



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

Computer Analysis of Pump Test Data

Thomas H. Kean, *Governor*
Robert E. Hughey, *Commissioner*

Computer Analysis of Pump Test Data

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

by
Jeffrey L. Hoffman, *Senior Geologist*

John W. Gaston, Jr., P.E., *Director*
Frank J. Markewicz, *Acting State Geologist*
Haig F. Kasabach, *Deputy State Geologist*

Division of Water Resources
CN 029
Trenton 08625

1983

CONTENTS

Abstract.....	v
I. Introduction.....	1
II. Assumptions.....	2
III. Theis Analysis.....	3
IV. Jacob Analysis.....	8
V. Calculated Recovery Analysis.....	9
VI. Residual Drawdown.....	11
VII. Data Input.....	12
VIII. Using CAPTD.....	14
IX. Example.....	20
References.....	29
Appendix A. Mathematical Basis for Theis Curve Matching Technique..... 31	
Appendix B. Sample Input Data.....	34
Appendix C. Sample Output.....	35
Appendix D. 1. EXEC Listing.....	39
Appendix D. 2. CAPGO Listing.....	41
Appendix D. 3. CAPTD Listing.....	45

ILLUSTRATIONS

Figure 1. Theis Type Curve and Data.....	4
Figure 2. Dt, D δ Pairs.....	6
Figure 3. Time - Drawdown Data Relationship.....	10
Figure 4. Data Input Format.....	13
Figure 5. CAPTD Flow Chart.....	16
Figure 6. Theis Analysis Plot.....	21
Figure 7. Jacob Analysis Plot.....	22
Figure 8. Jacob Analysis Plot.....	23
Figure 9. Jacob Analysis Plot.....	24
Figure 10. Calculated Recovery Analysis Plot.....	25
Figure 11. Calculated Recovery Analysis Plot.....	26
Figure 12. Residual Drawdown Plot.....	27
Figure 13. Residual Drawdown Plot.....	28
Table 1. F and G Values for Dt, D δ Definition.....	33

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:

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

where

s	= drawdown	[L]	(2A)
Q	= pumping rate	[L ³ /T]	(2B)
T	= transmissivity	[L ² /T]	(2C)
W(u)	= Theis well function	[-]	(2D)
u	= r ² S/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 s 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.

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

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

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.

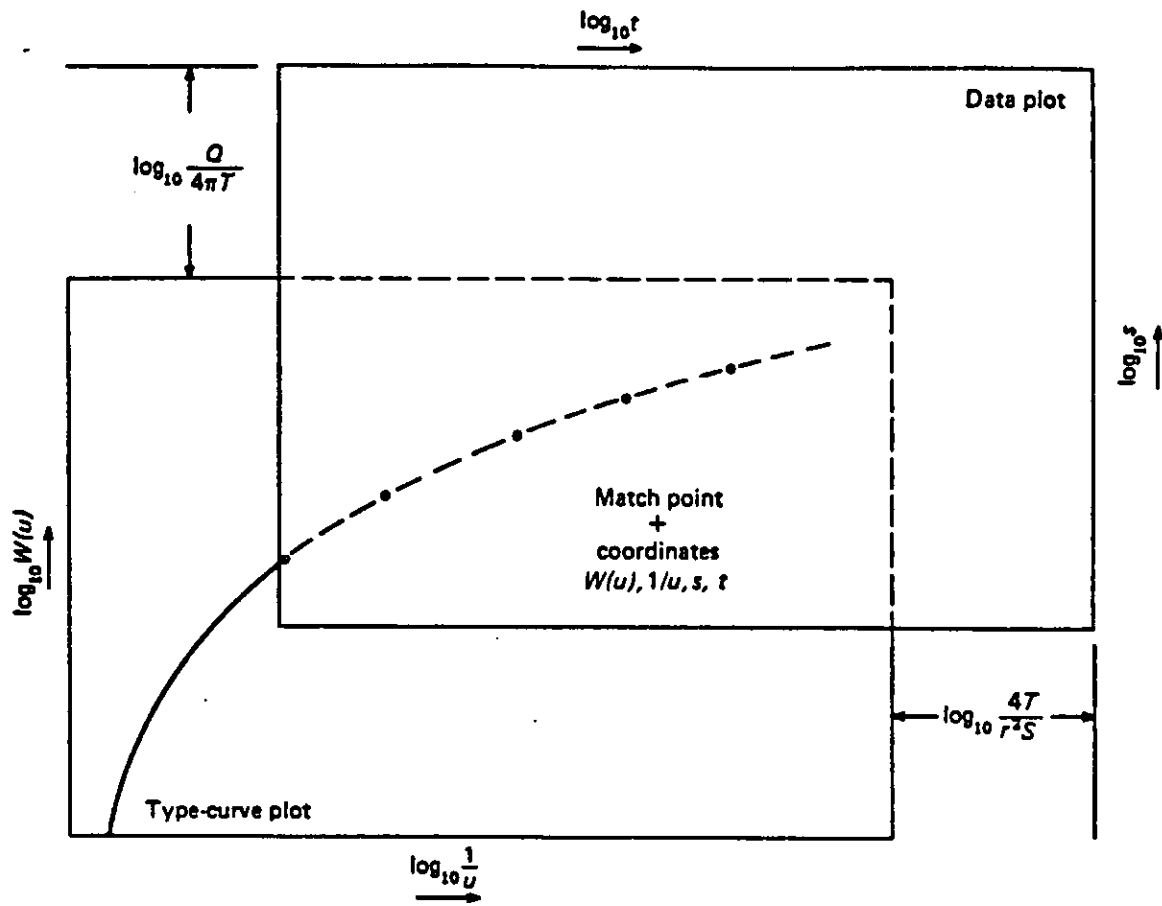


Figure 1. Theis Type Curve and Data (from Reed, 1980)

CAPTD performs the Theis analysis by shifting the $\log(t)$ and $\log(S)$ 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 $\log(S)$ data are overlain upon the Theis type curve at nine different positions by shifting the data values specific distances Δt and ΔS 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 S 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 R_4 and R_6 is greater than the sum of R_2 and R_8 the value of Δt is divided by 1.5. If the sum of R_2 and R_8 is greater the value of ΔS 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 , S , $1/u$, $W(u)$) are then substituted into equations 3 and 4 to generate values of T and S .

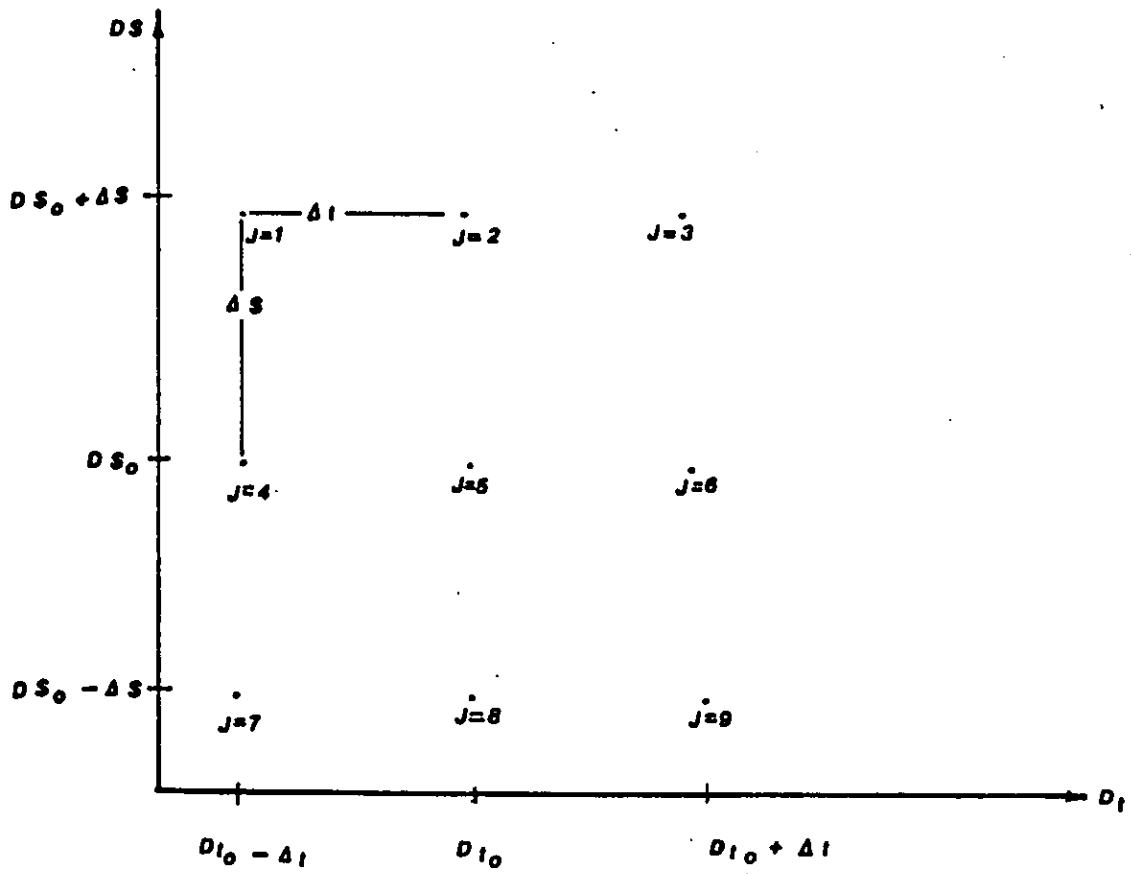


Figure 2. $D_t, D\$$ Pairs

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

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

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

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

Substituting equations 6 and 2e into 1 results in

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

This can be rewritten as

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

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

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

At the point where the drawdown is equal to 0.0, let the time value be equal to t_0 . At this point, equation 8 becomes

$$0 = \log \left(\frac{2.25 T t_0}{r^2 S} \right) \quad (10)$$

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

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

$$S = \frac{2.25 T t_0}{r^2 S} \quad (12)$$

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_0) is determined. Equations 11 and 12 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.

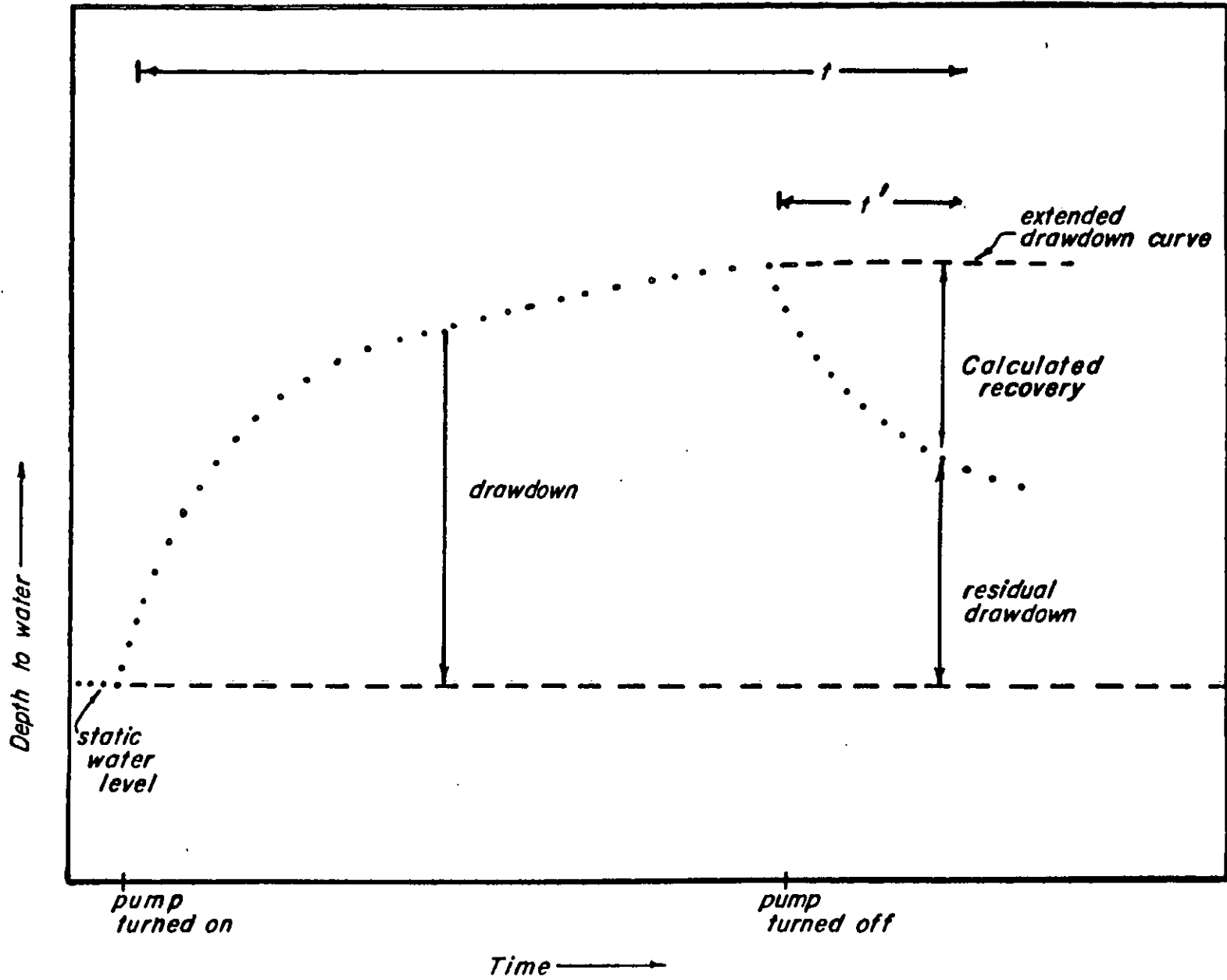


Figure 3. Time-Drawdown Data Relationships

VI. Residual Drawdown

The residual drawdown analysis is a method for calculating the aquifer transmissivity from the relationship between residual drawdown (s') 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). s' is plotted on the arithmetic scale of semi-log paper and t/t' is plotted on the log axis.

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

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

The slope of this line is thus:

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

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

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

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. Similarly, 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 observa- tion 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.

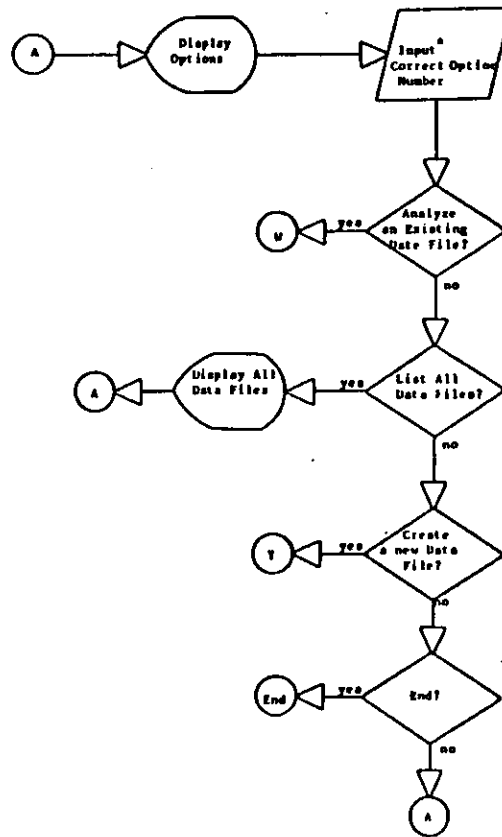
For the first calculated recovery analysis after a Theis or Jacob analysis, CAPTD automatically sets the time limits so as to include all recovery data. For each successive calculated recovery analysis the user sets the limits.

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.

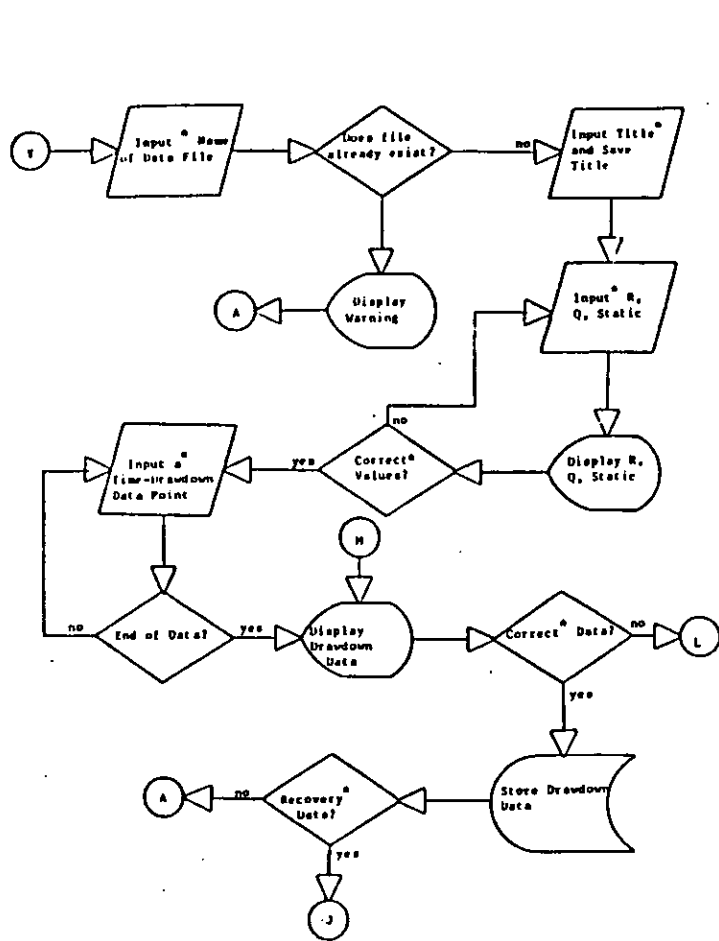
Figure 5 shows a flow chart which graphically displays the steps and options in CAPTD.

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.

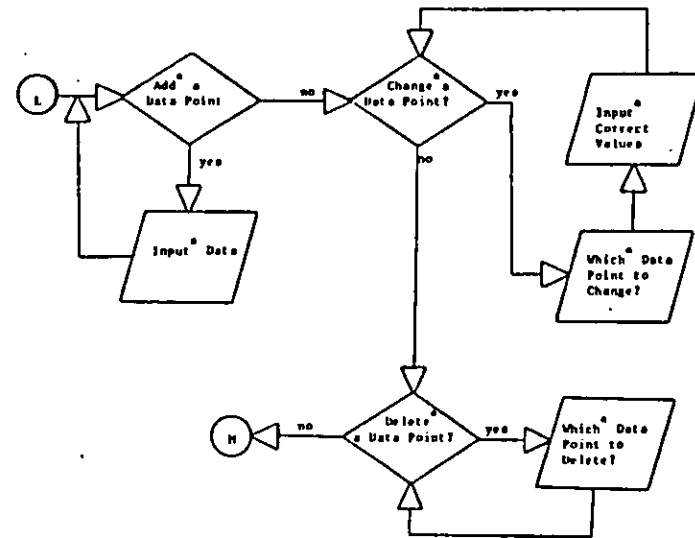
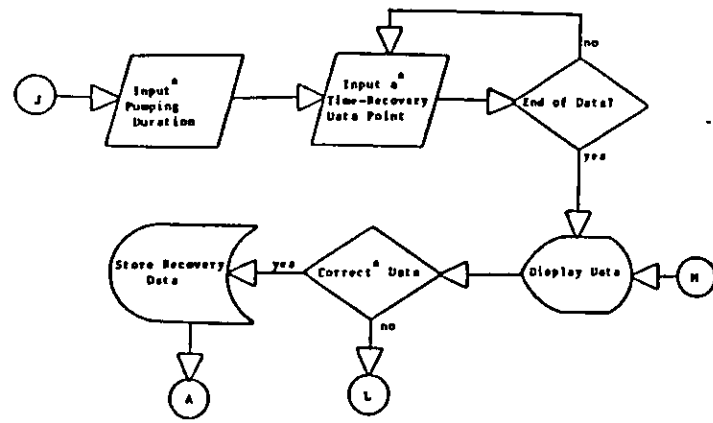


* Step requires user input from terminal

Figure 5. CAPTD Flow Chart



* Step requires user input from the terminal



* Step requires user input from the terminal

Figure 5 continued.

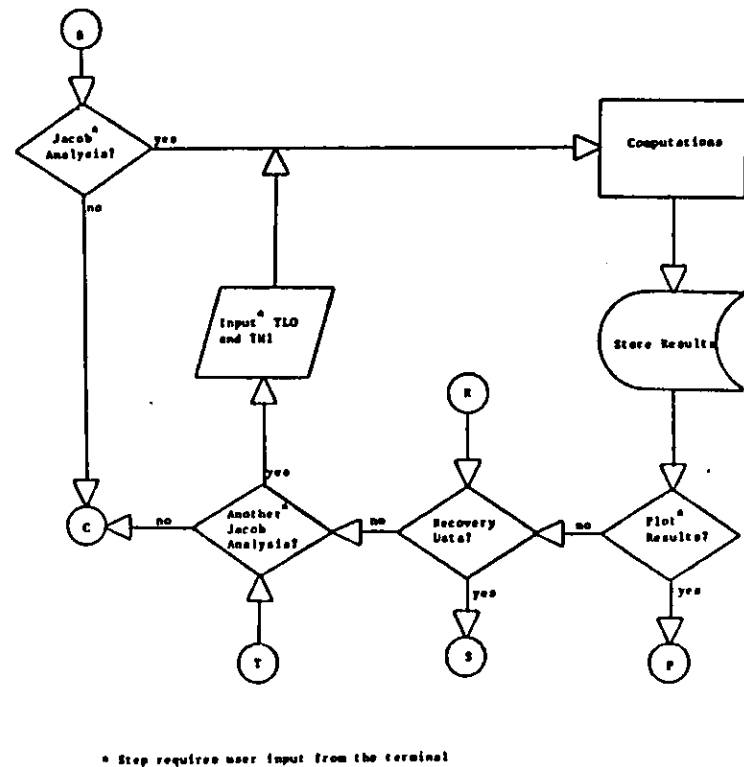
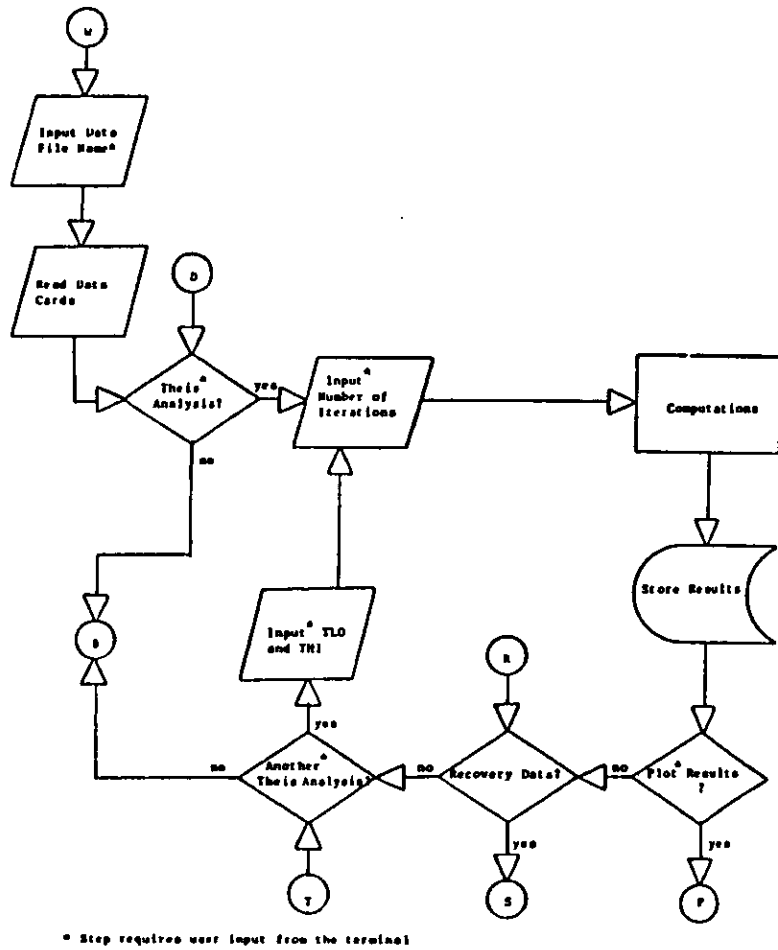
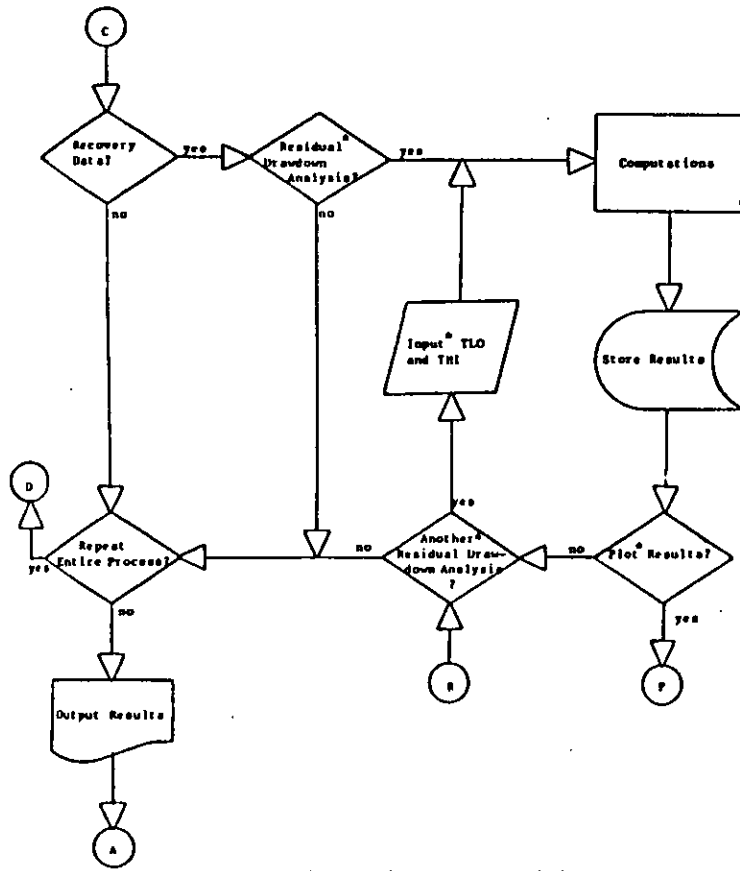
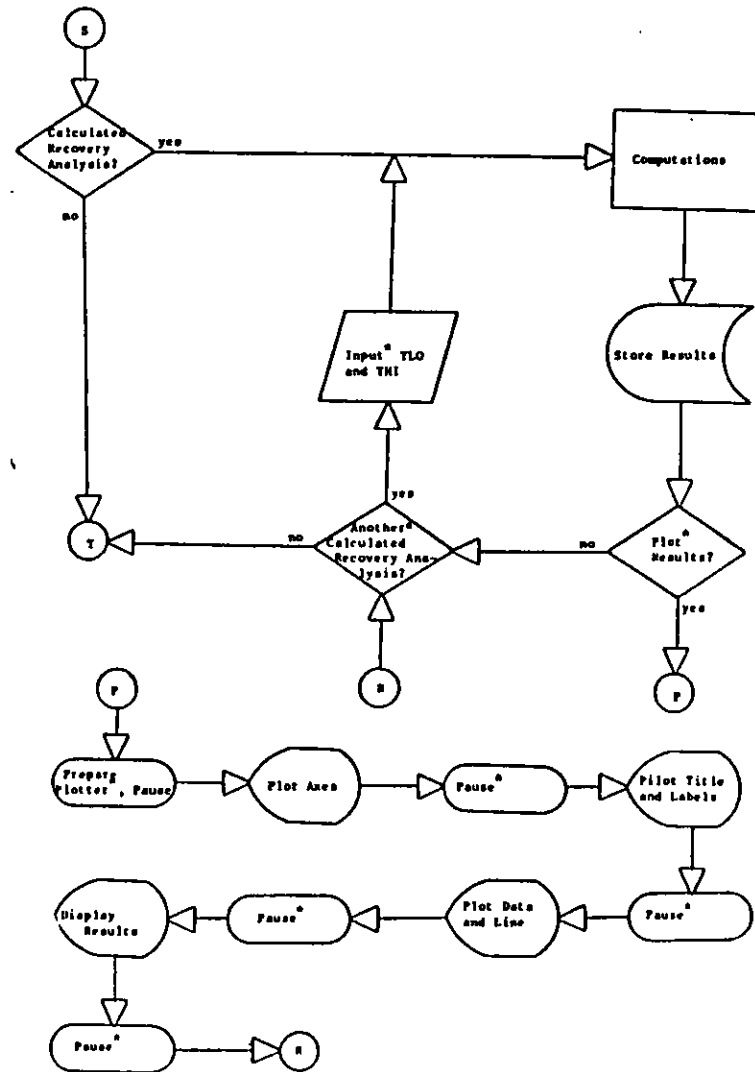


Figure 5 continued.



* Step requires user input from the terminal



* Step requires user input from the terminal

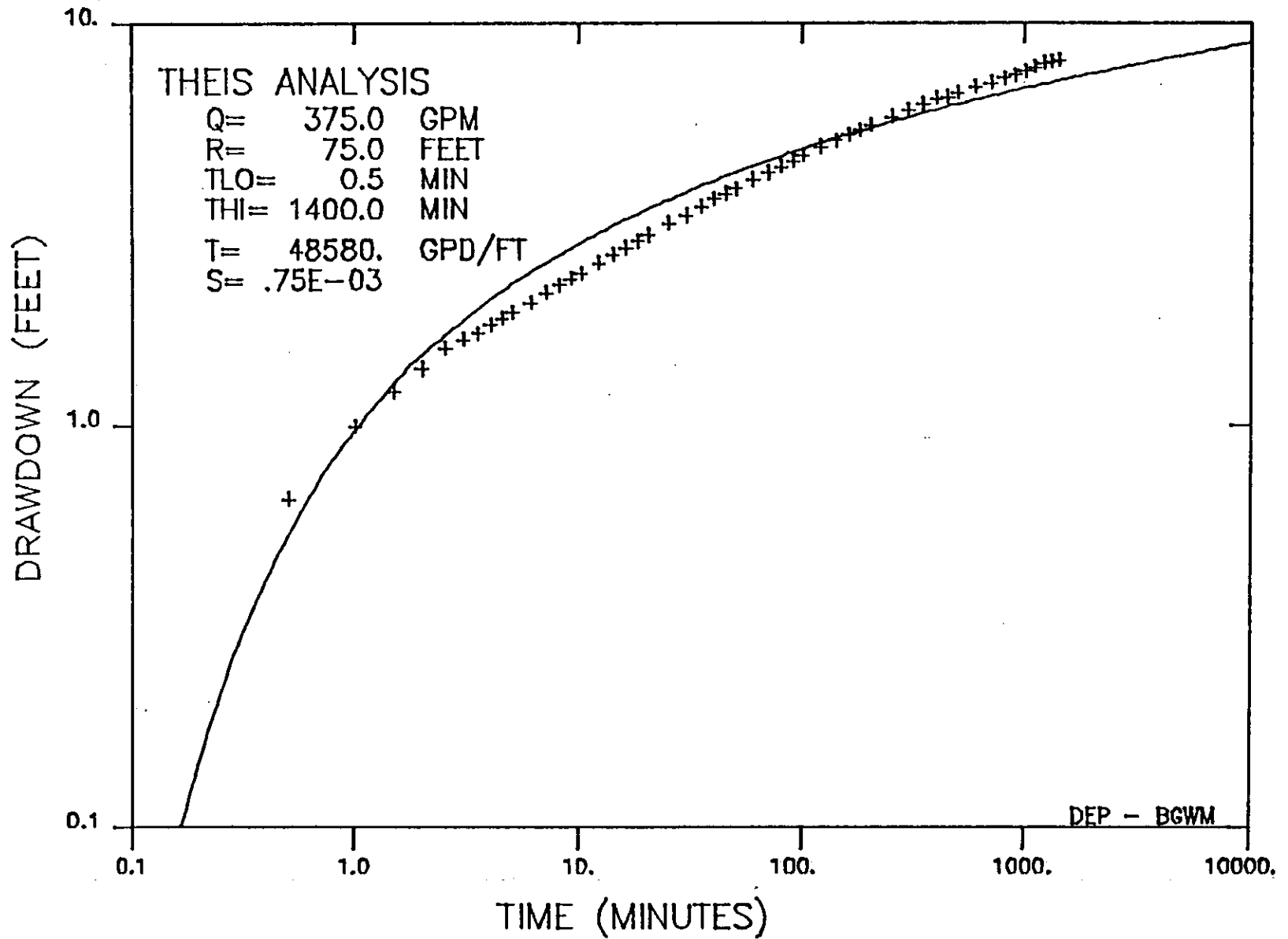
Figure 5 continued.

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.

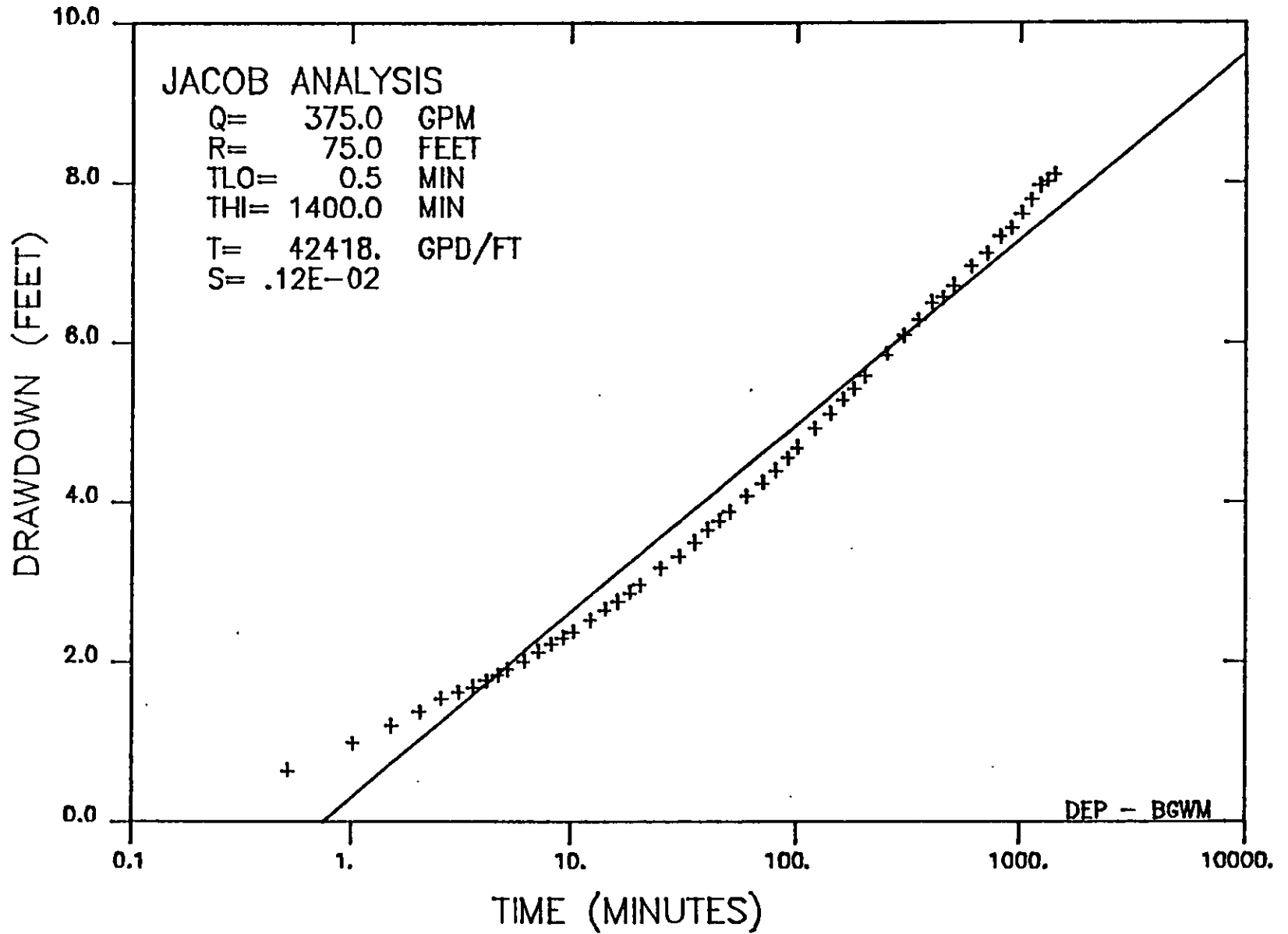
GREENPOND TW5 OW5

Figure 6. Theis Analysis Plot



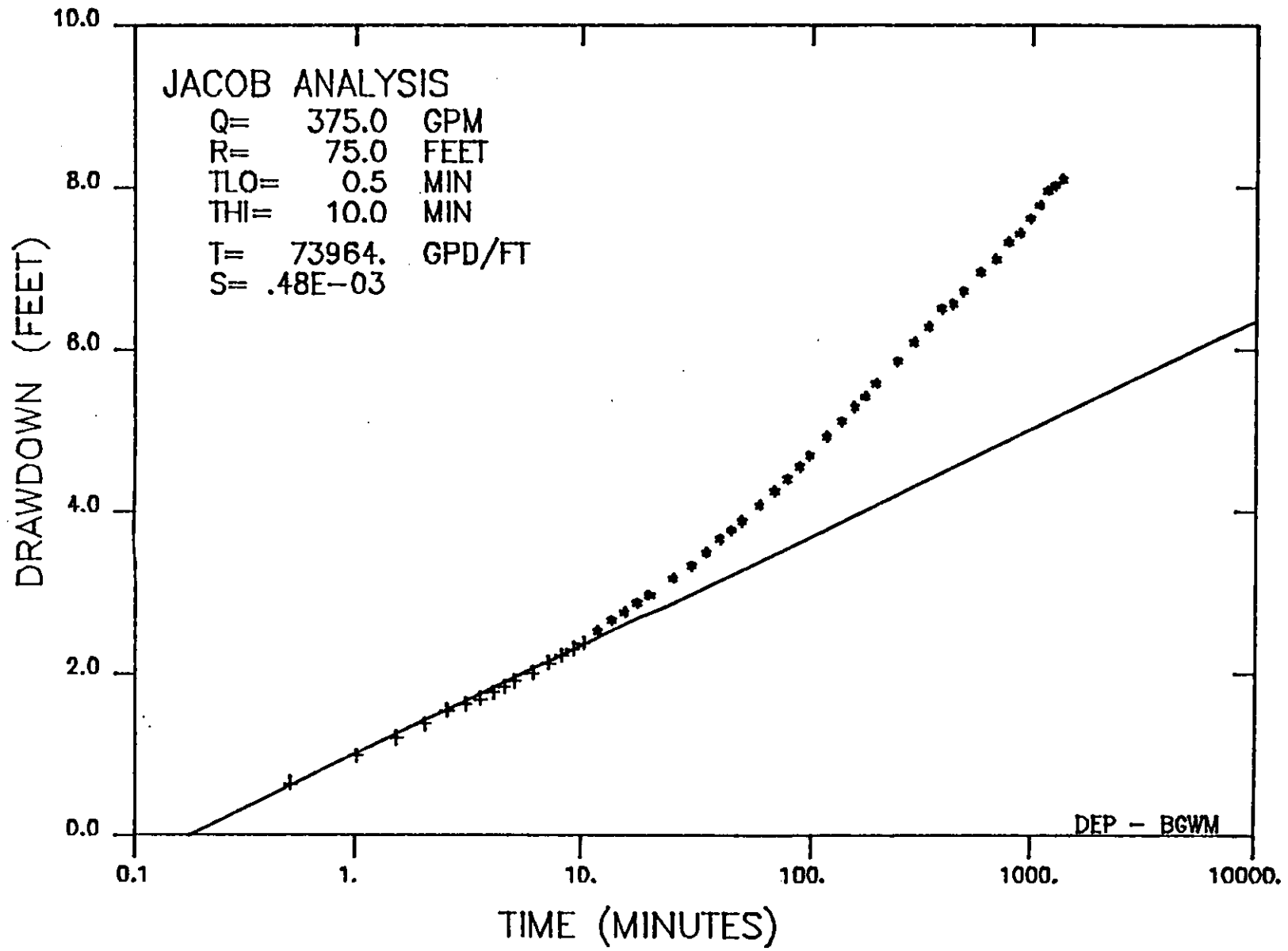
GREENPOND TW5 OW5

Figure 7. Jacob Analysis Plot



GREENPOND TW5 OW5

Figure 8. Jacob Analysis Plot



GREENPOND TW5 OW5

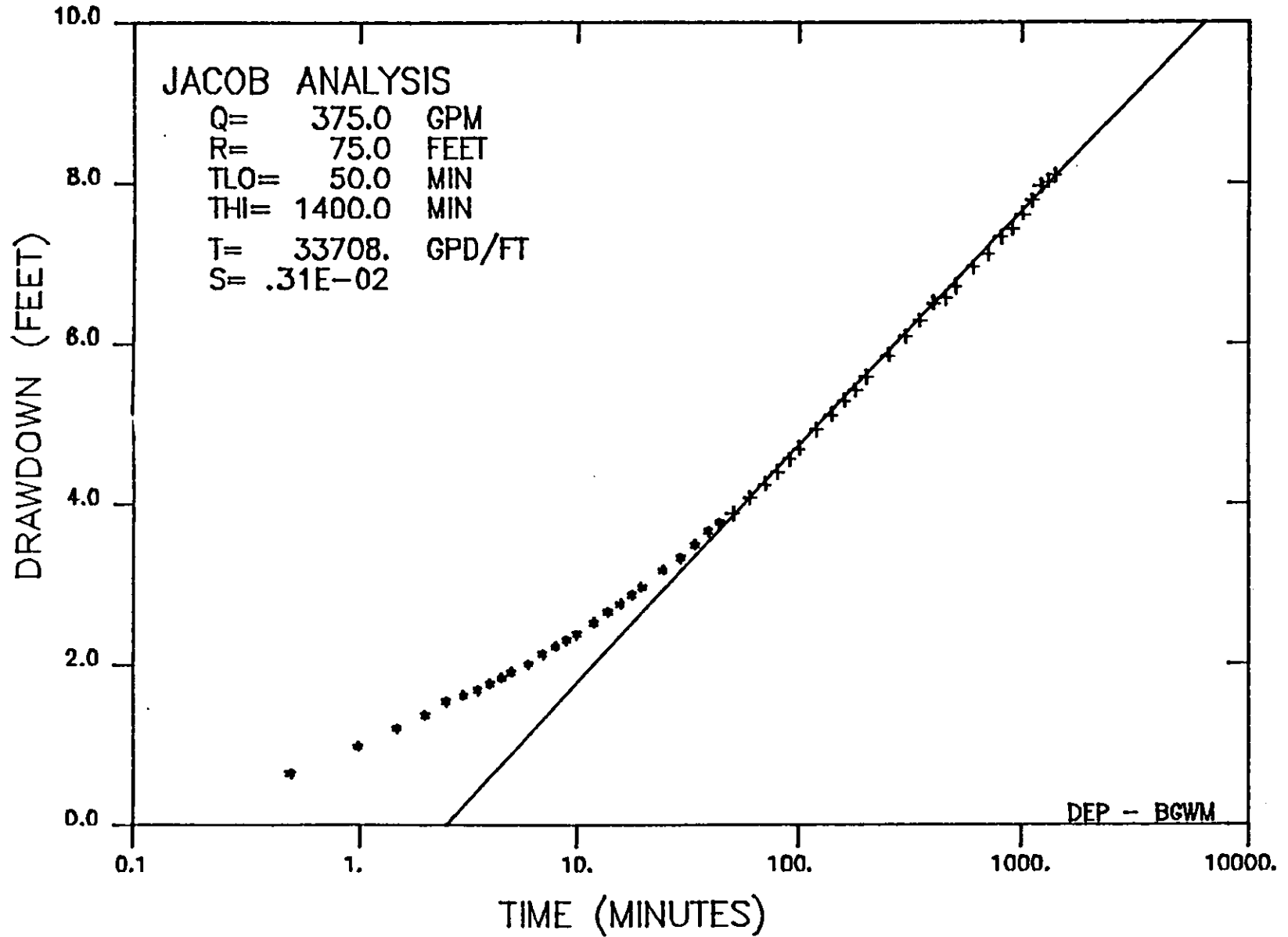
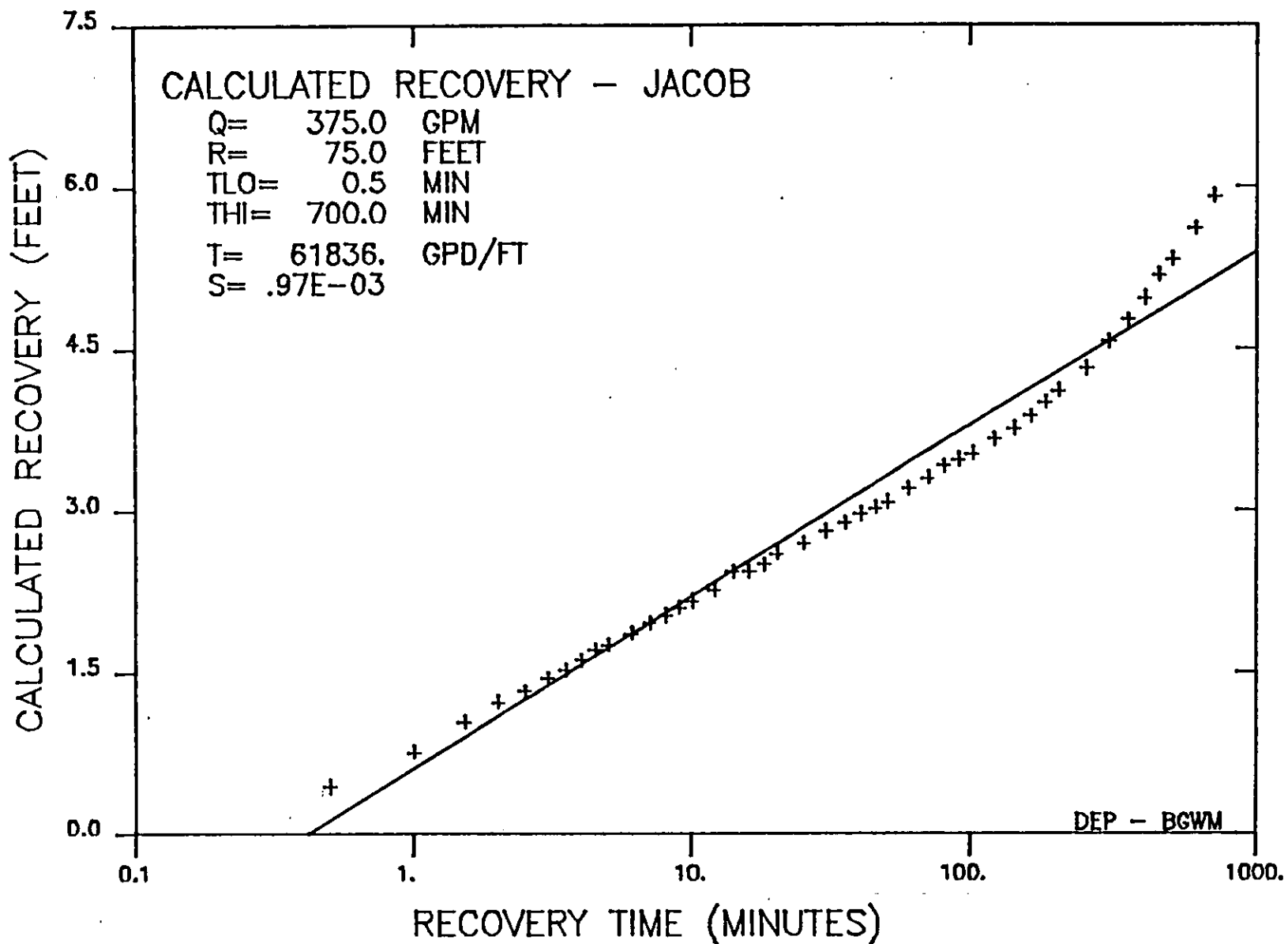


Figure 9. Jacob Analysis Plot

GREENPOND TW5 OW5

Figure 10. Calculated Recovery Analysis Plot



GREENPOND TW5 OW5

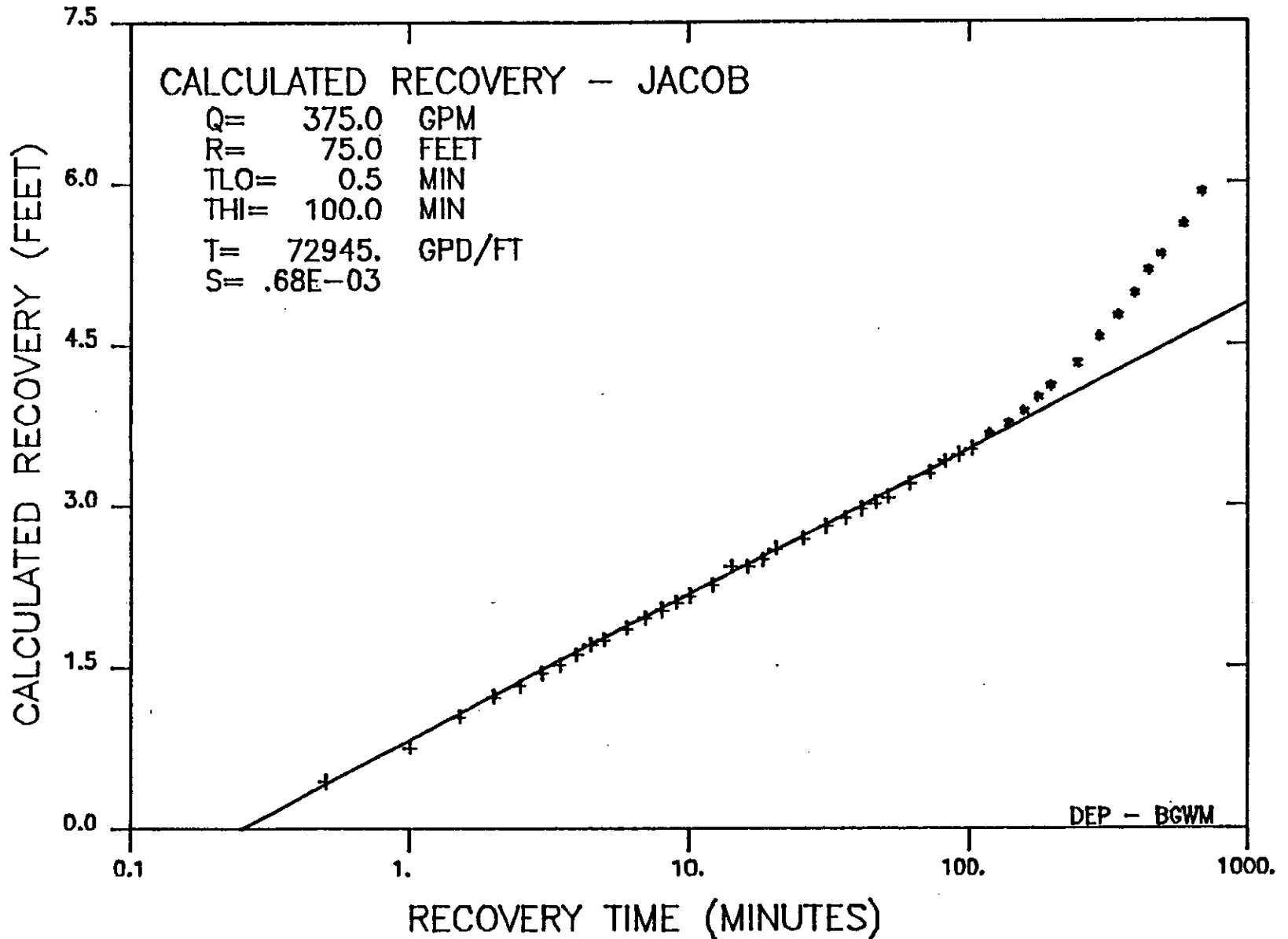
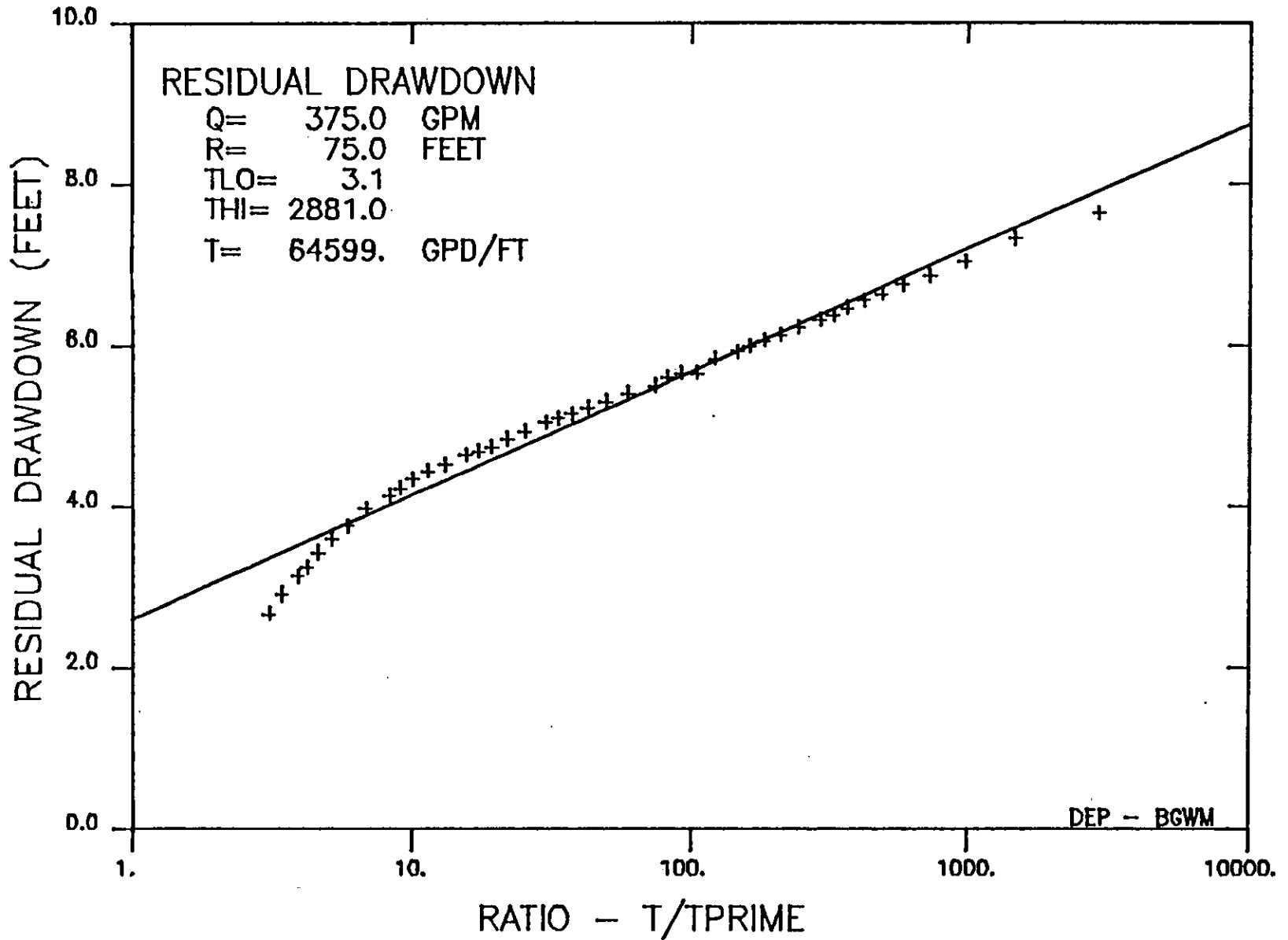


Figure 11. Calculated Recovery Analysis Plot

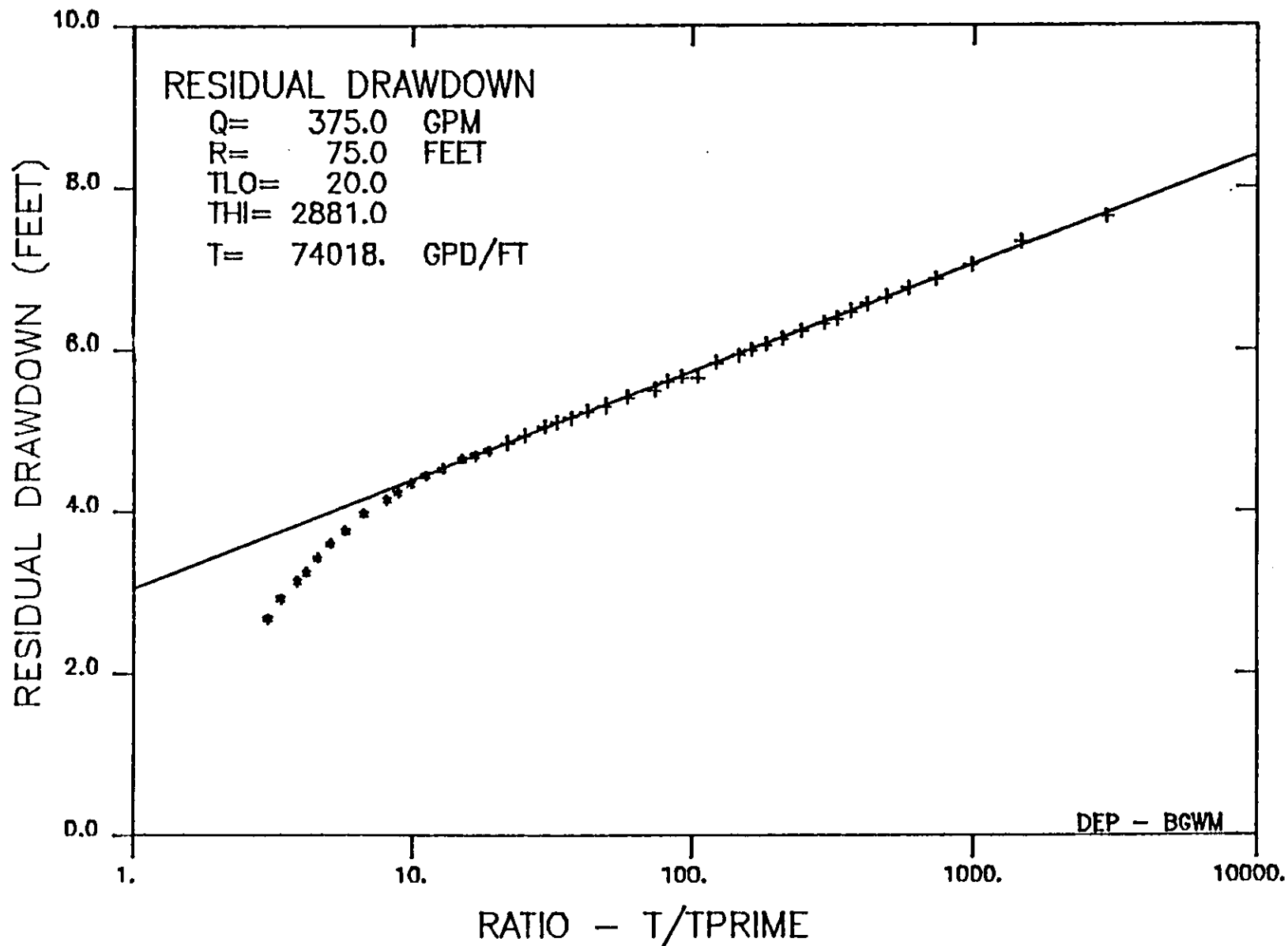
GREENPOND TW5 OW5

Figure 12. Residual Drawdown Plot



GREENPOND TW5 OW5

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:

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

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

where

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

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

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

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

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:

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

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

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

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:

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

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

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:

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

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

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 iteration.

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

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

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

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

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

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

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 and $D\$\circ$ are reassigned to be that $(Dt, D\$\circ)$ 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 and $D\$\circ$ 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 $\Delta\$\circ$ 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 , $D\$\circ$ definition

GREENPOND	TW5	DW5						
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

FEBRUARY 08, 1983

CGREENPOND TW5 0W5

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

```

&CCNTROL OFF NOMSG
CP LINK DEPGIS 193 193 KR 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 &I

&IF &I EQ 1 &GOTO -RUN

&IF &I EQ 2 &GOTO -LIST

&IF &I EQ 3 &GOTO -NEW

&IF &I EQ 4 &GOTO -STOP

&IF &I EQ STOP GO TO -STOP

&IF &I EQ END GO TO -STOP

&IF &I 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 PLOTTEKZ 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 8)
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
```

```

0001          REAL*4 T(200),S(200)
0002          REAL YN,YES/'YES '/,NO/'NO '/
0003          LOGICAL*1 TITLE(25),AMP/'C'/
C
C---INPUT TITLE AND PARANTERS
0004          WRITE (5,501)
0005          READ (5,502) (TITLE(I),I=2,25)
0006          TITLE(1)=AMP
0007          WRITE (6,601) (TITLE(I),I=1,25)
0008          10 WRITE (5,503)
0009          READ (5,*) R
0010          WRITE (5,504)
0011          READ (5,*) Q
0012          WRITE (5,505)
0013          READ (5,*) STATIC
0014          WRITE (5,507) R,Q,STATIC
0015          WRITE (5,508)
0016          READ (5,509) YN
0017          IF (YN.EQ.YES) GO TO 10
C---INPUT THE DATA POINTS
0018          WRITE (5,510)
0019          N=0
0020          20 N=N+1
0021          READ (5,*,END=30) T(N),S(N)
0022          GO TO 20
0023          30 REWIND 5
C---CHECK THE DATA POINTS
0024          N=N-1
0025          CALL CHECK (T,S,N)
C---OUTPUT THE DRAWDOWN DATA
0026          WRITE (6,602) R,Q,STATIC,N
0027          WRITE (6,603) (T(I),S(I),I=1,N)
C---RECOVERY DATA?
0028          WRITE (5,511)
0029          READ (5,509) YN
0030          IF (YN.NE.YES) GO TO 999
C---INPUT PARAMTER
0031          WRITE (5,512)
0032          READ (5,*) TPUMP
0033          WRITE (5,510)
C---INPUT THE DATA POINTS
0034          NREC=0
0035          40 NREC=NREC+1
0036          READ (5,*,END=50) T(NREC),S(NREC)
0037          GO TO 40
0038          50 REWIND 5
C---CHECK THE DATA POINTS
0039          NREC=NREC-1
0040          CALL CHECK (T,S,NREC)
0041          WRITE (6,605) TPUMP, NREC
0042          WRITE (6,603) (T(I),S(I),I=1,NREC)
C---FORMAT STATEMENTS
0043          501 FORMAT (I2X,'WHAT IS TITLE FOR PLOT (24 CHARACTERS OR LESS)?')
0044          502 FORMAT (Z5A1)
0045          503 FORMAT (2X,'WHAT IS R (FEET)?')
0046          504 FORMAT (2X,'WHAT IS Q (GPM)?')
0047          505 FORMAT (2X,'WHAT WAS STATIC DEPTH TO WATER (IN FEET) BEFORE',/
          65X,'PUMPING BEGAN?')
CAP00010
CAP00020
CAP00030
CAP00040
CAP00050
CAP00060
CAP00070
CAP00080
CAP00090
CAP00100
CAP00110
CAP00120
CAP00130
CAP00140
CAP00150
CAP00160
CAP00170
CAP00180
CAP00190
CAP00200
CAP00210
CAP00220
CAP00230
CAP00240
CAP00250
CAP00260
CAP00270
CAP00280
CAP00290
CAP00300
CAP00310
CAP00320
CAP00330
CAP00340
CAP00350
CAP00360
CAP00370
CAP00380
CAP00390
CAP00400
CAP00410
CAP00420
CAP00430
CAP00440
CAP00450
CAP00460
CAP00470
CAP00480
CAP00490
CAP00500
CAP00510
CAP00520
CAP00530
CAP00540
CAP00550
CAP00560
CAP00570
CAP00580

```

Appendix D?. CAPGO List

```
0048      507 FORMAT (2X,'R,Q,STATIC:',/3F10.2)          CAP00590
0049      508 FORMAT (2X,'CHANGE ANY OF THESE VALUES? (YES/NO)') CAP00600
0050      509 FORMAT (A4)                                  CAP00610
0051      510 FORMAT (/2X,'INPUT THE TIME-DEPTH TO WATER DATA PAIRS. AFTER',/ CAP00620
          & 2X,'EACH PAIR HIT THE "RETURN" KEY. TO SIGNAL THE END OF THE',/CAP00630
          & 2X,'DATA, INPUT A LINE WITH NO DATA ON IT.') CAP00640
0052      511 FORMAT (/2X,'ANY RECOVERY DATA? (YES/NO)') CAP00650
0053      512 FORMAT (2X,'HOW LONG DID THE PUMPING PERIOD LAST? (MINUTES)') CAP00660
0054      601 FORMAT (25A1)                                CAP00670
0055      602 FORMAT (3F10.2,110)                         CAP00680
0056      603 FORMAT (8F9.2)                               CAP00690
0057      604 FORMAT (F8.2,110)                           CAP00700
0058      605 FORMAT (F10.2,110)                          CAP00710
          C
0059      999 STOP                                         CAP00720
0060      END                                             CAP00730
                                         CAP00740
```

```

0001          SUBROUTINE CHECK (T,S,N)                                CAP00750
      C                                                                 CAP00760
0002          REAL T(200),S(200)                                    CAP00770
0003          REAL YN, YES/'YES '/, NO/'NO '/                       CAP00780
0004          INTEGER IN(200),KK(3)                                  CAP00790
      C                                                                 CAP00800
0005          DO 10 I=1,200                                          CAP00810
0006             10  IN(I)=I                                          CAP00820
      C---OUTPUT DATA POINTS TO SCREEN                               CAP00830
0007             20  WRITE (5,501)                                    CAP00840
0008                NROW=N/3                                          CAP00850
0009                IREM=N-3*NROW                                       CAP00860
0010                K1=NROW+1                                           CAP00870
0011                KK(1)=0                                             CAP00880
0012                KK(2)=NROW                                           CAP00890
0013                IF (IREM.GE.1) KK(2)=KK(2)+1                       CAP00900
0014                KK(3)=KK(2)+NROW                                       CAP00910
0015                IF (IREM.GE.2) KK(3)=KK(3)+1                       CAP00920
0016                WRITE (5,502) ((IN(I+KK(J)),T(I+KK(J)),S(I+KK(J)),J=1,3),I=1,NROW) CAP00930
0017                IF (IREM.GT.0) WRITE (5,502) ((IN(K1+KK(J)),T(K1+KK(J)),S(K1+KK(J)),
      6          ,J=1,IREM)                                           CAP00940
      C---CHANGE DATA?                                             CAP00950
0018                WRITE (5,503)                                       CAP00960
0019                READ (5,510) YN                                       CAP00970
0020                IF (YN.NE.YES) GO TO 999                             CAP00980
      C---ADD A DATA PAIR                                          CAP00990
0021                30  WRITE (5,504)                                    CAP01000
0022                READ (5,510) YN                                       CAP01010
0023                IF (YN.NE.YES) GO TO 40                             CAP01020
0024                N=N+1                                               CAP01030
0025                WRITE (5,505)                                       CAP01040
0026                READ (5,*) T(N),S(N)                                   CAP01050
0027                GO TO 30                                             CAP01060
      C---CHANGE A DATA PAIR                                       CAP01070
0028                40  WRITE (5,506)                                    CAP01080
0029                READ (5,510) YN                                       CAP01090
0030                IF (YN.NE.YES) GO TO 50                             CAP01100
0031                WRITE (5,507)                                       CAP01110
0032                READ (5,*) J                                          CAP01120
0033                WRITE (5,508) J                                       CAP01130
0034                READ (5,*) T(J),S(J)                                   CAP01140
0035                GO TO 40                                             CAP01150
      C---DELETE A DATA PAIR?                                       CAP01160
0036                50  WRITE (5,509)                                    CAP01170
0037                READ (5,510) YN                                       CAP01180
0038                IF (YN.NE.YES) GO TO 20                             CAP01190
0039                WRITE (5,511)                                       CAP01200
0040                READ (5,*) J                                          CAP01210
0041                DO 60 I=J,N                                           CAP01220
0042                   T(I)=T(I+1)                                       CAP01230
0043                   S(I)=S(I+1)                                       CAP01240
0044                60  CONTINUE                                         CAP01250
0045                N=N-1                                               CAP01260
0046                GO TO 50                                             CAP01270
      C---FORMAT STATEMENTS                                         CAP01280
0047                501 FORMAT (1/2X,'DATA POINTS',/3(3X,'*',5X,'TIME',5X,'DTW',3X)) CAP01290
0048                502 FORMAT (3(1X,13,2X,F8.2,2X,F6.2,2X))           CAP01300
0049                503 FORMAT (2X,'CHANGE ANY DATA? (YES/NO)')       CAP01310
                                                                    CAP01320

```

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 #',13,'?')	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	
0058	999 RETURN	CAP01410
0059	END	CAP01420 CAP01430


```

C 20. E1 - EXPONENTIAL INTEGRAL VWS00560
C VWS00570
C VWS00580
C VWS00590
C REFERENCES: VWS00600
C "ANALYSIS AND EVALUATION OF PUMPING TEST DATA," G. P. KRUSEMAN VWS00610
C AND N.A. DE KIDDER, 1979. VWS00620
C VWS00630
C "GROUND WATER AND WELLS," JOHNSON DIVISION, UCP INC., 1975. VWS00640
C VWS00650
C VWS00660
C PROGRAMMER: VWS00670
C JEFFREY L. HOFFMAN VWS00680
C NEW JERSEY DEPARTMENT OF ENVIRONMENTAL PROTECTION VWS00690
C DIVISION OF WATER RESOURCES VWS00700
C WATER QUALITY MANAGEMENT ELEMENT VWS00710
C BUREAU OF GROUND WATER MANAGEMENT VWS00720
C GROUND WATER RESOURCE EVALUATION SECTION VWS00730
C DECEMBER 1982 VWS00740
C VWS00750
C REVISED JUNE 1983 TO ACCOUNT FOR NEW GRAPHICS PROGRAMS VWS00760
C VWS00770
C VWS00780
C VWS00790
C VWS00800
C-----VWS00810
C VWS00820
C REAL T(200),S(200),LT(200),T2(200),S2(200),LT2(200) VWS00830
C REAL LS(200),ERP(200),LTX(200),LSX(200) VWS00840
C REAL YN, YES/'YES' /, NO/'NO' / VWS00850
C LOGICAL*1 TITLE(25),RECCVR VWS00860
C.....GRAPHING ARRAYS AND COMMON BLOCKS VWS00870
C DIMENSION IXY(2,6000), IPCINT(2,6000) VWS00880
C LOGICAL TEK, CAL VWS00890
C LOGICAL*1 LIFT(6000), STRING(130), ARG(130) VWS00900
C COMMON /SPUT/ TEK, CAL, NPATS, IXY, LIFT VWS00910
C COMMON /SPARE/ IPCINT, IPTR VWS00920
C COMMON /DEVICE/ IXLIM, IYLM, MAXXD, MAXYD, IXLIN, IYLIN VWS00930
C COMMON /PARS/ LEN, INDEX, NCHARS, NCMLEFT, ARG, STRING VWS00940
C.....END OF GRAPHICS REQUIREMENTS VWS00950
C COMMON /TITLES/ TITLE VWS00960
C COMMON /FLT/ XU,X1,YU,Y1 VWS00970
C COMMON /DATA/ N,R,L,TPUMP,NREC,STCR,TRAN,TRAN1,TRAN2,A,B,T0 VWS00980
C COMMON /FCINTS/ T,S,LT,T2,LS,ST,LT2 VWS00990
C COMMON /RANGE/ TLC,THI,SLC,SHI,T2LO,T2HI,S2LC,S2HI VWS01000
C COMMON /ICATA/ ERP,NPTS,NSLM VWS01010
C COMMON /ADD/ XADD,YADD,CX,DY,XORIG,YORIG VWS01020
C COMMON /FERR/ R1,R2,R3,R4,R5,R6,R7,R8,RR VWS01030
C COMMON /XTRA/ LTX,LSX VWS01040
C-----VWS01050
C VWS01060
C VWS01070
C---PREPARE PLOTTER TERMINAL VWS01080
C CALL FLTSET VWS01090
C CALL FLTERA VWS01100
C---SET UP PHYSICAL DIMENSIONS OF GRAPH, IN INCHES VWS01110

```

```

XC=1.7
X1=13.7
YC=1.2
Y1=5.2
C---READ IN DATA
CALL INPLT(RECOVR)
C---THEIS ANALYSIS?
10 WRITE (5,501)
READ (5,510) YN
IF (YN.EQ.YES) CALL THEIS(RECOVR)
C---JACCB ANALYSIS?
20 WRITE (5,502)
READ (5,510) YN
IF (YN.EQ.YES) CALL JACCB(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 (1/2X,'THEIS ANALYSIS? (YES/NO)')
502 FORMAT (1/2X,'JACCB ANALYSIS? (YES/NO)')
503 FORMAT (1/2X,'RESIDUAL DRAWDOWN ANALYSIS? (YES/NO)')
504 FORMAT (1/2X,'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 OUTPLT SENT TO UNIT 6
C
C CARD 1: TITLE
C LP TO 25 CHARACTERS
C THE FIRST CHARACTER IS A PLOTTING CONTROL CHARACTER
C THE USE OF A 'E' OR ' ' IS RECOMMENDED
C
C CARD 2:
C R- DISTANCE FROM PUMPING WELL TO OBSERVATION WELL (IN FEET)
C W- 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

```

VWS0111C
VWS0112C
VWS0113C
VWS0114C
VWS0115C
VWS0116C
VWS0117C
VWS0118C
VWS0119C
VWS0120C
VWS0121C
VWS0122C
VWS0123C
VWS0124C
VWS0125C
VWS0126C
VWS0127C
VWS0128C
VWS0129C
VWS0130C
VWS0131C
VWS0132C
VWS0133C
VWS0134C
VWS0135C
VWS0136C
VWS0137C
VWS0138C
VWS0139C
VWS0140C
VWS0141C
VWS0142C
VWS0143C
VWS0144C
VWS0145C
VWS0146C
VWS0147C
VWS0148C
VWS0149C
VWS0150C
VWS0151C
VWS0152C
VWS0153C
VWS0154C
VWS0155C
VWS0156C
VWS0157C
VWS0158C
VWS0159C
VWS0160C
VWS0161C
VWS0162C
VWS0163C
VWS0164C
VWS0165C

```

C CARD 3 TIME,CRAWDOWN DATA                                VNSC166C
C REALLY N CARDS HERE                                        VNSC167C
C EACH CARD HAS ON IT ONE DATA PAIR                       VNSC168C
C THE FIRST DATUM IS THE NUMBER OF MINUTES SINCE PUMPING  VNSC169C
C THE SECCND DATUM IS THE DEPTH TO WATER IN THE WELL AT   VNSC170C
C                                                           VNSC171C
C CARD 4                                                    VNSC172C
C TPUMP- TIME AT WHICH PUMPING STOPPED (MINUTES)          VNSC173C
C NREC- NUMBER OF TIME,RECCVRY DATA POINTS                VNSC174C
C                                                           VNSC175C
C CARD 5                                                    VNSC176C
C REALLY NREC CARDS HERE                                    VNSC177C
C EACH HAS ON IT ONE DATA PAIR                            VNSC178C
C THE FIRST DATUM IS THE NUMBER OF MINUTES SINCE PUMPING  VNSC179C
C THE SECCND DATUM IS THE DEPTH TO WATER IN THE WELL AT   VNSC180C
C                                                           VNSC181C
C VECTORS CREATED BY THIS SUBROUTINE                        VNSC182C
C T(I) - TIME SINCE PLMPNG BEGAN FOR ITH DATA PAIR , I=1  VNSC183C
C LT(I) - LOG (T(I))                                        VNSC184C
C S(I) - DRAWDCWN IN WELL FOR ITH DATA PAIR               VNSC185C
C LS(I) - LOG (S(I))                                       VNSC186C
C T2(J) - TIME SINCE PLMPNG STOPPED FOR JTH DATA PAIR,   VNSC187C
C LT2(J) - LOG (T2(J))                                     VNSC188C
C S2(J) - DRAWDCWN IN WELL FOR JTH DATA PAIR             VNSC189C
C                                                           VNSC190C
C SCALARS CREATED BY THIS SUBROUTINE                        VNSC191C
C TLC - MINIMUM VALUE OF THE VECTOR T                      VNSC192C
C THI - MAXIMUM VALUE OF THE VECTOR T                      VNSC193C
C SLC - MINIMUM VALUE OF THE VECTOR S                      VNSC194C
C SHI - MAXIMUM VALUE OF THE VECTOR S                      VNSC195C
C T2LC - MINIMUM VALUE OF THE VECTOR T2                   VNSC196C
C T2HI - MAXIMUM VALUE OF THE VECTOR T2                   VNSC197C
C S2LC - MINIMUM VALUE OF THE VECTOR S2                   VNSC198C
C S2HI - MAXIMUM VALUE OF THE VECTOR S2                   VNSC199C
C                                                           VNSC200C
C LOGICAL VARIABLES INITIALIZED BY THIS SUBROUTINE        VNSC201C
C RECCVR - THIS IS .TRUE. IF TIME-RECCVRY DATA WAS READ  VNSC202C
C IF IS .FALSE. OTHERWISE                                  VNSC203C
C                                                           VNSC204C
C                                                           VNSC205C
C                                                           VNSC206C
C ----- VNSC207C
C                                                           VNSC208C
C REAL T(200),S(200),LT(200),T2(200),S2(200),LT2(200),LS(200) VNSC209C
C REAL TEMP(200)                                           VNSC210C
C INTEGER IN(200),KK(3)                                     VNSC211C
C LOGICAL*1 TITLE(25),RECCVR                               VNSC212C
C DIMENSION IDATE(16)                                      VNSC213C
C COMMON /TITLES/ TITLE                                     VNSC214C
C COMMON /RANGE/ TLC,THI,SLC,SHI,T2LC,T2HI,S2LC,S2HI     VNSC215C
C COMMON /DATA/ N,R,C,TPUMP,NREC,STCR,TRAN,TRAN1,TRAN2,A,B,TO VNSC216C
C COMMON /PCINTS/ T,S,LT,LS,T2,S2,LT2                    VNSC217C
C                                                           VNSC218C
C ----- VNSC219C
C                                                           VNSC220C

```

```

C---DATE
  CALL MCDAYR(ICATE)
  WRITE (6,606) ICATE
C---INITIALIZE
  DC 5 I=1,200
    5 IN(I)=I
C
C---READ IN TIME, CRAWDOWN DATA
  READ (4,501) (TITLE(I),I=1,25)
  READ (4,*) R,G,STATIC,N
  TLC=100000.
  THI=-TLC
  SLC=TLC
  SHI=THI
  READ (4,*) (T(I),TEMP(I),I=1,N)
  DC 10 I=1,N
    S(I)=TEMP(I)-STATIC
    IF (T(I).LT.TLC) TLC=T(I)
    IF (T(I).GT.THI) THI=T(I)
    IF (S(I).LT.SLC) SLC=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,G,STATIC
  WRITE (6,602) N
  WRITE (6,604)
  NRCH=N/2
  IREM=N-3*NRCH
  K=NRCH+1
  KK(1)=0
  KK(2)=NRCH
  IF (IREM.GE.1) KK(2)=KK(2)+1
  KK(2)=KK(2)+NRCH
  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,NRCH)
  IF (IREM.GT.0) WRITE (6,605) (IN(K+KK(J)),T(K+KK(J)),TEMP(K+KK(J)),S(K+KK(J)),
  & ,S(K+KK(J)),J=1,IREM)
C---READ IN TIME, RECOVERY DATA
  RECCVF=.FALSE.
  READ (4,*,END=999) TPLP,NREC
  RECCVF=.TRUE.
  T2LC=100000.
  T2HI=-T2LC
  S2LC=T2LC
  S2HI=T2HI
  READ (4,*) (T2(I),TEMP(I),I=1,NREC)
  DC 20 I=1,NREC
    S2(I)=TEMP(I)-STATIC
    IF (T2(I).LT.T2LC) T2LC=T2(I)
    IF (T2(I).GT.T2HI) T2HI=T2(I)
    IF (S2(I).LT.S2LC) S2LC=S2(I)
    IF (S2(I).GT.S2HI) S2HI=S2(I)

```

VWS02210
VWS02220
VWS02230
VWS02240
VWS02250
VWS02260
VWS02270
VWS02280
VWS02290
VWS02300
VWS02310
VWS02320
VWS02330
VWS02340
VWS02350
VWS02360
VWS02370
VWS02380
VWS02390
VWS02400
VWS02410
VWS02420
VWS02430
VWS02440
VWS02450
VWS02460
VWS02470
VWS02480
VWS02490
VWS02500
VWS02510
VWS02520
VWS02530
VWS02540
VWS02550
VWS02560
VWS02570
VWS02580
VWS02590
VWS02600
VWS02610
VWS02620
VWS02630
VWS02640
VWS02650
VWS02660
VWS02670
VWS02680
VWS02690
VWS02700
VWS02710
VWS02720
VWS02730
VWS02740
VWS02750

```

      LT2(I)=ALOG10(T2(I))
20  CONTINUE
      WRITE (6,603) NREC,TPUPP
      WRITE (6,604)
      NRCW=NREC/3
      IREM=NREC-NRCW*3
      K=NRCW+1
      KK(1)=0
      KK(2)=NRCW
      IF (IREM.GE.1) KK(2)=KK(2)+1
      KK(3)=KK(2)+NRCW
      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,NRCW)
      IF (IREM.GT.0) WRITE (6,605) (IN(K+KK(J)),T2(K+KK(J)),TEMP(K+KK(J)),
& J=1,IREM)
C
501  FCRMAT (25A1)
600  FCRMAT (/5X,25A1)
601  FCRMAT (
& /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  FCRMAT (/5X,'TIME-RECOVERY DATA',
& /2X,' # OF DATA PAIRS = ',I10,
& /2X,' TIME PUMPING STOPPED = ',F10.2,' MIN')
604  FORMAT (/3(3A,' ',3X,' TIME (MIN) ',2X,' DTW (FEET) ',2X,' DRAWDOWN ',4X,
& ))
605  FCRMAT (3(1X,13,F10.2,2X,F9.2,2X,F9.2,7X))
606  FORMAT (/50X,A3,15A1)
C
999  RETURN
      END
      SUBROUTINE THEIS(RECOVF)
C
C
C      THEIS ANALYSIS
C
C
C-----
C
      REAL T(200),S(200),LT(200),LS(200),ERM(200),LTA(200),LSX(200)
      REAL T2(200),S2(200),LT2(200)
      REAL*8 L,WU,LINV
      LOGICAL*1 TITLE(25),RECCVR,RESTF,IEND
      LOGICAL*1 TX(30)/'T','I','M','E',' ',' ','M','I','N',' ','U','T','E',
& 'S',' ',10* '/'
      LOGICAL*1 TY(30)/'D','R','A','W','D','O','W','N',' ',' ','F','E',
& 'E','T',' ',15* '/'
      LOGICAL*1 TA(30)/'T','I','E','I','S',' ','A','N','A','L','Y','S',
& 'I','S',16* '/'
      REAL YN, YES/'YES' //, NO/'NO' //

```

CCMPCN /TITLES/ TITLE	VW503310
CCMPCN /DATA/ N,R,C,TPLMP,NREC,STCR,TRAN,TRAN1,TRAN2,A,B,TC	VW503320
CCMPCN /TCATA/ ERP,NPTS,NSUM	VW503330
CCMPCN /FCINTS/ T,S,LT,LS,T2,S2,LT2	VW503340
CCMPCN /RANGE/ TLC,THI,SLC,SHI,T2LC,T2HI,S2LC,S2HI	VW503350
CCMPCN /XTRA/ LTX,LSX	VW503360
CCMPCN /ACD/ XACC,YACD,DX,DY,AXORG,AYGRG	VW503370
C	VW503380
C-----	VW503390
C	VW503400
C	VW503410
C	VW503420
RESTF=.FALSE.	VW503430
C---SET UP MAX AND MIN	VW503440
XLO=ALOG10(TLC)	VW503450
XHI=ALOG10(THI)	VW503460
YLO=ALOG10(SLG)	VW503470
YHI=ALOG10(SHI)	VW503480
TMPKC=TLC	VW503490
TMPKI=THI	VW503500
GC TC 2C	VW503510
C	VW503520
C	VW503530
C	VW503540
C---WHAT ARE THE LOW AND HIGH CUTOFF TIMES FOR THIS RUN?	VW503550
10 WRITE (5,505)	VW503560
READ (5,*) TMPKC	VW503570
WRITE (5,506)	VW503580
READ (5,*) TMPKI	VW503590
20 NPTS=C	VW503600
XLOPKK=ALOG10(TMPKC)	VW503610
XHIPIK=ALOG10(TMPKI)	VW503620
C---SET MINIMUM NUMBER OF SUMMATIONS	VW503630
NSUM=50	VW503640
C---SET UP THE ANALYSIS ARRAYS	VW503650
DO 30 I=1,N	VW503660
IF (T(I).LT.TMPKC) GC TC 30	VW503670
IF (T(I).GT.TMPKI) GC TC 30	VW503680
NPTS=NPTS+1	VW503690
LTX(NPTS)=LT(I)	VW503700
LSX(NPTS)=LS(I)	VW503710
30 CONTINUE	VW503720
C---CALL THE SEARCH SUBROUTINES	VW503730
CALL SEARCH(IEND)	VW503740
IF (IEND) GO TO 8C	VW503750
C---CALCULATE TRANSMISSIVITY AND STORAGE	VW503760
PI=3.1415926	VW503770
UINV=10.**(-LTX(I)*XACC)	VW503780
U=1./UINV	VW503790
WC=LSX(I)+YACD	VW503800
WC=10.**(-WC)	VW503810
SX=10.**(-LSX(I))	VW503820
TRAN=C*WC/(4.*PI*SX*7.48)	VW503830
TRAN1=1440.*TRAN	VW503840
TRAN2=10771.*TRAN	VW503850

```

    TX1=10.** (LTX(1))
    STOR=4.*TRAN*TX1*C/(R*R)
C---PRINT OUT THE ANSWER
    WRITE (6,601)
    CALL FNTCLT(TMKLC,TKMHI,RESTF)
C---PLOT THIS ANALYSIS?
    WRITE (5,502)
    READ (5,510) YN
    IF (YN.EQ.NO) GO TO 60
C---PREPARE PLOTTER
    WRITE (5,503)
    READ (5,510,END=40) TRASH
    40 REWIND 5
C---CALL DATA PLOTTING ROUTINES
    CALL LLPLCT(XLO,XFI,YLC,YHI,XORIG,XMAX,XSCALE,YORIG,YMAX,YSCALE,
    & TITLE,IX,TY)
    CALL THSPLT(LT,LS,N,XORIG,XMAX,XSCALE,YORIG,YMAX,YSCALE,XLCPRK,
    & XHMRK)
    CALL RESLT(TMKLC,TKMHI,TA,RESTF)
C---VIEW RESULTS
    READ (5,510,END=50) TRASH
    50 REWIND 5
    CALL PLTERA
    60 IF (.NOT.RECCVR) GO TO 70
C---DO A CALCULATED RECOVERY ANALYSIS?
    WRITE (5,504)
    READ (5,510) YN
    IF (YN.EQ.YES) CALL THSREC
C---DO ANOTHER THEIS ANALYSIS?
    70 WRITE (5,507)
    READ (5,510) YN
    IF (YN.EQ.YES) GO TO 10
    GO TO 999
C---THEIS CURVE MATCHING ATTEMPT TOOK TOO LONG
    80 WRITE (5,506)
C
C      FORMAT STATEMENTS
C
    502 FORMAT (//2X,'PLOT THIS ANALYSIS? (YES/NO)')
    503 FORMAT (//2X,'PREPARE THE PLOTTER, TYPE "M", AND RETURN')
    504 FORMAT (//2X,'PERFORM A CALCULATED RECOVERY ANALYSIS ON THE RECOVERY
    &Y DATA',//5X,'BASED ON THE THEIS ANALYSIS RESULTS? (YES/NO)')
    505 FORMAT (//2X,'WHAT IS THE LOW TIME CUTOFF?')
    506 FORMAT (//2X,'WHAT IS THE HIGH TIME CUTOFF?')
    507 FORMAT (//2X,'DO ANOTHER THEIS ANALYSIS?')
    508 FORMAT (//2X,'THEIS MATCH TOOK TOO LONG. ATTEMPT STOPS.')
    510 FORMAT (A4)
    601 FORMAT (///1X,'THEIS ANALYSIS')
C
    999 RETURN
    END
    SLBRCLTINE THSREC
C
C      CALCULATED RECOVERY ANALYSIS BASED ON THE THEIS ANALYSIS
C

```

VNS03860
VNS03870
VNS03880
VNS03890
VNS03900
VNS03910
VNS03920
VNS03930
VNS03940
VNS03950
VNS03960
VNS03970
VNS03980
VNS03990
VNS04000
VNS04010
VNS04020
VNS04030
VNS04040
VNS04050
VNS04060
VNS04070
VNS04080
VNS04090
VNS04100
VNS04110
VNS04120
VNS04130
VNS04140
VNS04150
VNS04160
VNS04170
VNS04180
VNS04190
VNS04200
VNS04210
VNS04220
VNS04230
VNS04240
VNS04250
VNS04260
VNS04270
VNS04280
VNS04290
VNS04300
VNS04310
VNS04320
VNS04330
VNS04340
VNS04350
VNS04360
VNS04370
VNS04380
VNS04390
VNS04400

```

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 VECTOR SCALC(J).
C
C-----
C
REAL T(200),S(200),LT(200),T2(200),S2(200),LT2(200),SCALC(200)
REAL LS(200)
REAL*E U,WU
LOGICAL*1 TX(30)/'R','E','C','U','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',' ','S',
& '3* '/'
LOGICAL*1 TITLE(25),LINE,KESTF
REAL YA,YES/'YES','/','NO/'NC' '/'
COMMON /RANGE/ TLC,THI,SLC,SHI,T2LC,T2HI,S2LC,S2HI
COMMON /PCINTS/ T,S,LT,LS,T2,S2,LT2
COMMON /DATA/ N,R,G,TPUMP,NREC,STCR,TRAN,TRAN1,TRAN2,A,B,TC
COMMON /TITLES/ TITLE
C
C-----
C
C
C
C INITIALIZE PARAMETERS AND SET UP FOR FIRST RUN
C
C-----CALCULATE THE PREDICTED DRAWDOWN AND THE RECOVERY
LINE=.TRUE.
KESTF=.FALSE.
SCLC=1000000.
SCHI=0.01
PI=3.1415926
F1=C/(4*PI*TRAN*7.48)
F2=R*R*STCR/(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.SCLC) SCLC=SNEW
  IF (SNEW.GT.SCHI) SCHI=SNEW
10 CONTINUE
WRITE (6,602)
CALL TABLE(T2,SCALC,NREC)
XLC=ALOG10(T2LC)
XHI=ALOG10(T2HI)
TMKLC=T2LC
TMKHI=T2HI
GO TO 30
C

```



```

C          SET PARAMETERS FOR SUCCEEDING RUNS                                VAS04960
C                                                                                   VAS04970
C---WHAT ARE THE LOW AND HIGH CUTOFF TIMES?                                   VAS04980
  20 WRITE (5,501)                                                                VAS04990
     READ (5,*) TPKLG                                                             VAS05000
     WRITE (5,502)                                                                VAS05010
     READ (5,*) TPKHI                                                             VAS05020
  30 XLGMRK=ALCG10(TPKLG)                                                         VAS05030
     XHIMRK=ALCG10(TPKHI)                                                         VAS05040
C---COMPUTE THE RESULTS AND PRINT THEM OUT                                   VAS05050
  CALL TCHECK(LT2,SCALC,AREC,XLGMRK,XHIMRK)                                       VAS05060
  WRITE (6,601)                                                                    VAS05070
  CALL FNICLT(TPKLG,TPKHI,RESTF)                                                 VAS05080
C---PLOT THIS ANSWER?                                                       VAS05090
  WRITE (5,503)                                                                    VAS05100
  READ (5,510) YN                                                                  VAS05110
  IF (YN.EQ.NO) GO TO 60                                                         VAS05120
C---PREPARE PLOTTER                                                         VAS05130
  WRITE (5,504)                                                                    VAS05140
  READ (5,510,END=40) TRASH                                                       VAS05150
  40 REWIND 5                                                                      VAS05160
C---CALL DATA PLOTTING SUBROUTINES                                         VAS05170
  CALL SLPLOT(XLO,XHI,SCLO,SCHI,XDRIG,XMAX,XSCALE,YCRIG,YMAX,                    VAS05180
  6 YSCALE,TITLE,TA,TY)                                                         VAS05190
  CALL PLTLIN(LT2,SCALC,AREC,XLGMRK,XHIMRK,XDRIG,XMAX,XSCALE,YCRIG,           VAS05200
  6 YMAX,YSCALE,A,B,LINE)                                                       VAS05210
  CALL RESULT(TPKLG,TPKHI,TA,RESTF)                                             VAS05220
C---VIEW RESULTS                                                            VAS05230
  READ (5,510,END=50) TRASH                                                      VAS05240
  50 REWIND 5                                                                      VAS05250
  CALL PLTERA                                                                      VAS05260
C---DO ANOTHER CALCULATED RECOVERY ANALYSIS?                               VAS05270
  60 WRITE (5,505)                                                                VAS05280
  READ (5,510) YN                                                                  VAS05290
  IF (YN.EQ.YES) GO TO 20                                                         VAS05300
C                                                                                   VAS05310
C          FORMAT STATEMENTS                                                 VAS05320
C                                                                                   VAS05330
C  501 FORMAT (//2X,'WHAT IS THE LOW TIME CUTOFF?')                             VAS05340
C  502 FORMAT (//2X,'WHAT IS THE HIGH TIME CUTOFF?')                           VAS05350
C  503 FORMAT (//2X,'PLOT THIS ANALYSIS? (YES/NO)')                             VAS05360
C  504 FORMAT (//2X,'PREPARE PLOTTER, TYPE "C", AND RETURN?')                   VAS05370
C  505 FORMAT (//2X,'DO ANOTHER CALCULATED RECOVERY ANALYSIS? (YES/NO)')       VAS05380
C  510 FORMAT (A4)                                                                VAS05390
C  601 FORMAT (///1X,'CALCULATED RECOVERY ANALYSIS BASED ON PREVIOUS THEI        VAS05400
C     65 ANALYSIS')                                                              VAS05410
C  602 FORMAT (//5X,'TIME-CALCULATED RECOVERY VALUES BASED ON THE ABOVE AN     VAS05420
C     DALYSIS',//3(2X,'NC',7X,'T',9X,'RECOV',7X))                             VAS05430
C                                                                                   VAS05440
C  999 RETURN                                                                    VAS05450
C     END                                                                        VAS05460
C     SLBRoutine JAKE(RECOVR)                                                    VAS05470
C                                                                                   VAS05480
C                                                                                   VAS05490
C          JACOB ANALYSIS                                                       VAS05500

```

```

C
C      DC ALL WORK NECESSARY TO PERFORM A JACCB 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)
C      LOGICAL*1 TITLE(25),RESTF,RECOVR,LINE
C      LOGICAL*1 TX(30)/'T','I','M','E',' ','I','M','I','N','U','T','E',
C      & 'S',',',16*' //
C      LOGICAL*1 TY(30)/'D','R','A','M','D','O','N',' ','I','F','E',
C      & 'E','T',',',15*' //
C      LOGICAL*1 TA(30)/'J','A','C','B',' ','A','N','A','L','Y','S',
C      & 'I','S',',16*' //
C      REAL YN,YES/'YES '//,NC/'NC' //
C      COMMON /TITLES/ TITLE
C      COMMON /DATA/ N,R,L,TPCMP,NREC,STCR,TRAN,TRAN1,TRAN2,A,B,T0
C      COMMON /FCINIS/ T,S,LT,LS,T2,ST,LT2
C      COMMON /RANGE/ TLC,THI,SLC,SHI,T2LC,T2HI,S2LO,S2HI
C-----
C
C      INITIALIZE PARAMETERS FOR FIRST JACCB ANALYSIS
C
C      LINE=.TRUE.
C      RESTF=.FALSE.
C      XLG=ALGGIC(TLC)
C      XHI=ALGGIC(THI)
C      TPKLC=TLC
C      TPKHI=THI
C      GO TO 20
C
C      SET PARAMETERS FOR SUCCEEDING JACCB ANALYSES
C
C----ANOTHER JACCB ANALYSIS?
C      10 WRITE (5,501)
C      READ (5,510) YN
C      IF (YN.NE.YES) GO TO 999
C----WHAT ARE LOW AND HI LIMITING TIMES FOR THIS ANALYSIS?
C      WRITE (5,502)
C      READ (5,*) TPKLC
C      WRITE (5,503)
C      READ (5,*) TPKHI
C      20 XLCMRK=ALGGIC(TPKLC)
C      XHIMRK=ALGGIC(TPKHI)
C----COMPLETE THE RESULTS AND PRINT THEM OUT
C      CALL TCHECKILT,S,N,XLCMRK,XHIMRK)
C      WRITE (6,601)
C      CALL FNTCLT(TPKLC,TPKHI,RESTF)
C----PLOT THIS ANALYSIS?
C      WRITE (5,504)
C      READ (5,510) YN
C      IF (YN.EQ.NO) GO TO 50
C----PREPARE PLOTTER

```

VWS05510
VWS05520
VWS05530
VWS05540
VWS05550
VWS05560
VWS05570
VWS05580
VWS05590
VWS05600
VWS05610
VWS05620
VWS05630
VWS05640
VWS05650
VWS05660
VWS05670
VWS05680
VWS05690
VWS05700
VWS05710
VWS05720
VWS05730
VWS05740
VWS05750
VWS05760
VWS05770
VWS05780
VWS05790
VWS05800
VWS05810
VWS05820
VWS05830
VWS05840
VWS05850
VWS05860
VWS05870
VWS05880
VWS05890
VWS05900
VWS05910
VWS05920
VWS05930
VWS05940
VWS05950
VWS05960
VWS05970
VWS05980
VWS05990
VWS06000
VWS06010
VWS06020
VWS06030
VWS06040
VWS06050

```

WRITE (5,505) VWS0606C
READ (5,510,END=3C) TRASH VWS0607C
30 REWIND 5 VWS0608C
C---CALL DATA PLOTTING SUBROUTINES VWS0609C
CALL SLPLCT(XLO,XHI,SLC,SFI,XORIG,XMAX,XSCALE,YORIG,YMAX,YSCALE, VWS0610C
& TITLE,TX,TY) VWS0611C
CALL FLTLIN(LT,S,N,XLCMRK,XHIMRK,XCRIG,XMAX,XSCALE,YCRIG,YMAX, VWS0612C
& YSCALE,A,B,LINE) VWS0613C
CALL RESLT(TMKLG,TPKHI,TA,RESTF) VWS0614C
C---VIEW RESULTS VWS0615C
READ (5,510,END=4C) TRASH VWS0616C
40 REWIND 5 VWS0617C
CALL FLTERA VWS0618C
C---IF THERE IS RECOVERY DATA, DO A CALCULATED RECOVERY ANALYSIS? VWS0619C
50 IF (.NOT.RECCVR) GO TO 10 VWS0620C
WRITE (5,506) VWS0621C
READ (5,510) YN VWS0622C
IF (YN.EQ.YES) CALL CALREC VWS0623C
GO TO 1C VWS0624C
C VWS0625C
C FORMAT STATEMENTS VWS0626C
C VWS0627C
501 FORMAT (/2X,'ANOTHER JACOB ANALYSIS? (YES/NO)') VWS0628C
502 FORMAT (/2X,'WHAT IS THE LOW TIME? ') VWS0629C
503 FORMAT (/2X,'WHAT IS THE HIGH TIME?') VWS0630C
504 FORMAT (/2X,'PLOT THIS ANALYSIS? (YES/NO)') VWS0631C
505 FORMAT (/2X,'PREPARE PLOTTER, TYPE "JM", AND RETURN') VWS0632C
506 FORMAT (/2X,'PERFORM A CALCULATED RECOVERY ANALYSIS ON THE RECOVERY VWS0633C
& DATA',/5X,'BASED ON THE JACOB ANALYSIS RESULTS? (YES/NO)') VWS0634C
510 FORMAT (A4) VWS0635C
601 FORMAT (///1X,'JACOB ANALYSIS') VWS0636C
C VWS0637C
999 RETURN VWS0638C
END VWS0639C
SUBROUTINE CALREC VWS0640C
C VWS0641C
C VWS0642C
C THIS SUBPROGRAM PERFORMS A CALCULATED RECOVERY ANALYSIS. VWS0643C
C IT TAKES THE LINEAR VALUES OF A AND B (AS CALCULATED IN SUBROUTINE VWS0644C
C JAKE) AND EXTENDS THE LINE. TIME-DRAWDOWN VALUES AT THE RECOVERY VWS0645C
C TIMES ARE CALCULATED AND THE DIFFERENCE FROM THE OBSERVED RECOVERY VWS0646C
C DATA CALCULATED. THE CALCULATED RECOVERY VALUES ARE STORED IN VWS0647C
C THE VECTOR SCALC(J). VWS0648C
C VWS0649C
C VWS0650C
C----- VWS0651C
C VWS0652C
REAL T(200),S(200),LT(200),T2(200),S2(200),LT2(200),SCALC(200) VWS0653C
REAL LS(200) VWS0654C
LOGICAL*1 TX(30)/'R','E','C','O','V','E','R','Y',' ','T','I','M', VWS0655C
& 'E',' ',' ','M','I','N','U','T','E','S',' ',7*' / VWS0656C
LOGICAL*1 TY(30)/'C','A','L','C','U','L','A','T','E','D', VWS0657C
& 'E','C','O','V','E','R','Y',' ','F','E','E','T',' ',4*' / VWS0658C
LOGICAL*1 TA(30)/'C','A','L','C','U','L','A','T','E','D', VWS0659C
& 'E','C','O','V','E','R','Y',' ','-',' ','J','A','C','O','B', VWS0660C

```

```

6 3* */
LOGICAL*1 TITLE(25),LINE,RESTF
REAL YN,YES/'YES'/,NC/'NC' /
COMMON /RANGE/ T1C,T1H,SLC,SHI,T2LO,T2HI,S2LC,S2HI
COMMON /PCINTS/ T,S,LT,LS,T2,S2,LT2
COMMON /DATA/ N,R,W,TPLMP,NREC,STCR,TRAN,TRAN1,TRAN2,A,B,TO
COMMON /TITLES/ TITLE
VMS0661G
VMS0662C
VMS0663C
VMS0664C
VMS0665C
VMS0666E
VMS0667C
VMS0668E
VMS0669E
C-----
C
C      INITIALIZE PARAMETES AND SET UP FOR FIRST RLn THROUGH
VMS0670C
VMS0671C
VMS0672D
VMS0673D
VMS0674C
VMS0675C
VMS0676C
VMS0677C
VMS0678E
VMS0679C
VMS0680C
VMS0681C
VMS0682C
VMS0683C
VMS0684C
VMS0685D
VMS0686C
VMS0687C
VMS0688E
VMS0689C
VMS0690C
VMS0691D
VMS0692C
VMS0693D
VMS0694C
VMS0695C
VMS0696C
VMS0697C
VMS0698C
VMS0699C
VMS0700C
VMS0701D
VMS0702C
VMS0703C
VMS0704C
VMS0705C
VMS0706D
VMS0707D
VMS0708C
VMS0709C
VMS0710C
VMS0711C
VMS0712C
VMS0713C
VMS0714C
VMS0715C
C
C      SET PARAMETERS FOR SUCCEEDING RUNS
C
C---WHAT ARE THE LCN AND HI CUTOFF TIMES FOR THIS RUN?
20 WRITE (5,501)
   READ (5,*) TMKLC
   WRITE (5,502)
   READ (5,*) TMKHI
30 XLDMRK=ALOG10(TMKLC)
   XHIMRK=ALOG10(TMKHI)
C---COMPUTE THE RESULTS AND PRINT THEM OUT
   CALL TCHECK(LT2,SCALC,NREC,XLDMRK,XHIMRK)
   WRITE (6,601)
   CALL PNTCLT(TMKLC,TKKHI,RESTF)
C---PLOT THIS ANALYSIS?
   WRITE (5,503)
   READ (5,510) YN
   IF (YN.EC.ND) GO TO 60
C---PREPARE THE PLOTTER
   WRITE (5,504)
   READ (5,510,END=4C) TRASH
40 REWIND 5
C---CALL THE DATA PLOTTING SUBROUTINES
   CALL SLPLCT(ALU,XHI,SCLC,SCHI,XORIG,XMAX,XSCALE,YORIG,YMAX,

```

```

      & YSCALE, TITLE, TA, TY)                                VWS07160
      CALL PLTLINILT2, SCALC, NREC, XLOMRK, XHIMRK, XORIG, XMAX, XSCALE, YORIG, VWS07170
      & YMAX, YSCALE, A, B, LINE)                             VWS07180
      CALL RESULT(TMKLG, TMKHI, TA, RESTF)                   VWS07190
C---VIEW THE RESULTS                                        VWS07200
      READ (5, 510, END=50) TRASH                           VWS07210
      50 REWIND 5                                            VWS07220
      CALL PLTERA                                           VWS07230
C---DO ANOTHER CALCULATED RECOVERY ANALYSIS?              VWS07240
      60 WRITE (5, 505)                                       VWS07250
      READ (5, 510) YN                                       VWS07260
      IF (YN.EQ.YES) GO TO 20                                VWS07270
C                                                           VWS07280
C      FORMAT STATEMENTS                                     VWS07290
C                                                           VWS07300
      501 FORMAT (1/2X, 'WHAT IS THE LOW TIME CUTOFF?')     VWS07310
      502 FORMAT (1/2X, 'WHAT IS THE HIGH TIME CUTOFF?')   VWS07320
      503 FORMAT (1/2X, 'PLOT THIS ANALYSIS? (YES/NO)')     VWS07330
      504 FORMAT (1/2X, 'PREPARE PLOTTER, TYPE "C", AND RETURN') VWS07340
      505 FORMAT (1/2X, 'DO ANOTHER CALCULATED RECOVERY ANALYSIS? (YES/NO)') VWS07350
      510 FORMAT (A4)                                         VWS07360
      601 FORMAT (///1X, 'CALCULATED RECOVERY ANALYSIS BASED ON PREVIOUS JACCVWS07370
      && ANALYSIS')                                           VWS07380
      602 FORMAT (1/5X, 'TIME-CALCULATED RECOVERY VALUES BASED ON ABOVE ANALYSVWS07390
      &IS', //3(2X, 'NO', 7X, 'T', 9X, 'RECOV', 7X))       VWS07400
C                                                           VWS07410
      999 RETURN                                             VWS07420
      END                                                    VWS07430
      SUBROUTINE RESIC                                       VWS07440
C                                                           VWS07450
C                                                           VWS07460
C      RESIDUAL DRAWDOWN ANALYSIS                           VWS07470
C                                                           VWS07480
C                                                           VWS07490
C      THE VECTOR TCALC(IJ) CONTAINS THE T/T' VALUES.     VWS07500
C      TCLU AND TCHI CONTAIN THE EXTREME VALUES OF TCALC(IJ) VWS07510
C                                                           VWS07520
C                                                           VWS07530
C                                                           VWS07540
C-----                                                    VWS07550
C                                                           VWS07560
      REAL T(200), S(200), LT(200), T2(200), S2(200), LT2(200), TCALC(200) VWS07570
      REAL LS(200), TLC(200)                                  VWS07580
      LOGICAL*1 TX(30)/'R', 'A', 'T', 'I', 'D', 'U', 'A', 'L', 'D', 'R', 'A', VWS07590
      & 'R', 'I', 'M', 'E', 14*' '/'                               VWS07600
      LOGICAL*1 TY(30)/'R', 'E', 'S', 'I', 'D', 'U', 'A', 'L', 'D', 'R', 'A', VWS07610
      & 'D', 'O', 'N', 'E', 'F', 'E', 'T', 6*' '/'                 VWS07620
      LOGICAL*1 TA(30)/'R', 'E', 'S', 'I', 'D', 'U', 'A', 'L', 'D', 'R', 'A', VWS07630
      & 'D', 'O', 'N', 'E', 13*' '/'                               VWS07640
      LOGICAL*1 TITLE(25), LINE, RESTF                       VWS07650
      REAL YN, YES/'YES', 'NO/'NO'                          VWS07660
      COMMON /RANGE/ TLC, THI, SLD, SHI, T2LC, T2HI, S2LC, S2HI VWS07670
      COMMON /PCINTS/ T, S, LT, LS, T2, S2, LT2             VWS07680
      COMMON /DATA/ N, R, C, TPLMP, NREC, STCR, TRAN, TRAN1, TRAN2, A, B, TO VWS07690
      COMMON /TITLES/ TITLE                                  VWS07700

```

```

C-----
C
C      SET UP PARAMETERS FOR FIRST RUN
C
      LINE=.TRUE.
      RESTF=.TRUE.
C---CALCULATE THE T/T' VALUES AND STORE IN T/CALC(J)
      T/CALC=1000000.
      TCHI=-T/CALC
      DO 10 I=1,NREC
          TCC(I)=(T2(I)+TPUMP)/T2(I)
          T/CALC(I)=ALOG10(TCC(I))
          IF (T/CALC(I).LT.T/CALC) T/CLO=T/CALC(I)
          IF (T/CALC(I).GT.TCHI) TCHI=T/CALC(I)
      10 CONTINUE
      WRITE (6,602)
      CALL TABLE(TCC,S2,NREC)
      XLC=T/CALC
      XHI=TCHI
      XLUMRK=T/CALC
      XHIMRK=TCHI
      T/MKLG=10.**(T/CALC)
      T/MKHI=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,*) T/MKLG
          WRITE (5,502)
          READ (5,*) T/MKHI
          XLUMRK=ALOG10(T/MKLG)
          XHIMRK=ALOG10(T/MKHI)
C---COMPLETE THE RESULTS AND PRINT THEM OUT
      30 CALL T/CHECK(T/CALC,S2,NREC,XLUMRK,XHIMRK)
          WRITE (6,601)
          CALL PRINT(T/MKLG,T/MKHI,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 SLPLCT(XLO,XHI,S2LC,S2HI,XORIG,XMAX,XSCALE,YORIG,YMAX,
          & YSCALE,TITLE,TA,TY)
          CALL PLTLINITCALC,S2,NREC,XLUMRK,XHIMRK,XORIG,XMAX,XSCALE,YORIG,
          & YMAX,YSCALE,A,B,LINE)
          CALL RESLLT(T/MKLG,T/MKHI,TA,RESTF)
C---VIEW RESULTS
          READ (5,510,END=50) TRASH

```

VNS07710
VNS07720
VNS07730
VNS07740
VNS07750
VNS07760
VNS07770
VNS07780
VNS07790
VNS07800
VNS07810
VNS07820
VNS07830
VNS07840
VNS07850
VNS07860
VNS07870
VNS07880
VNS07890
VNS07900
VNS07910
VNS07920
VNS07930
VNS07940
VNS07950
VNS07960
VNS07970
VNS07980
VNS07990
VNS08000
VNS08010
VNS08020
VNS08030
VNS08040
VNS08050
VNS08060
VNS08070
VNS08080
VNS08090
VNS08100
VNS08110
VNS08120
VNS08130
VNS08140
VNS08150
VNS08160
VNS08170
VNS08180
VNS08190
VNS08200
VNS08210
VNS08220
VNS08230
VNS08240
VNS08250

```

50 REWIND 5                                VMS08260
   CALL FLTERA                              VMS08270
C---DO ANOTHER RESIDUAL DRAWDOWN ANALYSIS? VMS08280
   GO WRITE (5,505)                         VMS08290
   READ (5,510) YN                          VMS08300
   IF (YN.EQ.YES) GO TO 2C                  VMS08310
C                                           VMS08320
C   FORMAT STATEMENTS                       VMS08330
C                                           VMS08340
501 FORMAT (/2X,'WHAT IS T/T'' LOW?')      VMS08350
502 FORMAT (/2X,'WHAT IS T/T'' HI?')      VMS08360
503 FORMAT (/2X,'PLCT THIS ANALYSIS? (YES/NO)') VMS08370
504 FORMAT (/2X,'PREPARE PLOTTER, TYPE "R", AND RETURN') VMS08380
505 FORMAT (/2X,'DO ANOTHER RESIDUAL DRAWDOWN ANALYSIS? (YES/NO)') VMS08390
510 FORMAT (A4)                             VMS08400
601 FORMAT (///1X,'RESIDUAL DRAWDOWN ANALYSIS') VMS08410
602 FORMAT (/5X,'T/T''-RESIDUAL DRAWDOWN VALUES FOR THE FOLLOWING ANALYSIS',/31' # T/T'' RESID ',3X)) VMS08420
C                                           VMS08430
C                                           VMS08440
999 RETURN                                  VMS08450
   END                                       VMS08460
   SUBROUTINE TCHECK(LT,S,NUM,TLCMRK,THIMRK) VMS08470
C                                           VMS08480
C   CALCULATION ROUTINE                     VMS08490
C                                           VMS08500
C   DATA INPUT:                            VMS08510
C                                           VMS08520
C   LT(I) - A VECTOR OF LENGTH NUM CONTAINING THE LOG TIME VALUES VMS08530
C   S(I) - A VECTOR OF LENGTH NUM CONTAINING THE DRAWDOWN VALUES VMS08540
C   NUM - NUMBER OF VALUES IN LT(I) AND S(I) (NUM<=100) VMS08550
C   TLCMRK - THE LOW CUTOFF VALUE FOR THE LOG TIME VALUES VMS08560
C   THIMRK - THE HIGH CUTOFF VALUE FOR THE LOG TIME VALUES VMS08570
C                                           VMS08580
C   RESULTS:                                VMS08590
C   TRAN - TRANSMISSIVITY (SQ FEET/MIN)     VMS08600
C   TRAN1 - TRANSMISSIVITY (SQ FEET/DAY)    VMS08610
C   TRAN2 - TRANSMISSIVITY (GALLONS/DAY/FOOT) VMS08620
C   STCR - STORAGE COEFFICIENT (DIMENSIONLESS) VMS08630
C   A,B - BEST FIT ON A LINEAR REGRESSION OF ALL LT VALUES VMS08640
C         BETWEEN TLCMRK AND THIMRK TO FIT THE EQUATION VMS08650
C         S(I) = A + B*LT(I)                VMS08660
C   TO - TIME AT WHICH LT EQUALS 0 ( TO = 10.**(-A/B) ) VMS08670
C                                           VMS08680
C   NOTES:                                  VMS08690
C   1. ALL RESULTS ARE STORED IN THE COMMON BLOCK /DATA/ VMS08700
C   2. THE VECTORS LT AND S MAY CONTAIN UP TO 100 VALUES VMS08710
C                                           VMS08720
-----VMS08730
C                                           VMS08740
   REAL S(200),LT(200),X(200),Y(200)      VMS08750
   COMMON /DATA/ N,R,C,TPLMP,NREC,STCR,TRAN,TRAN1,TRAN2,A,B,TO VMS08760
C                                           VMS08770
-----VMS08780
C                                           VMS08790
C---PULL OUT ALL VALUES BETWEEN THE LIMITS TLCMRK AND THIMRK VMS08800

```



```

COMMON /SPARE/ IPCINT, IPTR                                VMS09360
COMMON /DEVICE/ IXLIM, IYLIM, MAXXO, MAXYO, IXMIN, IYMIN  VMS09370
COMMON /PARS/ LEN, INDEX, NCHARS, NUMLFT, ARG, STRING    VMS09380
COMMON /FLT/ XU,X1,Y0,Y1                                  VMSU9350
C                                                         VMSU9400
C-----
C                                                         VMS09410
C                                                         VMS09420
C          FLCT CLT POINTS                                VMS09430
C                                                         VMS09440
C---PAUSE                                                 VMS09450
          REAC (5,510,END=5) TRASH                         VMS09460
          5 REWIND 5                                       VMS09470
C---SET PLOTTING CHARACTERS                              VMS09480
          DO 10 I=1,N                                       VMS09490
              MARK=PLLS                                     VMS09500
              IF ((X(I).LT.XLCMRK).OR.(X(I).GT.XHIMRK)) MARK=STAR VMS09510
              XLCC= XG + XSCALE*(X(I)-XORIG) - 0.075      VMS09520
              YLCC= YC + YSCALE*(Y(I)-YORIG) - 0.075      VMS09530
              CALL SYMBOL(XLCC,YLCC,0.15,MARK,0.,1)        VMS09540
              NPNTS = 0                                     VMS09550
          10 CONTINUE                                       VMS09560
C                                                         VMS09570
C          PLCT CLT LINE                                    VMS09580
C                                                         VMS09590
C---PLOT CLT LINE AT ALL?                                 VMS09600
          IF (.NOT.LINE) GO TO 999                          VMS09610
C---DETERMINE LEFT END POINT                              VMS09620
          XVAL=XORIG                                        VMS09630
          YVAL=A + B*XVAL                                    VMS09640
          IF (YVAL.GE.YORIG) GO TO 20                       VMS09650
          YVAL=YORIG                                        VMS09660
          XVAL=(YVAL-A)/B                                    VMS09670
          20 XLCC1=XO+XSCALE*(XVAL-XORIG)                   VMS09680
              YLCC1=Y0+YSCALE*(YVAL-YORIG)                 VMS09690
              CALL FLTPCV(XLCC1,YLCC1,3)                   VMS09700
C---DETERMINE RIGHT END POINT                             VMS09710
          XVAL=XMAX                                         VMS09720
          YVAL=A + B*XVAL                                    VMS09730
          IF (YVAL.LE.YMAX) GO TO 30                       VMS09740
          YVAL=YMAX                                         VMS09750
          XVAL=(YVAL-A)/B                                    VMS09760
          30 XLCC2=XO+XSCALE*(XVAL-XORIG)                   VMS09770
              YLCC2=Y0+YSCALE*(YVAL-YORIG)                 VMS09780
              CALL FLTPCV(XLCC2,YLCC2,2)                   VMS09790
C---TRACE OVER LINE A SECOND TIME FOR DARKNESS          VMS09800
          CALL FLTPCV(XLCC1,YLCC1,2)                       VMS09810
          CALL PLTPCV(XLCC1,YLCC1,-999)                    VMS09820
C---FORMATS                                              VMS09830
          510 FORMAT (A4)                                   VMS09840
C                                                         VMS09850
          999 RETURN                                       VMS09860
              END                                         VMS09870
              SUBPCLTINE SLPLCT(XLO,XHI,YLO,YHI,XORIG,XMAX,XSCALE,YORIG,YMAX,
              & YSCALE,IITLE,XTITLE,YTITLE)                VMS09880
          C                                                         VMS09890

```

```

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          DATA INPUT:
C          XLC      - LOG OF LOWEST X VALUE DATA POINTED
C          XHI      - LOG OF HIGHEST X VALUE DATA POINTED
C          YLC      - 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          RESULTS:
C          XCRIG    - 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          YCRIG    - LOWEST Y VALUE ON THE Y AXIS
C          YMAX     - HIGHEST Y VALUE ON THE Y AXIS
C          YSCALE   - Y SCALING FACTOR
C
C-----
C          LOGICAL*1 TITLE(25),XTITLE(30),YTITLE(30),CENT(10)
C          LOGICAL*1 XTIC(10)/10** '//,YTIC(9)/9** '//,BLANK/' '//,CHECK
C          LOGICAL*1 CENTX(30)/30** '//,CENTY(30)/30** '//,AMP/'&'//,BAR/'|'//
C          LOGICAL*1 CENTA(25)/25** '//
C          DIMENSION IX(2,6000), IPCINT(2,6000)
C          LOGICAL TEK, CAL
C          LOGICAL*1 LIFT(6000), STRING(130), ARG(130)
C          COMMON /SPOT/ TEK, CAL, NPNTS, IX, LIFT
C          COMMON /SPARE/ IPCINT, IPTK
C          COMMON /DEVICE/ IXLIM, IYLM, MAXXG, MAXYO, Ixmin, IYMIN
C          COMMON /PARKS/ LEN, INDEX, NCHARS, NMLFT, ARG, STRING
C          COMMON /FLT/ X0,X1,Y0,Y1
C-----
C
C-----CLEAR PLOTTER SCREEN
C          CALL PLTERA
C-----SET POSITIONS FOR TIC MARKS AND TITLES
C          XSYM=X0-1.30
C          YSYM=Y0-0.45
C          XDIFF=X1-X0

```

```

YCDIFF=Y1-Y0
XT=X0-0.2
YT=Y0-0.2
XT1=X1-C.2
YT1=Y1-C.2
XTNAME=YC-0.45
C---DRAW ECRCEF
DC 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 CCNTINUE
    CALL PLTMOV(X0,Y0,-999)
C
C    SET UP X AXIS
C
C---DETERMINE X AXIS PARAMETERS
IXLC=INT(XL0)
IF (XLC.LT.0.0) IXLC=I>LO-1
IXHI=INT(XHI)+1
IF (XHI.LT.0.0) IXHI=I>HI-1
XORIG=IXLC
XMAX=IXHI
ICDIFF=XHI-IXLC
IDIFF1=ICDIFF+1
XSCALE=XCDIFF/IDIFF1
NPNTS=0
C---PLOT OUT X TIC VALUES
DC 30 I=1,IDIFF1
    X=X0+(I-1)*XSCALE
    CALL PLTMOV(X,YT,3)
    CALL PLTMOV(X,YC,2)
    CALL PLTMOV(X,YT1,3)
    CALL PLTMOV(X,Y1,2)
    CALL PLTMOV(0.,C.,-999)
    XVAL=IC.**((IXLC+I-1)
    IDEC=2
    IF (XVAL.GE.0.055555) IDEC=1
    IF (XVAL.GE.1.0) IDEC=0
    CALL LNPAKF(XVAL,XTIC,BINARY,10,IDEC,CHECK)
    XNEW=X-C.9
    DC 20 J=1,IC
        CENT(J)=BLANK
    CALL CENTER(CENT,1,IC,XTIC,0,10)
    CALL SYMBOL(XNEW,YSYM,0.15,CENT,C.,10)
    NPNTS=C
30 CCNTINUE
C
C    LABEL Y AXIS
C
C---DETERMINE Y AXIS PARAMETERS
YCRIG=0.C
JFCWER=ALCG10(YHI)

```

VWS10460
VWS10470
VWS10480
VWS10490
VWS10500
VWS10510
VWS10520
VWS10530
VWS10540
VWS10550
VWS10560
VWS10570
VWS10580
VWS10590
VWS10600
VWS10610
VWS10620
VWS10630
VWS10640
VWS10650
VWS10660
VWS10670
VWS10680
VWS10690
VWS10700
VWS10710
VWS10720
VWS10730
VWS10740
VWS10750
VWS10760
VWS10770
VWS10780
VWS10790
VWS10800
VWS10810
VWS10820
VWS10830
VWS10840
VWS10850
VWS10860
VWS10870
VWS10880
VWS10890
VWS10900
VWS10910
VWS10920
VWS10930
VWS10940
VWS10950
VWS10960
VWS10970
VWS10980
VWS10990
VWS11000

```

IF (JPCWER.LT.0) JPCWER=JPCWER-1
TEST=YHI/(10.**JPCWER)
IFAC=(TEST/2.5)+1
YMAX=IFAC*2.5*(10.**JPCWER)
IF (TEST.EQ.1) YMAX=YHI
DIFF=YMAX-YORIG
YSCALE=YC/DIFF
DELTAY=DIFF/5.
NPNTS=0
C---PLOT CUT AND LABEL Y TIC MARKS
IDEC=1
IF (YMAX.LE.1.0) IDEC=2
DO 40 J=1,6
Y=(J-1)*DELTAY*YSCALE + YC
CALL PLTMUV(XT,Y,3)
CALL PLTMGV(AC,Y,2)
CALL PLTMUV(XT1,Y,3)
CALL PLTMOV(X1,Y,2)
CALL PLTMOV(U.,L.,-999)
YVAL=YCRIG+(J-1)*DELTAY
CALL UNPAKFIYVAL,YTIC,BINARY,7,IDEC,CHECK)
CALL SYMBOL(XSYM,Y,U.15,YTIC,U.,9)
NPNTS=C
40 CONTINUE
C
C PLCT TITLES
C
C---PAUSE
READ (5,510,END=50) TRASH
50 REWIND 5
C---PLOT MAIN TITLE
X=XC-1.0
Y=Y1+C.3
CALL CENTER(CENTTA,1,25,TITLE,U,25)
CALL SYMECL(X,Y,U.50,CENTTA,U.,25)
NPNTS=0
C---PLOT X-AXIS TITLE
X=XC+1.8
Y=YC-1.0
CALL CENTER(CENTX,1,30,XTITLE,U,30)
CALL SYMECL(X,Y,U.25,CENTX,U.,30)
NPNTS=0
C---PLOT Y-AXIS TITLE
X=XC-1.0
Y=YC+C.3
CALL CENTER(CENTY,1,30,YTITLE,U,30)
CALL SYMECL(X,Y,U.25,CENTY,U.,30)
NPNTS=0
C---FORMATS
510 FORMAT (A4)
C
999 RETURN
END
SUBROUTINE RESULT(IMKLC,IMKHI,TTOP,RESTF)
C

```

VWS11010
VWS11020
VWS11030
VWS11040
VWS11050
VWS11060
VWS11070
VWS11080
VWS11090
VWS11100
VWS11110
VWS11120
VWS11130
VWS11140
VWS11150
VWS11160
VWS11170
VWS11180
VWS11190
VWS11200
VWS11210
VWS11220
VWS11230
VWS11240
VWS11250
VWS11260
VWS11270
VWS11280
VWS11290
VWS11300
VWS11310
VWS11320
VWS11330
VWS11340
VWS11350
VWS11360
VWS11370
VWS11380
VWS11390
VWS11400
VWS11410
VWS11420
VWS11430
VWS11440
VWS11450
VWS11460
VWS11470
VWS11480
VWS11490
VWS11500
VWS11510
VWS11520
VWS11530
VWS11540
VWS11550

```

C                                     VMS1156C
C           DISPLAY THE RESULTS                                     VMS1157C
C                                     VMS1158C
C           INPUT DATA:                                         VMS1159C
C           TPKLG - THE LOW TIME CUTOFF VALUE                   VMS1160C
C           TPKHI - THE HIGH TIME CUTOFF VALUE                  VMS1161C
C           TTCP - TITLE FOR TOP OF RESULT SECTION              VMS1162C
C           RESTF - LOGICAL VARIABLE, =.TRUE. IF RESULTS COME FROM A
C                   RESIDUAL CRASHDOWN ANALYSIS (AND THUS NO STORAGE
C                   WAS CALCULATED), =.FALSE. OTHERWISE         VMS1163C
C                                     VMS1164C
C                                     VMS1165C
C                                     VMS1166C
C                                     VMS1167C
C-----VMS1168C
C                                     VMS1169C
C           LOGICAL*1 CHECK,RESTF                                VMS1170C
C           LOGICAL*1 TTCP(30),CN(2),RN(2),TN(2),SN(2),U1(3),U2(4),U3(6) VMS1171C
C           LOGICAL*1 N1(5),N2(5),N3(5),N4(7),N5(9),N6(9)       VMS1172C
C           LOGICAL*1 TA(4),TB(4),LT(3),ASSOC(10)              VMS1173C
C           DATA LN/'Q',I='/', RN/'F',I='/', TN/'T',I='/', SN/'S',I='/'
C           DATA L1/'G',I='P',I='M',/, L2/'F',I='E',I='E',I='T',/, U3/'G',I='P',I='D',I='/',I='F',
C           I='T',/
C           DATA N1/'S',I='/', N2/'S',I='/',N3/'S',I='/',N4/'7',I='/'
C           DATA N5/'S',I='/', N6/'S',I='/'
C           DATA TA/'T',I='L',I='C',I='/',TB/'T',I='H',I='I',I='/', LT/'M',I='I',I='N',/
C           DATA ASSOC/'C',I='E',I='P',I='',I='-',I='',I='N',I='J',I='G',I='S',/
C           COMMON /PLTPRM/ XLG,XHI,XCRIG,XMAX,XSCALE,YLC,YHI,YORIG,YMAX,
C           EYSCALE
C           DIMENSION IXY(2,6000), IPCINT(2,6000)
C           LOGICAL TEK, CAL
C           LOGICAL*1 LIFT(6000), STRING(130), ARG(130)
C           COMMON /SPUT/ TEK, CAL, NPPTS, IXY, LIFT
C           COMMON /SPAKE/ IPLINT, IPTR
C           COMMON /CEVICE/ IXLIM, IYLIM, MAXX0, MAXY0, IYMIN, IYMIN
C           COMMON /FARS/ LEN, INDEX, NCHARS, NUMLFT, ARG, STRING
C           COMMON /FLT/ XU,XI,YG,YI
C           COMMON /DATA/ N,R,Q,TPLMP,NREC,STCR,TRAN,TRAN1,TRAN2,A,B,TC
C                                     VMS1192C
C-----VMS1193C
C                                     VMS1194C
C                                     VMS1195C
C                                     VMS1196C
C           L---PAUSE
C           READ (5,501,END=5) TRASH
C           5 REWIND 5
C           VMS1197C
C           VMS1198C
C           VMS1199C
C           VMS1200C
C           VMS1201C
C           VMS1202C
C           VMS1203C
C           VMS1204C
C           VMS1205C
C           VMS1206C
C           VMS1207C
C           VMS1208C
C           VMS1209C
C           VMS1210C
C           VMS1211C
C           VMS1212C
C           VMS1213C
C           VMS1214C
C           VMS1215C
C           VMS1216C
C           VMS1217C
C           VMS1218C
C           VMS1219C
C           VMS1220C
C           VMS1221C
C           VMS1222C
C           VMS1223C
C           VMS1224C
C           VMS1225C
C           VMS1226C
C           VMS1227C
C           VMS1228C
C           VMS1229C
C           VMS1230C
C           VMS1231C
C           VMS1232C
C           VMS1233C
C           VMS1234C
C           VMS1235C
C           VMS1236C
C           VMS1237C
C           VMS1238C
C           VMS1239C
C           VMS1240C
C           VMS1241C
C           VMS1242C
C           VMS1243C
C           VMS1244C
C           VMS1245C
C           VMS1246C
C           VMS1247C
C           VMS1248C
C           VMS1249C
C           VMS1250C
C           VMS1251C
C           VMS1252C
C           VMS1253C
C           VMS1254C
C           VMS1255C
C           VMS1256C
C           VMS1257C
C           VMS1258C
C           VMS1259C
C           VMS1260C
C           VMS1261C
C           VMS1262C
C           VMS1263C
C           VMS1264C
C           VMS1265C
C           VMS1266C
C           VMS1267C
C           VMS1268C
C           VMS1269C
C           VMS1270C
C           VMS1271C
C           VMS1272C
C           VMS1273C
C           VMS1274C
C           VMS1275C
C           VMS1276C
C           VMS1277C
C           VMS1278C
C           VMS1279C
C           VMS1280C
C           VMS1281C
C           VMS1282C
C           VMS1283C
C           VMS1284C
C           VMS1285C
C           VMS1286C
C           VMS1287C
C           VMS1288C
C           VMS1289C
C           VMS1290C
C           VMS1291C
C           VMS1292C
C           VMS1293C
C           VMS1294C
C           VMS1295C
C           VMS1296C
C           VMS1297C
C           VMS1298C
C           VMS1299C
C           VMS1300C

```


C		VWS12660
C	INPUT DATA:	VWS12670
C	Y - VECTOR OF LENGTH N	VWS12680
C	X - VECTOR OF LENGTH N	VWS12690
C		VWS12700
C		VWS12710
C	OUTPLT	VWS12720
C	A - INTERCEPT	VWS12730
C	B - SLOPE	VWS12740
C		VWS12750
C	-----	VWS12760
C		VWS12770
C	REAL*4 X(200),Y(200)	VWS12780
C		VWS12790
C	-----	VWS12800
C		VWS12810
C		VWS12820
C	XAVG=C.	VWS12830
C	YAVG=C.	VWS12840
C	DO 10 I=1,N	VWS12850
C	XAVG=XAVG+X(I)	VWS12860
C	YAVG=YAVG+Y(I)	VWS12870
C	10 CONTINUE	VWS12880
C	XAVG=XAVG/N	VWS12890
C	YAVG=YAVG/N	VWS12900
C	TCP=C,C	VWS12910
C	BCT=C,C	VWS12920
C	DO 20 I=1,N	VWS12930
C	TCF=TCF+(X(I)-XAVG)*Y(I)	VWS12940
C	BCT=BCT+(X(I)-XAVG)**2	VWS12950
C	20 CONTINUE	VWS12960
C	B=TCP/BCT	VWS12970
C	A=YAVG-B*XAVG	VWS12980
C		VWS12990
C	999 RETRN	VWS13000
C	END	VWS13010
C	SUBROUTINE LLPLCT(XLO,XHI,YLO,YHI,XORIG,XMAX,XSCALE,YORIG,YMAX,	VWS13020
C	YSCALE,TITLE,XTITLE,YTITLE)	VWS13030
C		VWS13040
C		VWS13050
C	LOG-LOG PLOT	VWS13060
C	1. CALCULATE SCALING FACTORS FOR OTHER SUBROUTINES	VWS13070
C	2. PLOT OUT THE BORDER, TIC MARKS, AND TITLES	VWS13080
C	3. VALUES FOR XG, XI, YG, AND YI HAVE TO BE PASSED IN THROUGH	VWS13090
C	THE COMMON BLOCK /PLT/	VWS13100
C		VWS13110
C		VWS13120
C		VWS13130
C	DATA INPLT:	VWS13140
C	XLC - LOG OF LOWEST X VALUE TO BE PLOTTED	VWS13150
C	XHI - LOG OF HIGHEST X VALUE TO BE PLOTTED	VWS13160
C	YLC - LOG OF LOWEST Y VALUE TO BE PLOTTED	VWS13170
C	YHI - LOG OF HIGHEST Y VALUE TO BE PLOTTED	VWS13180
C	TITLE - MAIN TITLE FOR PLOT	VWS13190
C	XTITLE - X-AXIS TITLE	VWS13200
C	YTITLE - Y-AXIS TITLE	

```

C
C
C
C      RESULTS:
C      XCRIG - LOG CF NUMBER AT ORIGIN OF X AXIS
C      XMAX  - LOG CF NUMBER AT HIGH END OF X AXIS
C      XSCALE - X AXIS SCALING FACTOR
C      YCRIG - LOG CF NUMBER AT ORIGIN OF Y AXIS
C      YMAX  - LOG CF NUMBER AT HIGH END OF Y AXIS
C      YSCALE - Y AXIS SCALING FACTOR
C
C-----
C
C      LOGICAL*1 TITLE(25),XTITLE(30),YTITLE(30),CENT(10)
C      LOGICAL*1 XTIC(10)/10**' ',YTIC(9)/9**' ',BLANK/' ' ',CHECK
C      LOGICAL*1 CENTX(30)/30**' ',CENTY(30)/30**' ',AMP/'&' ',BAR/'|' /
C      LOGICAL*1 CENTA(25)/25**' ' /
C      DIMENSION IAY(2,6000), IPCINT(2,6000)
C      LOGICAL TEK, CAL
C      LOGICAL*1 LIFT(6000), STRING(130), ARG(130)
C      COMMON /SPOT/ TEK, CAL, NPNTS, IXY, LIFT
C      COMMON /SPARE/ IPCINT, IPTK
C      COMMON /DEVICE/ IXLIM, IYLIM, MAXXC, MAXYO, IXLIN, IYMIN
C      COMMON /PAKS/ LEN, INDEX, NCHARS, NUMLFT, ARG, STRING
C      COMMON /PLT/ XU,X1,Y0,Y1
C
C-----
C
C
C
C---CLEAR PLOTTER SCREEN
C      CALL PLTERA
C---SET PARAMETERS FOR TITLES AND TIC MARKS
C      XSYM=X0-1.30
C      YSYM=Y0-0.45
C      XDIFF=X1-X0
C      YDIFF=Y1-Y0
C      XT=X0-0.2
C      YT=Y0-0.2
C      XT1=X1-0.2
C      YT1=Y1-0.2
C      XTNAME=Y0-0.45
C---DRAW ECRDEF
C      DO 10 I=1,2
C          CALL PLTMOV(X0,Y0,3)
C          CALL PLTMOV(X1,Y0,2)
C          CALL PLTMOV(X1,Y1,2)
C          CALL PLTMOV(X0,Y1,2)
C          CALL PLTMOV(X0,Y0,2)
C      10 CONTINUE
C      CALL PLTMV(C.,C.,-999)
C
C      DRAW X-AXIS
C
C---DETERMINE SPACINGS ON X-AXIS
C      IXL=INT(XL0)
C      IF (XL0.LT.0.0) IXL=I>LC-1

```



```

IXHI=INT(IXHI)+1
IF (XHI.LT.0.C) 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,YC,2)
  CALL PLTMOV(X,YT1,3)
  CALL PLTMOV(X,Y1,2)
  CALL PLTMOV(O.,C.,-999)
  XVAL=IC.** (IXLO+I-1)
  ICEC=2
  IF (XVAL.GE.0.C99999) IDEC=1
  IF (XVAL.GT.1.C) IDEC=0
  CALL UNPAK(XVAL,XTIC,BINARY,10,IDEC,CHECK)
  XNEH=X-C.9
  DC 20 J=1,IC
    20  CENT(J)=BLANK
    CALL CENTER(CENT,1,IC,XTIC,0,10)
    CALL SYMBOL(XNEH,YSYM,C.15,CENT,C.,10)
    NPNTS=C
  30 CONTINUE
C
C LABEL Y-AXIS
C
C---DETERMINE SPACINGS ON Y AXIS
JYLC=INT(YLO)
IF (YLO.LT.0.C) JYLC=JYLC-1
JYHI=INT(YHI)+1
IF (YHI.LT.0.C) JYHI=JYHI-1
YCRIG=JYLC
YMAX=JYHI
JDIFF=JYHI-JYLC
JDIFF1=JDIFF+1
YSCALE=YDIFF/JDIFF
NPNTS=0
C---PLOT AND LABEL Y TIC MARKS
DC 40 J=1,JDIFF1
  Y=YC+(J-1)*YSCALE
  CALL PLTMOV(XT,Y,3)
  CALL PLTMOV(XC,Y,2)
  CALL PLTMOV(XT1,Y,3)
  CALL PLTMOV(X1,Y,2)
  CALL PLTMOV(O.,C.,-999)
  YVAL=IC.** (JYLC+J-1)
  ICEC=2
  IF (YVAL.GT.0.C999) IDEC=1
  IF (YVAL.GT.1.C) IDEC=0
  CALL UNPAK(YVAL,YTIC,BINARY,7,IDEC,CHECK)

```

VMS13760
VMS13770
VMS13780
VMS13790
VMS13800
VMS13810
VMS13820
VMS13830
VMS13840
VMS13850
VMS13860
VMS13870
VMS13880
VMS13890
VMS13900
VMS13910
VMS13920
VMS13930
VMS13940
VMS13950
VMS13960
VMS13970
VMS13980
VMS13990
VMS14000
VMS14010
VMS14020
VMS14030
VMS14040
VMS14050
VMS14060
VMS14070
VMS14080
VMS14090
VMS14100
VMS14110
VMS14120
VMS14130
VMS14140
VMS14150
VMS14160
VMS14170
VMS14180
VMS14190
VMS14200
VMS14210
VMS14220
VMS14230
VMS14240
VMS14250
VMS14260
VMS14270
VMS14280
VMS14290
VMS14300

```

      CALL SYMBOL(XSYM,Y,0.15,YTIC,0.,9)
      NPNTS=C
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+C.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+C.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 TFSPLT(X,Y,NUM,XORIG,XMAX,XSCALE,YCRIG,YMAX,YSCALE,
      & XLCMRK,XHIMRK)
C
C
C      THIS ANALYSIS CURVE PLOTTING
C
C
C      DATA INPLT:
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      YCRIG  - LOG OF NUMBER AT ORIGIN OF Y AXIS
C      YMAX   - LOG OF NUMBER AT END OF Y AXIS
C      YSCALE - Y SCALING FACTOR
C      XLCMRK - LOW TIME CUTOFF FOR ANALYSIS
C      XHIMRK - HIGH TIME CUTOFF FOR ANALYSIS
C
C      NOTES:
C      1. THE VALUES XC,X1,Y0, AND Y1 HAVE TO BE PASSED IN BY THE
C      COMMON BLOCK /PLT/

```

VWS14310
VWS14320
VWS14330
VWS14340
VWS14350
VWS14360
VWS14370
VWS14380
VWS14390
VWS14400
VWS14410
VWS14420
VWS14430
VWS14440
VWS14450
VWS14460
VWS14470
VWS14480
VWS14490
VWS14500
VWS14510
VWS14520
VWS14530
VWS14540
VWS14550
VWS14560
VWS14570
VWS14580
VWS14590
VWS14600
VWS14610
VWS14620
VWS14630
VWS14640
VWS14650
VWS14660
VWS14670
VWS14680
VWS14690
VWS14700
VWS14710
VWS14720
VWS14730
VWS14740
VWS14750
VWS14760
VWS14770
VWS14780
VWS14790
VWS14800
VWS14810
VWS14820
VWS14830
VWS14840
VWS14850


```

20 DC 40 I=1,251
  XVAL=XCRIG+(I-1)*DELTA
  UINV=1C.***(XVAL+XADD)
  U=1./UINV
  CALL E1(U,NU)
  YVAL=DLGIG(WU)-YADD
  IF (YVAL.GT.YMAX) GC TO 50
  IF (YVAL.LT.YCRIG) GC TO 40
  XLCC=XC+XSCALE*(XVAL-XCRIG)
  YLCC=YC+YSCALE*(YVAL-YCRIG)
  IF (IFLAG.EQ.1) GC TO 30
  IFLAG=1
  CALL PLIMOV(XLCC,YLCC,3)
30 CALL FLTMOV(XLCC,YLCC,2)
40 CONTINUE
50 CALL FLTMOV(C.,C.,-999)
C---FORMATS
510 FORMAT (A4)
C
999 RETURN
  END
  SUBROUTINE PNTOUT(TIMLC,TIMHI,RESTF)
C
C      PRINT OUT THE RESULTS
C
C      DATA INPUT:
C      TIMLC - 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      LOGICAL*1 RESTF
C      COMMON /DATA/ N,K,G,TPLMP,NREC,STCR,TRAN,TRAN1,TRAN2,A,B,TO
C-----
C
C      WRITE (6,601) TIMLC,TIMHI
C      WRITE (6,602) TRAN2
C      IF (.NOT.RESTF) WRITE (6,603) STCR
C
C      FORMAT STATEMENTS
C
601 FORMAT (/5A,'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      OUTPLT VECTORS A AND B (EACH LENGTH N) IN 3 COLUMNS

```

VNS1541C
VNS1542C
VNS1543C
VNS1544C
VNS1545C
VNS1546C
VNS1547C
VNS1548C
VNS1549C
VNS1550C
VNS1551C
VNS1552C
VNS1553C
VNS1554C
VNS1555C
VNS1556C
VNS1557C
VNS1558C
VNS1559C
VNS1560C
VNS1561C
VNS1562C
VNS1563C
VNS1564C
VNS1565C
VNS1566C
VNS1567C
VNS1568C
VNS1569C
VNS1570C
VNS1571C
VNS1572C
VNS1573C
VNS1574C
VNS1575C
VNS1576C
VNS1577C
VNS1578C
VNS1579C
VNS1580C
VNS1581C
VNS1582C
VNS1583C
VNS1584C
VNS1585C
VNS1586C
VNS1587C
VNS1588C
VNS1589C
VNS1590C
VNS1591C
VNS1592C
VNS1593C
VNS1594C
VNS1595C

```

C                                     VnS15960
      REAL A(200),B(200)                VnS15970
      INTEGER K(200),KK(3)              VnS15980
C---INITIALIZE TABLE PARAMETERS        VnS15990
      DC 10 I=1,200                     VnS16000
      10  K(I)=I                         VnS16010
          NRCH=N/3                       VnS16020
          IREM=N-3*NRCH                  VnS16030
          KI=NRCH+1                      VnS16040
          KK(1)=0                        VnS16050
          KK(2)=NRCH                    VnS16060
          IF (IREM.GE.1) KK(2)=KK(2)+1  VnS16070
          KK(3)=KK(2)+NRCH              VnS16080
          IF (IREM.GE.2) KK(3)=KK(3)+1  VnS16090
C---OUTPUT THE TABLE                  VnS16100
      WRITE (6,601) ((K(I+KK(J)),A(I+KK(J)),B(I+KK(J)),J=1,3),I=1,NRCH) VnS16110
      IF (IREM.GT.0) WRITE (6,601) (K(KI+KK(J)),A(KI+KK(J)),B(KI+KK(J)), VnS16120
      & J=1,IREM)                       VnS16130
C                                     VnS16140
      601 FLMAT (3(1X,13,1X,F10.2,2X,F10.2,6X)) VnS16150
      999 RETURN                          VnS16160
      END                                  VnS16170
      SUBROUTINE SEARCH(IEND)             VnS16180
C                                     VnS16190
C      SEARCH THE SOLUTION SPACE        VnS16200
C                                     VnS16210
C-----                                VnS16220
C                                     VnS16230
      REAL*4 T(200),S(200),LT(200),LS(200),ERM(200),L,R,STCR,TRAN VnS16240
      INTEGER NPTS                        VnS16250
      LOGICAL*1 IEND                     VnS16260
      COMMON /TITLES/ TITLE              VnS16270
      COMMON /DATA/ N,R,L,TPUMP,NREC,STCR,TRAN,TRAN1,TRAN2,A,B,TU VnS16280
      COMMON /ICATA/ ERM,NPTS,NSUM       VnS16290
      COMMON /XTRA/ LT,LS                 VnS16300
      COMMON /ACC/ XADD,YADD,DX,DY,XCRIG,YCRIG VnS16310
      COMMON /FERR/ R1,R2,R3,R4,R5,R6,R7,R8,R9,RR VnS16320
C                                     VnS16330
C-----                                VnS16340
C                                     VnS16350
C      IEND=.FALSE.                     VnS16360
      RRCLD=999999999.                   VnS16370
      XCRIG=0.0                           VnS16380
      YCRIG=-1.0                          VnS16390
      DX=2.0                               VnS16400
      DY=2.0                               VnS16410
      INDEX=0                              VnS16420
      JFLAG=0                              VnS16430
      LCGF=0                               VnS16440
      RR=RRCLD                             VnS16450
C                                     VnS16460
C-----INITIALIZE                      VnS16470
      10 YADD=YCRIG+DY                    VnS16480
      CALL SETLP(A,B,C)                   VnS16490
      R1=A                                 VnS16500

```

R2=E	VWS16510
R3=C	VWS16520
YADC=YCRIG	VWS16530
CALL SETLP(A,B,C)	VWS16540
R4=A	VWS16550
R5=B	VWS16560
R6=C	VWS16570
YADC=YORIG-DY	VWS16580
CALL SETLP(A,B,C)	VWS16590
R7=A	VWS16600
R8=B	VWS16610
R9=C	VWS16620
C	VWS16630
C---COMPARE	VWS16640
20 CALL MIN(INDEX)	VWS16650
LCCF=LCCF+1	VWS16660
CHANGE=AE5((FRGLD-RR)/FRGLD)	VWS16670
IF (LCCF.LT.NSUM) GC TC 30	VWS16680
IF (LOUP.GT.250) GC TC 50	VWS16690
IF (CHANGE.LE.0.001) GC TC 999	VWS16700
30 RRGLD=RR	VWS16710
GC TO (1,2,3,4,5,6,7,8,9),INDEX	VWS16720
C	VWS16730
1 CONTINUE	VWS16740
YCRIG=YCRIG+CY	VWS16750
XCRIG=XCRIG-CX	VWS16760
R6=R2	VWS16770
R5=R5	VWS16780
R5=R1	VWS16790
R8=R4	VWS16800
YADC=YCRIG+DY	VWS16810
CALL SETLP(A,B,C)	VWS16820
R1=A	VWS16830
R2=E	VWS16840
R3=C	VWS16850
XADC=XCRIG-DX	VWS16860
YADC=YCRIG	VWS16870
CALL COMFLT(R4,JFLAG)	VWS16880
YADC=YCRIG-DY	VWS16890
CALL COMFLT(R7,JFLAG)	VWS16900
GC TC 20	VWS16910
C	VWS16920
2 CONTINUE	VWS16930
YORIG=YCRIG+CY	VWS16940
R7=R4	VWS16950
R8=R5	VWS16960
R9=R6	VWS16970
R4=R1	VWS16980
R5=R2	VWS16990
R6=R3	VWS17000
YADC=YCRIG+DY	VWS17010
CALL SETLP(A,B,C)	VWS17020
R1=A	VWS17030
R2=E	VWS17040
R3=C	VWS17050

	GC TC 20	VWS17060
C		VWS17070
	3 CCNTINUE	VWS17080
	XCRIG=XCRIG+CX	VWS17090
	YCRIG=YCRIG+CY	VWS17100
	R4=R2	VWS17110
	R7=R5	VWS17120
	R5=R3	VWS17130
	R8=R6	VWS17140
	YADD=YCRIG+DY	VWS17150
	CALL SETLP(A,B,C)	VWS17160
	R1=A	VWS17170
	R2=B	VWS17180
	R3=C	VWS17190
	XADC=XCRIG+DX	VWS17200
	YADC=YCRIG	VWS17210
	CALL COMPUT(R8,JFLAG)	VWS17220
	YADC=YCRIG-DY	VWS17230
	CALL CCPLT(R9,JFLAG)	VWS17240
	GC TC 20	VWS17250
		VWS17260
C		VWS17270
	4 CONTINUE	VWS17280
	XCRIG=XCRIG-CX	VWS17290
	R3=R2	VWS17300
	R6=R5	VWS17310
	R9=R8	VWS17320
	R2=R1	VWS17330
	R5=R4	VWS17340
	R8=R7	VWS17350
	XADC=XCRIG-DX	VWS17360
	YADC=YADC+DY	VWS17370
	CALL CCPLT(R1,JFLAG)	VWS17380
	YADC=YCRIG	VWS17390
	CALL CCPLT(R4,JFLAG)	VWS17400
	YADC=YCRIG-DY	VWS17410
	CALL CCPLT(R7,JFLAG)	VWS17420
	GC TO 20	VWS17430
		VWS17440
C		VWS17450
	5 CCNTINUE	VWS17460
	ERRX=R4+R6	VWS17470
	ERRY=R2+R8	VWS17480
	IF (ERRX.GE.ERRY) DX=DX/1.5	VWS17490
	IF (ERRY.GE.ERRX) CY=CY/1.5	VWS17500
	GC TO 10	VWS17510
		VWS17520
C		VWS17530
	6 CCNTINUE	VWS17540
	XCRIG=XCRIG+CX	VWS17550
	R1=R2	VWS17560
	R4=R5	VWS17570
	R7=R8	VWS17580
	R2=R3	VWS17590
	R5=R6	VWS17600
	R8=R9	VWS17610
	XADC=XCRIG+DX	VWS17620
	YADC=YCRIG+DY	VWS17630

	CALL CCPLT(R3,JFLAG)	VWS17610
	YADC=YCRIG	VWS17620
	CALL CCPLT(R6,JFLAG)	VWS17630
	YADC=YORIG-DY	VWS17640
	CALL CCPLT(R9,JFLAG)	VWS17650
	GO TO 20	VWS17660
C		VWS17670
	7 CONTINUE	VWS17680
	XCRIG=XCRIG-DX	VWS17690
	YORIG=YCRIG-DY	VWS17700
	R2=R4	VWS17710
	R3=R5	VWS17720
	R6=R8	VWS17730
	R5=R7	VWS17740
	XADC=XORIG-DX	VWS17750
	YADC=YCRIG+DY	VWS17760
	CALL CCPLT(R1,JFLAG)	VWS17770
	YADC=YORIG	VWS17780
	CALL CCPLT(R4,JFLAG)	VWS17790
	YADC=YORIG-DY	VWS17800
	CALL SETLP(A,B,C)	VWS17810
	R7=A	VWS17820
	R8=B	VWS17830
	R9=C	VWS17840
	GO TO 20	VWS17850
C		VWS17860
	8 CONTINUE	VWS17870
	YCRIG=YCRIG-DY	VWS17880
	R1=R4	VWS17890
	R2=R5	VWS17900
	R3=R6	VWS17910
	R4=R7	VWS17920
	R5=R8	VWS17930
	R6=R9	VWS17940
	YADC=YCRIG-DY	VWS17950
	CALL SETLP(A,B,C)	VWS17960
	R7=A	VWS17970
	R8=B	VWS17980
	R9=C	VWS17990
	GO TO 20	VWS18000
C		VWS18010
	9 CONTINUE	VWS18020
	XCRIG=XCRIG+DX	VWS18030
	YCRIG=YCRIG-DY	VWS18040
	R1=R5	VWS18050
	R2=R6	VWS18060
	R4=R8	VWS18070
	R5=R9	VWS18080
	YADC=YCRIG-DY	VWS18090
	CALL SETLP(A,B,C)	VWS18100
	R7=A	VWS18110
	R8=B	VWS18120
	R9=C	VWS18130
	XADC=XCRIG+DX	VWS18140
	YADC=YCRIG	VWS18150


```

CALL COMPUT(R6,JFLAG)                                VhS18160
YADD=YCRIG+DY                                        VhS18170
CALL COMPUT(R3,JFLAG)                                VhS18180
GC TC 20                                             VhS18190
C---TGD MANY ITERATIONS, STOP HERE                   VhS18200
50 IEND=.TRUE.                                       VhS18210
C                                                     VhS18220
999 RETLRN                                           VhS18230
END                                                  VhS18240
SUBROUTINE SETUP(A,B,C)                              VhS18250
C                                                     VhS18260
C                                                     VhS18270
-----
C                                                     VhS18280
C                                                     VhS18290
COMMON /ACC/ XACC,YACC,DX,DY,XCRIG,YCRIG            VhS18300
C                                                     VhS18310
-----
C                                                     VhS18320
C                                                     VhS18330
JFLAG=0                                             VhS18340
XACC=XCRIG-DX                                       VhS18350
CALL COMFLT(A,JFLAG)                                VhS18360
XACC=XCRIG                                           VhS18370
CALL COMPUT(B,JFLAG)                                VhS18380
XACC=XCRIG+DX                                       VhS18390
CALL COMFLT(C,JFLAG)                                VhS18400
C                                                     VhS18410
999 RETLRN                                           VhS18420
END                                                  VhS18430
SUBROUTINE MIN(IN)                                   VhS18440
C                                                     VhS18450
C                                                     VhS18460
-----
C                                                     VhS18470
C                                                     VhS18480
COMMON /ERR/ R1,R2,R3,R4,R5,R6,R7,R8,R9,RR         VhS18490
C                                                     VhS18500
-----
C                                                     VhS18510
C                                                     VhS18520
RR=AMIN1(R1,R2,R3,R4,R5,R6,R7,R8,R9)              VhS18530
IF (RR.EQ.R1) IN=1                                    VhS18540
IF (RR.EQ.R2) IN=2                                    VhS18550
IF (RR.EQ.R3) IN=3                                    VhS18560
IF (RR.EQ.R4) IN=4                                    VhS18570
IF (RR.EQ.R5) IN=5                                    VhS18580
IF (RR.EQ.R6) IN=6                                    VhS18590
IF (RR.EQ.R7) IN=7                                    VhS18600
IF (RR.EQ.R8) IN=8                                    VhS18610
IF (RR.EQ.R9) IN=9                                    VhS18620
C                                                     VhS18630
999 RETLRN                                           VhS18640
END                                                  VhS18650
SUBROUTINE COMPUT(RESIC,JFLAG)                      VhS18660
C                                                     VhS18670
C COMPUTE THE RESIC                                  VhS18680
C                                                     VhS18690
C                                                     VhS18700

```

```

C-----VNS18710
C                                             VNS18720
REAL*4 T(200),S(200),LT(200),LS(200),ERM(200),U,R,STCR,TRAN VNS18730
REAL*8 U,NU,GINV                                             VNS18740
INTEGER NPTS                                                 VNS18750
COMMON /TITLES/ TITLE                                       VNS18760
COMMON /DATA/ N,R,C,TFLMP,NREC,STCR,TRAN,TRAN1,TRAN2,A,B,TO VNS18770
COMMON /TCATA/ ERM,NPTS,NSUM                                 VNS18780
COMMON /XTRA/ LT,LS                                         VNS18790
COMMON /ACD/ XADD,YADD,CX,DY,XORIG,YCRIG                   VNS18800
COMMON /FERK/ R1,R2,R3,R4,R5,R6,R7,R8,R9,RA                VNS18810
C-----VNS18820
C                                             VNS18830
C                                             VNS18840
C                                             VNS18850
C                                             VNS18860
C                                             VNS18870
C                                             VNS18880
C                                             VNS18890
C                                             VNS18900
C                                             VNS18910
C                                             VNS18920
C                                             VNS18930
C                                             VNS18940
C                                             VNS18950
C                                             VNS18960
C                                             VNS18970
C                                             VNS18980
C                                             VNS18990
C                                             VNS19000
C                                             VNS19010
C                                             VNS19020
C                                             VNS19030
C                                             VNS19040
C                                             VNS19050
C                                             VNS19060
C                                             VNS19070
C                                             VNS19080
C                                             VNS19090
C                                             VNS19100
C                                             VNS19110
C                                             VNS19120
C                                             VNS19130
C                                             VNS19140
C                                             VNS19150
C                                             VNS19160
C                                             VNS19170
C                                             VNS19180
C                                             VNS19190
C                                             VNS19200
C                                             VNS19210
C                                             VNS19220
C                                             VNS19230
C                                             VNS19240
C                                             VNS19250

STCRE=0.
DO 10 I=1,NPTS
  GINV=10.**(-LT(I)+XADD)
  U=1./GINV
  CALL E1(U,NU)
  IF (NU.LE.G.0) NU=1.E-50
  NU=DLOG10(NU)
  WLPRED=LS(I)+YADD
  RES=(NU-WLPRED)**2
  STCRE=STCRE+RES
  IF (JFLAG.NE.1) GO TO 10
  ERM(I)=RES
10 CONTINUE
RESID=STCRE
IF (JFLAG.NE.1) GO TO 599
DO 20 I=1,NPTS
  ERM(I)=100.*ERM(I)/RESID
20 CONTINUE
599 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     ALX - 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-F,G-Z)

```

```

Z=0. VNS19260
IF (Y-1.) 2,1,1 VNS19270
1 CCNTINLE VNS19280
S=1.00/Y VNS19290
AUX=1.00 -S VNS19300
6*((S+3.27735000)*S+2.05215000)*S+2.7094790-1)/(((S* VNS19310
61.07255300+S.71694300)*S+0.94523900)*S+2.59388800)*S+2.7094960-1) VNS19320
XY=-Y VNS19330
IF (XY.GE.-10) Z=AUX*S*DEXP(XY) VNS19340
557 RETLRN VNS19350
2 IF (Y+3.) 6,6,3 VNS19360
3 AUX=((((17.1224520-7*Y-1.7663450-6)*Y+2.9284530-5)*Y-2.3353790-4 VNS19370
6)*Y+1.6641560-3)*Y-1.0415760-2)*Y+5.5556820-2)*Y-2.500010-1)*Y VNS19380
6+5.5599990-1 VNS19390
Z=-1.075 VNS19400
IF (Y) 4,998,4 VNS19410
4 CCNTINLE VNS19420
Z=(Y*AUX-CLOG(0A0S(Y))-5.7721570-C1) VNS19430
598 RETLRN VNS19440
6 IF (Y+9.) 8,8,7 VNS19450
7 CCNTINLE VNS19460
ALX=1.00 VNS19470
6-((((5.1762450-2*Y+3.06103700)*Y+3.24300501)*Y+2.24423402)*Y VNS19480
6+2.48669702)/(((Y+3.95510100)*Y+3.09394401)*Y+2.20381801)*Y VNS19490
6+1.00783702) VNS19500
GO TO 5 VNS19510
8 CCNTINLE VNS19520
S=5.00/Y VNS19530
ALX=1.00-S*(((S+7.0598240-1)*S-7.2710150-1)*S-1.08069300)/(((S VNS19540
6*2.51675000+1.12292701)*S+5.92140500)*S-0.06670200)*S-9.72421000) VNS19550
9 CCNTINLE VNS19560
XY=-Y VNS19570
IF (XY.GE.-10) Z=AUX*CEXP(XY)/Y VNS19580
C 559 RETLRN VNS19590
END VNS19600

```

