New Jersey Geological Survey
Open File Report No. 83-2

Computer Analysis of Pump Test Data

New Jersey Department of Environmental Protection • Division of Water Resources

Computer Analysis of Pump Test Data

New Jersey Geological Survey
Open File Report No. 83-2

by
Jeffrey L. Hoffman, *Senior Geologist*

Division of Water Resources
CN 029
Trenton 08625

1983

CONTENTS

ILLUSTRATIONS

Abstract

The computer program CAPTD (Computer Analysis of Pump Test Data) is an interactive package for performing Theis, Jacob, calculated recovery, and residual drawdown analyses. Transmissivity and, except in the residual drawdown analysis, storage coefficient are calculated. Analyses can be performed for an entire data set or for a specific time interval within a data set. Data points and fitted curves can be plotted in a form suitable for publication. The program has been run under the CMS system of an IBM 370 computer. The graphics capability and plotting programs are not an indispensable part of CAPTD; CAPTD can be run without their use, and the data and results displayed in tabular form.

I. Introduction

An aquifer pumping test is a means of determining aquifer characteristics by pumping water from a production well and measuring drawdown in nearby observation wells. Much of the work involved in the data analysis is amenable to computer automated procedures. The mathematical calculations, the plotting of axes, titles, and points, and the fitting of straight lines and curves are all processes which a computer can perform accurately and quickly. If a plotter is used the computer can prepare graphics suitable for publication.

Computer Analysis of Pump Test Data (CAPTD) is an interactive computer program package written to facilitate analysis of data gathered during a water resource evaluation of Pleistocene valley-fill deposits in northern New Jersey (Canace and others, in preparation). The program is written in the computer language FORTRAN and has been run successfully under the CMS operating system of an IBM 370 computer.

CAPTD can produce plots of the data and best fit lines suitable for publication. However, this requires access to graphics programs supported by the New Jersey Department of Environmental Protection, Office of Science and Research. Mounting CAPTD at another site and getting the graphics to work may require the aid of a computer programmer. However, CAPTD can work without the graphics since all the results are automatically saved in an output file. By answering 'no' to all questions concerning graphics, CAPTD can be made to work at a site which does not have any graphics capabilities.

CAPTD consists of three subprograms. The first, an EXEC program, defines the necessary data files and presents the user with the available options and starts whichever options the user requests. The second program, CAPGO, prompts the user for the information necessary to fill any data files defined by the EXEC program. The third program, called CAPTD, *per se*, prompts for commands, then performs the analyses requested and displays the results.

A user not interested in reviewing the theory behind the pump test methods may skip directly to Section VIII (Using CAPTD) for instructions on how to run the program.

II. Assumptions

"It is now desperately easy to employ a mathematical or statistical technique without any understanding of the basic assumptions made by the technique or of the limitations to the technique in its practical interpretation." (Jeffers, 1973).

The mathematical analysis of aquifer pumping test data is made tractable only by making several assumptions concerning the aquifer, the wells, and the pumping conditions. For all analysis methods performed by CAPTD the following assumptions must be made:

- The pumping well and observation wells are completed in the same aquifer.
- The aquifer is uniform, homogeneous, isotropic, of uniform thickness, and of infinite areal extent.
- The aquifer is confined.
- Prior to pumping the piezometric surface is horizontal.
- The pumping rate is kept constant.
- The well completely penetrates the aquifer and is screened throughout the aquifer.
- There is no recharge. The effect of this is that the cone of depression does not stabilize and the measured drawdown curve does not reach an equilibrium level.
- A decline in head produces an instantaneous release of water from storage.
- Storage of water in the casing of the pumping well is negligible.

In order to perform a Jacob, calculated recovery, or residual drawdown analysis one further assumption is required.

- The value of u (see Section III) is less than 0.01.

If any of the assumptions are violated, the data will not exactly fit the predicted time-drawdown curve. In practice the assumptions are never completely met, and so calculated aquifer parameters are approximations to actual values. The interpreter must be aware of the physical conditions in the field and how they differ from the ideal, assumed conditions. These deviations must be taken into account when interpreting the results.

III. Theis Analysis

The Theis analysis (Kruseman and De Ridder, 1979) is a graphical method for computing the transmissivity (T) and storage coefficient (S) of an aquifer from the relationship between time-drawdown data recorded at an observation well and the Theis type curve. The Theis equation, which describes drawdown in a confined aquifer, is written as:

$$\$ = \frac{Q}{4\pi T} W(u) \qquad (1)$$

where

$\$$	= drawdown	[L]	(2A)
Q	= pumping rate	$[L^3/T]$	(2B)
T	= transmissivity	$[L^2/T]$	(2C)
W(u)	= Theis well function	[-]	(2D)
u	= $r^2S/4Tt$	[-]	(2E)
r	= distance from pumping well to observation well	[L]	(2F)
S	= storage coefficient	[-]	(2G)
t	= time	[T]	(2H)

The Theis type curve is obtained by plotting W(u) on the vertical axis and 1/u on the horizontal axis of log-log graph paper. If the assumptions of the Theis analysis are met, then plotting time-drawdown data on log-log paper of the same scale will result in a curve identical to the Theis type curve but shifted in position (figure 1). A best fit between the time-drawdown data and the Theis curve is obtained by laying the data over the type curve and shifting it parallel to the vertical and horizontal axes without rotation. A match point common to both the Theis curve and the time-drawdown data is chosen. This point will have coordinates t and $ for time and drawdown and 1/u and W(u) on the Theis type curve. These values are substituted into equations 3 and 4 (which are equations 1 and 2E rewritten) to obtain T and S.

$$T = \frac{Q}{4\pi \$} W(u) \qquad (3)$$

$$S = \frac{u4Tt}{r^2} \qquad (4)$$

Theis type curves are available commercially at varying scales. However the user must still plot the data, overlay the type curve, pick a match point, and compute the results. Descriptions of this process are in Kruseman and De Ridder (1979), Reed (1980), and other standard references. For an aquifer pump test with many observation wells, the performing of numerous analyses may be a very time consuming process.

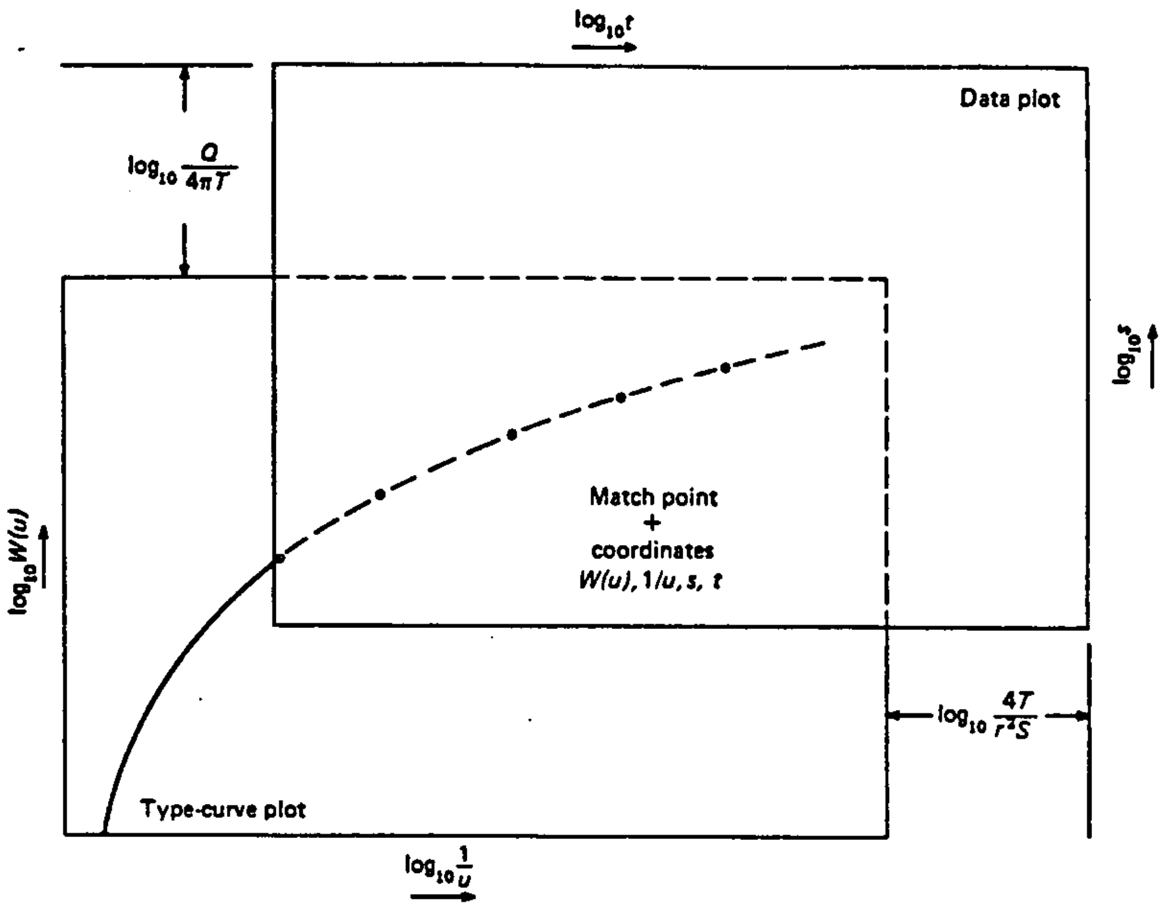


Figure 1. Theis Type Curve and Data (from Reed, 1980)

CAPTD performs the Theis analysis by shifting the log(t) and log($) data to a best fit position on the Theis type curve by the following trial and error method (see Appendix A for a more rigorous mathematical description of this process):

1) In an initial iteration, log(t) and log($) data are overlain upon the Theis type curve at nine different positions by shifting the data values specific distances Δt and $\Delta \$$ as shown in figure 2. These positions, called 'shift positions' can be viewed as occurring at the corners, line midpoints, and center of a rectangular box.

Each point shown in figure 2 can be visualized as that point on the 1/u, W(u) plot where the first time-drawdown data point would appear for the particular shift.

2) For each of the nine shift positions the overall difference between all data points and the Theis type curve is quantified as the sum of the squared distance, parallel to the vertical axis, between each shifted data point and the Theis type curve. This sum is defined as the residual R.

3) Each of the nine shift positions generates a residual. These residuals are compared and that position which generates the least residual is taken to be the closest match of the data to the Theis type curve for the iteration.

4) For the next iteration that shift position which produced the minimum residual in the previous iteration becomes the center of the box of shift positions (figure 2). The box thus moves with the center of the box becoming that point which produces the minimum residual. After the box is shifted the nine new shift positions are calculated and the program returns to step 2.

5) If the smallest residual from step 4 arises from the center point of the box, the box is not shifted for the next iteration. However, either the value of t or $ is changed. The sum of residuals arising from positions 4 and 6 (figure 2) is compared to the sum from 2 and 8. If the sum of R4 and R6 is greater than the sum of R2 and R8 the value of Δt is divided by 1.5. If the sum of R2 and R8 is greater the value of $\Delta \$$ is divided by 1.5. This procedure locates that direction in which the residual is increasing most rapidly and shortens the box in that direction.

6) 50 iterations are performed. After this minimum number of iterations has been reached, the minimum residual for each iteration is compared to the minimum residual of the previous iteration. When the change in residuals between successive iterations is less than 0.1% the best fit of the data points to the Theis curve is determined to have been reached. If the iteration process has not reached a minimum residual after 250 iterations the program stops and issues a message saying that the data do not fit a Theis type curve and meaningful values of T and S cannot be generated.

7) Once the best fit of the data to the Theis type curve has been found the position of the first time-drawdown data point on the 1/u vs. W(u) plot is determined. The point's four coordinates(t, $, 1/u, W(u)) are then substituted into equations 3 and 4 to generate values of T and S.

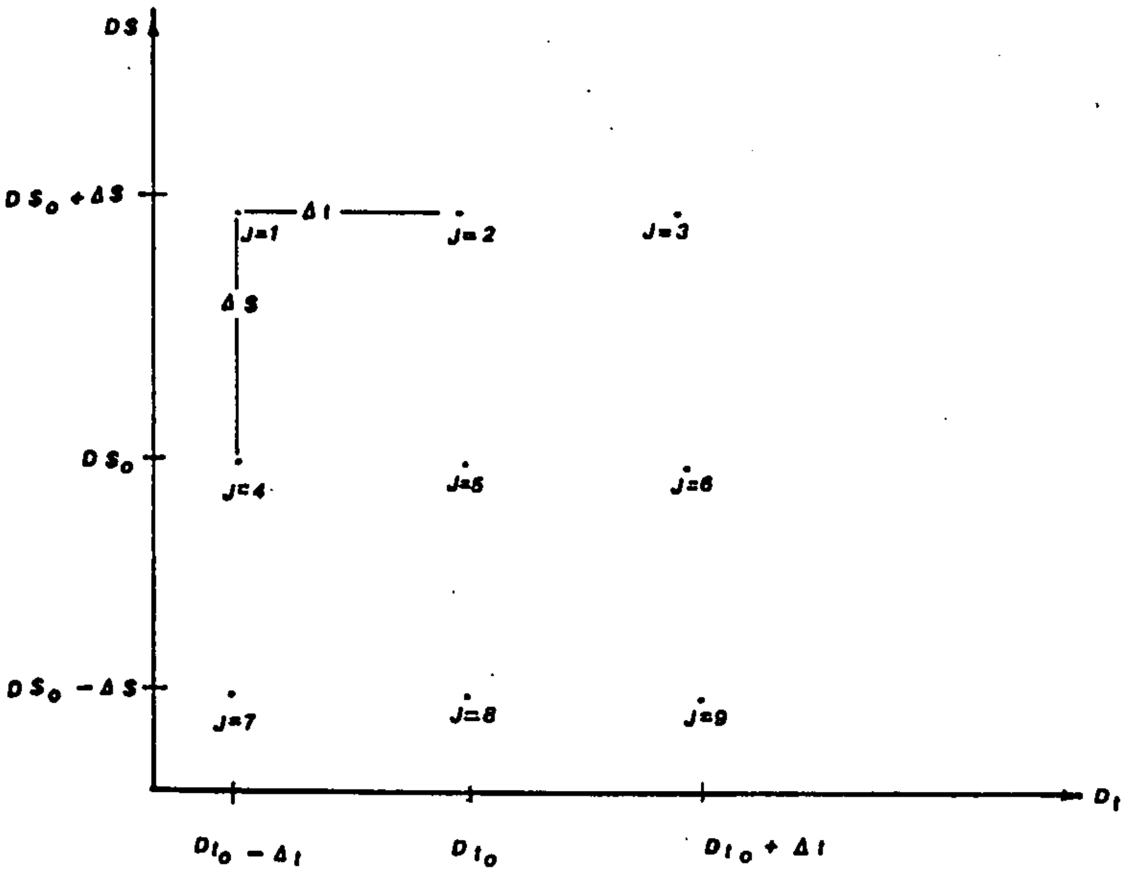


Figure 2. Dt, D$ Pairs

The method described here is not dependent upon any specific numerical properties of the Theis function. A different method (McElwee, 1980) uses the derivatives of the Theis function and of the calculated residuals to determine where to shift the Theis type curve. The method using derivatives is quicker than the trial and error method presented here but requires an initial estimate of the transmissivity and storage and may not converge to a solution if the data points fit the Theis curve poorly.

IV. Jacob Analysis

The Jacob time-drawdown analysis is a method of obtaining T and S from the slope and intercept of a line fitted to pump test data plotted on semi-log paper (Kruseman and De Ridder, 1979). The method is based on an expansion of the Theis well function. This expansion is

$$W(u) = -0.5772 - \ln(u) + u - \frac{u^2}{2!2} + \frac{u^3}{3!3} - \ldots \qquad (5)$$

If u is less than 0.01 the Theis function can be approximated as

$$W(u) = 0.5772 - \ln(u). \qquad (6)$$

Substituting equations 6 and 2e into 1 results in

$$\$ = \frac{Q}{4\pi T}\left(-0.5772 - \ln\left(\frac{r^2 S}{4Tt}\right)\right). \qquad (7)$$

This can be rewritten as

$$\$ = \frac{2.3Q}{4\pi T}\log\left(\frac{2.25Tt}{r^2 S}\right) \qquad (8)$$

A plot of \$ vs. log(t) will result in a straight line if all the assumptions listed in section III are met. The slope of the line can be expressed as

$$\text{slope} = \frac{2.3\,Q}{4\pi T} \qquad (9)$$

At the point where the drawdown is equal to 0.0, let the time value be equal to t_o. At this point, equation 8 becomes

$$0 = \log\left(\frac{2.25Tt_o}{r^2 S}\right) \qquad (10)$$

Equation 9 and 10 can be rewritten to yield T and S.

$$T = \frac{2.3Q}{4\pi(\text{slope})} \qquad (11)$$

$$S = \frac{2.25Tt_o}{r^2 S} \qquad (12)$$

Once the data points have been plotted on semi-log paper and the best fit line drawn through the data points, the slope and the x-axis intercept value (t_o) is determined. Equations 17 and 18 then readily yield estimates of the transmissivity and the storage. CAPTD uses the standard least squares linear regression technique to determine the slope and intercept of the best fit line to the \$, log(t) data points. CAPTD then calculates the aquifer transmissivity and the storage coefficient from these parameters.

V. Calculated Recovery Analysis

The calculated recovery analysis is mathematically identical to the Jacob analysis except that T and S are obtained from calculated recovery values. Calculated recovery is the distance between measured residual drawdown and an extrapolation of drawdown measured during pumping (figure 3). Plotting of calculated recovery against time since pumping stopped (t') on semi-log paper will produce a straight line if the assumptions listed in Section II are met (Johnson, 1975).

In order for CAPTD to compute the calculated recovery, it must first extrapolate the time-drawdown curve. This is done using predicted drawdown calculated from the Theis drawdown function (equation 1) with T and S values from the preceding Theis or Jacob analysis of the drawdown data. Measured drawdown is then subtracted from the predicted drawdown to give the calculated recovery. Least squares linear regression is used to obtain a best fit line to log(t) vs. calculated recovery. The slope and intercept of this line are used to find T and S exactly as was done in the Jacob analysis procedure.

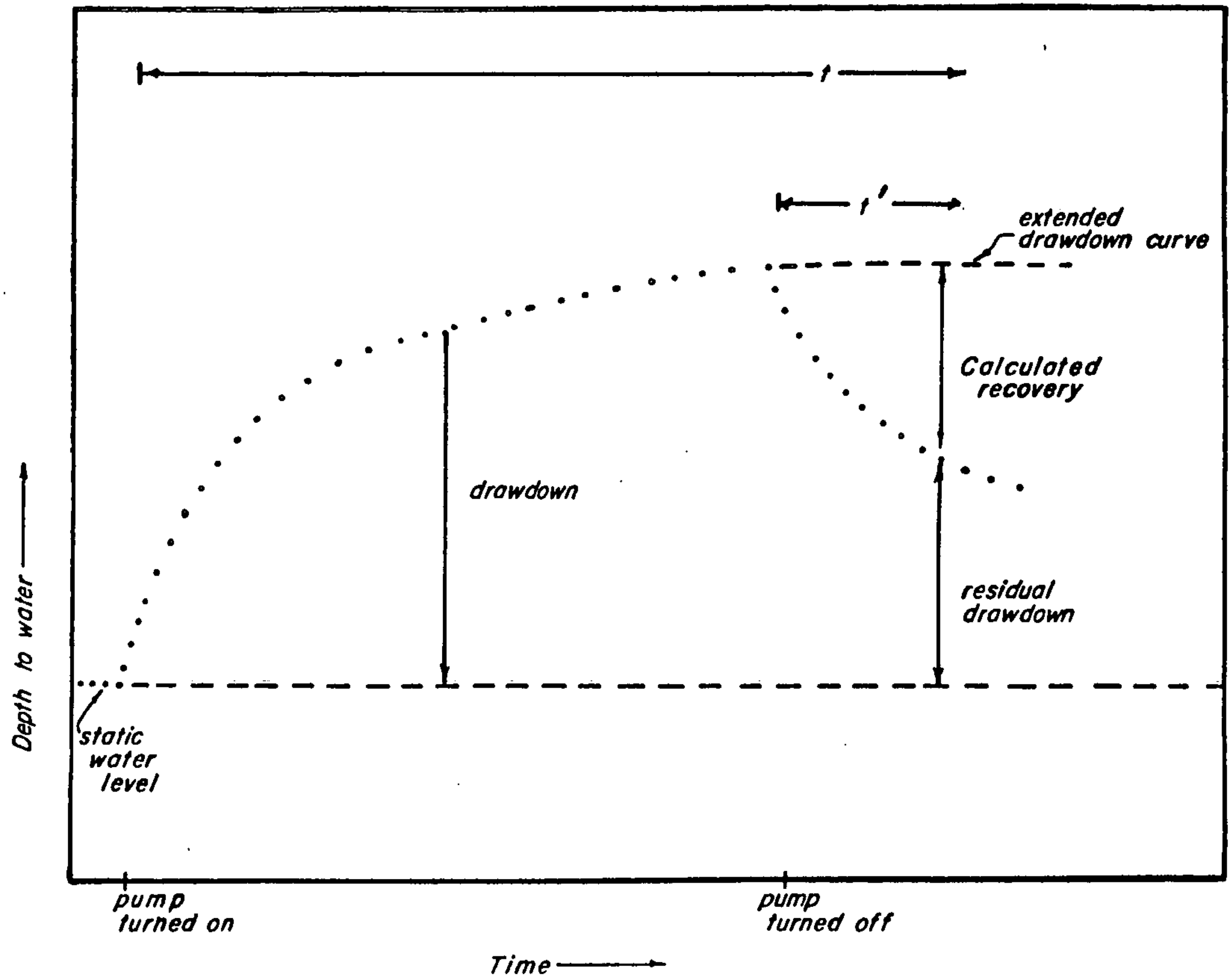


Figure 3. Time-Drawdown Data Relationships

VI. Residual Drawdown

The residual drawdown analysis is a method for calculating the aquifer transmissivity from the relationship between residual drawdown ($') and t/t' where t is the time since the pumping started and t' is the time since pumping ended (figure 3). Residual drawdown is the distance from the original, static head before pumping began to the water level at some time after the pumping ends (figure 3). $' is plotted on the arithmetic scale of semi-log paper and t/t' is plotted on the log axis.

On semi-log paper $' vs. t/t' will be a straight line if the assumptions listed in Section II are met. The equation for this line is:

$$\$' = \frac{2.3Q}{4\pi T} \log (t/t') \qquad (13)$$

The slope of this line is thus:

$$\text{slope} = \frac{2.3Q}{4\pi T} \qquad (14)$$

Transmissivity can be calculated from the slope of the best fit line. No estimate of the storage coefficient is possible when using this method.

In estimating transmissivity by the residual drawdown method CAPTD first calculates the t/t' values and then residual drawdown values. A straight line is fitted to those two variable using a standard least square regression technique. The slope of the resulting line is then substituted into the equation

$$T = \frac{2.3Q}{4\pi (\text{slope})} \qquad (15)$$

to calculate the transmissivity.

VII. Data Input

A data file must be created to hold the data needed by CAPTD. There are two ways to do this. A user with a working knowledge of CMS can create a file according to the formatting instructions which follow. But if the user wishes, CAPTD will help create the necessary data file. In this case the user merely has to answer the questions asked and CAPTD will automatically set up the data file.

If the user will always allow CAPTD to set up the data file the next three paragraphs may be skipped. However, the important notes which end this section are indeed important and must be read.

The input data format is shown in figure 4. The first card contains the title line. The first column is a plotting control symbol which controls the main title font. A blank space or an '&' is recommended. Columns 2-25 are for the title. Any characters in columns 26-80 are ignored. The second card holds four pieces of data: 1) the distance from the pumping well to the observation well in feet; 2) the pumping rate in gallons per minute; 3) the depth in feet to water in the observation well prior to pumping, and; 4) the number of drawdown observations made in the observation well. At least one space must separate each of the four data items.

The drawdown data are paired on the following cards. Each data pair, consisting of the observation time in minutes and the corresponding depth to water in feet, should appear on a different card. Thus, if 27 is specified on card 2 as the number of data pairs, 27 cards will follow.

Recovery data, if any, are entered after the last time-drawdown pair has been entered. The recovery data are added onto the file without any blank cards or other delimitation. The first card holds the pumping time in minutes and the number of time-recovery data pairs. The next series of cards holds the data pairs, each pair appearing on a new card. The first item on each card is the time in minutes since pumping began. The second item on the same card is the observed depth to water, in feet, at that time.

Important Notes:

1. Even though data are referred to as time-drawdown and time-recovery data pairs, do not enter actual drawdown or recovery data. The user enters the raw time and depth to water data. CAPTD subtracts the static head entered on card 2 from all depth to water values to arrive at drawdown and recovery values. In the special case where the original depth to water data is not available and only the drawdown is known, CAPTD can be used by setting the value of the static head on card 2 equal to 0.0.

2. No time value can be 0.0. The first time-drawdown datum must be at a time greater than zero. Similarily, the first time-recovery datum must also be at a time greater than zero.

3. If a Theis analysis is desired, all depth to water values for the drawdown data must be greater than the static depth to water value. Physically, the water level in the observation well must have dropped during the pump test; it may not rise above the static level. This is necessary because the log of the drawdown is taken during the Theis analysis. If the

water level rises above the static level, drawdown will have a negative value. When the computer tries to take the log of a negative number it stops, issues an error message, and quits.

Card #	Variable Name	Meaning
1	GC (in column 1)	graphics control character a blank or '&' is recommended
	TITLE (columns 2-25)	a title to appear at top of all plots
2	R	distance from pumping well to observation well (feet)
	Q	pumping rate (gpm)
	STATIC	depth to water in observation well prior to pumping (feet)
	N	number of time-drawdown data pairs
2 + 1 through 2 + N	time-drawdown data	each card has one data pair - time (minutes) and depth to water (feet)
3 + N	TPUMP	time pumping stopped (minutes)
	NREC	number of time-recovery data pairs
4 + N through 4 + N + NREC	time-recovery data	each card has one data pair - time since pumping stopped (minutes) and depth to water (feet)

Note: Cards 3 + N and on only present if recovery data is available.

Figure 4. Data input Format

VIII. Using CAPTD

After logging onto the computer and creating an input data file as specified in Section VII, the user types VWSCAPTD followed by pressing the return key. From this point on the user doesn't initiate anything, the computer asks questions and the user merely has to answer. The user must be careful when answering questions on the terminal. Hitting extraneous keys, misspelling 'yes' or 'no', or holding down a key too long may result in errors.

A series of options will appear on the screen. This is called the menu. The options are:

1. End.
2. List all CAPTD data files.
3. Create a new CAPTD data file.
4. Run CAPTD on an existing data file.

The user inputs the number of the requested option. If '1' is input the program will end. If '2' is input all CAPTD data files (all files with a file type of CAPDATA) will be listed on the screen.

If option '3' is selected the computer will ask all the questions necessary to create a data file. The first question asked under option 3 is what is to be the name of the new data file. The user here must input an alphanumeric name of 8 or fewer characters. This name should be recorded so that it won't be forgotten. The computer then goes on to ask all the questions necessary to create the new data file. A user thus could create a data file, and then run an analysis on it.

If option '4' is chosen the first question asked is: "What is the name of the data file?" Once this is entered CAPTD will go to that file, open it, and read all the data.

After each analysis CAPTD asks if a plot should be made, and then tells the user to prepare the plotter. If the plotter is not attached the plot will appear only on the terminal screen. The plot is output in four steps. First the axes are plotted, then the title and axis labels, then the data and best fit line, and finally the calculated results. After each step the program will pause. Hitting the return key will cause the program to go onto the next step. After everything has been plotted out CAPTD will stop, allowing the user to view the screen and, if necessary, remove the graph from the plotter bed. Pressing the return key will start CAPTD up again.

After each analysis and plot of results CAPTD allows the user to change the time limits on the data. It is very important to realize that CAPTD will look at all data within the time limits and assign the same relative importance to each data point. Thus if an aquifer pumping test has 100 data points during late time and 5 points during early time, an analysis using all the points will match the late time data much better than the early time data. The user may adjust the values of TLO and THI (the time limits) to exclude data points. Alternatively the user may chose simply to eliminate data points by not entering them into the computer. Note: The first time through an analysis all of the data points will automatically be included. On successive analyses the user may specify the values of TLO and THI.

CAPTD starts off by asking if a Theis analysis is requested. If not, it goes to the Jacob analysis. If yes, CAPTD then begins its computations. The Theis analysis may take up to a minute on a day when a lot of people are using the computer system.

If recovery data are available the user will be asked after each Theis analysis if a calculated recovery analysis is requested. If no, CAPTD asks if another Theis analysis should be run and if so what are the time limits which bound the data to be included in the new analysis.

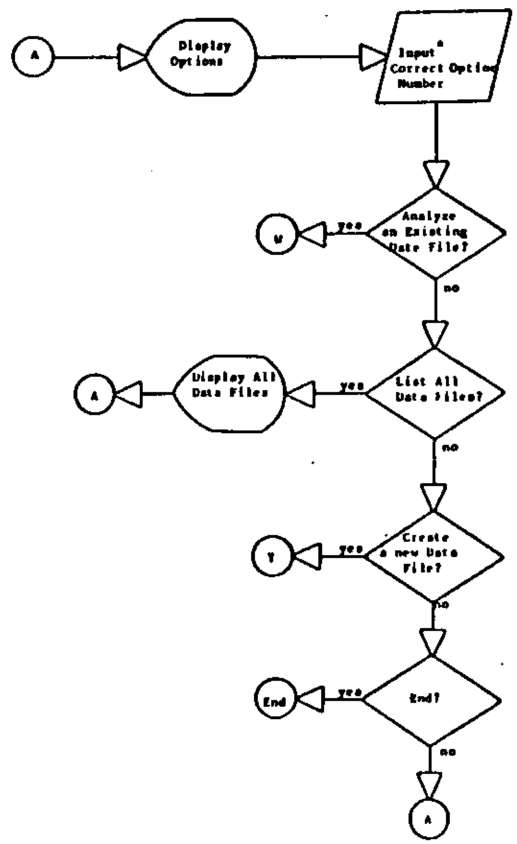
For the first calculated recovery analysis after a Theis or Jacob analysis, CAPTD automatically sets the time limits so as to include all recovery data. For each successive calculated recovery analysis the user sets the limits.

The Jacob analysis procedure exactly follows the Theis analysis procedure.

If recovery data are available, CAPTD asks if a residual drawdown analysis should be done after the last Jacob analysis. The residual drawdown analysis procedure is the same as the Jacob analysis procedure.

Figure 5 shows a flow chart which graphically displays the steps and options in CAPTD.

CAPTD creates, on a separate file, a written copy of all the analyses. This output file, while not graphically displaying the data points and results, still contains the results. This file is automatically printed at the end of each session.



* Step requires user input from terminal

Figure 5. CAPTD Flow Chart

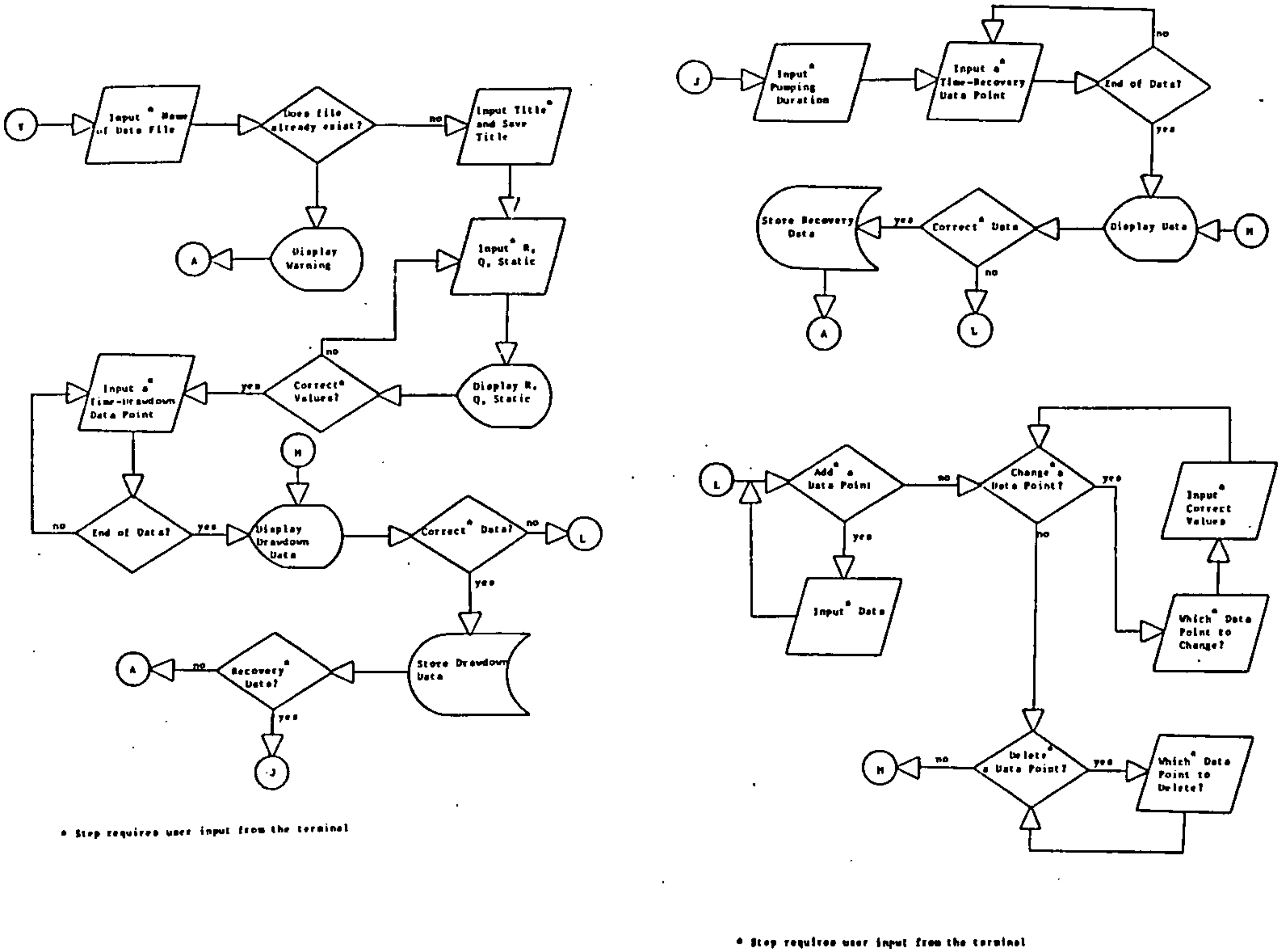
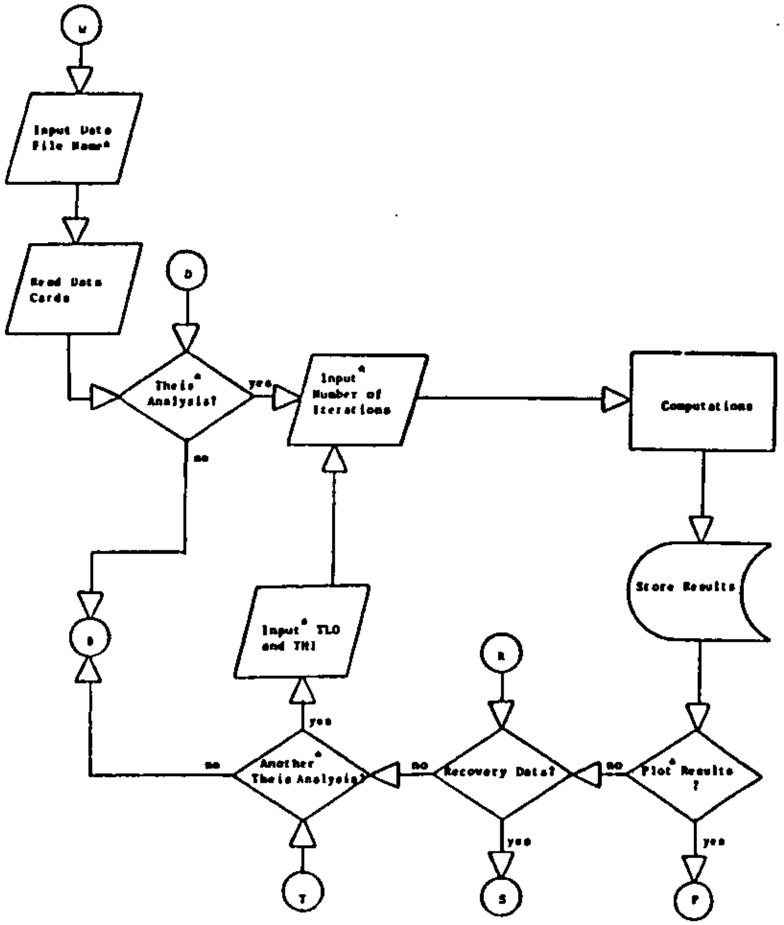
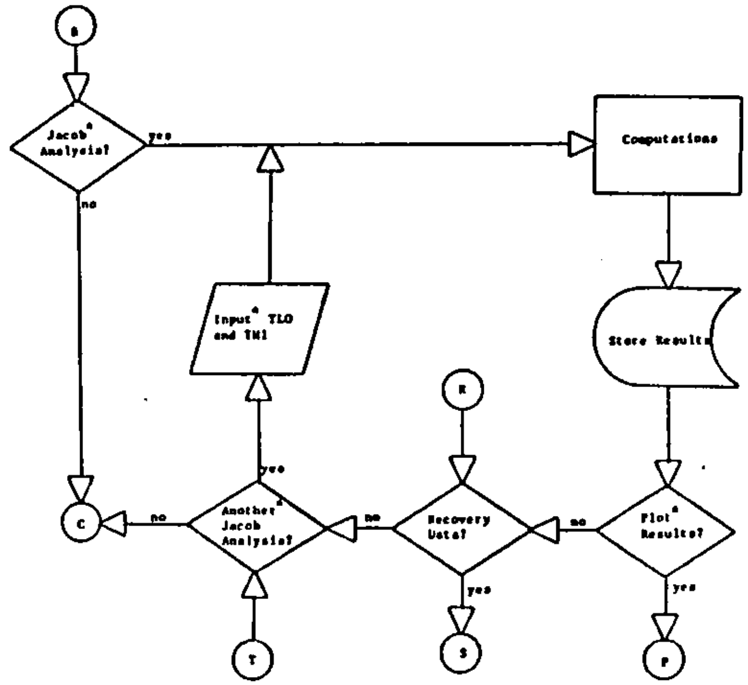


Figure 5 continued.

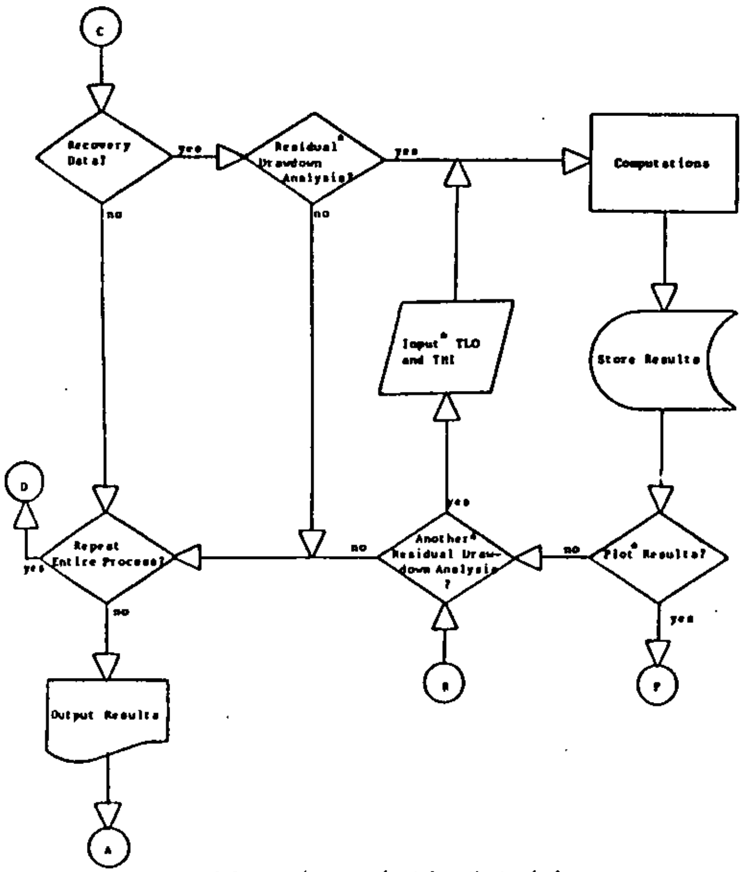


* Step requires user input from the terminal

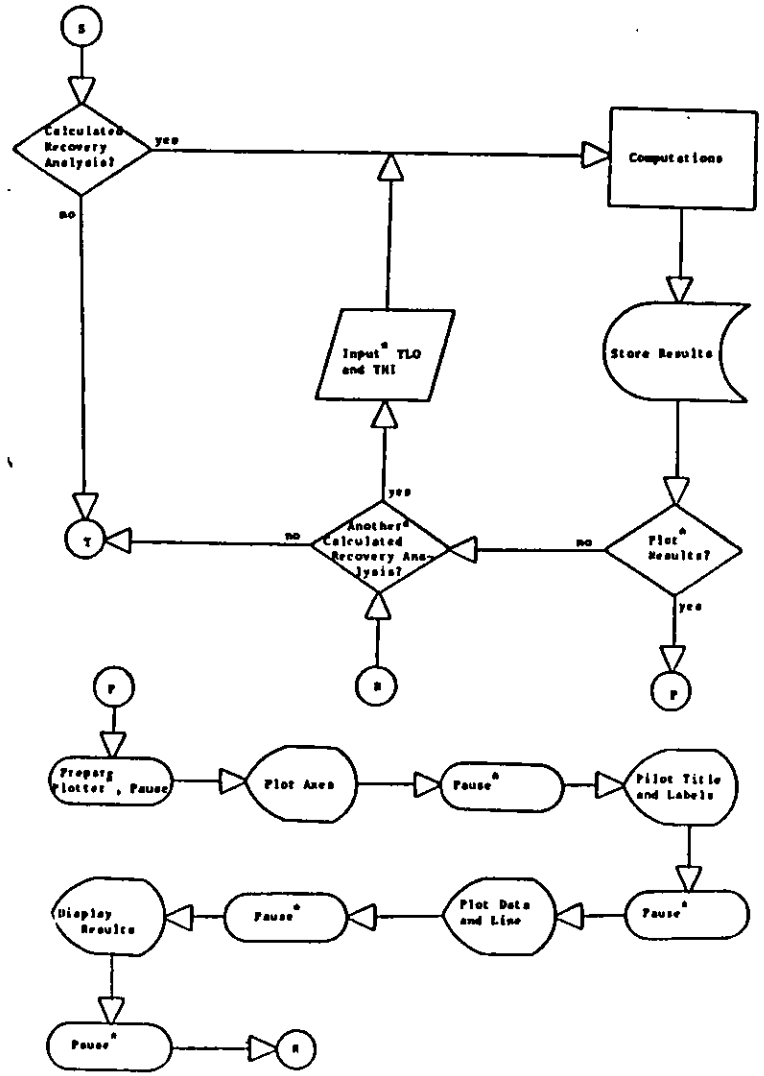


* Step requires user input from the terminal

Figure 5 continued.



* Step requires user input from the terminal



* Step requires user input from the terminal

Figure 5 continued.

X. Example

Figures 6 through 13 show graphs made during one session of CAPTD. The data file used is shown in Appendix B. The output file generated is in Appendix C.

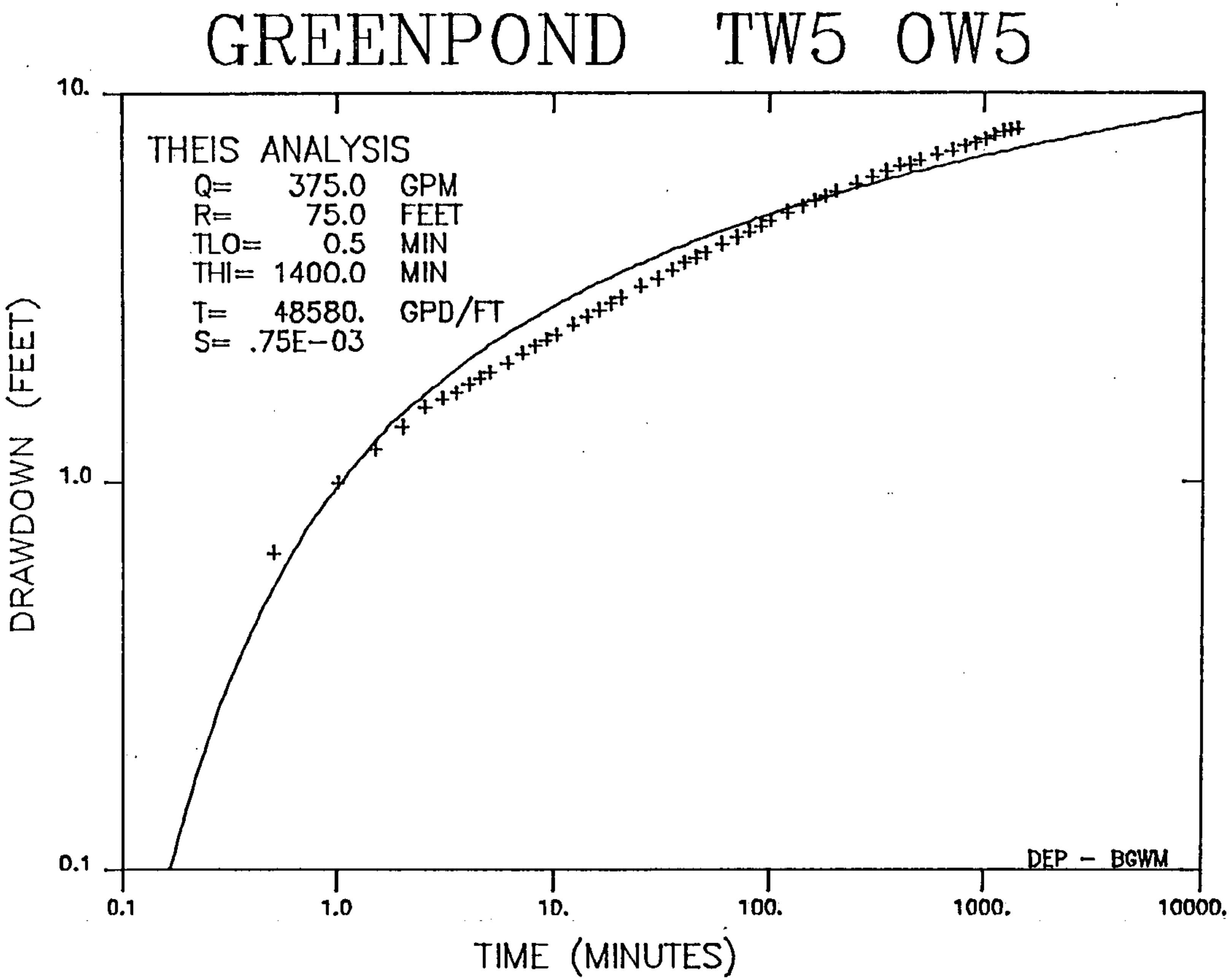


Figure 6. Theis Analysis Plot

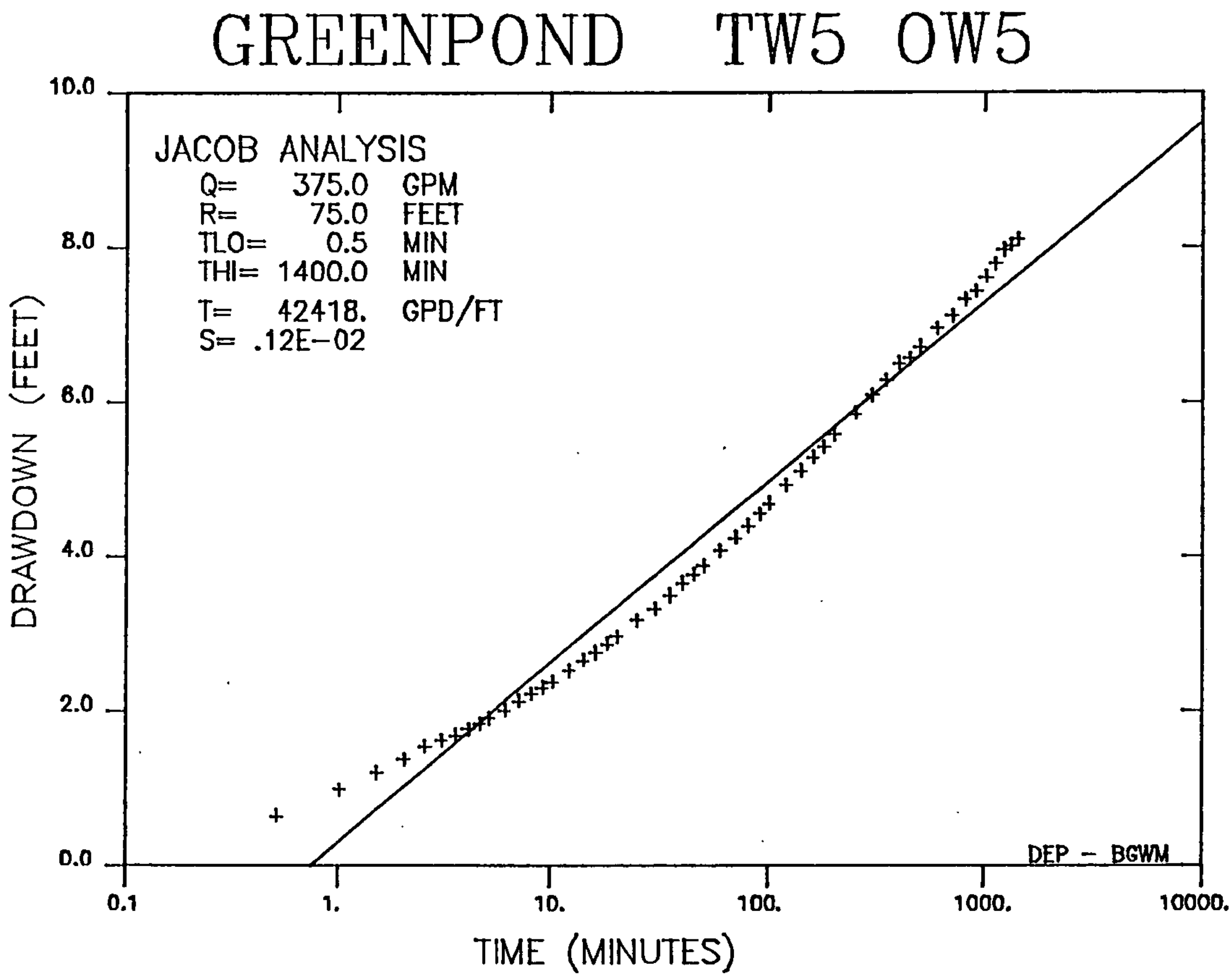


Figure 7. Jacob Analysis Plot

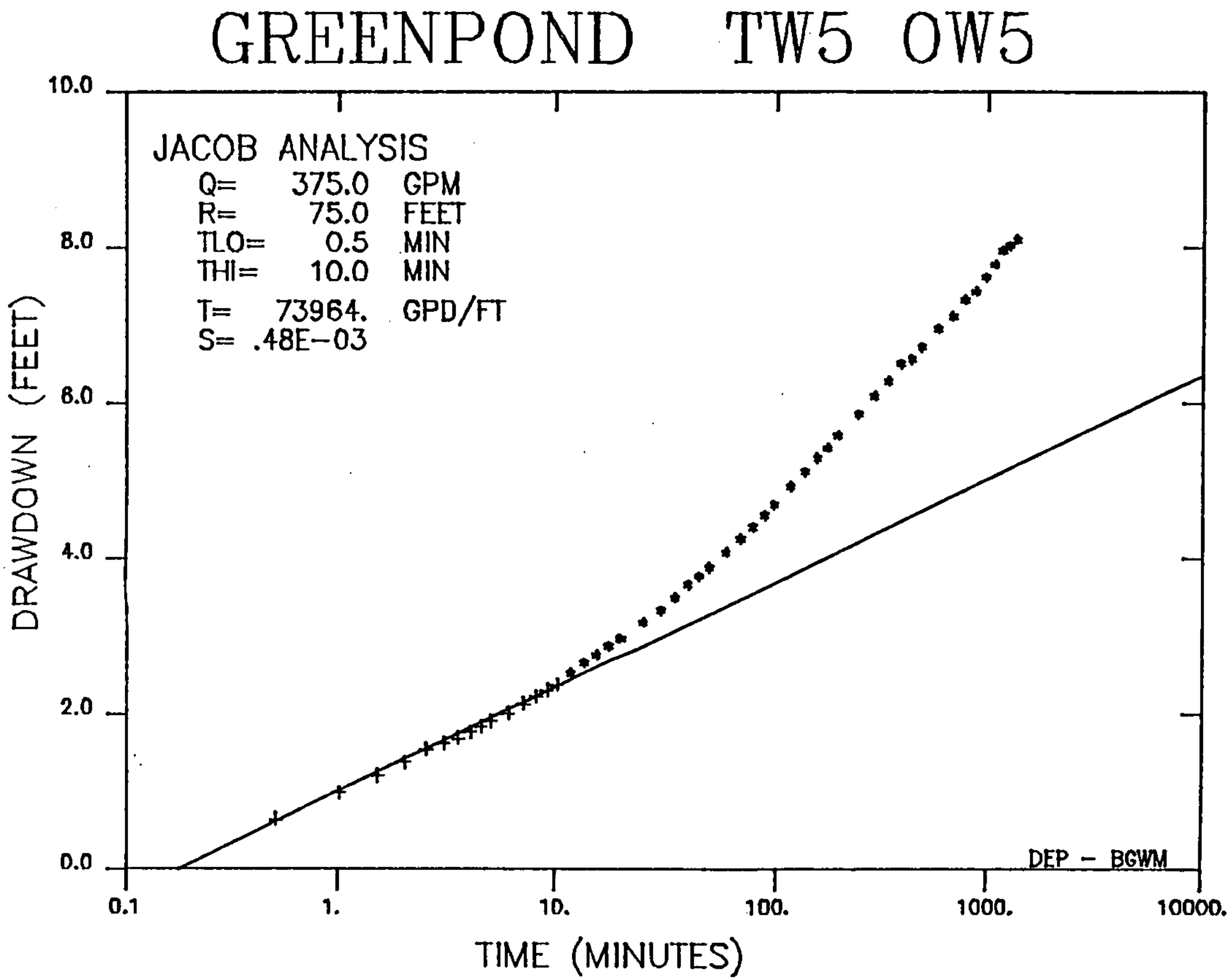


Figure 8. Jacob Analysis Plot

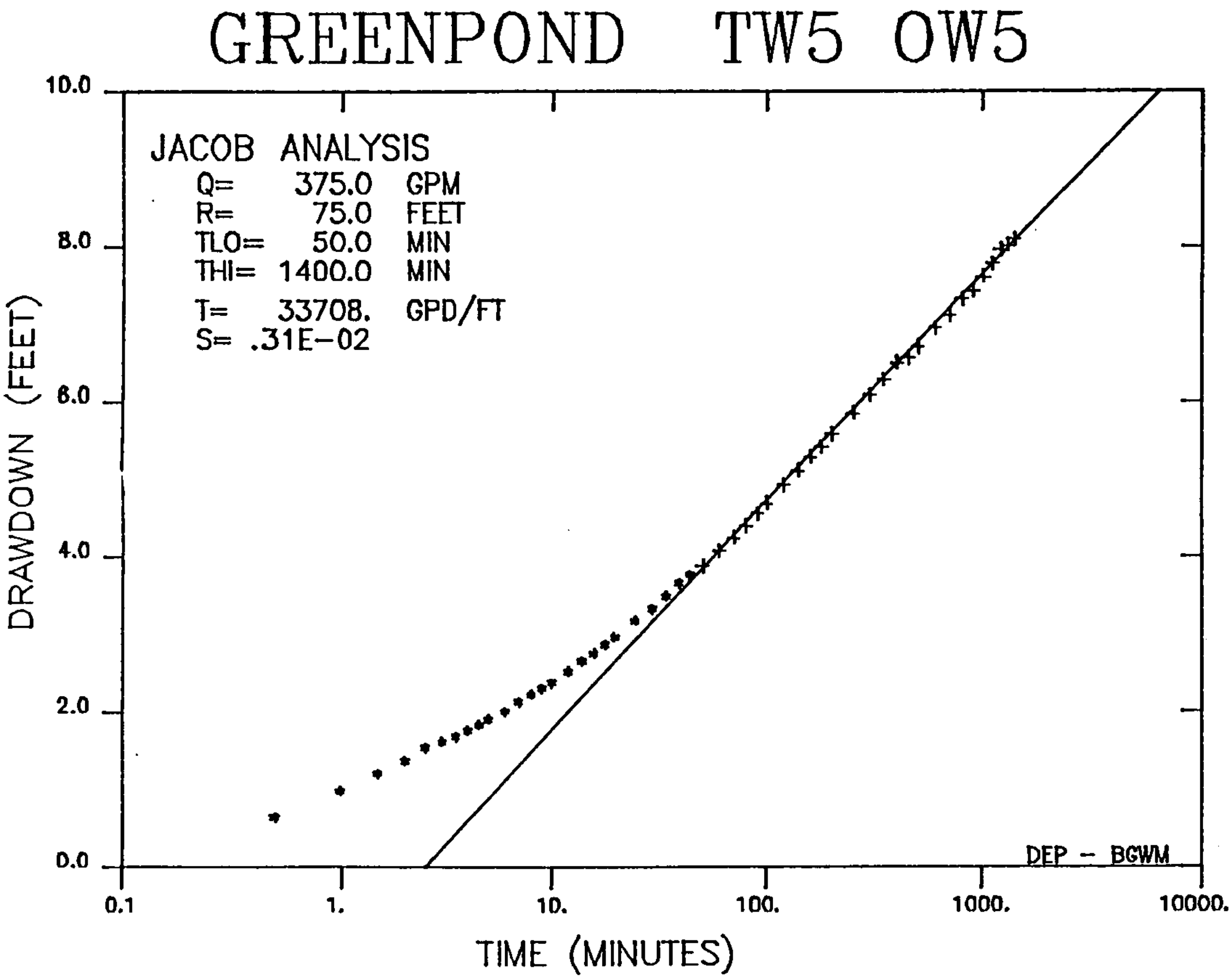


Figure 9. Jacob Analysis Plot

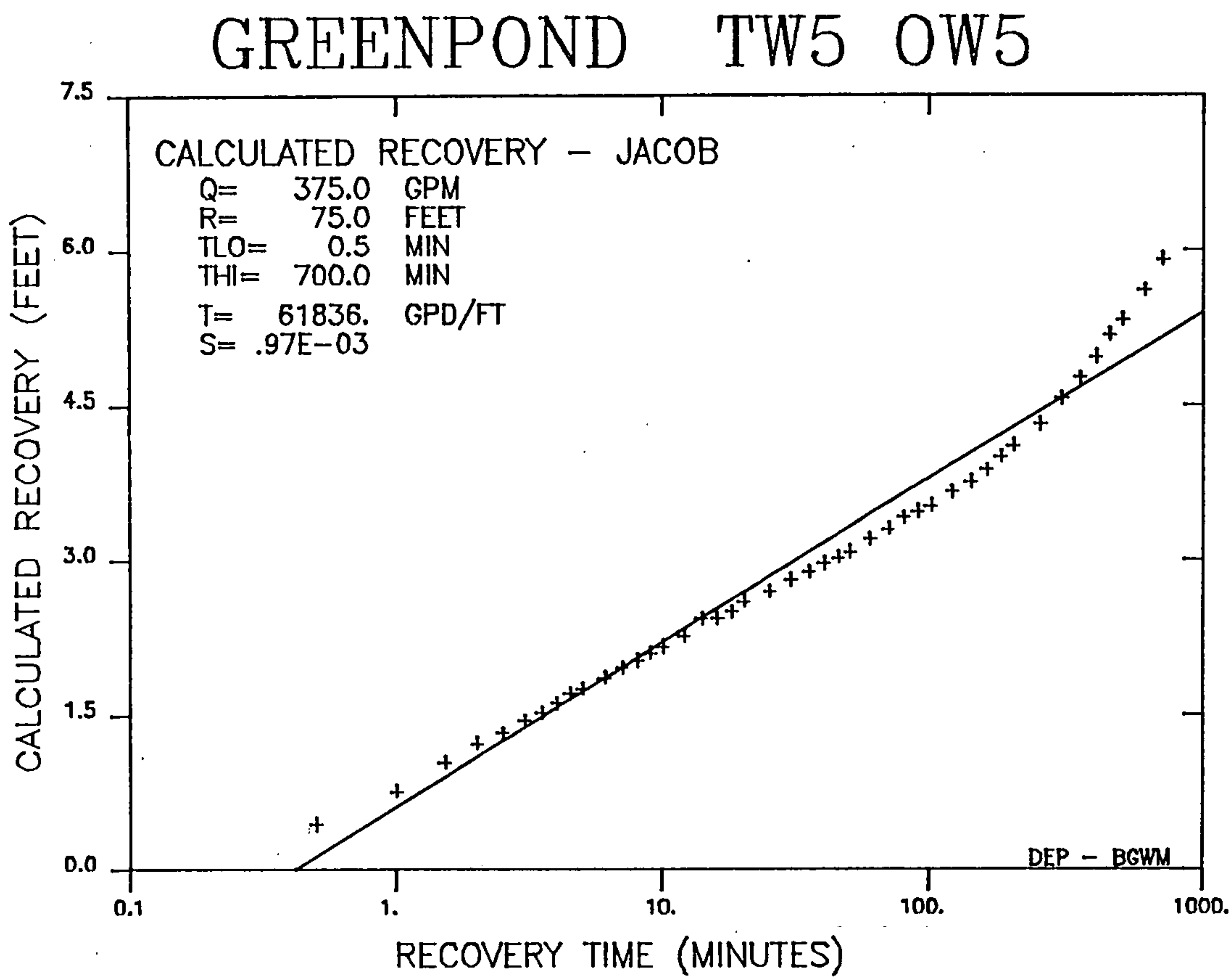


Figure 10. Calculated Recovery Analysis Plot

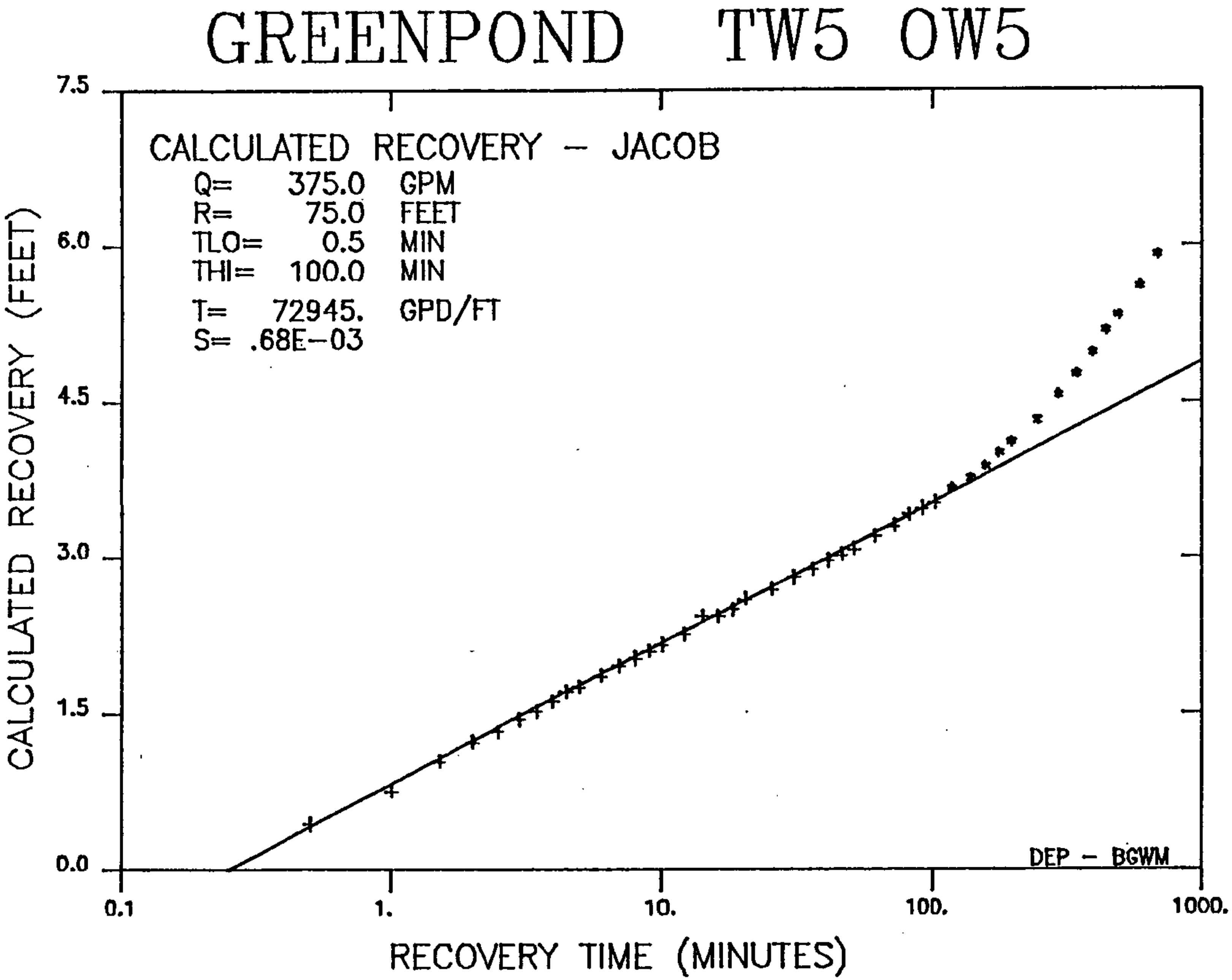


Figure 11. Calculated Recovery Analysis Plot

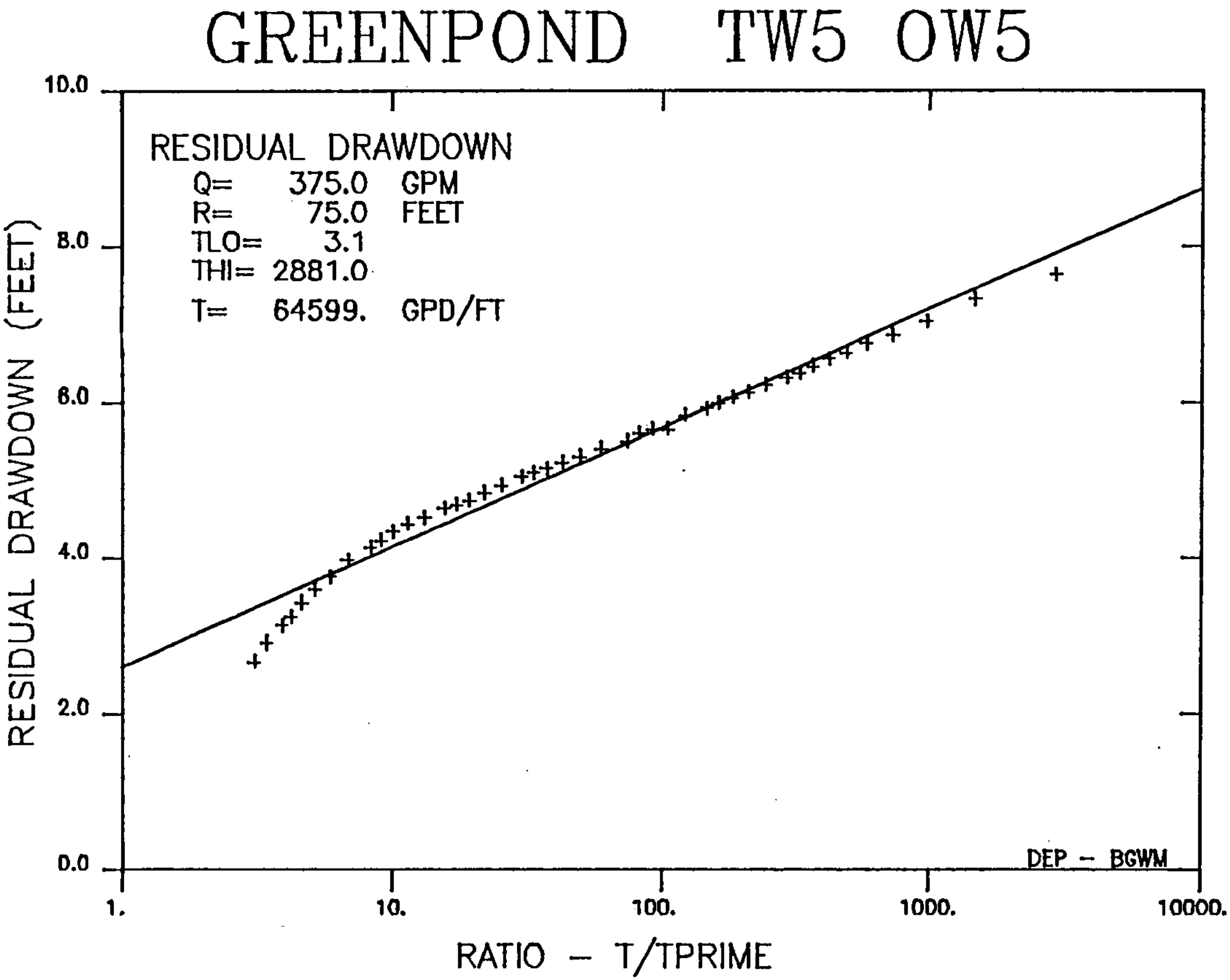


Figure 12. Residual Drawdown Plot

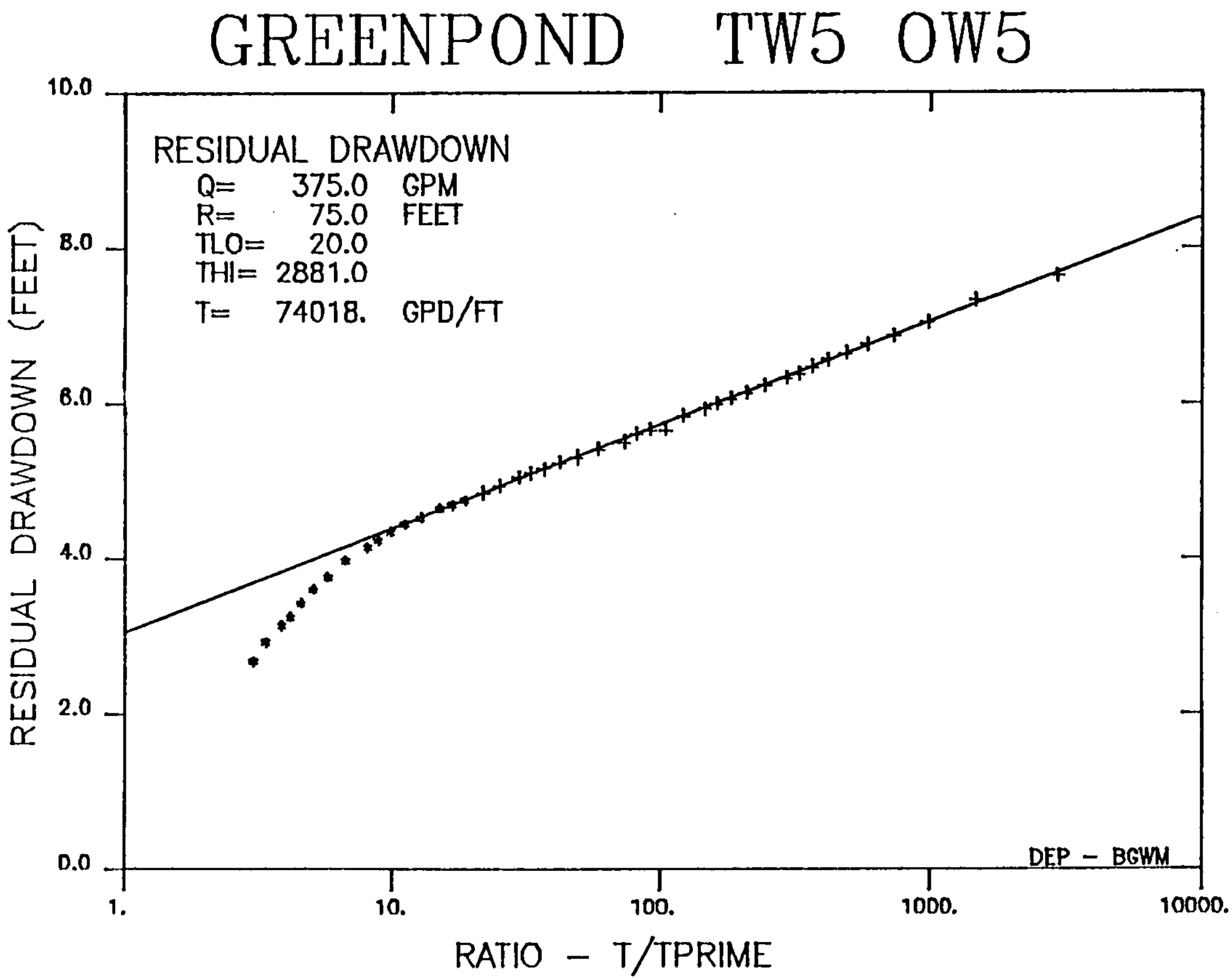


Figure 15. Residual Drawdown Plot

References

Bear, J., 1979, Hydraulics of Groundwater: McGraw-Hill International Book Company, New York, 576 p.

Canace, R., Hutchinson, W., Saunders, W., and Andres, K.G., 1983, Results of the 1980-81 Drought Emergency Ground Water Investigation in Morris and Passaic Counties, New Jersey: New Jersey Geological Survey, Open File Report, in preparation, Trenton.

Jeffers, J.N.R., 1973, Systems Modeling and Analysis in Resource Management: Journal of Environmental Management, vol. 1, pp. 13-28.

Johnson, E.E., 1975, Ground Water and Wells: Johnson Division, UOP Inc., Saint Paul, Minn., 440 p.

Kruseman, G.P., and De Ridder, N.A., 1979, Analysis and Evaluation of Pumping Test Data: Bulletin 11, International Institute for Land Reclamation and Improvement, Wageningen, The Netherlands, 200 p.

McElwee, Carl D., 1980, The Theis Equation: Evaluation, Sensitivity to Storage and Transmissivity and Automated Fit of Pump Test Data: Kansas Geological Survey, Lawrence, Kansas, 39 p.

Reed, J.E., 1980, Type Curves for Selected Problems of Flow to Wells in Confined Aquifers: Chapter B3, Techniques of Water Resources - Investigations of the United States Geological Survey, USGS, Arlington, Virginia, 106 p.

Appendix A. Mathematical Basis for Theis Curve Matching Technique

The mathematical basis for the Theis type curve matching technique outlined in Section IV is relatively simple. When the time-drawdown data is plotted on log-log paper and overlain on a Theis type curve it is apparent that for the i'th data point:

$$Lu_i = Lt_i + Dt \qquad (16)$$

$$LW_i = L\$_i + D\$ \qquad (17)$$

where

$$Lt_i = \log(t_i) \qquad (18)$$

$$L\$_i = \log(\$_i) \qquad (19)$$

$$LW_i = \log(W(u_i)) \qquad (20)$$

$$Lu_i = \log(1/u_i) \qquad (21)$$

and Dt and D$ are the amounts by which the log (1/u) and log(W(u)) axes are offset from the log(t) and log($) axes. The superposition of the time-drawdown data and the shifting of the axes relative to each other (keeping the log($) and log(1/u) axes parallel, and doing the same for the log(t) and log(W(u)) axes) can be viewed as altering the values of Dt and D$.

For a particular pair of Dt and D$ each data pair (t,$) can be viewed as falling on the Lt and L$ plot, and also on the Lu, LW plot. On the Lu, LW plot, however, one other point is of interest. The value of Lu can be manipulated to yield a value of u. This value of u yields a W(U) value which can be plotted on the Lu, LW graph. This new point is a point on the Theis curve. Let the point on the Theis curve corresponding to the i'th data point be designated PW_i. Then:

$$PW_i = \log(W((10^{Lu_i})^{-1})) \qquad (22)$$

The residual R is defined as the sum of the source of the distance between the Theis type curve and each data point, or:

$$R = \sum_{i=1}^{n} (PW_i - LW_i)^2 \qquad (23)$$

where n is the number of data points.

Changing the values of Dt and D$ will change the residual. Those values of Dt and D$ which minimize the residual will result in the best fit of the data to the Theis type curve.

The procedure for finding the minimum residual is iterative. Each iteration begins with the establishment of 8 Dt,D$ pairs centered around a ninth Dt, D$ pair (figure 2). Each value of Dt and D$ can be expressed as:

$$Dt_j = Dt_o + f_j \Delta t \qquad (24)$$

$$D\$_j = D\$_o + g_j \Delta \$ \qquad (25)$$

where j ranges from 1 to 9 and the values of f and g are defined in table 1. For j=5, $f_5=0$ and $g_5=0$, or:

$$Dt_5 = Dt_o \qquad (26)$$

$$D\$_5 = D\$_o \qquad (27)$$

Dt_5, $D\$_5$ thus is at the center of the nine 'shift positions.'

The preceding equations can be rewritten taking into account the steps which occur each interation.

$$Dt_j = Dt_o + f_j \Delta t \qquad (28)$$

$$D\$_j = D\$_o + g_j \Delta \$ \qquad (29)$$

$$Lu_{ij} = Lt_i + Dt_j \qquad (30)$$

$$LW_{ij} = L\$_i + D\$_j \qquad (31)$$

$$PW_{ij} = \log(W((10^{Lu_{ij}})^{-1})) \qquad (32)$$

$$R_j = \sum_{i=1}^{n} (PW_{ij} - LW_{ij})^2 \qquad (33)$$

where i varies from 1 to n and j varies from 1 to 9. Each iteration thus will produce 9 residuals.

The residual arising from each (Dt,D$) pair is compared to the other residuals from that iteration. That pair which produces the minimum residual is noted. The values of Dt_o and $D\$_o$ are reassigned to be that (Dt,D$) pair which produces the least residual. The iteration process then starts over.

If the minimum residual is produced by j=5 (figure 2) reassigning Dt_o and $D\$_o$ will have no effect. In this special case the sum of R2 and R8 is compared to the sum of R4 and R6. If R2 plus R8 is greater, the value of $\Delta\$$ is divided by 1.5. If R4 plus R6 is greater, the value of Δt is divided by 1.5. This procedure identifies that direction (either horizontal or vertical) in which the residual is increasing most rapidly and decreases the search area in that direction.

A minimum of 50 iterations takes place. After this minimum is reached the minimum residual from each iteration is compared to the minimum residual of the preceding iteration. If the minimum residual changes by less than 0.1% the iteration process stops; if not, it continues. A maximum of 250 iterations is allowed to take place before the program halts and issues a statement that the data do not fit a Theis curve and no solution is available.

j	f_j	g_j
1	-1	+1
2	0	+1
3	+1	+1
4	-1	0
5	0	0
6	+1	0
7	-1	-1
8	0	-1
9	+1	-1

Table 1. F and G Values for Dt, D$ definition

```
&GREENPOND  TW5 OW5
    75.0      375.0       3.94          51
    0.50       4.60       1.00        4.94        1.50        5.16        2.00        5.33
    2.50       5.50       3.00        5.58        3.50        5.64        4.00        5.72
    4.50       5.79       5.00        5.86        6.00        5.96        7.00        6.08
    8.00       6.18       9.00        6.26       10.00        6.33       12.00        6.48
   14.00       6.61      16.00        6.71       18.00        6.82       20.00        6.92
   25.00       7.13      30.00        7.28       35.00        7.45       40.00        7.61
   45.00       7.72      50.00        7.83       60.00        8.03       70.00        8.20
   80.00       8.35      90.00        8.50      100.00        8.64      120.00        8.87
  140.00       9.06     160.00        9.24      180.00        9.37      200.00        9.53
  250.00       9.80     300.00       10.04      350.00       10.23      400.00       10.45
  450.00      10.51     500.00       10.66      600.00       10.90      700.00       11.06
  800.00      11.27     900.00       11.38     1000.00       11.56     1100.00       11.73
 1200.00      11.91    1300.00       11.97     1400.00       12.05
  1440.00          44
    0.50      11.59       1.00       11.27        1.50       10.99        2.00       10.81
    2.50      10.71       3.00       10.59        3.50       10.51        4.00       10.42
    4.50      10.33       5.00       10.28        6.00       10.18        7.00       10.08
    8.00      10.01       9.00        9.94       10.00        9.88       12.00        9.78
   14.00       9.61      16.00        9.61       18.00        9.55       20.00        9.45
   25.00       9.36      30.00        9.25       35.00        9.18       40.00        9.10
   45.00       9.05      50.00        9.00       60.00        8.88       70.00        8.79
   80.00       8.69      90.00        8.64      100.00        8.59      120.00        8.47
  140.00       8.39     160.00        8.29      180.00        8.18      200.00        8.09
  250.00       7.92     300.00        7.71      350.00        7.55      400.00        7.38
  450.00       7.20     500.00        7.09      600.00        6.87      700.00        6.63
```

Appendix B. Sample Input Data

FILE: TW5OW5 OUTPUT A NEW JERSEY DEPARTMENT OF TRANSPORTATION - CMS PAGE 001

FEBRUARY 8, 1983

&GREENPOND TW5 OW5

R = 75.00 FEET
PUMPING RATE = 375.00 GPM
STATIC WATER LEVEL = 3.94 FEET

TIME-DRAWDOWN DATA
OF DATA PAIRS = 51

#	TIME (MIN)	DTW (FEET)	DRAWDOWN	#	TIME (MIN)	DTW (FEET)	DRAWDOWN	#	TIME (MIN)	DTW (FEET)	DRAWDOWN
1	0.50	4.60	0.66	18	16.00	6.71	2.77	35	180.00	9.37	5.43
2	1.00	4.94	1.00	19	18.00	6.82	2.88	36	200.00	9.53	5.59
3	1.50	5.16	1.22	20	20.00	6.92	2.98	37	250.00	9.80	5.86
4	2.00	5.33	1.39	21	25.00	7.13	3.19	38	300.00	10.04	6.10
5	2.50	5.50	1.56	22	30.00	7.28	3.34	39	350.00	10.23	6.29
6	3.00	5.58	1.64	23	35.00	7.45	3.51	40	400.00	10.45	6.51
7	3.50	5.64	1.70	24	40.00	7.61	3.67	41	450.00	10.51	6.57
8	4.00	5.72	1.78	25	45.00	7.72	3.78	42	500.00	10.66	6.72
9	4.50	5.79	1.85	26	50.00	7.83	3.89	43	600.00	10.90	6.96
10	5.00	5.86	1.92	27	60.00	8.03	4.09	44	700.00	11.06	7.12
11	6.00	5.96	2.02	28	70.00	8.20	4.26	45	800.00	11.27	7.33
12	7.00	6.08	2.14	29	80.00	8.35	4.41	46	900.00	11.38	7.44
13	8.00	6.18	2.24	30	90.00	8.50	4.56	47	1000.00	11.56	7.62
14	9.00	6.26	2.32	31	100.00	8.64	4.70	48	1100.00	11.73	7.79
15	10.00	6.33	2.39	32	120.00	8.87	4.93	49	1200.00	11.91	7.97
16	12.00	6.48	2.54	33	140.00	9.06	5.12	50	1300.00	11.97	8.03
17	14.00	6.61	2.67	34	160.00	9.24	5.30	51	1400.00	12.05	8.11

TIME-RECOVERY DATA:
OF DATA PAIRS = 44
TIME PUMPING STOPPED = 1440.00 MIN

#	TIME (MIN)	DTW (FEET)	DRAWDOWN	#	TIME (MIN)	DTW (FEET)	DRAWDOWN	#	TIME (MIN)	DTW (FEET)	DRAWDOWN
1	0.50	11.59	7.65	16	12.00	9.78	5.84	31	100.00	8.59	4.65
2	1.00	11.27	7.33	17	14.00	9.61	5.67	32	120.00	8.47	4.53
3	1.50	10.99	7.05	18	16.00	9.61	5.67	33	140.00	8.39	4.45
4	2.00	10.81	6.87	19	18.00	9.55	5.61	34	160.00	8.29	4.35
5	2.50	10.71	6.77	20	20.00	9.45	5.51	35	180.00	8.18	4.24
6	3.00	10.59	6.65	21	25.00	9.36	5.42	36	200.00	8.09	4.15
7	3.50	10.51	6.57	22	30.00	9.25	5.31	37	250.00	7.92	3.98
8	4.00	10.42	6.48	23	35.00	9.18	5.24	38	300.00	7.71	3.77
9	4.50	10.33	6.39	24	40.00	9.10	5.16	39	350.00	7.55	3.61
10	5.00	10.28	6.34	25	45.00	9.05	5.11	40	400.00	7.38	3.44
11	6.00	10.18	6.24	26	50.00	9.00	5.06	41	450.00	7.20	3.26
12	7.00	10.08	6.14	27	60.00	8.88	4.94	42	500.00	7.09	3.15
13	8.00	10.01	6.07	28	70.00	8.79	4.85	43	600.00	6.87	2.93
14	9.00	9.94	6.00	29	80.00	8.69	4.75	44	700.00	6.63	2.69
15	10.00	9.88	5.94	30	90.00	8.64	4.70				

THEIS ANALYSIS

TIME CUTOFF LIMITS:
LOW = 0.50 MINUTES
HIGH = 1400.00 MINUTES
TRANSMISSIVITY = 48580. GPD/FT
STORAGE = 0.752E-03

JACOB ANALYSIS

TIME CUTOFF LIMITS:
LOW = 0.50 MINUTES
HIGH = 1400.00 MINUTES
TRANSMISSIVITY = 42418. GPD/FT
STORAGE = 0.117E-02

JACOB ANALYSIS

TIME CUTOFF LIMITS:
LOW = 0.50 MINUTES
HIGH = 10.00 MINUTES
TRANSMISSIVITY = 73964. GPD/FT
STORAGE = 0.485E-03

TIME-CALCULATED RECOVERY VALUES BASED ON ABOVE ANALYSIS

NO	T	RECOV	NO	T	RECOV	NO	T	RECOV
1	0.50	-2.42	16	12.00	-0.61	31	100.00	0.62
2	1.00	-2.10	17	14.00	-0.44	32	120.00	0.74
3	1.50	-1.82	18	16.00	-0.44	33	140.00	0.83
4	2.00	-1.64	19	18.00	-0.38	34	160.00	0.94
5	2.50	-1.54	20	20.00	-0.28	35	180.00	1.06
6	3.00	-1.42	21	25.00	-0.18	36	200.00	1.15
7	3.50	-1.34	22	30.00	-0.07	37	250.00	1.34
8	4.00	-1.25	23	35.00	0.00	38	300.00	1.57
9	4.50	-1.16	24	40.00	0.08	39	350.00	1.74
10	5.00	-1.11	25	45.00	0.13	40	400.00	1.93
11	6.00	-1.01	26	50.00	0.19	41	450.00	2.12
12	7.00	-0.91	27	60.00	0.31	42	500.00	2.25
13	8.00	-0.84	28	70.00	0.40	43	600.00	2.50
14	9.00	-0.77	29	80.00	0.51	44	700.00	2.77
15	10.00	-0.71	30	90.00	0.56			

CALCULATED RECOVERY ANALYSIS BASED ON PREVIOUS JACOB ANALYSIS

TIME CUTOFF LIMITS:
LOW = 0.50 MINUTES
HIGH = 700.00 MINUTES

TRANSMISSIVITY = 64647. GPD/FT
STORAGE = 0.684E-01

JACOB ANALYSIS

TIME CUTOFF LIMITS:
LOW = 50.00 MINUTES
HIGH = 1400.00 MINUTES
TRANSMISSIVITY = 33708. GPD/FT
STORAGE = 0.311E-02

TIME-CALCULATED RECOVERY VALUES BASED ON ABOVE ANALYSIS

NO	T	RECOV	NO	T	RECOV	NO	T	RECOV
1	0.50	0.45	16	12.00	2.27	31	100.00	3.54
2	1.00	0.77	17	14.00	2.44	32	120.00	3.67
3	1.50	1.05	18	16.00	2.45	33	140.00	3.77
4	2.00	1.23	19	18.00	2.51	34	160.00	3.89
5	2.50	1.33	20	20.00	2.61	35	180.00	4.01
6	3.00	1.45	21	25.00	2.70	36	200.00	4.12
7	3.50	1.53	22	30.00	2.82	37	250.00	4.33
8	4.00	1.63	23	35.00	2.89	38	300.00	4.57
9	4.50	1.72	24	40.00	2.98	39	350.00	4.77
10	5.00	1.77	25	45.00	3.03	40	400.00	4.97
11	6.00	1.87	26	50.00	3.09	41	450.00	5.19
12	7.00	1.97	27	60.00	3.21	42	500.00	5.33
13	8.00	2.04	28	70.00	3.31	43	600.00	5.62
14	9.00	2.11	29	80.00	3.42	44	700.00	5.92
15	10.00	2.17	30	90.00	3.48			

CALCULATED RECOVERY ANALYSIS BASED ON PREVIOUS JACOB ANALYSIS

TIME CUTOFF LIMITS:
LOW = 0.50 MINUTES
HIGH = 700.00 MINUTES
TRANSMISSIVITY = 61836. GPD/FT
STORAGE = 0.969E-03

CALCULATED RECOVERY ANALYSIS BASED ON PREVIOUS JACOB ANALYSIS

TIME CUTOFF LIMITS:
LOW = 0.50 MINUTES
HIGH = 100.00 MINUTES
TRANSMISSIVITY = 72945. GPD/FT
STORAGE = 0.676E-03

T/T'-RESIDUAL DRAWDOWN VALUES FOR THE FOLLOWING ANALYSIS

#	T/T'	RESID	#	T/T'	RESID	#	T/T'	RESID

1	2881.00	7.65	16	121.00	5.84	31	15.40	4.65
2	1441.00	7.33	17	103.86	5.67	32	13.00	4.53
3	961.00	7.05	18	91.00	5.67	33	11.29	4.45
4	721.00	6.87	19	81.00	5.61	34	10.00	4.35
5	577.00	6.77	20	73.00	5.51	35	9.00	4.24
6	481.00	6.65	21	58.60	5.42	36	8.20	4.15
7	412.43	6.57	22	49.00	5.31	37	6.76	3.98
8	361.00	6.48	23	42.14	5.24	38	5.80	3.77
9	321.00	6.39	24	37.00	5.16	39	5.11	3.61
10	289.00	6.34	25	33.00	5.11	40	4.60	3.44
11	241.00	6.24	26	29.80	5.06	41	4.20	3.26
12	206.71	6.14	27	25.00	4.94	42	3.88	3.15
13	181.00	6.07	28	21.57	4.85	43	3.40	2.93
14	161.00	6.00	29	19.00	4.75	44	3.06	2.69
15	145.00	5.94	30	17.00	4.70			

RESIDUAL DRAWDOWN ANALYSIS

TIME CUTOFF LIMITS:
LOW = 3.06 MINUTES
HIGH = 2880.99 MINUTES
TRANSMISSIVITY = 64599. GPD/FT

RESIDUAL DRAWDOWN ANALYSIS

TIME CUTOFF LIMITS:
LOW = 20.00 MINUTES
HIGH = 2881.00 MINUTES
TRANSMISSIVITY = 74018. GPD/FT

```
&CONTROL OFF NOMSG
CP LINK DEPGIS 193 193 RR READ
ACC 193 C
CP TERM LINESIZE 255
CP TERM ATTN OFF
CP SPOOL PRINTER CONT
SET BLIP OFF
CP TERM LINEDEL OFF
CP TERM LINEND OFF
CP TERM ESCAPE OFF
CP TERM BELL ON
&BEGTYPE
         C A P T D
  COMPUTER ANALYSIS OF PUMP TEST DATA
&ENDTYPE
-EXP
&BEGTYPE
OPTIONS:        1. RUN CAPTD ON AN EXISTING DATA FILE.
                2. LIST ALL CAPTD DATA FILES.
                3. CREATE A NEW CAPTD DATA FILE.
                4. END
INPUT THE NUMBER OF THE REQUESTED OPTION.
&ENDTYPE
&READ VARS &1
&IF &1 EQ 1 &GOTO -RUN
&IF &1 EQ 2 &GOTO -LIST
&IF &1 EQ 3 &GOTO -NEW
&IF &1 EQ 4 &GOTO -STOP
&IF &1 EQ STOP GO TO -STOP
&IF &1 EQ END  GO TO -STOP
&IF &1 EQ QUIT GO TO -STOP
&GOTO -EXP
-RUN
&TYPE
&TYPE WHAT IS THE NAME OF THE DATA FILE?
&READ VARS &NAME
FI 4 DISK &NAME CAPDATA A
FI 6 DISK &NAME OUTPUT A (RECFM FA LRECL 133 BLOCK 132
LOAD CAPTD PLOTTEK2 PLOTCHAR BLOCK(START
PRINT &NAME OUTPUT A (CC)
&TYPE
&TYPE THE FILE  "&NAME OUTPUT A"  HAS BEEN PRINTED OUT
&TYPE
&GOTO -EXP
-LIST
&TYPE
&TYPE  THE EXISTING DATA FILES FOR CAPTD ARE:
&TYPE
L * CAPDATA A
&GOTO -EXP
-NEW
&TYPE
&TYPE WHAT IS THE FILE NAME OF THE NEW DATA FILE TO BE CREATED?
&TYPE  REMEMBER THIS NAME. WRITE IT DOWN.
&READ VARS &NAME
```

Appendix D1. EXEC Listing

```
STATE &NAME CAPDATA A
&IF &RETCODE NE 0 &GOTO -OK
&TYPE
&TYPE THE FILE "&NAME CAPDATA A " ALREADY EXISTS.
&TYPE   START THE PROCESS OVER AGAIN.
&GOTO -EXP
-OK
&TYPE
FI 6 DISK &NAME CAPDATA A (RECFM FA LRECL 80 BLOCK 80
LOAD CAPGO (START
&GOTO -EXP
-STOP
REL C
DET 193
CP TERM BELL OFF
CP TERM LINESIZE 80
CP SPOOL PRINTER NOCONT
CP TERM ATTN ON
CP SET LINEDIT ON
SET BLIP ON
&EXIT
```

```
      REAL*4 T(200),S(200)
      REAL YN,YES/'YES '/,NO/'NO  '/
      LOGICAL*1 TITLE(25),AMP/'&'/
C
C---INPUT TITLE AND PARAMTERS
      WRITE (5,501)
      READ (5,502) (TITLE(I),I=2,25)
      TITLE(1)=AMP
      WRITE (6,601) (TITLE(I),I=1,25)
   10 WRITE (5,503)
      READ (5,*) R
      WRITE (5,504)
      READ (5,*) Q
      WRITE (5,505)
      READ (5,*) STATIC
      WRITE (5,507) R,Q,STATIC
      WRITE (5,508)
      READ (5,509) YN
      IF (YN.EQ.YES) GO TO 10
C---INPUT THE DATA POINTS
      WRITE (5,510)
      N=0
   20 N=N+1
      READ (5,*,END=30) T(N),S(N)
      GO TO 20
   30 REWIND 5
C---CHECK THE DATA POINTS
      N=N-1
      CALL CHECK (T,S,N)
C---OUTPUT THE DRAWDOWN DATA
      WRITE (6,602) R,Q,STATIC,N
      WRITE (6,603) (T(I),S(I),I=1,N)
C---RECOVERY DATA?
      WRITE (5,511)
      READ (5,509) YN
      IF (YN.NE.YES) GO TO 999
C---INPUT PARAMTER
      WRITE (5,512)
      READ (5,*) TPUMP
      WRITE (5,510)
C---INPUT THE DATA POINTS
      NREC=0
   40 NREC=NREC+1
      READ (5,*,END=50) T(NREC),S(NREC)
      GO TO 40
   50 REWIND 5
C---CHECK THE DATA POINTS
      NREC=NREC-1
      CALL CHECK (T,S,NREC)
      WRITE (6,605) TPUMP, NREC
      WRITE (6,603) (T(I),S(I),I=1,NREC)
C---FORMAT STATEMENTS
  501 FORMAT (/2X,'WHAT IS TITLE FOR PLOT (24 CHARACTERS OR LESS)?')
  502 FORMAT (25A1)
  503 FORMAT (2X,'WHAT IS R (FEET)?')
  504 FORMAT (2X,'WHAT IS Q (GPM)?')
  505 FORMAT (2X,'WHAT WAS STATIC DEPTH TO WATER (IN FEET) BEFORE',/
     &5X,'PUMPING BEGAN?')
```

Appendix D2. CAPGO List

```
0048           507 FORMAT (2X,'R,Q,STATIC:',/3F10.2)
0049           508 FORMAT (2X,'CHANGE ANY OF THESE VALUES? (YES/NO)')
0050           509 FORMAT (A4)
0051           510 FORMAT (/2X,'INPUT THE TIME-DEPTH TO WATER DATA PAIRS.  AFTER',/
                 &  2X,'EACH PAIR HIT THE "RETURN" KEY.  TO SIGNAL THE END OF THE',/
                 &  2X,'DATA, INPUT A LINE WITH NO DATA ON IT.')
0052           511 FORMAT (/2X,'ANY RECOVERY DATA? (YES/NO)')
0053           512 FORMAT (2X,'HOW LONG DID THE PUMPING PERIOD LAST? (MINUTES)')
0054           601 FORMAT (25A1)
0055           602 FORMAT (3F10.2,I10)
0056           603 FORMAT (8F9.2)
0057           604 FORMAT (F8.2,I10)
0058           605 FORMAT (F10.2,I10)
              C
0059           999 STOP
0060               END
```

```
0001             SUBROUTINE CHECK (T,S,N)
           C
0002             REAL T(200),S(200)
0003             REAL YN, YES/'YES '/, NO/'NO  '/
0004             INTEGER IN(200),KK(3)
           C
0005             DO 10 I=1,200
0006        10     IN(I)=I
           C---OUTPUT DATA POINTS TO SCREEN
0007          20 WRITE (5,501)
0008             NROW=N/3
0009             IREM=N-3*NROW
0010             K1=NROW+1
0011             KK(1)=0
0012             KK(2)=NROW
0013             IF (IREM.GE.1) KK(2)=KK(2)+1
0014             KK(3)=KK(2)+NROW
0015             IF (IREM.GE.2) KK(3)=KK(3)+1
0016             WRITE (5,502) ((IN(I+KK(J)),T(I+KK(J)),S(I+KK(J)),J=1,3),I=1,NROW)
0017             IF (IREM.GT.0) WRITE (5,502) (IN(K1+KK(J)),T(K1+KK(J)),S(K1+KK(J))
                &  ,J=1,IREM)
           C---CHANGE DATA?
0018             WRITE (5,503)
0019             READ (5,510) YN
0020             IF (YN.NE.YES) GO TO 999
           C---ADD A DATA PAIR
0021          30 WRITE (5,504)
0022             READ (5,510) YN
0023             IF (YN.NE.YES) GO TO 40
0024             N=N+1
0025             WRITE (5,505)
0026             READ (5,*) T(N),S(N)
0027             GO TO 30
           C---CHANGE A DATA PAIR
0028          40 WRITE (5,506)
0029             READ (5,510) YN
0030             IF (YN.NE.YES) GO TO 50
0031             WRITE (5,507)
0032             READ (5,*) J
0033             WRITE (5,508) J
0034             READ (5,*) T(J),S(J)
0035             GO TO 40
           C---DELETE A DATA PAIR?
0036          50 WRITE (5,509)
0037             READ (5,510) YN
0038             IF (YN.NE.YES) GO TO 20
0039             WRITE (5,511)
0040             READ (5,*) J
0041             DO 60 I=J,N
0042               T(I)=T(I+1)
0043               S(I)=S(I+1)
0044          60 CONTINUE
0045             N=N-1
0046             GO TO 50
           C---FORMAT STATEMENTS
0047         501 FORMAT (/2X,'DATA POINTS',/3(3X,'#',5X,'TIME',5X,'DTW',3X))
0048         502 FORMAT (3(1X,I3,2X,F8.2,2X,F6.2,2X))
0049         503 FORMAT (2X,'CHANGE ANY DATA? (YES/NO)')
```

```
0050        504 FORMAT (2X,'ADD A DATA PAIR? (YES/NO)')                        CAP01330
0051        505 FORMAT (2X,'INPUT THIS NEW DATA PAIR.')                        CAP01340
0052        506 FORMAT (2X,'CHANGE A DATA PAIR? (YES/NO)')                     CAP01350
0053        507 FORMAT (2X,'WHAT IS # OF DATA PAIR TO CHANGE?')                CAP01360
0054        508 FORMAT (2X,'WHAT IS CORRECT TIME AND DTW FOR #',I3,'?')        CAP01370
0055        509 FORMAT (2X,'DELETE A DATA PAIR? (YES/NO)')                     CAP01380
0056        510 FORMAT (A4)                                                    CAP01390
0057        511 FORMAT (2X,'WHAT IS # OF THE DATA PAIR TO DELETE?')            CAP01400
          C                                                                    CAP01410
0058        999 RETURN                                                         CAP01420
0059            END                                                            CAP01430
```

```
C
C
C
C
C
C    CCCCCCCC          AAA        PPPPPPPPP    TTTTTTTTTTT    DDDDDDDDD
C   CCCCCCCCCC        AA AA       PPPPPPPPPP   TTTTTTTTTTT    DDDDDDDDDD
C   CC      CC       AA   AA      PP      PP       TT         DD      DD
C   CC              AA     AA     PP      PP       TT         DD      DD
C   CC             AA       AA    PPPPPPPPP        TT         DD      DD
C   CC             AAAAAAAAAAA    PPPPPPPP         TT         DD      DD
C   CC             AAAAAAAAAAA    PP               TT         DD      DD
C   CC      CC     AA       AA    PP               TT         DD      DD
C   CC      CC     AA       AA    PP               TT         DD      DD
C   CCCCCCCCCC     AA       AA    PP               TT         DDDDDDDDDD
C    CCCCCCCC      AA       AA    PP               TT         DDDDDDDDD
C
C
C
C
C
C
C           **  COMPUTER ANALYSIS OF PUMP TEST DATA **
C
C
C
C
C     A GRAPHICAL METHOD FOR ANALYZING TIME-DRAWDOWN DATA AND PRODUCING
C     REPORT QUALITY GRAPHS. CAPTD CAN USE THE THEIS, JACOB, CALCULATED
C     RECOVERY AND RESIDUAL DRAWDOWN METHODS FOR CALCULATING TRANSMISSI-
C     VITY AND STORAGE. SEE THE SUBROUTINE INPUT FOR A DESCRIPTION OF
C     THE INPUT DATA. SEE THE USERS MANUAL FOR AN EXPLINATION OF HOW
C     TO USE CAPTD AND THE ASSUMPTIONS INVOLVED.
C
C
C   SUBROUTINES:
C     1.  INPUT   -  READ IN THE DATA
C     2.  THEIS   -  THEIS ANALYSIS COMMANDS
C     3.  THSREC  -  CALCULATED RECOVERY ANALYSIS BASED ON THEIS
C     4.  JAKE    -  JACOB ANALYSIS COMMANDS
C     5.  CALREC  -  CALCULATED RECOVERY ANALYSIS BASED ON JACOB
C     6.  RESID   -  RESIDUAL DRAWDOWN ANALYSIS
C     7.  TCHECK  -  CALCULATES T AND S VALUES AFTER LINEAR REGRESSION
C     8.  PLTLIN  -  DATA AND LINE PLOTTING
C     9.  SLPLOT  -  SEMI-LOG PLOTTING
C    10.  RESULT  -  DISPLAY RESULTS ON THE PLOT
C    11.  LINREG  -  CALCULATIONS FOR LINEAR REGRESSION
C    12.  LLPLOT  -  LOG-LOG PLOTTING
C    13.  THSPLT  -  THEIS ANALYSIS CURVE PLOTTING
C    14.  PNTOUT  -  PRINTING RESULTS
C    15.  TABLE   -  TABLE GENERATION
C    16.  SEARCH  -  SEARCHING THEIS SOLUTION SPACE
C    17.  SETUP   -  PREPARE FOR CALCULATIONS FOR FINDING THEIS SOLUTION
C    18.  MIN     -  FIND MIN OF 9 NUMBERS
C    19.  COMPUT  -  COMPUTE THE RESIDUAL OF 9 THEIS FITS
```

Appendix D3. CAPTD Listing

```
C     20.  E1       -  EXPONENTIAL INTEGRAL
C
C
C
C     REFERENCES:
C        "ANALYSIS AND EVALUATION OF PUMPING TEST DATA," G. P. KRUSEMAN
C           AND N.A. DE RIDDER, 1979.
C
C        "GROUND WATER AND WELLS," JOHNSON DIVISION, UOP INC., 1975.
C
C
C     PROGRAMMER:
C           JEFFREY L. HOFFMAN
C           NEW JERSEY DEPARTMENT OF ENVIRONMENTAL PROTECTION
C           DIVISION OF WATER RESOURCES
C           WATER QUALITY MANAGEMENT ELEMENT
C           BUREAU OF GROUND WATER MANAGEMENT
C           GROUND WATER RESOURCE EVALUATION SECTION
C           DECEMBER 1982
C
C           REVISED JUNE 1983 TO ACCOUNT FOR NEW GRAPHICS PROGRAMS
C
C
C
C-----------------------------------------------------------------------
C
      REAL T(200),S(200),LT(200),T2(200),S2(200),LT2(200)
      REAL LS(200),ERR(200),LTX(200),LSX(200)
      REAL YN, YES/'YES '/, NO/'NO  '/
      LOGICAL*1 TITLE(25),RECOVR
C............GRAPHING ARRAYS AND COMMON BLOCKS
      DIMENSION IXY(2,6000), IPOINT(2,6000)
      LOGICAL TEK, CAL
      LOGICAL*1 LIFT(6000), STRING(130), ARG(130)
      COMMON /SPOT/ TEK, CAL, NPNTS, IXY, LIFT
      COMMON /SPARE/ IPOINT, IPTR
      COMMON /DEVICE/ IXLIM, IYLIM, MAXXO, MAXYO, IXMIN, IYMIN
      COMMON /PARS/ LEN, INDEX, NCHARS, NUMLFT, ARG, STRING
C............END OF GRAPHICS REQUIREMENTS
      COMMON /TITLES/ TITLE
      COMMON /PLT/ X0,X1,Y0,Y1
      COMMON /DATA/ N,R,Q,TPUMP,NREC,STOR,TRAN,TRAN1,TRAN2,A,B,T0
      COMMON /POINTS/ T,S,LT,T2,LS,ST,LT2
      COMMON /RANGE/ TLO,THI,SLO,SHI,T2LO,T2HI,S2LO,S2HI
      COMMON /IDATA/ ERR,NPTS,NSUM
      COMMON /ADD/ XADD,YADD,DX,DY,XORIG,YORIG
      COMMON /PERR/ R1,R2,R3,R4,R5,R6,R7,R8,RR
      COMMON /XTRA/ LTX,LSX
C
C-----------------------------------------------------------------------
C
C---PREPARE PLOTTER TERMINAL
      CALL PLTSET
      CALL PLTERA
C---SET UP PHYSICAL DIMENSIONS OF GRAPH, IN INCHES
```

```
      XC=1.7
      X1=13.7
      YC=1.2
      Y1=5.2
C---READ IN DATA
      CALL INPUT(RECOVR)
C---THEIS ANALYSIS?
   10 WRITE (5,501)
      READ (5,510) YN
      IF (YN.EQ.YES) CALL THEIS(RECOVR)
C---JACOB ANALYSIS?
   20 WRITE (5,502)
      READ (5,510) YN
      IF (YN.EQ.YES) CALL JAKE(RECOVR)
      IF (.NOT.RECOVR) GO TO 30
C---RESIDUAL DRAWDOWN ANALYSIS?
      WRITE (5,503)
      READ (5,510) YN
      IF (YN.EQ.YES) CALL RESID
C---START PROCESS ALL OVER AGAIN?
   30 WRITE (5,504)
      READ (5,510) YN
      IF (YN.EQ.YES) GO TO 10
C
C
  501 FORMAT (/2X,'THEIS ANALYSIS? (YES/NO)')
  502 FORMAT (/2X,'JACOB ANALYSIS? (YES/NO)')
  503 FORMAT (/2X,'RESIDUAL DRAWDOWN ANALYSIS? (YES/NO)')
  504 FORMAT (/2X,'START PROCESS ALL OVER AGAIN? (YES/NO)')
  510 FORMAT (A4)
C
  999 STOP
      END
      SUBROUTINE INPUT(RECOVR)
C
C
C                   INPUT ALL DATA
C
C   IMPORTANT NOTE:
C          DATA IS READ FROM UNIT 4
C          UNIT 5 IS THE TERMINAL AND IS INTERACTIVELY USED
C          WRITTEN OUTPUT SENT TO UNIT 6
C
C
C CARD 1: TITLE
C      UP TO 25 CHARACTERS
C      THE FIRST CHARACTER IS A PLOTTING CONTROL CHARACTER
C      THE USE OF A '&' OR ' ' IS RECOMMENDED
C
C CARD 2:
C    R-        DISTANCE FROM PUMPING WELL TO OBSERVATION WELL (IN FEET)
C    Q-        PUMPING RATE (IN GALLONS PER MINUTE)
C    STATIC-   STATIC WATER DEPTH IN WELL BEFORE PUMPING BEGINS
C    N-        NUMBER OF TIME, DRAWDOWN DATA POINTS
C
```

```
C CARD 3  TIME,DRAWDOWN DATA
C      REALLY N CARDS HERE
C      EACH CARD HAS ON IT ONE DATA PAIR
C      THE FIRST DATUM IS THE NUMBER OF MINUTES SINCE PUMPING STARTED
C      THE SECOND DATUM IS THE DEPTH TO WATER IN THE WELL AT THAT TIME
C
C CARD 4
C    TPUMP-    TIME AT WHICH PUMPING STOPPED   (MINUTES)
C    NREC-     NUMBER OF TIME,RECOVERY DATA POINTS
C
C CARD 5
C      REALLY NREC CARDS HERE
C      EACH HAS ON IT ONE DATA PAIR
C      THE FIRST DATUM IS THE NUMBER OF MINUTES SINCE PUMPING STOPPED
C      THE SECOND DATUM IS THE DEPTH TO WATER IN THE WELL AT THAT TIME
C
C
C  VECTORS CREATED BY THIS SUBROUTINE
C     T(I)   -  TIME SINCE PUMPING BEGAN FOR ITH DATA PAIR , I=1 TO N
C     LT(I)  -  LOG (T(I))
C     S(I)   -  DRAWDOWN IN WELL FOR ITH DATA PAIR
C     LS(I)  -  LOG (S(I))
C     T2(J)  -  TIME SINCE PUMPING STOPPED FOR JTH DATA PAIR, J=1,NREC
C     LT2(J) -  LOG (T2(J))
C     S2(J)  -  DRAWDOWN IN WELL FOR JTH DATA PAIR
C
C  SCALARS CREATED BY THIS SUBROUTINE
C     TLO    -  MINIMUM VALUE OF THE VECTOR T
C     THI    -  MAXIMUM VALUE OF THE VECTOR T
C     SLO    -  MINIMUM VALUE OF THE VECTOR S
C     SHI    -  MAXIMUM VALUE OF THE VECTOR S
C     T2LO   -  MINIMUM VALUE OF THE VECTOR T2
C     T2HI   -  MAXIMUM VALUE OF THE VECTOR T2
C     S2LO   -  MINIMUM VALUE OF THE VECTOR S2
C     S2HI   -  MAXIMUM VALUE OF THE VECTOR S2
C
C  LOGICAL VARIBLES INITIALIZED BY THIS SUBROUTINE
C     RECOVR -  THIS IS .TRUE. IF TIME-RECOVERY DATA WAS READ IN
C               IF IS .FALSE. OTHERWISE
C
C
C-----------------------------------------------------------------------
C
      REAL T(200),S(200),LT(200),T2(200),S2(200),LT2(200),LS(200)
      REAL TEMP(200)
      INTEGER IN(200),KK(3)
      LOGICAL*1 TITLE(25),RECOVR
      DIMENSION IDATE(16)
      COMMON /TITLES/ TITLE
      COMMON /RANGE/ TLO,THI,SLO,SHI,T2LO,T2HI,S2LO,S2HI
      COMMON /DATA/ N,R,L,TPUMP,NREC,STOR,TRAN,TRAN1,TRAN2,A,B,TO
      COMMON /POINTS/ T,S,LT,LS,T2,S2,LT2
C
C-----------------------------------------------------------------------
C
```

```
C---DATE
      CALL MCDAYR(IDATE)
      WRITE (6,606) IDATE
C---INITIALIZE
      DO 5 I=1,200
    5   IN(I)=I
C
C---READ IN TIME, DRAWDOWN DATA
      READ (4,501) (TITLE(I),I=1,25)
      READ (4,*) R,Q,STATIC,N
      TLO=1000000.
      THI=-TLO
      SLO=TLO
      SHI=THI
      READ (4,*) (T(I),TEMP(I),I=1,N)
      DO 10 I=1,N
        S(I)=TEMP(I)-STATIC
        IF (T(I).LT.TLO) TLO=T(I)
        IF (T(I).GT.THI) THI=T(I)
        IF (S(I).LT.SLO) SLO=S(I)
        IF (S(I).GT.SHI) SHI=S(I)
        LT(I)=ALOG10(T(I))
        LS(I)=ALOG10(S(I))
   10 CONTINUE
      WRITE (6,600) (TITLE(I),I=1,25)
      WRITE (6,601) R,Q,STATIC
      WRITE (6,602) N
      WRITE (6,604)
      NROW=N/3
      IREM=N-3*NROW
      K=NROW+1
      KK(1)=0
      KK(2)=NROW
      IF (IREM.GE.1) KK(2)=KK(2)+1
      KK(3)=KK(2)+NROW
      IF (IREM.GE.2) KK(3)=KK(3)+1
      WRITE (6,605) ((IN(I+KK(J)),T(I+KK(J)),TEMP(I+KK(J)),S(I+KK(J)),
     &  J=1,3),I=1,NROW)
      IF (IREM.GT.0) WRITE (6,605) (IN(K+KK(J)),T(K+KK(J)),TEMP(K+KK(J))
     &  ,S(K+KK(J)),J=1,IREM)
C---READ IN TIME, RECOVERY DATA
      RECOVR=.FALSE.
      READ (4,*,END=999) TPUMP,NREC
      RECOVR=.TRUE.
      T2LO=1000000.
      T2HI=-T2LO
      S2LO=T2LO
      S2HI=T2HI
      READ (4,*) (T2(I),TEMP(I),I=1,NREC)
      DO 20 I=1,NREC
        S2(I)=TEMP(I)-STATIC
        IF (T2(I).LT.T2LO) T2LO=T2(I)
        IF (T2(I).GT.T2HI) T2HI=T2(I)
        IF (S2(I).LT.S2LO) S2LO=S2(I)
        IF (S2(I).GT.S2HI) S2HI=S2(I)
```

```
      LT2(I)=ALOG10(T2(I))
   20 CONTINUE
      WRITE (6,603) NREC,TPUMP
      WRITE (6,604)
      NROW=NREC/3
      IREM=NREC-NROW*3
      K=NROW+1
      KK(1)=0
      KK(2)=NROW
      IF (IREM.GE.1) KK(2)=KK(2)+1
      KK(3)=KK(2)+NROW
      IF (IREM.GE.2) KK(3)=KK(3)+1
      WRITE (6,605) ((IN(I+KK(J)),T2(I+KK(J)),TEMP(I+KK(J)),S2(I+KK(J)),
     & J=1,3),I=1,NROW)
      IF (IREM.GT.0) WRITE (6,605) (IN(K+KK(J)),T2(K+KK(J)),TEMP(K+KK(J
     & )),S2(K+KK(J)),J=1,IREM)
C
  501 FORMAT (25A1)
  600 FORMAT (//5X,25A1)
  601 FORMAT (
     &  /2X,'                        R = ',F10.2,' FEET',
     &  /2X,'            PUMPING RATE = ',F10.2,' GPM',
     &  /2X,'      STATIC WATER LEVEL = ',F10.2,' FEET')
  602 FORMAT (/5X,'TIME-DRAWDOWN DATA',
     &  /2X,'       # OF DATA PAIRS = ',I10)
  603 FORMAT (//5X,'TIME-RECOVERY DATA:',
     &  /2X,'       # OF DATA PAIRS = ',I10,
     &  /2X,'TIME PUMPING STOPPED = ',F10.2,' MIN')
  604 FORMAT (/3(3X,'#',3X,'TIME (MIN)',2X,'DTW (FEET)',2X,'DRAWDOWN',4X
     &  ))
  605 FORMAT (3(1X,I3,F10.2,2X,F9.2,2X,F9.2,7X))
  606 FORMAT (/50X,A3,15A1)
C
  999 RETURN
      END
      SUBROUTINE THEIS(RECOVR)
C
C
C        THEIS ANALYSIS
C
C
C
C------------------------------------------------------------------------
C
      REAL T(200),S(200),LT(200),LS(200),ERM(200),LTX(200),LSX(200)
      REAL T2(200),S2(200),LT2(200)
      REAL*8 L,WU,LINV
      LOGICAL*1 TITLE(25),RECOVR,RESTF,IEND
      LOGICAL*1 TX(30)/'T','I','M','E',' ','(','M','I','N','U','T','E',
     & 'S',')',16*' '/
      LOGICAL*1 TY(30)/'D','R','A','W','D','O','W','N',' ','(','F','E',
     & 'E','T',')',15*' '/
      LOGICAL*1 TA(30)/'T','H','E','I','S',' ','A','N','A','L','Y','S',
     & 'I','S',16*' '/
      REAL YN, YES/'YES '/, NO/'NO  '/
```

```
      COMMON /TITLES/ TITLE
      COMMON /DATA/ N,R,Q,TPUMP,NREC,STOR,TRAN,TRAN1,TRAN2,A,B,TC
      COMMON /IDATA/ ERR,NPTS,NSUM
      COMMON /POINTS/ T,S,LT,LS,T2,S2,LT2
      COMMON /RANGE/ TLO,THI,SLO,SHI,T2LO,T2HI,S2LO,S2HI
      COMMON /XTRA/ LTX,LSX
      COMMON /ACD/ XACC,YACC,DX,DY,AXORG,AYORG
C
C-----------------------------------------------------------------------
C
C          SET UP PARAMTERS FOR FIRST RUN
C
      RESTF=.FALSE.
C---SET UP MAX AND MIN
      XLO=ALOG10(TLO)
      XHI=ALOG10(THI)
      YLO=ALOG10(SLO)
      YHI=ALOG10(SHI)
      TMKLO=TLO
      TMKHI=THI
      GO TO 20
C
C         SET PARAMETERS FOR SUCCEEDING RUNS
C
C---WHAT ARE THE LOW AND HIGH CUTOFF TIMES FOR THIS RUN?
   10 WRITE (5,505)
      READ (5,*) TMKLO
      WRITE (5,506)
      READ (5,*) TMKHI
   20 NPTS=0
      XLOMRK=ALOG10(TMKLO)
      XHIMRK=ALOG10(TMKHI)
C---SET MINIMUM NUMBER OF SUMMATIONS
      NSUM=50
C---SET UP THE ANALYSIS ARRAYS
      DO 30 I=1,N
        IF (T(I).LT.TMKLO) GO TO 30
        IF (T(I).GT.TMKHI) GO TO 30
        NPTS=NPTS+1
        LTX(NPTS)=LT(I)
        LSX(NPTS)=LS(I)
   30 CONTINUE
C---CALL THE SEARCH SUBROUTINES
      CALL SEARCH(IEND)
      IF (IEND) GO TO 80
C---CALCULATE TRANSMISSIVITY AND STORAGE
      PI=3.1415926
      UINV=10.**(LTX(1)+XACC)
      U=1./UINV
      WU=LSX(1)+YACC
      WU=10.**(WU)
      SX1=10.**(LSX(1))
      TRAN=Q*WU/(4.*PI*SX1*7.48)
      TRAN1=1440.*TRAN
      TRAN2=10771.*TRAN
```

```
      TX1=10.**(LTX(1))                                                 VWS03860
      STOR=4.*TRAN*TX1*L/(R*R)                                          VWS03870
C---PRINT OUT THE ANSWER                                                VWS03880
      WRITE (6,601)                                                     VWS03890
      CALL PNTCUT(TMKLO,TMKHI,RESTF)                                    VWS03900
C---PLOT THIS ANALYSIS?                                                 VWS03910
      WRITE (5,502)                                                     VWS03920
      READ (5,510) YN                                                   VWS03930
      IF (YN.EQ.NO) GO TO 60                                            VWS03940
C---PREPARE PLOTTER                                                     VWS03950
      WRITE (5,503)                                                     VWS03960
      READ (5,510,END=40) TRASH                                         VWS03970
   40 REWIND 5                                                          VWS03980
C---CALL DATA PLOTTING ROUTINES                                         VWS03990
      CALL LLPLOT(XLO,XHI,YLO,YHI,XORIG,XMAX,XSCALE,YORIG,YMAX,YSCALE,  VWS04000
     & TITLE,TX,TY)                                                     VWS04010
      CALL THSPLT(LT,LS,N,XORIG,XMAX,XSCALE,YORIG,YMAX,YSCALE,XLOMRK,   VWS04020
     & XHIMRK)                                                          VWS04030
      CALL RESULT(TMKLO,TMKHI,TA,RESTF)                                 VWS04040
C---VIEW RESULTS                                                        VWS04050
      READ (5,510,END=50) TRASH                                         VWS04060
   50 REWIND 5                                                          VWS04070
      CALL PLTERA                                                       VWS04080
   60 IF (.NOT.RECOVR) GO TO 70                                         VWS04090
C---DO A CALCULATED RECOVERY ANALYSIS?                                  VWS04100
      WRITE (5,504)                                                     VWS04110
      READ (5,510) YN                                                   VWS04120
      IF (YN.EQ.YES) CALL THSREC                                        VWS04130
C---DO ANOTHER THEIS ANALYSIS?                                          VWS04140
   70 WRITE (5,507)                                                     VWS04150
      READ (5,510) YN                                                   VWS04160
      IF (YN.EQ.YES) GO TO 10                                           VWS04170
      GO TO 999                                                         VWS04180
C---THEIS CURVE MATCHING ATTEMPT TOOK TOO LONG                          VWS04190
   80 WRITE (5,508)                                                     VWS04200
C                                                                       VWS04210
C        FORMAT STATEMENTS                                              VWS04220
C                                                                       VWS04230
  502 FORMAT (/2X,'PLOT THIS ANALYSIS? (YES/NO)')                       VWS04240
  503 FORMAT (/2X,'PREPARE THE PLOTTER, TYPE "T", AND RETURN')          VWS04250
  504 FORMAT (/2X,'PERFORM A CALCULATED RECOVERY ANALYSIS ON THE RECOVERVWS04260
     &Y DATA',/5X,'BASED ON THE THEIS ANALYSIS RESULTS? (YES/NO)')      VWS04270
  505 FORMAT (/2X,'WHAT IS THE LOW TIME CUTOFF?')                       VWS04280
  506 FORMAT (/2X,'WHAT IS THE HIGH TIME CUTOFF?')                      VWS04290
  507 FORMAT (/2X,'DO ANOTHER THEIS ANALYSIS?')                         VWS04300
  508 FORMAT (/2X,'THEIS MATCH TOOK TOO LONG. ATTEMPT STOPS.')          VWS04310
  510 FORMAT (A4)                                                       VWS04320
  601 FORMAT (///1X,'THEIS ANALYSIS')                                   VWS04330
C                                                                       VWS04340
  999 RETURN                                                            VWS04350
      END                                                               VWS04360
      SUBROUTINE THSREC                                                 VWS04370
C                                                                       VWS04380
C         CALCULATED RECOVERY ANALYSIS BASED ON THE THEIS ANALYSIS      VWS04390
C                                                                       VWS04400
```

```
C
C
C     THE THEIS CURVE CALCULATED IN THE SUBROUTINE THEIS IS EXTENDED
C     AND CALCULATED RECOVERY VALUES ARE COMPUTED AND STORED IN THE
C     VECOR SCALC(J).
C
C-----------------------------------------------------------------------
C
      REAL T(200),S(200),LT(200),T2(200),S2(200),LT2(200),SCALC(200)
      REAL LS(200)
      REAL*8 U,WU
      LOGICAL*1 TX(30)/'R','E','C','O','V','E','R','Y',' ','T','I','M',
     & 'E',' ','(','M','I','N','U','T','E','S',')',7*' '/
      LOGICAL*1 TY(30)/'C','A','L','C','U','L','A','T','E','D',' ','R',
     & 'E','C','O','V','E','R','Y',' ','(','F','E','E','T',')',4*' '/
      LOGICAL*1 TA(30)/'C','A','L','C','U','L','A','T','E','D',' ','R',
     & 'E','C','O','V','E','R','Y',' ','-',' ','T','H','E','I','S',
     & 3*' '/
      LOGICAL*1 TITLE(25),LINE,RESTF
      REAL YN,YES/'YES '/,NO/'NO  '/
      COMMON /RANGE/ TLO,THI,SLO,SHI,T2LO,T2HI,S2LO,S2HI
      COMMON /POINTS/ T,S,LT,LS,T2,S2,LT2
      COMMON /DATA/ N,R,Q,TPUMP,NREC,STOR,TRAN,TRAN1,TRAN2,A,B,TO
      COMMON /TITLES/ TITLE
C
C-----------------------------------------------------------------------
C
C
C        INITIALIZE PARAMETERS AND SET UP FOR FIRST RUN
C
C---CALCULATE THE PREDICTED DRAWDOWN AND THE RECOVERY
      LINE=.TRUE.
      RESTF=.FALSE.
      SCLO=1000000.
      SCHI=0.01
      PI=3.1415926
      F1=Q/(4*PI*TRAN*7.48)
      F2=R*R*STOR/(4.*TRAN)
      DO 10 I=1,NREC
        U=F2/(T2(I)+TPUMP)
        CALL E1(U,WU)
        SPRED=F1*WU
        SNEW=SPRED - S2(I)
        SCALC(I)=SNEW
        IF (SNEW.LT.SCLO) SCLO=SNEW
        IF (SNEW.GT.SCHI) SCHI=SNEW
   10 CONTINUE
      WRITE (6,602)
      CALL TABLE(T2,SCALC,NREC)
      XLO=ALOG10(T2LO)
      XHI=ALOG10(T2HI)
      TMKLO=T2LO
      TMKHI=T2HI
      GO TO 30
C
```

```
C        SET PARAMETERS FOR SUCCEEDING RUNS
C
C---WHAT ARE THE LOW AND HIGH CUTOFF TIMES?
   20 WRITE (5,501)
      READ (5,*) TMKLO
      WRITE (5,502)
      READ (5,*) TMKHI
   30 XLOMRK=ALOG10(TMKLO)
      XHIMRK=ALOG10(TMKHI)
C---COMPUTE THE RESULTS AND PRINT THEM OUT
      CALL TCHECK(LT2,SCALC,NREC,XLOMRK,XHIMRK)
      WRITE (6,601)
      CALL PNTCLT(TMKLO,TMKHI,RESTF)
C---PLOT THIS ANSWER?
      WRITE (5,503)
      READ (5,510) YN
      IF (YN.EQ.NO) GO TO 60
C---PREPARE PLOTTER
      WRITE (5,504)
      READ (5,510,END=40) TRASH
   40 REWIND 5
C---CALL DATA PLOTTING SUBROUTINES
      CALL SUPLOT(XLO,XHI,SCLO,SCHI,XORIG,XMAX,XSCALE,YORIG,YMAX,
     & YSCALE,TITLE,TX,TY)
      CALL PLTLIN(LT2,SCALC,NREC,XLOMRK,XHIMRK,XORIG,XMAX,XSCALE,YORIG,
     & YMAX,YSCALE,A,B,LINE)
      CALL RESULT(TMKLO,TMKHI,TA,RESTF)
C---VIEW RESULTS
      READ (5,510,END=50) TRASH
   50 REWIND 5
      CALL FLTERA
C---DO ANOTHER CALCULATED RECOVERY ANALYSIS?
   60 WRITE (5,505)
      READ (5,510) YN
      IF (YN.EQ.YES) GO TO 20
C
C         FORMAT STATEMENTS
C
  501 FORMAT (/2X,'WHAT IS THE LOW TIME CUTOFF?')
  502 FORMAT (/2X,'WHAT IS THE HIGH TIME CUTOFF?')
  503 FORMAT (/2X,'PLOT THIS ANALYSIS? (YES/NO)')
  504 FORMAT (/2X,'PREPARE PLOTTER, TYPE "C", AND RETURN')
  505 FORMAT (/2X,'DO ANOTHER CALCULATED RECOVERY ANALYSIS? (YES/NO)')
  510 FORMAT (A4)
  601 FORMAT (///1X,'CALCULATED RECOVERY ANALYSIS BASED ON PREVIOUS THEI
     &S ANALYSIS')
  602 FORMAT (/5X,'TIME-CALCULATED RECOVERY VALUES BASED ON THE ABOVE AN
     &ALYSIS',//3(2X,'NO',7X,'T',9X,'RECOV',7X))
C
  999 RETURN
      END
      SUBROUTINE JAKE(RECOVR)
C
C
C                 JACOB ANALYSIS
```

```
C
C       DO ALL WORK NECESSARY TO PERFORM A JACOB ANALYSIS ON
C       THE INPUT DATA.
C
C
C-----------------------------------------------------------------------
C
      REAL T(200),S(200),LT(200),T2(200),S2(200),LT2(200),LS(200)
      LOGICAL*1 TITLE(25),RESTF,RECOVR,LINE
      LOGICAL*1 TX(30)/'T','I','M','E',' ','(','M','I','N','U','T','E',
     & 'S',')',16*' '/
      LOGICAL*1 TY(30)/'D','R','A','W','D','O','W','N',' ','(','F','E',
     & 'E','T',')',15*' '/
      LOGICAL*1 TA(30)/'J','A','C','O','B',' ','A','N','A','L','Y','S',
     & 'I','S',16*' '/
      REAL YN,YES/'YES '/,NO/'NO  '/
      COMMON /TITLES/ TITLE
      COMMON /DATA/ N,R,W,TPUMP,NREC,STOR,TRAN,TRAN1,TRAN2,A,B,TO
      COMMON /POINTS/ T,S,LT,LS,T2,ST,LT2
      COMMON /RANGE/ TLO,THI,SLO,SHI,T2LO,T2HI,S2LO,S2HI
C
C-----------------------------------------------------------------------
C
C       INITIALIZE PARAMETERS FOR FIRST JACOB ANALYSIS
C
      LINE=.TRUE.
      RESTF=.FALSE.
      XLO=ALOG10(TLO)
      XHI=ALOG10(THI)
      TMKLO=TLO
      TMKHI=THI
      GO TO 20
C
C       SET PARAMETERS FOR SUCCEEDING JACOB ANLAYSES
C
C---ANOTHER JACOB ANALYSIS?
   10 WRITE (5,501)
      READ (5,510) YN
      IF (YN.NE.YES) GO TO 999
C---WHAT ARE LOW AND HI LIMITING TIMES FOR THIS ANALYSIS?
      WRITE (5,502)
      READ (5,*) TMKLO
      WRITE (5,503)
      READ (5,*) TMKHI
   20 XLOMRK=ALOG10(TMKLO)
      XHIMRK=ALOG10(TMKHI)
C---COMPUTE THE RESULTS AND PRINT THEM OUT
      CALL TCHECK(LT,S,N,XLOMRK,XHIMRK)
      WRITE (6,601)
      CALL PNTOUT(TMKLO,TMKHI,RESTF)
C---PLOT THIS ANALYSIS?
      WRITE (5,504)
      READ (5,510) YN
      IF (YN.EQ.NO) GO TO 50
C---PREPARE PLOTTER
```

```
      WRITE (5,505)
      READ (5,510,END=30) TRASH
   30 REWIND 5
C---CALL DATA PLOTTING SUBROUTINES
      CALL SLPLOT(XLO,XHI,SLO,SHI,XORIG,XMAX,XSCALE,YORIG,YMAX,YSCALE,
     & TITLE,TX,TY)
      CALL PLTLIN(LT,S,N,XLOMRK,XHIMRK,XORIG,XMAX,XSCALE,YORIG,YMAX,
     & YSCALE,A,B,LINE)
      CALL RESULT(TMKLO,TMKHI,TA,RESTF)
C---VIEW RESULTS
      READ (5,510,END=40) TRASH
   40 REWIND 5
      CALL PLTERA
C---IF THERE IS RECOVERY DATA, DO A CALCULATED RECOVERY ANALYSIS?
   50 IF (.NOT.RECOVR) GO TO 10
      WRITE (5,506)
      READ (5,510) YN
      IF (YN.EQ.YES) CALL CALREC
      GO TO 10
C
C        FORMAT STATEMENTS
C
  501 FORMAT (/2X,'ANOTHER JACOB ANALYSIS? (YES/NO)')
  502 FORMAT (/2X,'WHAT IS THE LOW TIME? ')
  503 FORMAT (/2X,'WHAT IS THE HIGH TIME?')
  504 FORMAT (/2X,'PLOT THIS ANALYSIS? (YES/NO)')
  505 FORMAT (/2X,'PREPARE PLOTTER, TYPE "J", AND RETURN')
  506 FORMAT (/2X,'PERFORM A CALCULATED RECOVERY ANALYSIS ON THE RECOVER
     &Y DATA',/5X,'BASED ON THE JACOB ANALYSIS RESULTS? (YES/NO)')
  510 FORMAT (A4)
  601 FORMAT (///1X,'JACOB ANALYSIS')
C
  999 RETURN
      END
      SUBROUTINE CALREC
C
C
C     THIS SUBPROGRAM PERFORMS A CALCULATED RECOVERY ANALYSIS.
C     IT TAKES THE LINEAR VALUES OF A AND B (AS CALCULATED IN SUBROUTINE
C     JAKE) AND EXTENDS THE LINE. TIME-DRAWDOWN VALUES AT THE RECOVERY
C     TIMES ARE CALCULATED AND THE DIFFERENCE FROM THE OBSERVED RECOVERY
C     DATA CALCULATED.  THE CALCULATED RECOVERY VALUES ARE STORED IN
C     THE VECTOR SCALC(J).
C
C
C---------------------------------------------------------------------
C
      REAL T(200),S(200),LT(200),T2(200),S2(200),LT2(200),SCALC(200)
      REAL LS(200)
      LOGICAL*1 TX(30)/'R','E','C','O','V','E','R','Y',' ','T','I','M',
     & 'E',' ','(','M','I','N','U','T','E','S',')',7*' '/
      LOGICAL*1 TY(30)/'C','A','L','C','U','L','A','T','E','D',' ','R',
     & 'E','C','O','V','E','R','Y',' ','(','F','E','E','T',')',4*' '/
      LOGICAL*1 TA(30)/'C','A','L','C','U','L','A','T','E','D',' ','R',
     & 'E','C','O','V','E','R','Y',' ','-',' ','J','A','C','O','B',
```

```
     &   3*' '/
      LOGICAL*1 TITLE(25),LINE,RESTF
      REAL YN,YES/'YES '/,NO/'NO  '/
      COMMON /RANGE/ TLO,THI,SLO,SHI,T2LO,T2HI,S2LO,S2HI
      COMMON /POINTS/ T,S,LT,LS,T2,S2,LT2
      COMMON /DATA/ N,R,W,TPUMP,NREC,STOR,TRAN,TRAN1,TRAN2,A,B,TO
      COMMON /TITLES/ TITLE
C
C-----------------------------------------------------------------
C
C        INITIALIZE PARAMETES AND SET UP FOR FIRST RUN THROUGH
C
      LINE=.TRUE.
      RESTF=.FALSE.
      SCLO=1000000.
      SCHI=-SCLO
C---CALCULATE THE PREDICTED DRAWDOWN, AND THE RECOVERY
      DO 10 I=1,NREC
        STIME=A + B*ALOG10(T2(I)+TPUMP)
        SNEW=STIME - S2(I)
        SCALC(I)=SNEW
        IF (SNEW.LT.SCLO) SCLO=SNEW
        IF (SNEW.GT.SCHI) SCHI=SNEW
   10 CONTINUE
      WRITE (6,602)
      CALL TABLE(T2,SCALC,NREC)
      XLO=ALOG10(T2LO)
      XHI=ALOG10(T2HI)
      TMKLO=T2LO
      TMKHI=T2HI
      GO TO 30
C
C        SET PARAMETERS FOR SUCCEEDING RUNS
C
C---WHAT ARE THE LOW AND HI CUTOFF TIMES FOR THIS RUN?
   20 WRITE (5,501)
      READ (5,*) TMKLO
      WRITE (5,502)
      READ (5,*) TMKHI
   30 XLOMRK=ALOG10(TMKLO)
      XHIMRK=ALOG10(TMKHI)
C---COMPUTE THE RESULTS AND PRINT THEM OUT
      CALL TCHECK(LT2,SCALC,NREC,XLOMRK,XHIMRK)
      WRITE (6,601)
      CALL PNTOUT(TMKLO,TMKHI,RESTF)
C---PLOT THIS ANALYSIS?
      WRITE (5,503)
      READ (5,510) YN
      IF (YN.EQ.NO) GO TO 60
C---PREPARE THE PLOTTER
      WRITE (5,504)
      READ (5,510,END=40) TRASH
   40 REWIND 5
C---CALL THE DATA PLOTTING SUBROUTINES
      CALL SUPLOT(XLO,XHI,SCLO,SCHI,XORIG,XMAX,XSCALE,YORIG,YMAX,
```

```
     & YSCALE,TITLE,TX,TY)
      CALL PLTLIN(LT2,SCALC,NREC,XLOMRK,XHIMRK,XORIG,XMAX,XSCALE,YORIG,
     & YMAX,YSCALE,A,B,LINE)
      CALL RESULT(TMKLO,TMKHI,TA,RESTF)
C---VIEW THE RESULTS
      READ (5,510,END=50) TRASH
   50 REWIND 5
      CALL PLTERA
C---DO ANOTHER CALCULATED RECOVERY ANALYSIS?
   60 WRITE (5,505)
      READ (5,510) YN
      IF (YN.EQ.YES) GO TO 20
C
C        FORMAT STATEMENTS
C
  501 FORMAT (/2X,'WHAT IS THE LOW TIME CUTOFF?')
  502 FORMAT (/2X,'WHAT IS THE HIGH TIME CUTOFF?')
  503 FORMAT (/2X,'PLOT THIS ANALYSIS? (YES/NO)')
  504 FORMAT (/2X,'PREPARE PLOTTER, TYPE "C", AND RETURN')
  505 FORMAT (/2X,'DO ANOTHER CALCULATED RECOVERY ANALYSIS? (YES/NO)')
  510 FORMAT (A4)
  601 FORMAT (///1X,'CALCULATED RECOVERY ANALYSIS BASED ON PREVIOUS JACO
     &B ANALYSIS')
  602 FORMAT (/5X,'TIME-CALCULATED RECOVERY VALUES BASED ON ABOVE ANALYS
     &IS',//3(2X,'NO',7X,'T',9X,'RECOV',7X))
C
  999 RETURN
      END
      SUBROUTINE RESID
C
C
C               RESIDUAL DRAWDOWN ANALYSIS
C
C
C     THE VECTOR TCALC(J) CONTAINS THE T/T' VALUES.
C     TCLO AND TCHI CONTAIN THE EXTREME VALUES OF TCALC(J)
C
C
C
C-----------------------------------------------------------------------
C
      REAL T(200),S(200),LT(200),T2(200),S2(200),LT2(200),TCALC(200)
      REAL LS(200),TCC(200)
      LOGICAL*1 TX(30)/'R','A','T','I','O',' ','-',' ','T','/','T','P',
     & 'R','I','M','E',14*' '/
      LOGICAL*1 TY(30)/'R','E','S','I','D','U','A','L',' ','D','R','A',
     & 'W','D','O','W','N',' ','(','F','E','E','T',')',6*' '/
      LOGICAL*1 TA(30)/'R','E','S','I','D','U','A','L',' ','D','R','A',
     & 'W','D','O','W','N',13*' '/
      LOGICAL*1 TITLE(25),LINE,RESTF
      REAL YN,YES/'YES '/,NO/'NO  '/
      COMMON /RANGE/ TLO,THI,SLO,SHI,T2LO,T2HI,S2LO,S2HI
      COMMON /POINTS/ T,S,LT,LS,T2,S2,LT2
      COMMON /DATA/ N,R,Q,TPUMP,NREC,STOR,TRAN,TRAN1,TRAN2,A,B,TO
      COMMON /TITLES/ TITLE
```

```
C
C-----------------------------------------------------------------------
C
C       SET UP PARAMETERS FOR FIRST RUN
C
      LINE=.TRUE.
      RESTF=.TRUE.
C---CALCULATE THE T/T' VALUES AND STORE IN TCALC(J)
      TCLO=1000000.
      TCHI=-TCLO
      DO 10 I=1,NREC
        TCC(I)=(T2(I)+TPUMP)/T2(I)
        TCALC(I)=ALOG10(TCC(I))
        IF (TCALC(I).LT.TCLO) TCLO=TCALC(I)
        IF (TCALC(I).GT.TCHI) TCHI=TCALC(I)
   10 CONTINUE
      WRITE (6,602)
      CALL TABLE(TCC,S2,NREC)
      XLO=TCLO
      XHI=TCHI
      XLOMRK=TCLO
      XHIMRK=TCHI
      TMKLO=10.**(TCLO)
      TMKHI=10.**(TCHI)
      GO TO 30
C
C       SET UP PARAMTERS FOR SUCCEEDING RUNS
C
C---SET LOW AND HIGH T/T' VALUES
   20 WRITE (5,501)
      READ (5,*) TMKLO
      WRITE (5,502)
      READ (5,*) TMKHI
      XLOMRK=ALOG10(TMKLO)
      XHIMRK=ALOG10(TMKHI)
C---COMPUTE THE RESULTS AND PRINT THEM OUT
   30 CALL TCHECK(TCALC,S2,NREC,XLOMRK,XHIMRK)
      WRITE (6,601)
      CALL PNTOUT(TMKLO,TMKHI,RESTF)
C---PLOT THIS ANALYSIS?
      WRITE (5,503)
      READ (5,510) YN
      IF (YN.EQ.NO) GO TO 60
C---PREPARE PLOTTER
      WRITE (5,504)
      READ (5,510,END=40) TRASH
   40 REWIND 5
C---CALL THE DATA PLOTTING SUBROUTINES
      CALL SUPLOT(XLO,XHI,S2LO,S2HI,XORIG,XMAX,XSCALE,YORIG,YMAX,
     &  YSCALE,TITLE,TX,TY)
      CALL PLTLIN(TCALC,S2,NREC,XLOMRK,XHIMRK,XORIG,XMAX,XSCALE,YORIG,
     &  YMAX,YSCALE,A,B,LINE)
      CALL RESULT(TMKLO,TMKHI,TA,RESTF)
C---VIEW RESULTS
      READ (5,510,END=50) TRASH
```

```
   50 REWIND 5
      CALL FLTERA
C---DO ANOTHER RESIDUAL DRAWDOWN ANALYSIS?
   60 WRITE (5,505)
      READ (5,510) YN
      IF (YN.EQ.YES) GO TO 20
C
C         FORMAT STATEMENTS
C
  501 FORMAT (/2X,'WHAT IS T/T'' LOW?')
  502 FORMAT (/2X,'WHAT IS T/T'' HI?')
  503 FORMAT (/2X,'PLOT THIS ANALYSIS? (YES/NO)')
  504 FORMAT (/2X,'PREPARE PLOTTER, TYPE "R", AND RETURN')
  505 FORMAT (/2X,'DO ANOTHER RESIDUAL DRAWDOWN ANALYSIS? (YES/NO)')
  510 FORMAT (A4)
  601 FORMAT (///1X,'RESIDUAL DRAWDOWN ANALYSIS')
  602 FORMAT (/5X,'T/T''-RESIDUAL DRAWDOWN VALUES FOR THE FOLLOWING ANAL
     &YSIS',//3('   #        T/T''          RESID     ',3X))
C
  999 RETURN
      END
      SUBROUTINE TCHECK(LT,S,NUM,TLOMRK,THIMRK)
C
C               CALCULATION ROUTINE
C
C      DATA INPUT:
C
C        LT(I)   -  A VECTOR OF LENGTH NUM CONTAINING THE LOG TIME VALUES
C        S(I)    -  A VECTOR OF LENGTH NUM CONTAINING THE DRAWDOWN VALUES
C        NUM     -  NUMBER OF VALUES IN LT(I) AND S(I)    (NUM<=100)
C        TLOMRK  -  THE LOW CUTOFF VALUE FOR THE LOG TIME VALUES
C        THIMRK  -  THE HIGH CUTOFF VALUE FOR THE LOG TIME VALUES
C
C      RESULTS:
C        TRAN    -  TRANSMISSIVITY (SQ FEET/MIN)
C        TRAN1   -  TRANSMISSIVITY (SQ FEET/DAY)
C        TRAN2   -  TRANSMISSIVITY (GALLONS/DAY/FOOT)
C        STOR    -  STORAGE COEFFICIENT (DIMENSIONLESS)
C        A,B     -  BEST FIT ON A LINEAR REGRESSION OF ALL LT VALUES
C                     BETWEEN TLOMRK AND THIMRK TO FIT THE EQUATION
C                     S(I) = A + B*LT(I)
C        TO      -  TIME AT WHICH LT EQUALS 0 ( TO = 10.**(-A/B) )
C
C      NOTES:
C          1.  ALL RESULTS ARE STORED IN THE COMMON BLOCK /DATA/
C          2.  THE VECTORS LT AND S MAY CONTAIN UP TO 100 VALUES
C
C---------------------------------------------------------------------
C
      REAL S(200),LT(200),X(200),Y(200)
      COMMON /DATA/ N,R,Q,TPUMP,NREC,STOR,TRAN,TRAN1,TRAN2,A,B,TO
C
C---------------------------------------------------------------------
C
C---PULL OUT ALL VALUES BETWEEN THE LIMITS TLOMRK AND THIMRK
```

```
      NN=0
      DO 10 I=1,NUM
        IF (LT(I).LT.TLOMRK) GO TO 10
        IF (LT(I).GT.THIMRK) GO TO 10
        NN=NN+1
        X(NN)=LT(I)
        Y(NN)=S(I)
   10 CONTINUE
C---CALL THE LINEAR REGRESSION SUBROUTINE
      CALL LINREG(X,Y,NN,A,B)
C---CALCULATE THE RESULTS
      PI=3.14159
      QNEW=Q/7.48
      TRAN=2.3*QNEW/(4.*PI*B)
      TO=10**(-A/B)
      STOR=2.25*TRAN*TO/(R*R)
      TRAN1=1440.*TRAN
      TRAN2=10771.*TRAN
C
  999 RETURN
      END
      SUBROUTINE PLTLIN(X,Y,N,XLOMRK,XHIMRK,XORIG,XMAX,XSCALE,YORIG,
     & YMAX,YSCALE,A,B,LINE)
C
C
C         DATA AND LINE PLOTTING ROUTINE
C
C
C   REQUIRED PARAMETERS
C        X(I)    -   VECTOR OF LENGTH N CONTAINING X VALUES
C        Y(I)    -   VECTOR OF LENGTH N CONTAINING Y VALUES
C        N       -   NUMBER OF POINTS IN X(I) AND Y(I)
C        XLOMRK  -   LOW CUTOFF VALUE OF X POINTS USED TO DETERMINE LINE
C        XHIMRK  -   HIGH CUTOFF VALUE OF X POINTS USED TO DETERMINE LINE
C        XORIG   -   LOWEST NUMBER ON THE X AXIS
C        XMAX    -   HIGHEST NUMBER ON THE X AXIS
C        XSCALE  -   X SCALING FACTOR
C        YORIG   -   LOWEST NUMBER ON THE Y AXIS
C        YMAX    -   HIGHEST NUMBER ON THE Y AXIS
C        YSCALE  -   Y SCALING FACTOR
C        A       -   LINEAR REGRESSION PARAMETER
C        B       -   LINEAR REGRESSION PARAMETER
C                     ( Y = A + B*X )
C        LINE    -   A LOGICAL VARIABLE, =.TRUE. IF THE LINE IS TO BE
C                     PLOTTED, =.FALSE. IF NOT
C
C
C-----------------------------------------------------------------------
C
      REAL*4 X(200),Y(200)
      LOGICAL*1 MARK,STAR/'*'/,PLUS/'+'/,LINE
      DIMENSION IXY(2,6000), IPOINT(2,6000)
      LOGICAL TEK, CAL
      LOGICAL*1 LIFT(6000), STRING(130), ARG(130)
      COMMON /SPLT/ TEK, CAL, NPNTS, IXY, LIFT
```

```
      COMMON /SPARE/ IPOINT, IPTR
      COMMON /DEVICE/ IXLIM, IYLIM, MAXXO, MAXYO, IXMIN, IYMIN
      COMMON /PARS/ LEN, INDEX, NCHARS, NUMLFT, ARG, STRING
      COMMON /PLT/ X0,X1,Y0,Y1
C
C-----------------------------------------------------------------------
C
C         PLOT OUT POINTS
C
C---PAUSE
      READ (5,510,END=5) TRASH
    5 REWIND 5
C---SET PLOTTING CHARACTERS
      DO 10 I=1,N
        MARK=PLUS
        IF ((X(I).LT.XLOMRK).OR.(X(I).GT.XHIMRK)) MARK=STAR
        XLOC= XO + XSCALE*(X(I)-XORIG) - 0.075
        YLOC= YO + YSCALE*(Y(I)-YORIG) - 0.075
        CALL SYMBOL(XLOC,YLOC,0.15,MARK,0.,1)
        NPNTS = 0
   10 CONTINUE
C
C       PLOT OUT LINE
C
C---PLOT OUT LINE AT ALL?
      IF (.NOT.LINE) GO TO 999
C---DETERMINE LEFT END POINT
      XVAL=XORIG
      YVAL=A + B*XVAL
      IF (YVAL.GE.YORIG) GO TO 20
      YVAL=YORIG
      XVAL=(YVAL-A)/B
   20 XLOC1=XO+XSCALE*(XVAL-XORIG)
      YLOC1=YO+YSCALE*(YVAL-YORIG)
      CALL PLTMOV(XLOC1,YLOC1,3)
C---DETERMINE RIGHT END POINT
      XVAL=XMAX
      YVAL=A + B*XVAL
      IF (YVAL.LE.YMAX) GO TO 30
      YVAL=YMAX
      XVAL=(YVAL-A)/B
   30 XLOC2=XO+XSCALE*(XVAL-XORIG)
      YLOC2=YO+YSCALE*(YVAL-YORIG)
      CALL PLTMOV(XLOC2,YLOC2,2)
C---TRACE OVER LINE A SECOND TIME FOR DARKNESS
      CALL PLTMOV(XLOC1,YLOC1,2)
      CALL PLTMOV(XLOC1,YLOC1,-999)
C---FORMATS
  510 FORMAT (A4)
C
  999 RETURN
      END
      SUBROUTINE SLPLOT(XLO,XHI,YLO,YHI,XORIG,XMAX,XSCALE,YORIG,YMAX,
     &  YSCALE,TITLE,XTITLE,YTITLE)
C
```

```
C
C          SEMI-LOG PLOTTING ROUTINE
C
C    NOTES:
C        1. THE X AXIS IS THE LOG AXIS
C        2. THIS SUBROUTINE SETS UP THE SCALING FACTORS FOR OTHER
C              SUBROUTINES AND PLOTS OUT THE TITLES AND BORDER
C        3. THE VALUES OF X0, X1, Y0, AND Y1 ( THE PHYSICAL LIMITS OF
C              THE GRAPH IN INCHES) ARE PASSED IN IN THE COMMON BLOCK
C              /PLT/
C
C
C    DATA INPUT:
C        XLO       -  LOG OF LOWEST X VALUE DATA POINTTED
C        XHI       -  LOG OF HIGHEST X VALUE DATA POINTTED
C        YLO       -  LOWEST Y VALUE DATA POINT
C        YHI       -  HIGHEST Y VALUE DATA POINT
C        TITLE     -  GENERAL TITLE FOR PLOT
C        XTITLE    -  TITLE TO GO UNDER X AXIS
C        YTITLE    -  TITLE TO GO ALONG Y AXIS
C
C
C   RESULTS:
C        XORIG   -  LOG OF LOWEST X VALUE ON THE X AXIS
C        XMAX    -  LOG OF HIGHEST X VALUE ON THE Y AXIS
C        XSCALE  -  X SCALING FACTOR
C        YORIG   -  LOWEST Y VALUE ON THE Y AXIS
C        YMAX    -  HIGHEST Y VALUE ON THE Y AXIS
C        YSCALE  -  Y SCALING FACTOR
C
C
C-----------------------------------------------------------------------
C
      LOGICAL*1 TITLE(25),XTITLE(30),YTITLE(30),CENT(10)
      LOGICAL*1 XTIC(10)/10*' '/,YTIC(9)/9*' '/,BLANK/' '/,CHECK
      LOGICAL*1 CENTX(30)/30*' '/,CENTY(30)/30*' '/,AMP/'&'/,BAR/'|'/
      LOGICAL*1 CENTTA(25)/25*' '/
      DIMENSION IXY(2,6000), IPOINT(2,6000)
      LOGICAL TEK, CAL
      LOGICAL*1 LIFT(6000), STRING(130), ARG(130)
      COMMON /SPOT/ TEK, CAL, NPNTS, IXY, LIFT
      COMMON /SPARE/ IPOINT, IPTR
      COMMON /DEVICE/ IXLIM, IYLIM, MAXXO, MAXYO, IXMIN, IYMIN
      COMMON /PARS/ LEN, INDEX, NCHARS, NUMLFT, ARG, STRING
      COMMON /PLT/ X0,X1,Y0,Y1
C
C-----------------------------------------------------------------------
C
C
C---CLEAR PLOTTER SCREEN
      CALL PLTERA
C---SET POSITIONS FOR TIC MARKS AND TITLES
      XSYM=X0-1.30
      YSYM=Y0-0.45
      XDIFF=X1-X0
```

```
      YDIFF=Y1-Y0
      XT=X0-0.2
      YT=Y0-0.2
      XT1=X1-0.2
      YT1=Y1-0.2
      XTNAME=Y0-0.45
C---DRAW BORDER
      DO 10 I=1,2
        CALL PLTMOV(X0,Y0,3)
        CALL PLTMOV(X1,Y0,2)
        CALL PLTMOV(X1,Y1,2)
        CALL PLTMOV(X0,Y1,2)
        CALL PLTMOV(X0,Y0,2)
   10 CONTINUE
      CALL PLTMOV(X0,Y0,-999)
C
C        SET UP X AXIS
C
C---DETERMINE X AXIS PARAMETERS
      IXLO=INT(XLO)
      IF (XLO.LT.0.0) IXLO=IXLO-1
      IXHI=INT(XHI)+1
      IF (XHI.LT.0.0) IXHI=IXHI-1
      XORIG=IXLO
      XMAX=IXHI
      IDIFF=IXHI-IXLO
      IDIFF1=IDIFF+1
      XSCALE=XDIFF/IDIFF
      NPNTS=0
C---PLOT OUT X TIC VALUES
      DO 30 I=1,IDIFF1
        X=X0+(I-1)*XSCALE
        CALL PLTMOV(X,YT,3)
        CALL PLTMOV(X,Y0,2)
        CALL PLTMOV(X,YT1,3)
        CALL PLTMOV(X,Y1,2)
        CALL PLTMOV(0.,0.,-999)
        XVAL=10.**(IXLO+I-1)
        IDEC=2
        IF (XVAL.GE.0.099999) IDEC=1
        IF (XVAL.GE.1.0) IDEC=0
        CALL UNPAKF(XVAL,XTIC,BINARY,10,IDEC,CHECK)
        XNEW=X-0.9
        DO 20 J=1,10
   20     CENT(J)=BLANK
        CALL CENTER(CENT,1,10,XTIC,0,10)
        CALL SYMBOL(XNEW,YSYM,0.15,CENT,0.,10)
        NPNTS=0
   30 CONTINUE
C
C       LABEL Y AXIS
C
C---DETERMINE Y AXIS PARAMETERS
      YORIG=0.0
      JPOWER=ALOG10(YHI)
```

```
      IF (JPOWER.LT.0) JPOWER=JPOWER-1
      TEST=YHI/(10.**JPOWER)
      IFAC=(TEST/2.5)+1
      YMAX=IFAC*2.5*(10.**JPOWER)
      IF (TEST.EQ.1) YMAX=YHI
      DIFF=YMAX-YORIG
      YSCALE=YDIFF/DIFF
      DELTAY=DIFF/5.
      NPNTS=0
C---PLOT OUT AND LABEL Y TIC MARKS
      IDEC=1
      IF (YMAX.LE.1.0) IDEC=2
      DO 40 J=1,6
        Y=(J-1)*DELTAY*YSCALE + YO
        CALL PLTMOV(XT,Y,3)
        CALL PLTMOV(XO,Y,2)
        CALL PLTMOV(XT1,Y,3)
        CALL PLTMOV(X1,Y,2)
        CALL PLTMOV(0.,0.,-999)
        YVAL=YORIG+(J-1)*DELTAY
        CALL UNPAKF(YVAL,YTIC,BINARY,7,IDEC,CHECK)
        CALL SYMBOL(XSYM,Y,0.15,YTIC,0.,9)
        NPNTS=0
   40 CONTINUE
C
C       PLOT TITLES
C
C---PAUSE
      READ (5,510,END=50) TRASH
   50 REWIND 5
C---PLOT MAIN TITLE
      X=XO-1.0
      Y=Y1+0.3
      CALL CENTER(CENTTA,1,25,TITLE,0,25)
      CALL SYMBOL(X,Y,0.50,CENTTA,0.,25)
      NPNTS=0
C---PLOT X-AXIS TITLE
      X=XO+1.8
      Y=YO-1.0
      CALL CENTER(CENTX,1,30,XTITLE,0,30)
      CALL SYMBOL(X,Y,0.25,CENTX,0.,30)
      NPNTS=0
C---PLOT Y-AXIS TITLE
      X=XO-1.0
      Y=YO+0.3
      CALL CENTER(CENTY,1,30,YTITLE,0,30)
      CALL SYMBOL(X,Y,0.25,CENTY,90.,30)
      NPNTS=0
C---FORMATS
  510 FORMAT (A4)
C
  999 RETURN
      END
      SUBROUTINE RESULT(IMKLO,IMKHI,TTOP,RESTF)
C
```

```
C
C           DISPLAY THE RESULTS
C
C     INPUT DATA:
C        TPKLO   -  THE LOW TIME CUTOFF VALUE
C        TPKHI   -  THE HIGH TIME CUTOFF VALUE
C        TTOP    -  TITLE FOR TOP OF RESULT SECTION
C        RESTF   -  LOGICAL VARIABLE, =.TRUE. IF RESULTS COME FROM A
C                    RESIDUAL DRAWDOWN ANALYSIS (AND THUS NO STORAGE
C                    WAS CALCULATED), =.FALSE. OTHERWISE
C
C
C-----------------------------------------------------------------
C
      LOGICAL*1 CHECK,RESTF
      LOGICAL*1 TTOP(30),QN(2),RN(2),TN(2),SN(2),U1(3),U2(4),U3(6)
      LOGICAL*1 N1(9),N2(9),N3(9),N4(7),N5(9),N6(9)
      LOGICAL*1 TA(4),TB(4),LT(3),ASSOC(10)
      DATA QN/'Q','='/, RN/'R','='/, TN/'T','='/, SN/'S','='/
      DATA U1/'G','P','M'/, U2/'F','E','E','T'/, U3/'G','P','D','/','F',
     &'T'/
      DATA N1/9*' '/, N2/9*' '/,N3/9*' '/,N4/7*' '/
      DATA N5/9*' '/, N6/9*' '/
      DATA TA/'T','L','O','='/,TB/'T','H','I','='/, LT/'M','I','N'/
      DATA ASSOC/'C','E','P',' ','-',' ','N','J','G','S'/
      COMMON /PLTPRM/ XLO,XHI,XORIG,XMAX,XSCALE,YLO,YHI,YORIG,YMAX,
     &YSCALE
      DIMENSION IXY(2,6000), IPOINT(2,6000)
      LOGICAL TEK, CAL
      LOGICAL*1 LIFT(6000), STRING(130), ARG(130)
      COMMON /SPOT/ TEK, CAL, NPNTS, IXY, LIFT
      COMMON /SPARE/ IPOINT, IPTR
      COMMON /DEVICE/ IXLIM, IYLIM, MAXXO, MAXYO, IXMIN, IYMIN
      COMMON /PARS/ LEN, INDEX, NCHARS, NUMLFT, ARG, STRING
      COMMON /PLT/ XO,X1,YO,Y1
      COMMON /DATA/ N,R,Q,TPUMP,NREC,STOR,TRAN,TRAN1,TRAN2,A,B,TO
C
C-----------------------------------------------------------------
C
C
C---PAUSE
      READ (5,501,END=5) TRASH
    5 REWIND 5
C---INITIALIZE
      X=XO+0.3
      Y=Y1-0.7
C---TYPE OF ANALYSIS
      CALL SYMBOL(X,Y,0.25,TTOP,0.,30)
      NPNTS=0
C---Q
      X=X+.50
      XN=X+.3
      XU=XN+2.
      Y=Y-.35
      CALL SYMBOL(X,Y,0.2,QN,0.,2)
```

```
      CALL UNPAKF(C,N1,BINARY,9,1,CHECK)
      CALL SYMBOL(XN,Y,0.2,N1,0.,9)
      CALL SYMBOL(XU,Y,0.2,U1,0.,3)
      NPNTS=0
C---R
      Y=Y-.30
      CALL SYMBOL(X,Y,0.2,RN,0.,2)
      CALL UNPAKF(R,N2,BINARY,9,1,CHECK)
      CALL SYMBOL(XN,Y,0.2,N2,0.,9)
      CALL SYMBOL(XU,Y,0.2,U2,0.,4)
      NPNTS=0
C---TIME LIMITS
      Y=Y-.3
      CALL SYMBOL(X,Y,0.2,TA,0.,4)
      CALL UNPAKF(TMKLO,N5,BINARY,9,1,CHECK)
      CALL SYMBOL(XN,Y,0.2,N5,0.,9)
      IF (.NOT.RESTF) CALL SYMBOL(XU,Y,0.2,UT,0.,3)
      Y=Y-.3
      CALL SYMBOL(X,Y,0.2,TB,0.,4)
      CALL UNPAKF(TMKHI,N6,BINARY,9,1,CHECK)
      CALL SYMBOL(XN,Y,0.2,N6,0.,9)
      IF (.NOT.RESTF) CALL SYMBOL(XU,Y,0.2,UT,0.,3)
      NPNTS=0
C---TRANSMISSIVITY
   10 Y=Y-.40
      CALL SYMBOL(X,Y,0.2,TN,0.,2)
      CALL UNPAKF(TRAN2,N3,BINARY,9,0,CHECK)
      CALL SYMBOL(XN,Y,0.2,N3,0.,9)
      CALL SYMBOL(XU,Y,0.2,U3,0.,6)
      NPNTS=0
C---STORAGE
      IF (RESTF) GO TO 20
      Y=Y-.30
      XN=XN+0.32
      CALL SYMBOL(X,Y,0.2,SN,0.,2)
      CALL UNPAKE(STOR,N4,BINARY,0,7,2,CHECK)
      CALL SYMBOL(XN,Y,0.2,N4,0.,7)
      NPNTS=0
C---ASSOCIATION
   20 Y=Y0+0.05
      X=X1-1.9
      CALL SYMBOL(X,Y,0.15,ASSOC,0.,10)
      NPNTS=0
C
  501 FORMAT (A4)
C
  999 RETURN
      END
      SUBROUTINE LINREG(X,Y,N,A,B)
C
C  SOLUTION OF THE NORMAL EQUATIONS FOR LEAST SQUARES ESTIMATION
C  OF LINEAR REGRESSION PARAMETERS.  (I.E. SOLVE THE LINEAR REGRESSION
C  PROBLEM.)
C
C             Y = A + B*X
```

```
C
C     INPUT DATA:
C         Y - VECTOR OF LENGTH N
C         X - VECTOR OF LENGTH N
C
C
C     OUTPUT
C         A - INTERCEPT
C         B - SLOPE
C
C-----------------------------------------------------------------------
C
      REAL*4 X(200),Y(200)
C
C-----------------------------------------------------------------------
C
C
      XAVG=0.
      YAVG=0.
      DO 10 I=1,N
        XAVG=XAVG+X(I)
        YAVG=YAVG+Y(I)
   10 CONTINUE
      XAVG=XAVG/N
      YAVG=YAVG/N
      TOP=0.0
      BOT=0.0
      DO 20 I=1,N
        TOP=TOP+(X(I)-XAVG)*Y(I)
        BOT=BOT+(X(I)-XAVG)**2
   20 CONTINUE
      B=TOP/BOT
      A=YAVG-B*XAVG
C
  999 RETURN
      END
      SUBROUTINE LLPLOT(XLO,XHI,YLO,YHI,XORIG,XMAX,XSCALE,YORIG,YMAX,
     &  YSCALE,TITLE,XTITLE,YTITLE)
C
C
C       LOG-LOG PLOT
C         1. CALCULATE SCALING FACTORS FOR OTHER SUBROUTINES
C         2. PLOT OUT THE BORDER, TIC MARKS, AND TITLES
C         3. VALUES FOR X0, X1, Y0, AND Y1 HAVE TO BE PASSED IN THROUGH
C               THE COMMON BLOCK /PLT/
C
C
C       DATA INPUT:
C         XLO     -  LOG OF LOWEST X VALUE TO BE PLOTTED
C         XHI     -  LOG OF HIGHEST X VALUE TO BE PLOTTED
C         YLO     -  LOG OF LOWEST Y VALUE TO BE PLOTTED
C         YHI     -  LOG OF HIGHEST Y VALUE TO BE PLOTTED
C         TITLE   -  MAIN TITLE FOR PLOT
C         XTITLE  -  X-AXIS TITLE
C         YTITLE  -  Y-AXIS TITLE
```

```
C
C
C      RESULTS:
C        XORIG  -  LOG OF NUMBER AT ORIGIN OF X AXIS
C        XMAX   -  LOG OF NUMBER AT HIGH END OF X AXIS
C        XSCALE -  X AXIS SCALING FACTOR
C        YORIG  -  LOG OF NUMBER AT ORIGIN OF Y AXIS
C        YMAX   -  LOG OF NUMBER AT HIGH END OF Y AXIS
C        YSCALE -  Y AXIS SCALING FACTOR
C
C-----------------------------------------------------------------------
C
      LOGICAL*1 TITLE(25),XTITLE(30),YTITLE(30),CENT(10)
      LOGICAL*1 XTIC(10)/10*' '/,YTIC(9)/9*' '/,BLANK/' '/,CHECK
      LOGICAL*1 CENTX(30)/30*' '/,CENTY(30)/30*' '/,AMP/'&'/,BAR/'|'/
      LOGICAL*1 CENTTA(25)/25*' '/
      DIMENSION IXY(2,6000), IPOINT(2,6000)
      LOGICAL TEK, CAL
      LOGICAL*1 LIFT(6000), STRING(130), ARG(130)
      COMMON /SPOT/ TEK, CAL, NPNTS, IXY, LIFT
      COMMON /SPARE/ IPOINT, IPTR
      COMMON /DEVICE/ IXLIM, IYLIM, MAXXO, MAXYO, IXMIN, IYMIN
      COMMON /PARS/ LEN, INDEX, NCHARS, NUMLFT, ARG, STRING
      COMMON /PLT/ X0,X1,Y0,Y1
C
C-----------------------------------------------------------------------
C
C
C---CLEAR PLOTTER SCREEN
      CALL PLTERA
C---SET PARAMETERS FOR TITLES AND TIC MARKS
      XSYM=X0-1.30
      YSYM=Y0-0.45
      XDIFF=X1-X0
      YDIFF=Y1-Y0
      XT=X0-0.2
      YT=Y0-0.2
      XT1=X1-0.2
      YT1=Y1-0.2
      XTNAME=Y0-0.45
C---DRAW BORDER
      DO 10 I=1,2
        CALL PLTMOV(X0,Y0,3)
        CALL PLTMOV(X1,Y0,2)
        CALL PLTMOV(X1,Y1,2)
        CALL PLTMOV(X0,Y1,2)
        CALL PLTMOV(X0,Y0,2)
   10 CONTINUE
      CALL PLTMOV(0.,0.,-999)
C
C        DRAW X-AXIS
C
C---DETERMINE SPACINGS ON X-AXIS
      IXLO=INT(XLO)
      IF (XLO.LT.0.0) IXLO=IXLO-1
```

```
      IXHI=INT(XHI)+1
      IF (XHI.LT.0.0) IXHI=IXHI-1
      XORIG=IXLO
      XMAX=IXHI
      IDIFF=IXHI-IXLO
      IDIFF1=IDIFF+1
      XSCALE=XDIFF/IDIFF
      NPNTS=0
C---DRAW AND LABEL X TIC MARKS
      DO 30 I=1,IDIFF1
        X=XO+(I-1)*XSCALE
        CALL PLTMOV(X,YT,3)
        CALL PLTMOV(X,YO,2)
        CALL PLTMOV(X,YT1,3)
        CALL PLTMOV(X,Y1,2)
        CALL PLTMOV(0.,0.,-999)
        XVAL=10.**(IXLO+I-1)
        IDEC=2
        IF (XVAL.GE.0.099999) IDEC=1
        IF (XVAL.GT.1.0) IDEC=0
        CALL UNPAKF(XVAL,XTIC,BINARY,10,IDEC,CHECK)
        XNEW=X-0.9
        DO 20 J=1,10
   20     CENT(J)=BLANK
        CALL CENTER(CENT,1,10,XTIC,0,10)
        CALL SYMBOL(XNEW,YSYM,0.15,CENT,0.,10)
        NPNTS=0
   30 CONTINUE
C
C       LABEL Y-AXIS
C
C---DETERMINE SPACINGS ON Y AXIS
      JYLO=INT(YLO)
      IF (YLO.LT.0.0) JYLO=JYLO-1
      JYHI=INT(YHI)+1
      IF (YHI.LT.0.0) JYHI=JYHI-1
      YORIG=JYLO
      YMAX=JYHI
      JDIFF=JYHI-JYLO
      JDIFF1=JDIFF+1
      YSCALE=YDIFF/JDIFF
      NPNTS=0
C---PLOT AND LABEL Y TIC MARKS
      DO 40 J=1,JDIFF1
        Y=YO+(J-1)*YSCALE
        CALL PLTMOV(XT,Y,3)
        CALL PLTMOV(XO,Y,2)
        CALL PLTMOV(XT1,Y,3)
        CALL PLTMOV(X1,Y,2)
        CALL PLTMOV(0.,0.,-999)
        YVAL=10.**(JYLO+J-1)
        IDEC=2
        IF (YVAL.GT.0.0999) IDEC=1
        IF (YVAL.GT.1.0) IDEC=0
        CALL UNPAKF(YVAL,YTIC,BINARY,7,IDEC,CHECK)
```

```
          CALL SYMBOL(XSYM,Y,0.15,YTIC,0.,9)
          NPNTS=0
   40 CONTINUE
C
C       PLOT TITLES
C
C---PAUSE
        READ (5,510,END=50) TRASH
   50 REWIND 5
C---PLOT MAIN TITLE
        X=XC-1.0
        Y=Y1+0.3
        CALL CENTER(CENTTA,1,25,TITLE,0,25)
        CALL SYMBOL(X,Y,0.50,CENTTA,0.,25)
        NPNTS=0
C---PLOT X-AXIS TITLE
        X=XC+1.8
        Y=YC-1.0
        CALL CENTER(CENTX,1,30,XTITLE,0,30)
        CALL SYMBOL(X,Y,0.25,CENTX,0.,30)
        NPNTS=0
C---PLOT Y-AXIS TITLE
        X=XC-1.0
        Y=YC+0.3
        CALL CENTER(CENTY,1,30,YTITLE,0,30)
        CALL SYMBOL(X,Y,0.25,CENTY,90.,30)
        NPNTS=0
C---FORMATS
  510 FORMAT (A4)
C
  999 RETURN
        END
        SUBROUTINE THSPLT(X,Y,NUM,XORIG,XMAX,XSCALE,YORIG,YMAX,YSCALE,
      &  XLOMRK,XHIMRK)
C
C
C            THEIS ANALYSIS CURVE PLOTTING
C
C
C       DATA INPUT:
C           X       -   OBSERVED LOG TIME VALUES
C           Y       -   OBSERVED LOG DRAWDOWN VALUES
C           NUM     -   NUMBER OF POINTS IN X AND Y
C           XORIG   -   LOG OF NUMBER AT ORIGIN OF X AXIS
C           XMAX    -   LOG OF NUMBER AT END OF X AXIS
C           XSCALE  -   X SCALING FACTOR
C           YORIG   -   LOG OF NUMBER AT ORIGIN OF Y AXIS
C           YMAX    -   LOG OF NUMBER AT END OF Y AXIS
C           YSCALE  -   Y SCALING FACTOR
C           XLOMRK  -   LOW TIME CUTOFF FOR ANALYSIS
C           XHIMRK  -   HIGH TIME CUTOFF FOR ANALYSIS
C
C       NOTES:
C         1. THE VALUES XO,X1,YO, AND Y1 HAVE TO BE PASSED IN BY THE
C                 COMMON BLOCK /PLT/
```

```
C
C
C
C
C-----------------------------------------------------------------------
C
      REAL X(200),Y(200)
      REAL*8 U,UINV,WU
      LOGICAL*1 TITLE(25),XTITLE(30),YTITLE(30),CENT(10)
      LOGICAL*1 XTIC(10)/10*' '/,YTIC(9)/9*' '/,BLANK/' '/,CHECK
      LOGICAL*1 CENTX(30)/30*' '/,CENTY(30)/30*' '/,AMP/'&'/,BAR/'|'/
      LOGICAL*1 CENTTA(25)/25*' '/,PLUS/'+'/,AST/'*'/,SOUT
      DIMENSION IXY(2,6000), IPOINT(2,6000)
      LOGICAL TEK, CAL
      LOGICAL*1 LIFT(6000), STRING(130), ARG(130)
      COMMON /SPOT/ TEK, CAL, NPNTS, IXY, LIFT
      COMMON /SPARE/ IPOINT, IPTR
      COMMON /DEVICE/ IXLIM, IYLIM, MAXXO, MAXYO, IXMIN, IYMIN
      COMMON /PARS/ LEN, INDEX, NCHARS, NUMLFT, ARG, STRING
      COMMON /PLT/ X0,X1,Y0,Y1
      COMMON /POINTS/ T,S,LT,LS,T2,S2,LT2
      COMMON /ADD/ XADD,YADD,DX,DY,AXORG,AYORG
C
C-----------------------------------------------------------------------
C
C
C
C---PAUSE
      READ (5,510,END=5) TRASH
    5 REWIND 5
C---PLOT OUT POINTS
C---  THE FACTOR OF -0.075 ACCOUNTS FOR THE FACT THAT THE PLOTTING
C---   CHARACTERS DO NOT APPEAR EXACTLY AT THE POINT SPECIFIED
      DO 10 I=1,NUP
        SOUT=PLUS
        IF (X(I).LT.XLOMRK) SOUT=AST
        IF (X(I).GT.XHIMRK) SOUT=AST
        XLOC=X0+XSCALE*(X(I)-XORIG) - 0.075
        YLOC=Y0+YSCALE*(Y(I)-YORIG) - 0.075
        CALL SYMBOL(XLOC,YLOC,0.15,SOUT,0.,1)
        NPNTS=0
   10 CONTINUE
C---PLOT OUT THEIS CURVE
      IFLAG=0
      DELTAX=(XMAX-XORIG)/250.
      XVAL=XORIG
      UINV=10.**(XVAL+XADD)
      U=1./UINV
      CALL E1(U,WU)
      YVAL=DLOG10(WU)-YADD
      IF (YVAL.LT.YORIG) GO TO 20
      XLOC=X0+XSCALE*(XVAL-XORIG)
      YLOC=Y0+YSCALE*(YVAL-YORIG)
      CALL PLTMOV(XLOC,YLOC,2)
      IFLAG=1
```

```
   20 DO 40 I=1,251
        XVAL=XORIG+(I-1)*DELTAX
        UINV=10.**(XVAL+XADD)
        U=1./UINV
        CALL E1(U,WU)
        YVAL=ALOG10(WU)-YADD
        IF (YVAL.GT.YMAX) GO TO 50
        IF (YVAL.LT.YORIG) GO TO 40
        XLOC=XO+XSCALE*(XVAL-XORIG)
        YLOC=YO+YSCALE*(YVAL-YORIG)
        IF (IFLAG.EQ.1) GO TO 30
        IFLAG=1
        CALL PLTMOV(XLOC,YLOC,3)
   30   CALL PLTMOV(XLOC,YLOC,2)
   40 CONTINUE
   50 CALL PLTMOV(0.,0.,-999)
C---FORMATS
  510 FORMAT (A4)
C
  999 RETURN
      END
      SUBROUTINE PNTOUT(TIMLO,TIMHI,RESTF)
C
C         PRINT OUT THE RESULTS
C
C    DATA INPUT:
C       TIMLO  -  LOW CUTOFF TIME USED IN ANALYSIS
C       TIMHI  -  HIGH CUTOFF TIME USED IN ANALYSIS
C       RESTF  -  LOGICAL VARIABLE, =.TRUE. IF RESULTS COME FROM A
C                    RESIDUAL DRAWDOWN ANALYSIS
C
C-----------------------------------------------------------------------
C
      LOGICAL*1 RESTF
      COMMON /DATA/ N,R,Q,TPUMP,NREC,STOR,TRAN,TRAN1,TRAN2,A,B,TO
C
C-----------------------------------------------------------------------
C
      WRITE (6,601) TIMLO,TIMHI
      WRITE (6,602) TRAN2
      IF (.NOT.RESTF) WRITE (6,603) STOR
C
C              FORMAT STATEMENTS
C
  601 FORMAT (/5X,'TIME CUTOFF LIMITS:',
     &/10X,'LOW  = ',F10.2,' MINUTES',
     &/10X,'HIGH = ',F10.2,' MINUTES')
  602 FORMAT (5X,'TRANSMISSIVITY = ',F10.0,' GPD/FT')
  603 FORMAT (5X,'STORAGE        = ',E11.3)
C
  999 RETURN
      END
      SUBROUTINE TABLE(A,B,N)
C
C         OUTPUT VECTORS A AND B (EACH LENGTH N) IN 3 COLUMNS
```

```
C
      REAL A(200),B(200)
      INTEGER K(200),KK(3)
C---INITIALIZE TABLE PARAMETERS
      DO 10 I=1,200
   10   K(I)=I
      NROW=N/3
      IREM=N-3*NROW
      K1=NROW+1
      KK(1)=0
      KK(2)=NROW
      IF (IREM.GE.1) KK(2)=KK(2)+1
      KK(3)=KK(2)+NROW
      IF (IREM.GE.2) KK(3)=KK(3)+1
C---OUTPUT THE TABLE
      WRITE (6,601) ((K(I+KK(J)),A(I+KK(J)),B(I+KK(J)),J=1,3),I=1,NROW)
      IF (IREM.GT.0) WRITE (6,601) (K(K1+KK(J)),A(K1+KK(J)),B(K1+KK(J)),
     &  J=1,IREM)
C
  601 FORMAT (3(1X,I3,1X,F10.2,2X,F10.2,6X))
  999 RETURN
      END
      SUBROUTINE SEARCH(IEND)
C
C         SEARCH THE SOLUTION SPACE
C
C-----------------------------------------------------------------------
C
      REAL*4 T(200),S(200),LT(200),LS(200),ERM(200),Q,R,STOR,TRAN
      INTEGER NPTS
      LOGICAL*1 IEND
      COMMON /TITLES/ TITLE
      COMMON /DATA/ N,R,Q,TPUMP,NREC,STOR,TRAN,TRAN1,TRAN2,A,B,TO
      COMMON /TDATA/ ERM,NPTS,NSUM
      COMMON /XTRA/ LT,LS
      COMMON /ADD/ XADD,YADD,DX,DY,XORIG,YORIG
      COMMON /FERR/ R1,R2,R3,R4,R5,R6,R7,R8,R9,RR
C
C-----------------------------------------------------------------------
C
      IEND=.FALSE.
      RROLD=999999999.
      XORIG=0.0
      YORIG=-1.0
      DX=2.0
      DY=2.0
      INDEX=0
      JFLAG=0
      LOOP=0
      RR=RROLD
C
C---INITIALIZE
   10 YADD=YORIG+DY
      CALL SETUP(A,B,C)
      R1=A
```

```
      R2=B
      R3=C
      YADD=YORIG
      CALL SETUP(A,B,C)
      R4=A
      R5=B
      R6=C
      YADD=YORIG-DY
      CALL SETUP(A,B,C)
      R7=A
      R8=B
      R9=C
C
C---COMPARE
   20 CALL MIN(INDEX)
      LOOP=LOOP+1
      CHANGE=ABS((RROLD-RR)/RROLD)
      IF (LOOP.LT.NSUM) GO TO 30
      IF (LOOP.GT.250) GO TO 50
      IF (CHANGE.LE.0.001) GO TO 999
   30 RROLD=RR
      GO TO (1,2,3,4,5,6,7,8,9),INDEX
C
    1 CONTINUE
      YORIG=YORIG+DY
      XORIG=XORIG-DX
      R6=R2
      R9=R5
      R5=R1
      R8=R4
      YADD=YORIG+DY
      CALL SETUP(A,B,C)
      R1=A
      R2=B
      R3=C
      XADD=XORIG-DX
      YADD=YORIG
      CALL COMPUT(R4,JFLAG)
      YADD=YORIG-DY
      CALL COMPUT(R7,JFLAG)
      GO TO 20
C
    2 CONTINUE
      YORIG=YORIG+DY
      R7=R4
      R8=R5
      R9=R6
      R4=R1
      R5=R2
      R6=R3
      YADD=YORIG+DY
      CALL SETUP(A,B,C)
      R1=A
      R2=B
      R3=C
```

```
      GO TO 20
C
    3 CONTINUE
      XORIG=XORIG+DX
      YORIG=YORIG+DY
      R4=R2
      R7=R5
      R5=R3
      R8=R6
      YADD=YORIG+DY
      CALL SETUP(A,B,C)
      R1=A
      R2=B
      R3=C
      XADD=XORIG+DX
      YADD=YORIG
      CALL COMPUT(R6,JFLAG)
      YADD=YORIG-DY
      CALL COMPUT(R9,JFLAG)
      GO TO 20
C
    4 CONTINUE
      XORIG=XORIG-DX
      R3=R2
      R6=R5
      R9=R8
      R2=R1
      R5=R4
      R8=R7
      XADD=XORIG-DX
      YADD=YADD+DY
      CALL COMPUT(R1,JFLAG)
      YADD=YORIG
      CALL COMPUT(R4,JFLAG)
      YADD=YORIG-DY
      CALL COMPUT(R7,JFLAG)
      GO TO 20
C
    5 CONTINUE
      ERRX=R4+R6
      ERRY=R2+R8
      IF (ERRX.GE.ERRY) DX=DX/1.5
      IF (ERRY.GE.ERRX) DY=DY/1.5
      GO TO 10
C
    6 CONTINUE
      XORIG=XORIG+DX
      R1=R2
      R4=R5
      R7=R8
      R2=R3
      R5=R6
      R8=R9
      XADD=XORIG+DX
      YADD=YORIG+DY
```

```
      CALL COMPLT(R3,JFLAG)
      YADD=YORIG
      CALL COMPLT(R6,JFLAG)
      YADD=YORIG-DY
      CALL COMPLT(R9,JFLAG)
      GO TO 20
C
    7 CONTINUE
      XORIG=XORIG-DX
      YORIG=YORIG-DY
      R2=R4
      R3=R5
      R6=R8
      R9=R7
      XADD=XORIG-DX
      YADD=YORIG+DY
      CALL COMPLT(R1,JFLAG)
      YADD=YORIG
      CALL COMPLT(R4,JFLAG)
      YADD=YORIG-DY
      CALL SETUP(A,B,C)
      R7=A
      R8=B
      R9=C
      GO TO 20
C
    8 CONTINUE
      YORIG=YORIG-DY
      R1=R4
      R2=R5
      R3=R6
      R4=R7
      R5=R8
      R6=R9
      YADD=YORIG-DY
      CALL SETUP(A,B,C)
      R7=A
      R8=B
      R9=C
      GO TO 20
C
    9 CONTINUE
      XORIG=XORIG+DX
      YORIG=YORIG-DY
      R1=R5
      R2=R6
      R4=R8
      R5=R9
      YADD=YORIG-DY
      CALL SETUP(A,B,C)
      R7=A
      R8=B
      R9=C
      XADD=XORIG+DX
      YADD=YORIG
```

```
      CALL COMPUT(R6,JFLAG)
      YADD=YORIG+DY
      CALL COMPUT(R3,JFLAG)
      GO TO 20
C---TOO MANY ITERATIONS, STOP HERE
   50 IEND=.TRUE.
C
  999 RETURN
      END
      SUBROUTINE SETUP(A,B,C)
C
C
C-----------------------------------------------------------------------
C
      COMMON /ADD/ XADD,YADD,DX,DY,XORIG,YORIG
C
C-----------------------------------------------------------------------
C
      JFLAG=0
      XADD=XORIG-DX
      CALL COMPUT(A,JFLAG)
      XADD=XORIG
      CALL COMPUT(B,JFLAG)
      XADD=XORIG+DX
      CALL COMPUT(C,JFLAG)
C
  999 RETURN
        END
      SUBROUTINE MIN(IN)
C
C
C-----------------------------------------------------------------------
C
      COMMON /RERR/ R1,R2,R3,R4,R5,R6,R7,R8,R9,RR
C
C-----------------------------------------------------------------------
C
      RR=AMIN1(R1,R2,R3,R4,R5,R6,R7,R8,R9)
      IF (RR.EQ.R1) IN=1
      IF (RR.EQ.R2) IN=2
      IF (RR.EQ.R3) IN=3
      IF (RR.EQ.R4) IN=4
      IF (RR.EQ.R5) IN=5
      IF (RR.EQ.R6) IN=6
      IF (RR.EQ.R7) IN=7
      IF (RR.EQ.R8) IN=8
      IF (RR.EQ.R9) IN=9
C
  999 RETURN
      END
      SUBROUTINE COMPUT(RESID,JFLAG)
C
C       COMPUTE THE RESID
C
C
```

```
C-----------------------------------------------------------------------
C
      REAL*4 T(200),S(200),LT(200),LS(200),ERM(200),Q,R,STOR,TRAN
      REAL*8 U,WU,UINV
      INTEGER NPTS
      COMMON /TITLES/ TITLE
      COMMON /DATA/ N,R,Q,TPUMP,NREC,STOR,TRAN,TRAN1,TRAN2,A,B,TO
      COMMON /TDATA/ ERM,NPTS,NSUM
      COMMON /XTRA/ LT,LS
      COMMON /ADD/ XADD,YADD,DX,DY,XORIG,YORIG
      COMMON /PERM/ R1,R2,R3,R4,R5,R6,R7,R8,R9,RR
C
C-----------------------------------------------------------------------
C
      STORE=0.
      DO 10 I=1,NPTS
        UINV=10.**(LT(I)+XADD)
        U=1./UINV
        CALL E1(U,WU)
        IF (WU.LE.0.0) WU=1.E-50
        WU=DLOG10(WU)
        WUPRED=LS(I)+YADD
        RES=(WU-WUPRED)**2
        STORE=STORE+RES
        IF (JFLAG.NE.1) GO TO 10
        ERM(I)=RES
   10 CONTINUE
      RESID=STORE
      IF (JFLAG.NE.1) GO TO 999
      DO 20 I=1,NPTS
        ERM(I)=100.*ERM(I)/RESID
   20 CONTINUE
C
  999 RETURN
      END
      SUBROUTINE E1(Y,Z)
C
C
C        PURPOSE:  COMPUTE THE EXPONENTIAL INTEGRAL
C           THIS ROUTINE IS ACCURATE TO THE EIGHTH PLACE
C
C
C       DESCRIPTION OF PARAMETERS:
C             Y    - ARGUMENT OF EXPONENTIAL INTEGRAL
C             Z    - RESULT VALUE
C             AUX  - RESULTANT AUXILIARY VALUE
C
C
C        REMARKS:
C              Y GT 170 (LT -174) MAY CAUSE UNDERFLOW (OVERFLOW)
C               FOR Y=0, THE RESULT VALUE IS SET TO -1.0E75
C
C         THIS REQUIRES NO OTHER ROUTINE
C
       IMPLICIT REAL*8 (A-H,O-Z)
```

```
      Z=0.
      IF (Y-1.) 2,1,1
 1    CONTINUE
      S=1.D0/Y
      AUX=1.D0 -S
     &*(((S+3.377358D0)*S+2.052156D0)*S+2.709479D-1)/((((S*
     &1.072553D0+5.716943D0)*S+6.945239D0)*S+2.593888D0)*S+2.709496D-1)
      XY=-Y
      IF (XY.GE.-160) Z=AUX*S*DEXP(XY)
  997 RETURN
 2    IF (Y+3.) 6,6,3
 3    AUX=(((((((7.122452D-7*Y-1.766345D-6)*Y+2.928433D-5)*Y-2.335379D-4
     &)*Y+1.664156D-3)*Y-1.041576D-2)*Y+5.555682D-2)*Y-2.500001D-1)*Y
     &+9.999999D-1
      Z=-1.D75
      IF (Y) 4,998,4
 4    CONTINUE
      Z=(Y*AUX-CLOG(DABS(Y))-5.772157D-01)
  998 RETURN
 6    IF (Y+9.) 8,6,7
 7    CONTINUE
      AUX=1.D0
     &-((((5.176245D-2*Y+3.061037D0)*Y+3.243665D1)*Y+2.244234D2)*Y
     &+2.486697D2)/((((Y+3.995161D0)*Y+3.893944D1)*Y+2.263818D1)*Y
     &+1.807837D2)
      GO TO 9
 8    CONTINUE
      S=9.D0/Y
      AUX=1.D0-S*(((S+7.659824D-1)*S-7.271015D-1)*S-1.080693D0)/((((S
     &*2.516750D0+1.122927D1)*S+5.921405D0)*S-8.666702D0)*S-9.724216D0)
 9    CONTINUE
      XY=-Y
      IF (XY.GE.-160) Z=AUX*DEXP(XY)/Y
C
  999 RETURN
      END
```