(67,58)
      COMMON A,MNS,MEW,MAQ
      LEVEL 2,PA,HY,DQ,IG,AR
      A=25000.
      MEW=67
      MNS=58
      MAQ=2
C READ CONTROL DATA
      READ(60,90) IYST,IYEN
   90 FORMAT(1X,I3,2X,I3)
      READ(60,91) NA
      READ(60,91) MODE
   91 FORMAT(I1)
      TP(1)=10HRECALCULAT
      TP(2)=10HED BALANCE
      TP(3)=10H AQUIFER
      ENCODE(10,92,TPC) NA
   92 FORMAT(I2,8X)
      DECODE(10,93,TPC) TP(4)
   93 FORMAT(A10)
   99 FORMAT(A1)
C READ DATA
      CALL REDGEO(1)
      CALL ERANG(IRS,IRE,KRS,KRE)
      CALL REDPOT(3,IYST)
      CALL REDHYD(2)
C INITIALIZE POTENTIALS, DISCHARGES, AVERAGE OF VP
      OAVP=0.
      NAVP=0
      DO 1 N=1,MAQ
      DO 1 I=1,MEW
      DO 1 K=1,MNS
      NQ=N*2
      NP=NQ+1
      DQ(I,K,N)=0.
      PA(I,K,NQ)=PA(I,K,NP)
      IF(N.NE.NA) GO TO 1
      IF(HY(I,K,N).LE.0.) GO TO 1
      OAVP=OAVP+HY(I,K,N)
      NAVP=NAVP+1
    1 CONTINUE
      OAVP=OAVP/NAVP
      RAVP=0.
      NAVP=0
      NSTP=1
      CALL HISDIS(4,IYEN)
      IF(IYST.EQ.IYEN) GO TO 2
      NSTP=MAX0(1,IYEN-IYST)
      CALL SERDAT(MNS,MEW,3,IYEN,NA,1HP,AR,61)
      DO 3 I=1,MEW
      DO 3 K=1,MNS
```

```
      PA(I,K,NA*2+1)=AR(I,K)
      AR(I,K)=-9999.
    3 CONTINUE
    2 CONTINUE
      CALL EQUAL(IYST,IYEN,6)
C INITIALIZE STATISTICAL SUMS
      NSUM=0
      SOLB=SOLB2=SNEW=SNEW2=0.
C NODAL LOOP
      IST=IRS(NA)
      IEN=IRE(NA)
      DO 10 I=IST,IEN
      KST=KRS(I,NA)
      KEN=KRE(I,NA)
      DO 10 K=KST,KEN
      AR(I,K)=0.
      CALL VERCAL(I,K,NA,NSTP,OLB,WEB,MODE)
      IF(OLB.EQ.0..AND.WEB.EQ.0.) GO TO 10
      AR(I,K)=WEB
      SOLB=SOLB+OLB
      SOLB2=SOLB2+OLB**2
      SNEW=SNEW+WEB
      SNEW2=SNEW2+WEB**2
      NSUM=NSUM+1
      RAVP=RAVP+HY(I,K,NA)
      NAVP=NAVP+1
   10 CONTINUE
C CALCULATE STANDARD DEVS AND PRINT
      SIGO=SQRT(SOLB2/NSUM)
      SIGN=SQRT(SNEW2/NSUM)
      RAVP=RAVP/NAVP
      RAT=RAVP/OAVP
      WRITE(61,900) NA
  900 FORMAT(1H1,20X,*AQUIFER*,I3,/1X,20X,10(1H-))
      WRITE(61,903) IYST,IYEN
  903 FORMAT(1H0,*NEW VERT. PERMEAB. FOR PERIOD 1*,I3,* TO 1*,I3)
      WRITE(61,905) MODE
  905 FORMAT(1X,*MODE*,I3)
      WRITE(61,901) SOLB,SIGO
  901 FORMAT(1H0,*OLD BALANCE*,E12.6,* STAND.DEV.*,E12.6)
  902 FORMAT(1H ,*NEW BLALANCE*,E12.6,* STAND.DEV.*,E12.6)
      WRITE(61,902) SNEW,SIGN
      WRITE(61,904) RAT
  904 FORMAT(1H0,*RATIO NEW/OLD VP *,E13.6)
      CALL SMAMAP(MEW,MNS,61,AR,1HN,TP,4)
C WRITE NEW HYDRAULICS FILE
      CALL OUTHYD(5,IYEN)
      END
```

```
      SUBROUTINE VERCAL(IN,KN,NN,NSTP,OLB,WEB,MODE)
      DIMENSION IGA(10)
      COMMON/LEV/PA(67,58,5),VP(67,58,2),TE(67,58,2),TS(67,58,2),
     1SC(67,58,2),DQ(67,58,2),IG(67,58)
      COMMON A,MNS,MEW,MAQ
      LEVEL 2,PA,VP,TE,TS,SC,DQ,IG
      STP=NSTP*365.*86400.
      EPSI=0.0000001**2
C DETERMINE GEOMETRICAL STATUS
   90 FORMAT(I10)
   91 FORMAT(10I1)
      OLB=WEB=0.
      ENCODE(10,90,IGC) IG(IN,KN)
      DECODE(10,91,IGC)(IGA(L),L=1,10)
      IF(IGA(NN).LE.2) RETURN
C DETERMINE ADJOINING AQUIFER NO.
      NPD=NN
      NNP=NN-1
      NPU=0
      IF(NNP.LT.1) GO TO 2
      DO 1 L=1,NNP
      IF(IGA(L).GT.1) NPU=L
    1 CONTINUE
    2 CONTINUE
      DO 4 L=NN,10
      IF(IGA(L).GT.1.AND.NPD.EQ.NN) NPD=L
    4 CONTINUE
C ORIGINAL FLOWBALANCE
      H=PA(IN,KN,NN*2+1)
      FN=TS(IN,KN,NN)*(PA(IN,KN+1,NN*2+1)-H)
      FS=TS(IN,KN-1,NN)*(PA(IN,KN-1,NN*2+1)-H)
      FE=TE(IN,KN,NN)*(PA(IN+1,KN,NN*2+1)-H)
      FW=TE(IN-1,KN,NN)*(PA(IN-1,KN,NN*2+1)-H)
      FU=VP(IN,KN,NN)*(PA(IN,KN,NPU*2+1)-H)
      FD=VP(IN,KN,NPD)*(PA(IN,KN,NPD*2+1)-H)
      FQ=-DQ(IN,KN,NN)
      ST=SC(IN,KN,NN)*(PA(IN,KN,NN*2)-H)/STP*A**2
      OLB=FN+FS+FE+FW+FU+FD+FQ+ST
      WEB=OLB
C DETERMINE VERTICAL FLOW CORRECTION
      DC=PA(IN,KN,NPU*2+1)-H
      IF(ABS(DC).LT.EPSI) RETURN
      TC=VP(IN,KN,NN)
      FC=DC*TC
      COR=1.0
      FN=FC-OLB
      IF(ABS(FC).GT.EPSI) COR=FN/FC
      TN=AMAX1(EPSI,TC*COR)
      IF(MODE.EQ.1) TN=AMAX1(TN,TC*0.5)
      IF(MODE.EQ.3) TN=TC
      IF(MODE.EQ.1)TN=AMIN1(TN,TC*2.)
      IF(MODE.EQ.0) TN=TC
C LIMIT DOWN LEAKAGE
      IF(MODE.NE.0) GO TO 3
      IF(DC.LE.0.) GO TO 3
      TN=EPSI
      WRITE(61,900) IN,KN
```

```
  900 FORMAT(1X,20H***DOWN LEAKAGE NODE,2I3,* SET TO MINIM.*)
    3 CONTINUE
      FN=DC*TN
      WEB=OLB-FC+FN
      IF(ABS(WEB).LT.EPSI) WEB=0.
      VP(IN,KN,NN)=TN
      RETURN
      END
```

```
      SUBROUTINE NODBAL(IP,KP,NP,IS,KS,NS,LFL,SBC,SBN,TC,TN ,SBS,SBT,
     1NSTP)
      DIMENSION IGP(10),IGS(10)
      COMMON/LEV/PA(67,58,5),VP(67,58,2),TE(67,58,2),TS(67,58,2),
     1SC(67,58,2),DQ(67,58,2),IG(67,58)
      COMMON A,MNS,MEW,MAQ,ORF,AVER,SIG
      LEVEL 2,PA,VP,TE,TS,SC,DQ,IG
      STP=NSTP*365.*86400.
      SIG=0.
      SBC=SBN=0.
      SBS=SBT=0.
      TC=TN=0.
      EPSI=0.0000001**2
  899 FORMAT(1X,3I3)
C DETERMINE GEOMETRICAL STATUS EXCLUDE OUTSIDE NODES
   90 FORMAT(I10)
   91 FORMAT(10I1)
      ENCODE(10,90,IGC) IG(IP,KP)
      DECODE(10,91,IGC) (IGP(L),L=1,10)
      ENCODE(10,90,IGC) IG(IS,KS)
      DECODE(10,91,IGC) (IGS(L),L=1,10)
      HP=PA(IP,KP,NP*2+1)
      HS=PA(IS,KS,NS*2+1)
      DC=HS-HP
C DETERMINE ADJOINING AQUIFER NOS
      NPD=NP
      NNP=NP-1
      NPU=0
      IF(NNP.LT.1) GO TO 12
      DO 11 L=1,NNP
      IF(IGP(L).GT.1) NPU=L
   11 CONTINUE
   12 CONTINUE
      DO 13 L=NP,10
      IF(IGP(L).GT.1.AND.NPD.EQ.NP) NPD=L
   13 CONTINUE
      NSD=NS
      NNS=NS-1
      NSU=0
      IF(NNS.LT.1) GO TO 22
      DO 21 L=1,NNS
      IF(IGS(L).GT.1) NSU=L
   21 CONTINUE
   22 CONTINUE
      DO 23 L=NS,10
      IF(IGS(L).GT.1.AND.NSD.EQ.NS) NSD=L
   23 CONTINUE
C DETERMINE FLOW BALANCE PRINCIPAL NODE
      FN=TS(IP,KP,NP)*(PA(IP,KP+1,NP*2+1)-HP)
      IF(PA(IP,KP+1,NP*2+1).EQ.-9999.) FN=0.
      FS=TS(IP,KP-1,NP)*(PA(IP,KP-1,NP*2+1)-HP)
      IF(PA(IP,KP-1,NP*2+1).EQ.-9999.) FS=0.
      FE=TE(IP,KP,NP)*(PA(IP+1,KP,NP*2+1)-HP)
      IF(PA(IP+1,KP,NP*2+1).EQ.-9999.) FE=0.
      FW=TE(IP-1,KP,NP)*(PA(IP-1,KP,NP*2+1)-HP)
      IF(PA(IP-1,KP,NP*2+1).EQ.-9999.) FW=0.
      FU=VP(IP,KP,NP)*(PA(IP,KP,NPU*2+1)-HP)
```

```
      FD=VP(IP,KP,NPD)*(PA(IP,KP,NPD*2+1)-HP)
      FQ=-DQ(IP,KP,NP)
      ST=SC(IP,KP,NP)*(PA(IP,KP,NP*2)-HP)/STP*A**2
      SBC=FN+FS+FE+FW+FU+FD+FQ+ST
C DETERMINE FLOWBALANCE SECONDARY NODE
      FN=TS(IS,KS,NS)*(PA(IS,KS+1,NS*2+1)-HS)
      IF(PA(IS,KS+1,NS*2+1).EQ.-9999.) FN=0.
      FS=TS(IS,KS-1,NS)*(PA(IS,KS-1,NS*2+1)-HS)
      IF(PA(IS,KS-1,NS*2+1).EQ.-9999.) FS=0.
      FE=TE(IS,KS,NS)*(PA(IS+1,KS,NS*2+1)-HS)
      IF(PA(IS+1,KS,NS*2+1).EQ.-9999.) FE=0.
      FW=TE(IS-1,KS,NS)*(PA(IS-1,KS,NS*2+1)-HS)
      IF(PA(IS-1,KS,NS*2+1).EQ.-9999.) FW=0.
      FU=VP(IS,KS,NS)*(PA(IS,KS,NSU*2+1)-HS)
      FD=VP(IS,KS,NSD)*(PA(IS,KS,NSD*2+1)-HS)
      FQ=-DQ(IS,KS,NS)
      ST=SC(IS,KS,NS)*(PA(IS,KS,NS*2)-HS)/STP*A**2
      SBS=FN+FS+FE+FW+FU+FD+FQ+ST
      SBT=SBST=SBS
      SBN=SBCT=SBC
C DETERMINE CURRENT CONNECTOR TRANSM.
      ICO=MINO(IP,IS)
      KCO=MINO(KP,KS)
      NCO=MINO(NP,NS)
      IF(IP.EQ.IS.AND.KP.NE.KS) TC=TS(ICO,KCO,NCO)
      IF(IP.NE.IS.AND.KP.EQ.KS) TC=TE(ICO,KCO,NCO)
      IF(IP.EQ.IS.AND.KP.EQ.KS) TC=VP(ICO,KCO,NCO)
      IF(IGP(NP).LT.2.OR.IGS(NS).LT.2) TC=0.
C BOUNDARY ABSORPTION
      IF(IGP(NP).GT.2.AND.IGS(NS).GT.2) GO TO 30
C GENERAL EXCLUSIONS
      IF(IGP(NP).LT.2.OR.IGS(NS).LT.2) GO TO 10
      IF(IGS(NS).EQ.2.AND.LFL.EQ.0) GO TO 10
      IF(IGP(NP).EQ.2.AND.LFL.EQ.0) GO TO 10
      IF(IGP(NP).EQ.2.AND.IGS(NS).EQ.2) GO TO 10
      IF(IGP(NP).EQ.2) GO TO 31
      IF(DC.GT.0..AND.LFL.EQ.-1) GO TO 10
      IF(DC.LE.0..AND.LFL.EQ.1) GO TO 10
      SIG=ABS(DC)
      SBCT=SBCT+AVER*SIG*ORF
      SBST=SBCT*(-1)
      GO TO 30
   31 CONTINUE
      IF(DC.GT.0..AND.LFL.EQ.1) GO TO 10
      IF(DC.LE.0..AND.LFL.EQ.-1) GO TO 10
      SIG=ABS(DC)
      SBST=SBST+AVER*SIG*ORF
      SBCT=SBST*(-1)
   30 CONTINUE
C DETERMINE CORRECTIONS AND NEW BALANCE
      COR=100.
      FC=DC*TC
      FN=FC+(SBST-SBCT)/2.
      IF(ABS(FC).GT.EPSI) COR=FN/FC
      TN=AMAX1(EPSI,TC*COR)
C EXTERNALLY SET LIMIT TO TRANSM.
      TN=AMIN1(TN,0.1)
```

```
      FN=DC*TN
      SBN=SBC-FC+FN
      SBT=SBS+FC-FN
   10 CONTINUE
      IF(TN.EQ.0.) TN=TC
      IF(IGP(NP).LT.3) SBN=0.
      RETURN
      END
```

```
      SUBROUTINE CALMOD(MAXI,ORFI,CON,NCAR,IRS,IRE,KRS,KRE)
      DIMENSION IGA(10),IRS(2),IRE(2),KRS(67,2),KRE(67,2)
      COMMON/LEV/PA(67,58,5),VP(67,58,2),TE(67,58,2),TS(67,58,2)
     1,SC(67,58,2),DQ(67,58,2),IG(67,58)
      COMMON A,MNS,MEW,MAQ
      LEVEL 2,PA,VP,TE,TS,SC,DQ,IG
C GENERAL INITIALIZATIONS
      ORF=ORFI
      VMIN=0.000001**3
C MAIN LOOP
      DO 1 IT=1,MAXI
C INITIALIZE ERROR SUMS AND EXTREMES
      SUM=EMAX=0.
      NSU=0
C START OF INNER LOOPS
      DO 2 N=1,MAQ
      NA=N
      IST=IRS(N)
      IEN=IRE(N)
      DO 3 I=IST,IEN
      KST=KRS(I,N)
      KEN=KRE(I,N)
      DO 4 K=KST,KEN
C DECODE GEOMETRICAL DATA AND DETERMINE ADJOINING AQUIFERS
      ENCODE(10,90,IGC) IG(I,K)
   90 FORMAT(I10)
   91 FORMAT(10I1)
      DECODE(10,91,IGC) (IGA(L),L=1,10)
      IF(IGA(N).LE.2) GO TO 4
      NPD=NA
      NN=NA-1
      NPU=0
      IF(NN.LT.1) GO TO 10
      DO 5 L=1,NN
      IF(IGA(L).GT.1) NPU=L
    5 CONTINUE
   10 CONTINUE
      DO 6 L=N,10
      IF(IGA(L).GT.1.AND.NPD.EQ.NA) NPD=L
    6 CONTINUE
C ASSIGN POINT POTENTIALS
    7 NP=NA*2+1
      H=PA(I,K,NP)
      HI=PA(I,K,NP-1)
      HN=PA(I,K+1,NP)
      HS=PA(I,K-1,NP)
      HE=PA(I+1,K,NP)
      HW=PA(I-1,K,NP)
      HU=PA(I,K,NPU*2+1)
      HD=PA(I,K,NPD*2+1)
C ASSIGN HYDRAULIC PARAMETERS
      DN=TS(I,K,NA)
      DS=TS(I,K-1,NA)
      DE=TE(I,K,NA)
      DW=TE(I-1,K,NA)
      DU=VP(I,K,NA)
      DD=VP(I,K,NPD)
```

```
C APPLY FINITE DIFFERENCE EQUATION
      SH=HN*DN+HS*DS+HE*DE+HW*DW+HU*DU+HD*DD
      ST=DN+DS+DE+DW+DU+DD
      SD=0.
      SQ=DQ(I,K,NA)
      DCOF=0.1
      COR=AMIN1(H,PA(I,K,NP-1))
      IF(PA(I,K,NP-1).LE.0.) COR=H
      DELA=(SH+DCOF*COR)/(ST+DCOF)
      DEL=(SH-SQ)/ST-H
      SQA=(DELA-COR)*DCOF
      IF(SQA.GE.0..AND.SQA.LT.SQ) DEL=DELA-H
      PA(I,K,NP)=H+ORF*DEL
C CALCULATE CONVERGENCE CHECKS
      ER=ABS(PA(I,K,NP)-H)
      SUM=SUM+ER
      EMAX=AMAX1(EMAX,ER)
      NSU=NSU+1
    4 CONTINUE
    3 CONTINUE
    2 CONTINUE
C END OF INNER LOOPS
C BYPASS CALL CARRE IF NCAR=0
      IF(NCAR.LE.0) GO TO 8
      CALL CARRE(ORF,SUM,IT,NCAR,MAXI)
C CHECK FOR CONVERGENCE
    8 CHE=SUM/NSU
      IF(CHE.LT.CON) GO TO 9
    1 CONTINUE
    9 WRITE(61,900) IT,CHE,EMAX,ORF,ORFI
  900 FORMAT(1H0,*CONVERGED AFTER*,I4,* ITERATIONS*,* ERROR AVER.*,
     1F7.3,* MAX.*,F8.3,/1X,*OVER RELAXED BY*,F7.3,* RESET TO*,F7.3)
      ORF=ORFI
      RETURN
      END
```

```
      SUBROUTINE REXBAL(RCF,IP,KP,NP,IS,KS,NS,LFL,SBC,SBN,TC,TN ,SBS,
     1SBT,NSTP)
      DIMENSION IGP(10),IGS(10)
      COMMON/LEV/PA(67,58,5),VP(67,58,2),TE(67,58,2),TS(67,58,2),
     1SC(67,58,2),DQ(67,58,2),IG(67,58),AR(67,58)
      COMMON A,MNS,MEW,MAQ,ORF,AVER,SIG
      LEVEL 2,PA,VP,TE,TS,SC,DQ,IG,AR
      STP=NSTP*365.*86400.
      SIG=0.
      SBC=SBN=0.
      SBS=SBT=0.
      TC=TN=0.
      EPSI=0.0000001**2
  899 FORMAT(1X,3I3)
C DETERMINE GEOMETRICAL STATUS EXCLUDE OUTSIDE NODES
   90 FORMAT(I10)
   91 FORMAT(10I1)
      ENCODE(10,90,IGC) IS(IP,KP)
      DECODE(10,91,IGC) (IGP(L),L=1,10)
      ENCODE(10,90,IGC) IG(IS,KS)
      DECODE(10,91,IGC) (IGS(L),L=1,10)
      HP=PA(IP,KP,NP*2+1)
      HS=PA(IS,KS,NS*2+1)
      DC=HS-HP
C DETERMINE ADJOINING AQUIFER NOS
      NPD=NP
      NNP=NP-1
      NPU=0
      IF(NNP.LT.1) GO TO 12
      DO 11 L=1,NNP
      IF(IGP(L).GT.1) NPU=L
   11 CONTINUE
   12 CONTINUE
      DO 13 L=NP,10
      IF(IGP(L).GT.1.AND.NPD.EQ.NP) NPD=L
   13 CONTINUE
      NSD=NS
      NNS=NS-1
      NSU=0
      IF(NNS.LT.1) GO TO 22
      DO 21 L=1,NNS
      IF(IGS(L).GT.1) NSU=L
   21 CONTINUE
   22 CONTINUE
      DO 23 L=NS,10
      IF(IGS(L).GT.1.AND.NSD.EQ.NS) NSD=L
   23 CONTINUE
C DETERMINE FLOW BALANCE PRINCIPAL NODE
      FN=TS(IP,KP,NP)*(PA(IP,KP+1,NP*2+1)-HP)
      IF(PA(IP,KP+1,NP*2+1).EQ.-9999.) FN=0.
      FS=TS(IP,KP-1,NP)*(PA(IP,KP-1,NP*2+1)-HP)
      IF(PA(IP,KP-1,NP*2+1).EQ.-9999.) FS=0.
      FE=TE(IP,KP,NP)*(PA(IP+1,KP,NP*2+1)-HP)
      IF(PA(IP+1,KP,NP*2+1).EQ.-9999.) FE=0.
      FW=TE(IP-1,KP,NP)*(PA(IP-1,KP,NP*2+1)-HP)
      IF(PA(IP-1,KP,NP*2+1).EQ.-9999.) FW=0.
      FU=VP(IP,KP,NP)*(PA(IP,KP,NPU*2+1)-HP)
```

```
      FD=VP(IP,KP,NPD)*(PA(IP,KP,NPD*2+1)-HP)
      FQ=-DQ(IP,KP,NP)
      TARGET=HP
      IF(IGP(NP).NE.5) GO TO 24
      IF(PA(IP,KP,NP*2).LE.0) GO TO 25
      TARGET=PA(IP,KP,NP*2)+ABS(FQ)*50.
      TARGET=AMAX1(TARGET,HP)
   24 IF(IGP(NP).LT.3.OR.IGP(NP).EQ.5) GO TO 25
      IF(AR(IP,KP).LE.0.) GO TO 25
      TARGET=ABS(AR(IP,KP))
   25 HFP=HP-TARGET
      SBC=FN+FS+FE+FW+FU+FD+FQ
      SBC=SBC+HFP*RCF
C DETERMINE FLOWBALANCE SECONDARY NODE
      TT=TS(IS,KS,NS)+TS(IS,KS-1,NS)+TE(IS,KS,NS)+TE(IS,KS-1,NS)
      IF(AR(IS,KS).LE.0.) TT=0.
      FN=TS(IS,KS,NS)*(PA(IS,KS+1,NS*2+1)-HS)
      IF(PA(IS,KS+1,NS*2+1).EQ.-9999.) FN=0.
      FS=TS(IS,KS-1,NS)*(PA(IS,KS-1,NS*2+1)-HS)
      IF(PA(IS,KS-1,NS*2+1).EQ.-9999.) FS=0.
      FE=TE(IS,KS,NS)*(PA(IS+1,KS,NS*2+1)-HS)
      IF(PA(IS+1,KS,NS*2+1).EQ.-9999.) FE=0.
      FW=TE(IS-1,KS,NS)*(PA(IS-1,KS,NS*2+1)-HS)
      IF(PA(IS-1,KS,NS*2+1).EQ.-9999.) FW=0.
      FU=VP(IS,KS,NS)*(PA(IS,KS,NSU*2+1)-HS)
      FD=VP(IS,KS,NSD)*(PA(IS,KS,NSD*2+1)-HS)
      FQ=-DQ(IS,KS,NS)
      TARGET=HS
      IF(IGS(NS).NE.5) GO TO 26
      IF(PA(IS,KS,NS*2).LE.0.) GO TO 27
      TARGET=PA(IS,KS,NS*2)+ABS(FQ)*50.
      TARGET=AMAX1(TARGET,HS)
   26 IF(IGS(NS).LT.3.OR.IGS(NS).EQ.5) GO TO 27
      IF(AR(IS,KS).LE.0.) GO TO 27
      TARGET=ABS(AR(IS,KS))
   27 HFS=HS-TARGET
      SBS=FN+FS+FE+FW+FU+FD+FQ
      SBS=SBS+HFS*RCF
      SBT=SBST=SBS
      SBN=SBCT=SBC
C DETERMINE CURRENT CONNECTOR TRANSM.
      ICO=MINO(IP,IS)
      KCO=MINO(KP,KS)
      NCO=MINO(NP,NS)
      IF(IP.EQ.IS.AND.KP.NE.KS) TC=TS(ICO,KCO,NCO)
      IF(IP.NE.IS.AND.KP.EQ.KS) TC=TE(ICO,KCO,NCO)
      IF(IP.EQ.IS.AND.KP.EQ.KS) TC=VP(ICO,KCO,NCO)
      IF(IGP(NP).LT.2.OR.IGS(NS).LT.2) TC=0.
C BOUNDARY ABSORPTION
      IF(IGP(NP).GT.2.AND.IGS(NS).GT.2) GO TO 30
C GENERAL EXCLUSIONS
      IF(IGP(NP).LT.2.OR.IGS(NS).LT.2) GO TO 10
      IF(IGS(NS).EQ.2.AND.LFL.EQ.0) GO TO 10
      IF(IGP(NP).EQ.2.AND.LFL.EQ.0) GO TO 10
      IF(IGP(NP).EQ.2.AND.IGS(NS).EQ.2) GO TO 10
      IF(IGP(NP).EQ.2) GO TO 31
      IF(DC.GT.0..AND.LFL.EQ.-1) GO TO 10
```

```
      IF(DC.LE.0..AND.LFL.EQ.1) GO TO 10
      SIG=ABS(DC)
      SBCT=SBCT+AVER*SIG*ORF
      SBST=SBCT*(-1)
      GO TO 30
   31 CONTINUE
      IF(DC.GT.0..AND.LFL.EQ.1) GO TO 10
      IF(DC.LE.0..AND.LFL.EQ.-1) GO TO 10
      SIG=ABS(DC)
      SBST=SBST+AVER*SIG*ORF
      SBCT=SBST*(-1)
   30 CONTINUE
C DETERMINE CORRECTIONS AND NEW BALANCE
      COR=100.
      FC=TC*DC
      FN=FC+(SBST-SBCT)/2.
      IF(ABS(FC).GT.EPSI) COR=FN/FC
      TN=TC*COR
      TN=AMIN1(TN,0.5)
      TN=AMAX1(TN,0.00001)
      FN=TN*DC
      SBN=SBC-FC+FN
      SBT=SBS+FC-FN
   10 CONTINUE
      IF(TN.EQ.0.) TN=TC
      IF(IGP(NP).LT.3) SBN=0.
      RETURN
      END
```

```
      PROGRAM NODIS(INPUT,OUTPUT,FD05,SCRAT,DFIL,TAPE60=INPUT,TAPE61
     1=OUTPUT,TAPE10=FD05,TAPE20=SCRAT,TAPE30=DFIL,ART,TAPE40=ART,
     1FD01,TAPE5=FD01)
      DIMENSION NBON(40000),NOAC(4000),DIS(10,4000),NBS(10),NBN(10),
     1YR(200),TH(200),DI(200),DIA(100),AL(20),DIH(4000),NOIN(4000)
      DIMENSION VA(3),VB(3)
      COMMON /A/NBON
      LEVEL 2,NBON,DIS
      EQUIVALENCE (NBON,DIS)
      INTEGER DUM
      DATA NOAC/4000*0/
      DO 11 N=1,40000
      NBON(N)=0
   11 CONTINUE
      IO=61
      NOB1=1
      NOB2=1
      NOL1=67
      NOL2=58
      NOR1=NOL1-NOB1
      NOR2=NOL2-NOB2
C READ DISCHARGE DATA BY BORE
    2 KA=1
      KB=3
    5 READ(10,92) (NC,NV,NCC,NW,AQ,(YR(K),TH(K),DI(K),K=KA,KB))
   92 FORMAT(2X,I1,A5,I1,I6,A1,2X,3(2F2.0,F4.0,12X))
      IF(EOF(10)) 50,6
    6 IF(AQ.EQ.1HA.OR.AQ.EQ.1HZ.OR.AQ.EQ.1HY) GO TO 5
C CONVERT TIME AND DISCHARGE
      DO 7 K=KA,KB
      IF(YR(K).EQ.0.AND.DI(K).EQ.0) GO TO 7
      IF(YR(K).GT.74) YR(K)=1800.+YR(K)+TH(K)/12.
      IF(YR(K).LE.74) YR(K)=1900.+YR(K)+TH(K)/12.
      DI(K)=DI(K)/19.
      IF(K.GT.1.AND.YR(K).LE.YR(K-1)) YR(K)=0.
    7 CONTINUE
      IF(NCC.EQ.0) GO TO 8
      KA=KB+1
      KB=KA+2
      GO TO 5
C INTERPOLATION
C FIRST SECTION BEFORE READINGS
    8 DO 9 N=1,KB
      IF(YR(N).GT.0) GO TO 10
    9 CONTINUE
      GO TO 2
   10 KA=N
      IA=YR(N)
      DO 12 K=1880,IA
      DIA(K-1880+1)=0.
   12 CONTINUE
C MAIN PART INTERPOLATION
   19 KZ=KA+1
      IF(KZ.GT.KB) GO TO 14
      DO 13 K=KZ,KB
      IF(YR(K).GT.0) GO TO 15
   13 CONTINUE
```

```
      14 IA=YR(KA)
         DO 16 K=IA,1979
         DIA(K-1880+1)=DI(KA)
      16 CONTINUE
         GO TO 17
      15 GRA=(DI(K)-DI(KA))/(YR(K)-YR(KA))
         NA=YR(K)
         NZ=YR(KA)
         DO 18 I=NZ,NA
         DIA(I-1880+1)=DI(KA)+(I-NZ)*GRA
      18 CONTINUE
         KA=K
         GO TO 19
C REPLACE BORE BY NODE IDENT
      17 CONTINUE
C READ MATCHING MASTERCARD AND DETERMINE GRID
     110 READ(5,192) NVM,NWM,VA(1),VA(2),VA(3),VB(1),VB(2),VB(3)
     192 FORMAT(3X,A5,1X,I6,10X,F3.0,5F2.0)
         IF(EOF(5)) 50,111
     111 IF(NV.EQ.NVM.AND.NW.EQ.NWM) GO TO 112
         IF(NV.NE.NVM.OR.NWM.LT.NW) GO TO 110
         BACKSPACE 5
         WRITE(61,1903) NV,NW
    1903 FORMAT(1X,21H***MASTERCARD VOLUME  ,A5,* WELL*,I7,* NOT FOUND*)
         GO TO 2
     112 CONTINUE
         CALL GRIDNO(XEW,YNS,VB,VA,0)
         I=XEW+0.5
         K=YNS+0.5
         IF(I.LE.NOL1.AND.K.LE.NOL2.AND.I.GT.0.AND.K.GT.0) GO TO 118
         WRITE(61,1900) I,K,NW,NV
    1900 FORMAT(22H ***OUT OF RANGE GRID ,I4,1X,I4,* AT WELL*,I7,* VOL*,
        1A5)
         GO TO 2
     118 NN=(I-NOB1)*(NOR2+1)+K-NOB2+1
         NOAC(NN)=1
C WRITE SCRATCH FILE SCRAT
         WRITE(20,900) NN
     900 FORMAT(I6)
         WRITE(20,901) (DIA(N),N=1,100)
     901 FORMAT(10F10.3)
         GO TO 2
      50 REWIND 10
C ACCUMULATE NODES
         DO 51 ML=1,10
         DO 52 N=1,4000
         DO 53 NY=1,10
         DIS(NY,N)=0.
      53 CONTINUE
      52 CONTINUE
         REWIND 20
      55 READ(20,900) NN
         IF(EOF(20)) 57,56
      56 READ(20,901) (DIA(K),K=1,100)
         DO 54 K=1,10
         KK=(ML-1)*10+K
         DIS(K,NN)=DIS(K,NN)+DIA(KK)
```

```
   54 CONTINUE
      GO TO 55
C GROUP OUTPUT ON SCRATCH FILE FD05
   57 DO 60 K=1,10
      JK=(ML-1)*10+K+1880-1
      WRITE(30,902) JK
  902 FORMAT(1HY,I6)
      NAC=0
      DO 61 L=1,4000
      IF(NOAC(L).LT.1) GO TO 61
      NAC=NAC+1
      II=(L-1)/(NOR2+1)+NOB1
      JJ=L-(II-NOB1)*(NOR2+1)+NOB2-1
      KK=2
      WRITE(30,903)  II,JJ,KK,DIS(K,L)
  903 FORMAT(3I2,F10.3)
   61 CONTINUE
   60 CONTINUE
   51 CONTINUE
C FINAL OUTPUT
      REWIND 30
   70 READ(30,95) (AL(K),K=1,20)
   95 FORMAT(20A1)
      IF(EOF(30))80,71
   71 DO 72 K=1,NAC
      READ(30,904) (NOIN(K),DIH(K))
  904 FORMAT(I6,F10.3)
   72 CONTINUE
      WRITE(40,95) (AL(K),K=1,20)
      WRITE(40,905) (NOIN(K),DIH(K),K=1,NAC)
  905 FORMAT(5(1X,I6,F9.3))
      GO TO 70
   80 CONTINUE
      END
```

```
      PROGRAM VARGEO(INPUT,OUTPUT,GEO,NEW,TAPE60=INPUT,TAPE61=OUTPUT,
     1TAPE1=GEO,TAPE2=NEW)
      DIMENSION IG(67,58),JG(67,58,2),IGA(10)
      DIMENSION IA(12),JA(12),CO(12)
      MAO=2
      MNS=58
      MEW=67
C INITIALIZE
      DO 100 I=1,MEW
      DO 100 K=1,MNS
  100 IG(I,K)=0
C READ GEOMETRY CODES AND SPLIT
      DO 1 K=1,MNS
      READ(1,90) (IG(I,K),I=1,MEW)
   90 FORMAT(10X,7I10)
      IF(EOF(1)) 7,8
    8 CONTINUE
      DO 2 I=1,MEW
      ENCODE(10,91,IGC) IG(I,K)
   91 FORMAT(I10)
      DECODE(10,92,IGC) (IGA(L),L=1,10)
   92 FORMAT(10I1)
      JG(I,K,1)=IGA(1)
      JG(I,K,2)=IGA(2)
    2 CONTINUE
    1 CONTINUE
    7 CONTINUE
C MODIFY GEOMETRY BLOCK OR POINTWISE
      DO 3 N=1,2000
      NN=1
      READ(60,93) (NO,IV,(IA(K),JA(K),CO(K),K=1,12))
   93 FORMAT(I2,1X,I2,12(2I2,A1))
      WRITE(61,94) (NO,IV,(IA(K),JA(K),CO(K),K=1,12))
   94 FORMAT(1X,I2,1X,I2,12(2I2,A1))
      IF(EOF(60)) 10,4
    4 I1=I2=IA(NN)
      J1=J2=JA(NN)
      IF(CO(NN).NE.1H-) GO TO 5
      NN=NN+1
      I2=IA(NN)
      J2=JA(NN)
      IF(I2.EQ.0) I2=I1
      IF(J2.EQ.0) J2=J1
    5 N1=I1*J1
      N2=I2*J2
      IF(N1.LT.1.OR.N1.GT.MEW*MNS) GO TO 3
      IF(N2.LT.1.OR.N2.GT.MEW*MNS) GO TO 3
      DO 16 I=I1,I2
      DO 16 J=J1,J2
      JG(I,J,NO)=IV
   16 CONTINUE
      NN=NN+1
      IF(NN.GT.12) GO TO 3
      GO TO 4
    3 CONTINUE
   10 CONTINUE
C RECOMBINE CODES
```

```
      DO 12 I=1,MEW
      DO 12 K=1,MNS
      JG(I,K,1)=MAX0(JG(I,K,1),0)
      JG(I,K,1)=MIN0(JG(I,K,1),9)
      JG(I,K,2)=MIN0(JG(I,K,2),9)
      JG(I,K,2)=MAX0(JG(I,K,2),0)
      IG(I,K)=(JG(I,K,1)*10+JG(I,K,2))*100000000
   12 CONTINUE
      CALL OUTGEO(MAQ,MNS,MEW,IG,2)
      END
```

```
      PROGRAM VARHYD(INPUT,OUTPUT,OLD,NEW,TAPE60=INPUT,TAPE61=OUTPUT,
     1TAPE1=OLD,TAPE2=NEW)
      DIMENSION  IA(12),JA(12),CO(12)
      COMMON/LEV/PA(67,58,5),HY(67,58,8)
      COMMON A,MNS,MEW,MAQ
      LEVEL 2,PA,HY
      MNS=58
      MEW=67
      MAQ=2
C INITIALIZE
      DO 100 I=1,MEW
      DO 100 K=1,MNS
      DO 100 N=1,MAQ
      HY(I,K,N)=HY(I,K,N+2)=HY(I,K,N+4)=HY(I,K,N+6)=-9999.
  100 CONTINUE
C READ OLD DATA IF AVAILABLE
      READ(1,90) COD
   90 FORMAT(A1)
      IF(EOF(1)) 7,8
    8 CONTINUE
      REWIND 1
      CALL REDHYD(1)
    7 CONTINUE
C MODIFY POINT OR BLOCK
      DO 3 N=1,2000
      NN=1
      READ(60,93) (DT,NQ,TC,V,(IA(K),JA(K),CO(K),K=1,12))
   93 FORMAT(A2,1X,I2,1X,A1,F13.5,12(2I2,A1))
      WRITE(61,94) (DT,NQ,TC,V,(IA(K),JA(K),CO(K),K=1,12))
   94 FORMAT(1X,A2,1X,I2,1X,A1,F13.5,12(2I2,A1))
      IF(EOF(60)) 10,4
    4 IN=NQ
      IF(DT.EQ.2HTE) IN=2+NQ
      IF(DT.EQ.2HTS) IN=4+NQ
      IF(DT.EQ.2HSC) IN=6+NQ
      I1=I2=IA(NN)
      J1=J2=JA(NN)
      IF(CO(NN).NE.1H-) GO TO 5
      NN=NN+1
      I2=IA(NN)
      J2=JA(NN)
      IF(I2.EQ.0) I2=I1
      IF(J2.EQ.0) J2=J1
    5 N1=I1*J1
      N2=I2*J2
      IF(N1.LT.1.OR.N1.GT.MEW*MNS) GO TO 3
      IF(N2.LT.1.OR.N2.GT.MEW*MNS) GO TO 3
      DO 6 I=I1,I2
      DO 6 J=J1,J2
      IF(TC.EQ.1HS) HY(I,J,IN)=V
      IF(HY(I,J,IN).EQ.-9999.) GO TO 6
      IF(TC.EQ.1H+) HY(I,J,IN)=HY(I,J,IN)+V
      IF(TC.EQ.1H-) HY(I,J,IN)=HY(I,J,IN)-V
      IF(TC.EQ.1H*) HY(I,J,IN)=HY(I,J,IN)*V
      IF(TC.EQ.1H/) HY(I,J,IN)=HY(I,J,IN)/V
    6 CONTINUE
    3 CONTINUE
```

```
   10 CONTINUE
      CALL OUTHYD(2,880)
      END
```

```
      SUBROUTINE PLAYCO(IC,A,B,C,IVAR,MF)
      DIMENSION VAR(3),DUM(2),IAD(3),MAN(3),IE(3),IES(3)
      VAR(1)=A
      VAR(2)=B
      VAR(3)=C
      IVAR=0
      IF(IC.EQ.0) RETURN
C PRINT MESSAGE IF VALUE OUT OF RANGE >0
      DO 1 N=1,3
      AVA=ABS(VAR(N))
      IF(AVA.GT.0..AND.AVA.LT.0.0001) GO TO 2
      IF(AVA.GT.9999) GO TO 2
      GO TO 1
    2 WRITE(MF,900) VAR(N),N
  900 FORMAT(1X,*VALUE OF*,E12.5,* OUT OF CODING RANGE VAR:*,I2)
    1 CONTINUE
C DETERMINE SIGN CODES
      DO 3 N=1,3
      IAD(N)=0
      IF(VAR(N).LT.0.) IAD(N)=5
    3 CONTINUE
C SPLIT NOTATION
      DO 4 N=1,3
      ENCODE(10,92,DUM) VAR(N)
   92 FORMAT(E10.4)
      DECODE(10,93,DUM) MAN(N),IE(N)
      IE(N)=IE(N)-4
    4 CONTINUE
C NORMALIZE TO FINAL EXPONENT
      DO 5 N=1,3
      IE(N)=IE(N)*(-1)
      IES(N)=0
      IF(IE(N).LE.4) GO TO 6
      IES(N)=IE(N)-4
      IE(N)=4
    6 MAN(N)=MAN(N)/10**IES(N)
      IE(N)=IE(N)+IAD(N)
      MAN(N)=IABS(MAN(N))
    5 CONTINUE
   93 FORMAT(2X,I4,1X,I3)
C CONCATENATE
      ENCODE(18,94,DUM) (IC,(IE(N),MAN(N),N=1,3))
   94 FORMAT(1X,I1,3(I1,I4.4))
      DECODE(18,95,DUM) IVAR
   95 FORMAT(I18)
      RETURN
      END
```

```
      SUBROUTINE MANWRI(MAQ,MNS,MEW,IDIS,IYY,NF)
      DIMENSION IDIS(MEW,MNS,MAQ)
      IY=IYY-1000
      IF(IY.GE.1000) IY=IY-1000
      WRITE(NF,900) IY
  900 FORMAT(I3,I1,2I2,4I18)
      MM=MEW/4+1
      DO 1 N=1,MAQ
      DO 1 K=1,MNS
      DO 1 L=1,MM
      IA=(L-1)*4+1
      IZ=MINO(MEW,IA+3)
      IF(IDIS(IA,K,N).GT.0.OR.IDIS(IA+1,K,N).GT.0) GO TO 2
      IF(IDIS(IZ-1,K,N).GT.0.OR.IDIS(IZ,K,N).GT.0) GO TO 2
      GO TO 1
    2 WRITE(NF,900) (IY,N,K,IA,(IDIS(I,K,N),I=IA,IZ))
    1 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE DISRED(MAQ,MNS,MEW,DIS,NYY,MS,MA)
      DIMENSION DIS(MEW,MNS,MAQ),MS(MAQ),MA(MAQ),NI(5),NK(5),NN(5),
     1DI(5)
      NY=NYY-1000
      IF(NY.GE.1000) NY=NY-1000
      DO 1 N=1,MAQ
      DO 1 K=1,MNS
      DO 1 I=1,MEW
      DIS(I,K,N)=0.
    1 CONTINUE
      DO 2 N=1,MAQ
      DO 2 IS=1,2
C EXCLUDE UNAVAILABLE FILES
      IF(IS.EQ.1.AND.MS(N).EQ.0.) GO TO 2
      IF(IS.EQ.2.AND.MA(N).EQ.0.) GO TO 2
      NF=(N-1)*MAQ+IS
C SEARCH FOR DATA OF SPECIFIED YEAR
    3 READ(NF,90) YC,IY
   90 FORMAT(A1,3X,I3)
      IF(EOF(NF)) 100,4
    4 IF(IY-NY) 3,5,3
C READ DISCHARGE DATA
    5 READ(NF,91) (YC,(NI(L),NK(L),NN(L),DI(L),L=1,5))
   91 FORMAT(A1,5(3I2,F9.3,1X))
      IF(EOF(NF)) 80,6
    6 IF(YC.EQ.1HY) GO TO 80
      DO 7 L=1,5
      IF(NN(L).LE.0) GO TO 7
      IF(DI(L).LE.0.) GO TO 7
      I=NI(L)
      K=NK(L)
      NA=NN(L)
      DIS(I,K,NA)=DIS(I,K,NA)+DI(L)
    7 CONTINUE
      GO TO 5
   80 BACKSPACE NF
    2 CONTINUE
      RETURN
  100 WRITE(61,900) NY,NF
  900 FORMAT(1X,3H***,*DATA FOR YEAR 1*,I3,* NOT FOUND ON UNIT*,I3)
      RETURN
      END
```

```
      SUBROUTINE OUTFIL(NYY,NF,CC,W)
      DIMENSION W(67,58,2)
      COMMON A,MNS,MEW,MAQ
      NY=NYY
      IF(NY.GE.1000) NY=NY-1000
      IF(NY.GE.1000) NY=NY-1000
      MM=MEW/6+1
      DO 1 NA=1,MAQ
      DO 1 K=1,MNS
      DO 2 N=1,MM
      IA=(N-1)*6+1
      IZ=MIN0(IA+5,MEW)
      WRITE(NF,90) (NY,NA,K,IA,CC,(W(I,K,NA),I=IA,IZ))
   90 FORMAT(I3,1X,I1,2(1X,I2),1X,A1,1X,6E11.5)
    2 CONTINUE
    1 CONTINUE
      RETURN
      END
```

```
      PROGRAM GELEV(INPUT,OUTPUT,COR,GEL,MAS,TAPE60=INPUT,TAPE61=OUTPUT
     1,TAPE1=COR,TAPE2=GEL,TAPE3=MAS)
      DIMENSION EL(67,58,2),AR(67,58),VA(3),VB(3)
      COMMON/LEV/AR
      LEVEL 2,AR
      COMMON A,MNS,MEW,MAQ
      MNS=58
      MEW=67
      MAQ=2
C INITIALIZE FROM CORRECTIONS FILE
      DO 1 N=1,MAQ
      CC=1HG
      CALL SERDAT(MNS,MEW,1,-90,N,CC,AR,61)
      DO 2 K=1,MNS
      DO 2 I=1,MEW
      EL(I,K,N)=AR(I,K)
    2 CONTINUE
    1 CONTINUE
C OBTAIN GEL FROM MASTER CARD
   10 READ(3,92) (NV,NW,(VA(K),K=1,3),(VB(K),K=1,3),GE)
   92 FORMAT(3X,A5,1X,I6,10X,F3.0,5F2.0,1X,F5.1,1X,F5.0)
      IF(EOF(3)) 100,11
   11 GE=GE*.305
      CALL GRIDNO(XEW,YNS,VB,VA,0)
      I=XEW+0.5
      K=YNS+0.5
      IF(I.GT.MEW.OR.I.LT.1.OR.K.GT.MNS.OR.K.LT.1) GO TO 10
      IF(GE.LE.0.) GO TO 10
      IF(EL(I,K,1).LE.0.) EL(I,K,1)=GE
      IF(EL(I,K,2).LE.0.) EL(I,K,2)=GE
      GO TO 10
  100 CALL OUTFIL(970,2,1HG,EL)
      END
```

```
      SUBROUTINE GRIDNO(X,Y,XLA,YLO,DIR)
      DIMENSION XLA(3),YLO(3)
      DOUBLE R,ARC,AR,D
      IF(DIR.GT.0.) GO TO 100
C RATIONALIZE LONG. LAT.
      XLA(2)=XLA(2)+XLA(3)/60.
      XLA(1)=XLA(1)+XLA(2)/60.
      YLO(2)=YLO(2)+YLO(3)/60.
      YLO(1)=YLO(1)+YLO(2)/60.
C RADIUS
      R=460.+(32.5566-XLA(1))*4.433
C ARC
      ARC=(YLO(1)-144.)*0.008127
C COORDINATES
      AR=DABS(ARC)
      D=DSIN(AR)
      X=R*DSIGN(D,ARC)+40.5
      Y=R*DCOS(AR)-460.
      RETURN
  100 CONTINUE
C DIRECTION: GRID TO LONG.LAT.
C ARC AND RADIUS
      ARC=(X-40.5)/(Y+460.)
      AR=DABS(ARC)
      AR=DATAN(AR)
      ARC=DSIGN(AR,ARC)
      R=SQRT((X-40.5)**2+(Y+460.)**2)
C ANGLE
      ARC=ARC/0.008127
C LONG. LAT. DECIMAL
      YL=144+ARC
      XL=32.5566+(460.-R)/4.433
C SPLIT FRACTION INTO MINUTES SECONDS
      YLO(1)=AINT(YL)
      XLA(1)=AINT(XL)
      YL=YL-YLO(1)
      XL=XL-XLA(1)
      YLO(2)=AINT(YL*60.)
      XLA(2)=AINT(XL*60.)
      YL=YL-YLO(2)
      XL=XL-XLA(2)
      YLO(3)=AINT(YL*60.)
      XLA(3)=AINT(XL*60.)
      RETURN
      END
```

```
      SUBROUTINE TEMCOR(COR,TEM,TCOR,DWA,DWE)
      DIMENSION TCOR(20),TEM(100)
C DETERMINE AVERAGE TEMPERATURE
      ATEM=35
      M=0
      SUM=0
      DO 1 N=1,100
      IF(TEM(N).LE.0) GO TO 1
      M=M+1
      SUM=SUM+TEM(N)
    1 CONTINUE
      IF(M.LE.0) GO TO 2
      ATEM=SUM/M
    2 ITEM=ATEM/5+1
C CALCULATE TEMPERATURE CORRECTOR (ADDITIVE)
      ITEM=MIN0(ITEM,20)
      COR=DWA*TCOR(ITEM)-DWE
      RETURN
      END
```

INDEX