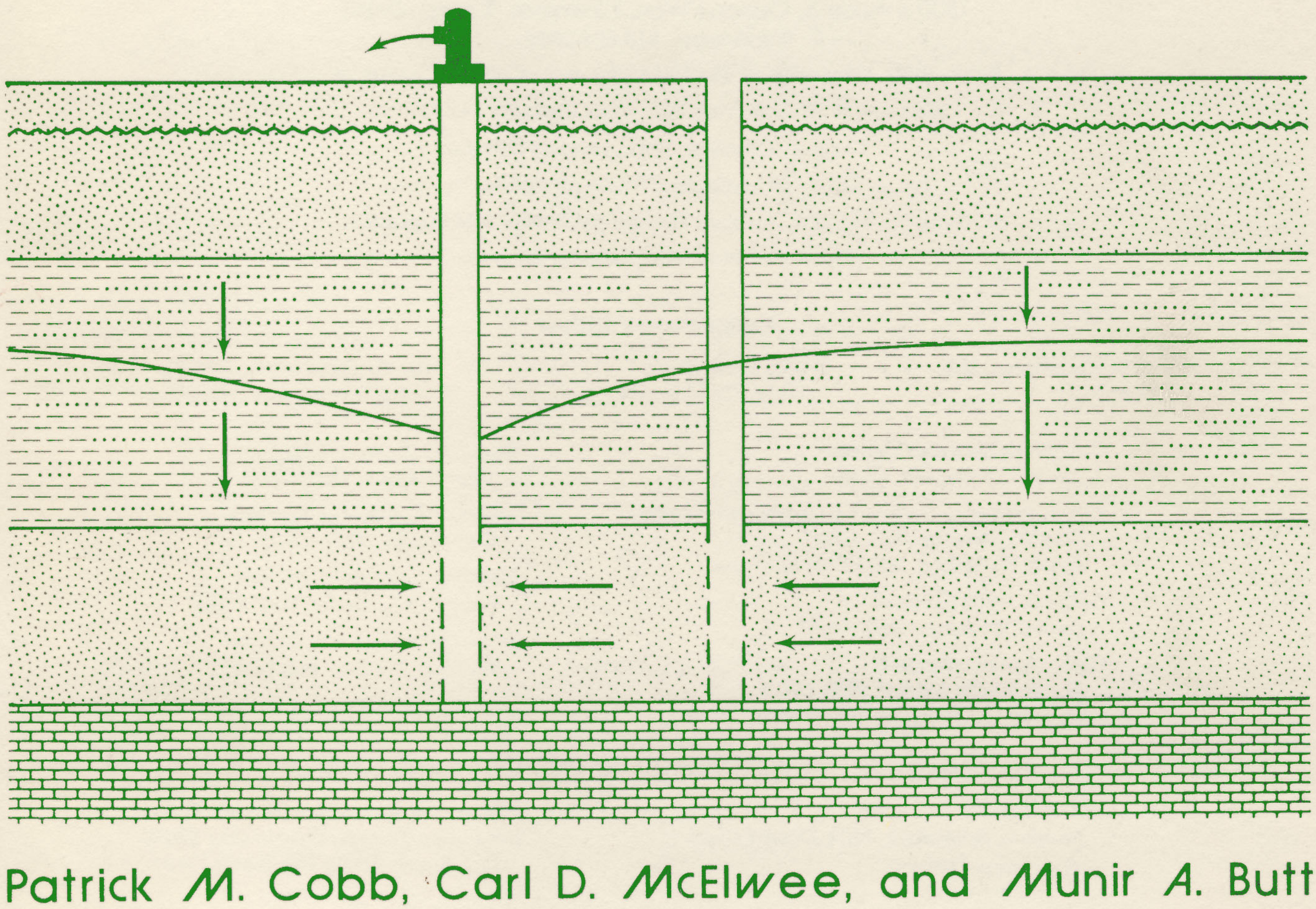
Groundwater Series 6

An Automated Numerical Evaluation of Leaky Aquifer Pumping Test Data: An Application of Sensitivity Analysis



Patrick M. Cobb, Carl D. McElwee, and Munir A. Butt

KANSAS Geological Survey 1982

An Automated Numerical Evaluation of Leaky Aquifer Pumping Test Data:

An Application of Sensitivity Analysis

by

Patrick M. Cobb, Carl D. McElwee, and Munir A. Butt

Kansas Geological Survey

Groundwater Series 6

1982

EXECUTIVE SUMMARY

The prediction of aquifer response to changes in water demand continues to be of increasing importance for groundwater management. Values for aquifer parameters must be determined in order to establish these demand-response relationships. A classic method for determining these aquifer parameters is to conduct a pumping test on an aquifer and take measurements of water-level declines versus time. These values are compared to standard aquifer type-curves to determine the type of aquifer being evaluated and the values of the parameters. The methodology outlined in this publication allows the user to analytically compare the pumping-test data with a general solution for a leaky-confined aquifer using modern numerical curve-fitting techniques that offer a bias-free solution and an estimation of solution accuracy. The values obtained may then be used to predict the demand-response behavior of the tested aquifer in the future.

CONTENTS

FIGURES

Figure

TABLES

Table

AN AUTOMATED NUMERICAL EVALUATION OF LEAKY AQUIFER PUMPING TEST DATA: AN APPLICATION OF SENSITIVITY ANALYSIS

Patrick M. Cobb, Carl D. McElwee, and Munir A. Butt[1]

ABSTRACT

The Kansas Geological Survey is pursuing an effort to automate some of the more common type-curve solutions for aquifer tests. This document discusses the results of the work done on the leaky artesian aquifer as defined by Jacob and Hantush (1955). The text covers the basic theory of the aquifer type, the numerical solution of the leaky artesian well function, W(U,r/B), and the methodology of achieving the "best fit" parameters. In keeping with our attempt to produce a user guide, we have included listings of all programs developed in this effort and examples of their use. Several figures are included that show examples of "best fit" solutions and their corresponding type-curve values. These comparisons indicate the generally satisfactory results produced by the regression algorithms documented here. The program documented here for leaky artesian aquifer drawdown functions in an acceptable fashion and could serve as the core for an analytical well-field simulator capable of handling that type of aquifer.

INTRODUCTION

This report is a detailed summary of the development and testing of a numerical algorithm designed to analyze leaky-aquifer pumping-test data by automated-fitting techniques, using sensitivity theory. It is the immediate successor to similar work done for a simple confined aquifer (McElwee, 1980a). It represents part of the effort of the Kansas Geological Survey to produce some practical tools for hydrologists.

The program discussed in this paper solves the parameter evaluation problem for an elementary leaky-artesian aquifer system as posed by Hantush and Jacob (1955). The situation considered in this work is not the most general configuration (see Hantush, 1960; Neuman and Witherspoon, 1969a); however, the limited number of available data sets tend to be applicable to this simple scheme. The limitations of the theory used here are outlined by

[1]Geohydrology Section, Kansas Geological Survey.

Neuman and Witherspoon (1969b). The methodology used in the present study involves sensitivity analysis and a least-squares fitting technique to analyze the time-drawdown data while satisfying the equations developed by Hantush and Jacob (1955). These techniques will be outlined in the text. More information may be found in McElwee (1980a, 1980b), McElwee and Yukler (1978), and Cobb and others (1978).

Because of the limited number of available data sets for this aquifer configuration, this technique is being published after extensive but not exhaustive testing. However, we have tested it for several hypothetical data sets and for seven real data sets readily available to us. At this point, we feel quite confident in the algorithm's capabilities. We hope that setting this algorithm out for public scrutiny will cause testing of new data sets and more thorough verification of the program. Using the available data sets, we have been able to establish that, for fairly smooth data sets (those that conform generally to the shape of the leaky type curves), the model has excellent convergence properties. Initial estimates of the storage coefficient, transmissivity, and leakage coefficient may be in the range of plus or minus three orders of magnitude of the correct value and still obtain successful convergence.

This method of pumping-test analysis does not remove the requirement of having an experienced hydrologist evaluate the local hydrogeology and pumping-test data to identify the aquifer type. However, once it is decided what aquifer configuration is being observed, this program will, in a quick and unbiased fashion, give an accurate assessment of the leaky-aquifer parameters within the limits of the theoretical approximations. After using this model for the pumping-test analysis, the hydrologist should always look at the root-mean-square (rms) deviation in drawdown and the "best fit" drawdowns calculated by the program. The experimental and theoretical drawdowns should not differ greatly anywhere and the rms deviation should be less than a few tenths of a foot for us to have confidence in the analysis. If this is not the case, one is probably not dealing with a simple leaky aquifer.

THEORY AND ANALYTICAL SOLUTION TO THE LEAKY-CONFINED-AQUIFER PROBLEM

The aquifer system defined by Hantush and Jacob (1955) (Fig. 1) is composed of a level, isotropic, homogeneous, porous medium of infinite areal extent. The lower aquifer boundary is assumed to be impervious, and the upper boundary to be a leaky confining bed. A source bed overlies the leaky confining bed. Water is derived from the aquifer by elastic expansion of the water and compression of the aquifer matrix as pumping occurs. Leakage through the semiconfining bed is assumed to be proportional to the drawdown in the semiconfined aquifer. No water is removed from storage in the semiconfining unit and no drawdown occurs in the source bed.

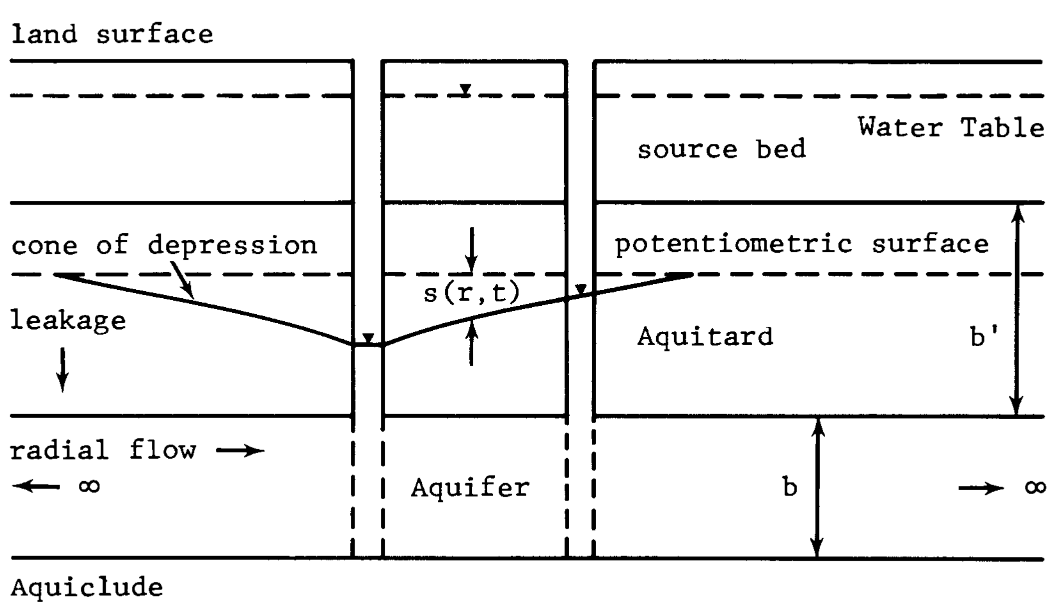


Figure 1. Definition of the problem (from Walton, 1970).

These assumptions lead to the following differential equation (Jacob, 1946)

1) $$\frac{\partial^2 s}{\partial r^2} + \frac{1}{r}\frac{\partial s}{\partial r} - \frac{s}{B^2} = (\frac{S}{T})\frac{\partial s}{\partial t}, \text{ where}$$

s(r,t) is the drawdown at any distance from the well at any time,

r is the radial distance measured from the well,

S is the storage coefficient of the artesian aquifer,

K and K' are the respective permeabilities of the artesian aquifer and the semiconfining bed,

b and b' are the respective thicknesses of the artesian aquifer and the semiconfining bed,

T = Kb is the transmissivity of the artesian aquifer,

K'/b' is the leakance of specific leakage of the semiconfining bed,

$B^2 = T/(K'/b')$,

Q is the well discharge.

With appropriate boundary conditions, an analytical solution can be obtained.

Hantush and Jacob (1955) give several solutions to equation (1) for different ranges of u (defined below) and r/B. In this program three equations are solved numerically in order to cover the broadest possible range of u and r/B. The equations are listed here, along with the appropriate ranges of u and r/B.

2) $$s = Q/4\pi T \bullet \int_u^\infty \exp(-y-z/y)dy/y$$

$$u = r^2S/4Tt,\ z = r^2/4B^2, \qquad u > 1.0, \quad \text{any value of } r/B$$

3) $$s = Q/4\pi T \bullet \left[2K_0(r/B) - \int_p^{\infty} \exp(-y-z/y)dy/y \right]$$

$$p = Tt/SB^2, \quad r^2B^2 \geq u \leq 1.0$$

4) $$s = Q/4\pi T \bullet \{ 2K_0(r/B) - I_0(r/B) \bullet -Ei(-r^2/4B^2u)$$

$$+ \exp(-r^2/4B^2u) \, [0.5772 + \ln(u) - Ei(-u)$$

$$-u + u\,(I_0(r/B)-1)/(r^2/4B^2)$$

$$- u^2 \sum_{n=1}^{\infty} \sum_{m=1}^{n} \frac{(-1)^{n+m}(n-m+1)!}{((n+2)!)^2} (r^2/4B^2)^m u^{n-m}]\}$$

$$(r/B)^2 \leq u \leq 1$$

where Ei(x) is the exponential integral, and I_0 and K_0 are the modified zero-order Bessel functions of the first and second kind, respectively. The numerical computational routines involving these functions were checked by generating the table published in Walton, 1970 (p. 146). This table could be produced accurately to the fourth decimal place.

NUMERICAL SOLUTION OF THE LEAKY-CONFINED-AQUIFER PROBLEM

Integral functions of the form

$$\int_0^{\infty} f(x)e^{-x}dx$$

may be approximated by the method of Laguerre integration:

$$\int_0^{\infty} f(x)e^{-x}dx \simeq \sum_{i=1}^{n} w_i f(x_i)$$

where the w_i's are weighting factors and the x_i's are the abscissas and correspond to the zeros of Laguerre polynomials. The method of solution and values of w_i and x_i are catalogued in Abramowitz and Stegun (1968).

To perform the integrations in equations 2 and 3, a transformation of variables must occur in order to make the limits of integration compatible with the Laguerre Polynomial method. This transformation is a straightforward substitution of the form y = x + n, where n takes the value of u for equation 2 and p for equation 3. The integrals take the form

$$G(r/B,n) = \int_0^\infty \exp\{-n-r^2/[4B^2(x+n)]\}/(x+n) \bullet \exp(-x) \bullet dx$$

and are solved numerically by the appropriate substitutions in the Laguerre integration formula.

The function G(r/B,n) was evaluated by Laguerre integration of order 15. We considered the possibility that G(r/B,n) might not be accurately approximated for all possible values of r/B and n. The roots of the Laguerre polynomial should sample the function to be integrated properly for desired accuracy. A scaling transformation was incorporated to change the range of abscissas over which the evaluation occured. The transformation was of the form x = ay such that:

$$f(y) = f(x/a)$$

and

$$\int_0^\infty h(x) \bullet \exp(-x) \bullet dx/a \simeq \frac{1}{a} \sum_i w_i h(x_i),$$

where h(x) = f(x/a)exp(1/a).

Results of numerical experiments indicate that the value of the scaling transformation ranging from 1.0 to 10.0 has no effect upon the integral when it is used simply to solve the drawdown equation. However, in the regression algorithm, a slight economy in number of iterations occurs when the scaling transformation is set equal to 5.0 (parameter AA in the function list, Appendix IV). This is a result of improvement in the evaluation of the higher-order decimal places.

The numerical solution of the exponential integral, Ei(x), is described in detail in McElwee (1980b, p. 3). Solutions for the modified Bessels functions of the first and second kinds, zero order, $I_0(x)$ and $K_0(x)$, were carried out in the manner of polynomial approximations. Abramowitz and Stegun (1968) catalog several forms for each function. Each form is suitable for a particular range of x. The solutions appear in the function list (Appendix IV) as Function AIO(RB) and AKO(RB).

The double summation in equation (4) is solved numerically by Function SUM(U,RB) and Function IFACT(L) (see Appendix IV). A truncated summation is performed, since only a finite number of terms are required to approximate a convergent function. Numerical experiments showed that n=5 (LIMIT=5 in SUM(U,RB) yields a very accurate value for the summation.

SENSITIVITY ANALYSIS

Parametric sensitivity analysis is a method of examining the stability of a mathematical representation of a dynamic system with respect to variations in the values of the system's physical parameters. The theoretical basis of this technique is outlined by Tomovic (1962), while the application to hydrologic problems has been examined by Vemuri and others (1969), McCuen (1973), and Yukler (1976).

In formulating the sensitivity analysis of the leaky-confined-aquifer problem, the following functional is useful:

$$F(h_{xx}, h_{yy}, h_t; \; S, T, L, Q) = 0$$

where $h_{xx} = \frac{\partial^2 h}{\partial x^2}$, $h_{yy} = \frac{\partial^2 h}{\partial y^2}$, $h_t = \frac{\partial h}{\partial t}$

h = hydraulic head

S = storage coefficient

T = transmissivity

L = inverse leakage coefficient (L = 1/B)

Q = pumpage

and whose solution may be written as $h = h(x,y,t;S,T,L,Q)$. Variations of any single parameter such as T produces a new formulation

$$F(h^*_{xx}, h^*_{yy}, h^*_t; S, T+\Delta T, L, Q) = 0$$

where ΔT is the incremental change in T and h* is the perturbed head. The solution to this expression is of the form $h^* = h^*(x,y,t;S,T+\Delta T,L,Q)$. The stability of the system to small changes in the parameter T may be expressed by

$$\frac{\Delta h}{\Delta T} = \frac{h^* - h}{\Delta T}.$$

If the limit to this expression exists as ΔT approaches zero, it may be written as

$$U_T(x,y,t;S,T,L,Q) = \frac{\partial h}{\partial T} = \lim_{\Delta T \to 0} \frac{\Delta h}{\Delta T}.$$

Similarly

$$U_S(x,y,t;S,T,L,Q) = \frac{\partial h}{\partial S} = \lim_{\Delta S \to 0} \frac{\Delta h}{\Delta S}$$

and

$$U_L(x,y,t;S,T,L,Q) = \frac{\partial h}{\partial L} = \lim_{\Delta L \to 0} \frac{\Delta h}{\Delta L},$$

which are respectively the sensitivity coefficient with respect to changes in S and the sensitivity coefficient with respect to changes in L.

The equation of drawdown is assumed to depend analytically upon the parameters S, T, and L; and S, T, L, and Q are independent of each other. Because of these assumptions, the function, $h^*(x,y,t;S,T+\Delta T,L,Q)$, which is perturbed in the parameter T, may be expanded in a Taylor's series (Tomovic, 1962), and if ΔT is small, all non-linear terms can be neglected

$$h^*(x,y,t;S,T+\Delta T,L,Q) = h(x,y,t;S,T,L,Q) + U_T \Delta T$$

where $U_T = \frac{\partial h}{\partial T}$. Thus, new hydraulic heads, resulting from incremental changes in T, can be computed directly if the unperturbed head is known and U_T can be computed. Similar expressions may be derived for perturbation with respect to S and L

$$h^*(x,y,t;S+\Delta S,T,L,Q) = h(x,y,t;S,T,L,Q) + U_S \Delta S$$

$$h^*(x,y,t;S,T,L+\Delta L,Q) = h(x,y,t;S,T,L,Q) + U_L \Delta L.$$

For this technique to be useful, it is only necessary to be able to compute U_S, U_T, and U_L, since $h(x,y,t;S,T,L,Q)$ may be computed by previously discussed techniques. This requirement may be satisfied by analytical or numerical techniques. In this work, it was found convenient to obtain U_S and U_T by direct analytical means and U_L by a numerical method.

Recall that the basic equation describing the solution to the leaky confined aquifer is

$$5) \qquad s = \frac{Q}{4\pi T} \int_u^\infty \frac{1}{y} \exp\left(-y - \frac{L^2 r^2}{4y}\right) dy, \qquad u = \frac{r^2 S}{4Tt}, \quad L = B^{-1}$$

By applying Leibnitz's rule for differentiating an integral (Hildebrand, 1962) it is easy to obtain the sensitivity coefficients with respect to S and T:

$$U_S = \frac{\partial s}{\partial S} = -\frac{Qr^2}{16\pi T^2 t}\left(\frac{1}{u}\exp(-u - \frac{L^2r^2}{4u})\right)$$

$$U_T = \frac{\partial s}{\partial T} = -\frac{Q}{4\pi T^2}\int_u^\infty \frac{1}{y}\exp(-y - \frac{L^2r^2}{4y})dy + \frac{Qr^2 S}{16\pi T^3 t}\left[\frac{1}{u}\exp(-u - \frac{L^2r^2}{4u})\right]$$

$$= -\frac{s}{T} + \frac{Qr^2S}{16\pi T^3 t}\left[\frac{1}{u}\exp(-u - \frac{L^2r^2}{4u})\right] = -\frac{s}{T} - \frac{S}{T}U_s.$$

These equations may be easily evaluated by standard numerical techniques on a high-speed computer.

We computed U_L by a direct numerical technique, rather than by formulating an analytical solution, because of a desire to conserve program simplicity while retaining computational accuracy. Note that the argument of the exponential within the integral of equation (5) contains the parameter L. Hence, differentiation would transform the entire function within the integral and would define

$$U_L = \int_u^\infty \frac{\partial}{\partial L}\left\{\exp(-y - L^2r^2/4y)/y\right\}dy$$

$$= \int_u^\infty \left\{-Lr^2/2y^2\right\}\exp(-y - L^2r^2/4y)dy.$$

Note that both U_S and U_T can be expressed in such a manner that, after the drawdown s is computed, no further numerical integration is required in that iteration. The sensitivity with respect to leakage, U_L, however can only be computed by numerical integration that would involve the formulation of a more complex function of Function SS(U,RB) (see Appendix IV). Since a numerical integration of W(u,r/B) was already operable (see Function W(u,RB), Appendix IV), the decision was made to generate U_L by finite difference approximation. The approximation

$$\partial s/\partial L \simeq \{s(L+\Delta L) - s(L-\Delta L)\}/2\Delta L$$

where

$$s(L\pm\Delta L) = Q/4\pi T \int_u^\infty \exp\{-y - r^2(L\pm\Delta L)^2/4y\}/y \bullet dy$$

becomes increasingly accurate as ΔL approaches zero. Satisfactory evaluation of U_L occurred for ΔL set equal to .01 L. The methodology for computing the sensitivity coefficients is now complete.

DISCUSSION OF THE LEAKY-AQUIFER SENSITIVITY COEFFICIENTS

Comparison of our results with similar work done by McElwee and Yukler (1978) for the fully confined aquifer (specifically figures 3, 5, 7, and 8 of that publication) will allow the reader to gain a better understanding of the differences and similarities of the time-dependent and space-dependent behavior of the two systems with respect to their principal hydrologic parameters.

The radial dependence of U_T is shown in Figure 2. The function diverges logarithmically near the well. U_T changes sign at some finite value of radius. This demonstrates that, when T is changed, the cone of depression deepens in some areas and shallows in others.

Figure 3 depicts the time dependence of positive values of U_T on variations in r and T. Note that U_T is inversely proportional to T. The curves represent a transmissivity of 24331 ft^2/day and ±20 percent of that value at a radius of 100 feet and of 24331 ft^2/day at a radius of 1000 feet. Note that all curves flatten after three to four days. This describes the steady condition caused by deriving the discharge Q totally from leakage.

The radial dependence of U_S is shown in Figure 4. This coefficient does not diverge at the well, nor does its sign change. It is inversely proportional to S. The constancy of algebraic sign indicates that as S changes

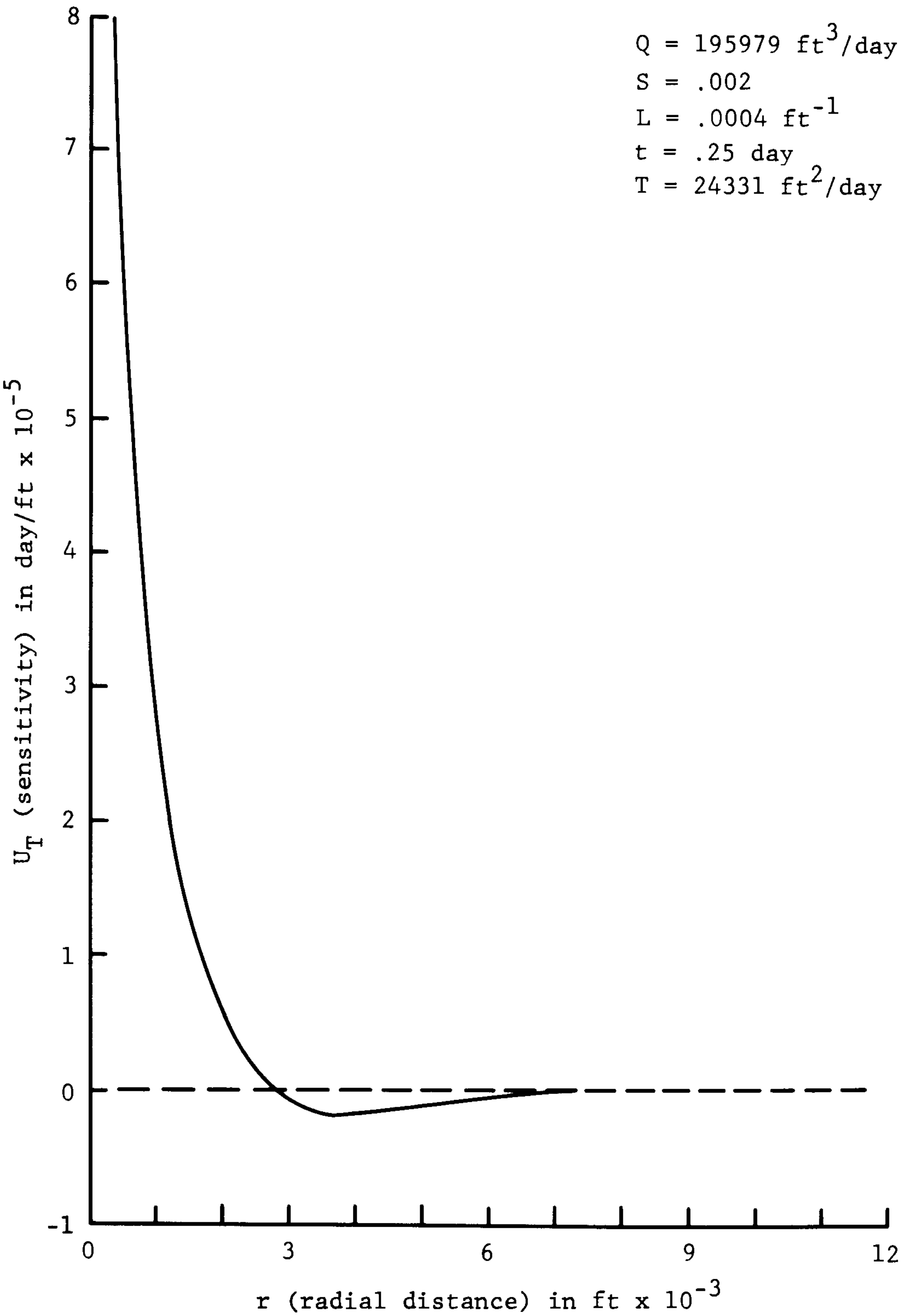


Figure 2. Radial dependence of U_T.

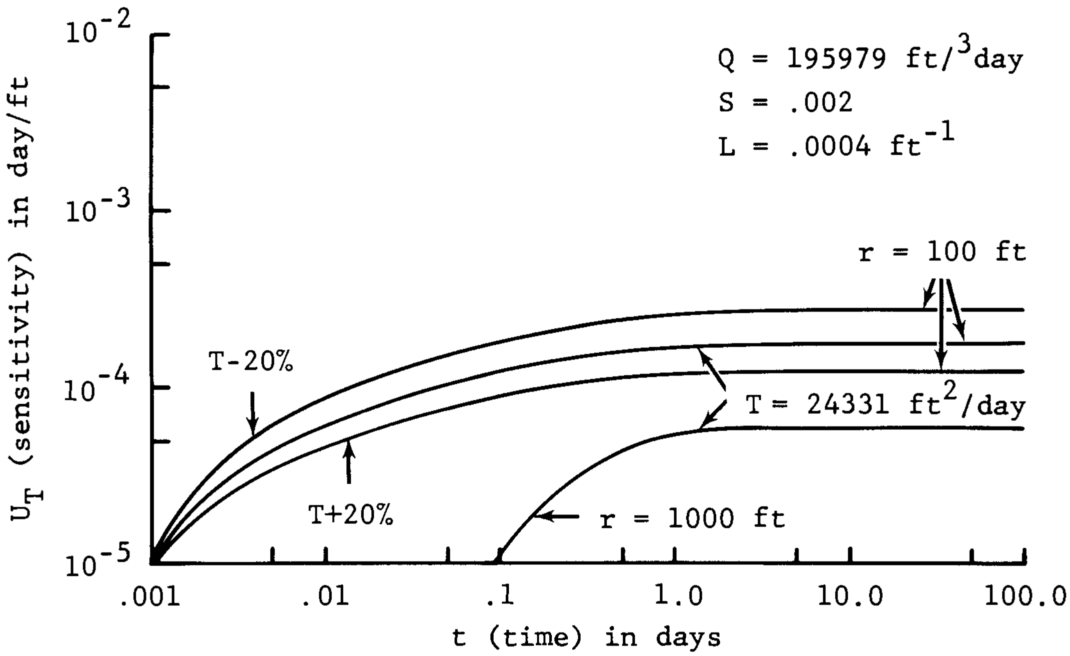


Figure 3. Effect of radius and transmissivity on the time dependence of U_T.

there is a general raising and lowering of the cone of depression.

The time dependence of U_S is presented in Figure 5. Radial variation is represented by three curves. Each curve reaches its maximum value for U_S at a time directly proportional to its radial value. At some finite value of time each curve approaches zero in value, indicating that a steady state is achieved. The differing nature of the curves is related to the fact that, until steady state is attained, there is a dual source supplying the pumpage, namely water released from storage and leakage. The curves roll over as leakage starts to dominate the source mechanism. U_S is zero outside the cone of depression and at any time t after steady state is attained.

Figure 6 shows the radial dependence of U_L. The sensitivity coefficient U_L does not diverge at the well and approaches zero for large values of r.

The time dependence of U_L is shown in Figure 7 for two values of r. All curves grow with time until a steady state is achieved where leakage is supplying the entire discharge Q. The set of curves labeled L = .0004 ft^{-1} and

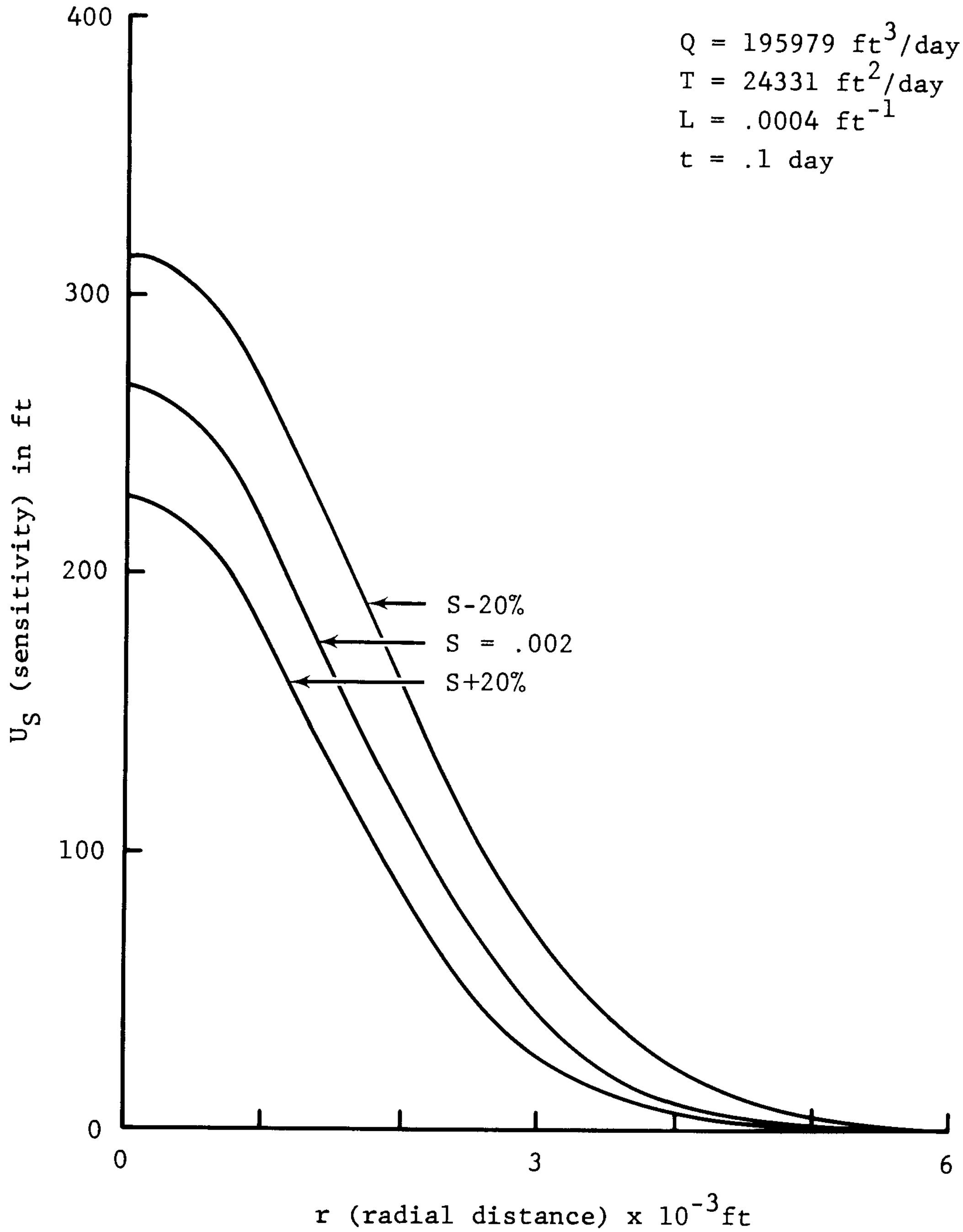


Figure 4. Effects of changes in S on the radial dependence of U_S.

±20 percent of that value is of interest. Observe that at t less than .6 days, U_L is directly proportional to L, while for t greater than .6 days, U_L is inversely proportional to L. As indicated before, Q is supplied by a dual source in the leaky artesian aquifer--water taken from storage in the aquifer

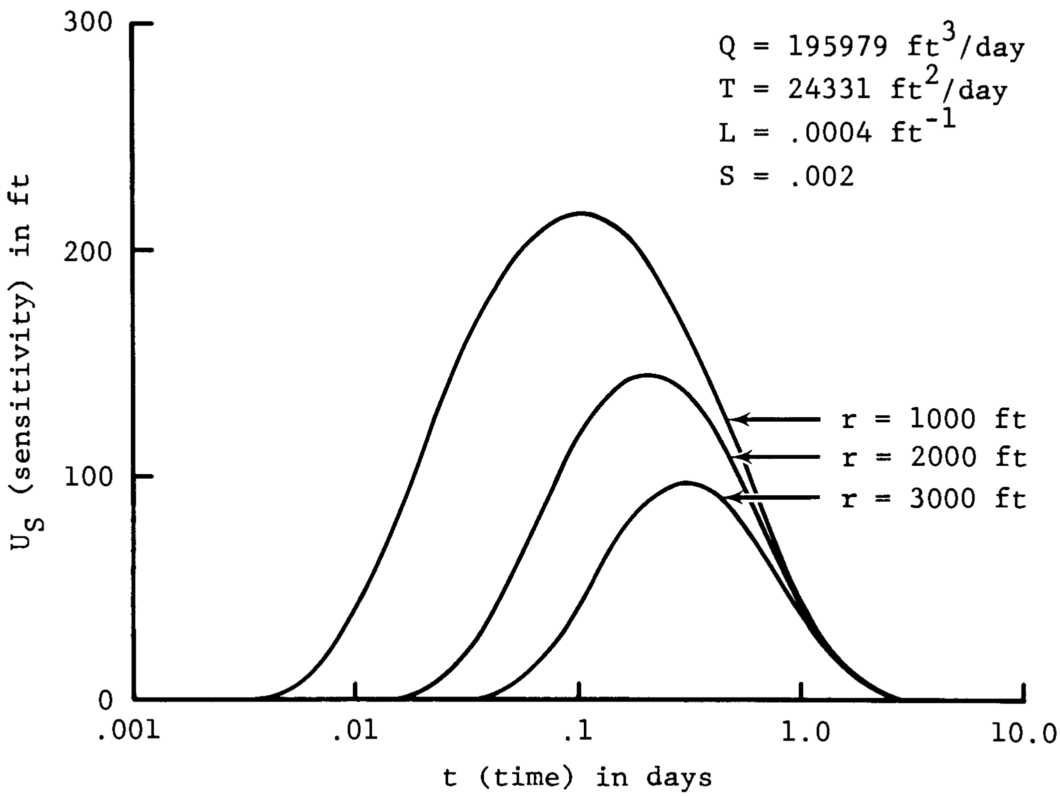


Figure 5. Effects of radius on the time dependence of U_S.

and water supplied by leakage through the aquitard. This dual source mechanism results in the changing dependence on L.

THE LEAST-SQUARES-FITTING PROCEDURE

The objective of any curve-fitting technique, whether performed manually or by machine, is to fit as well as possible a theoretical type curve to an experimental data set, evaluating in the process a corresponding set of physical parameters. To perform this task successfully, a mechanism is required for judging the error in the fit. Classical manual curve-fitting relies basically on the best "eyeball" fit. The machine method described here allows the error in fitting to be accurately and meaningfully determined as the rms error.

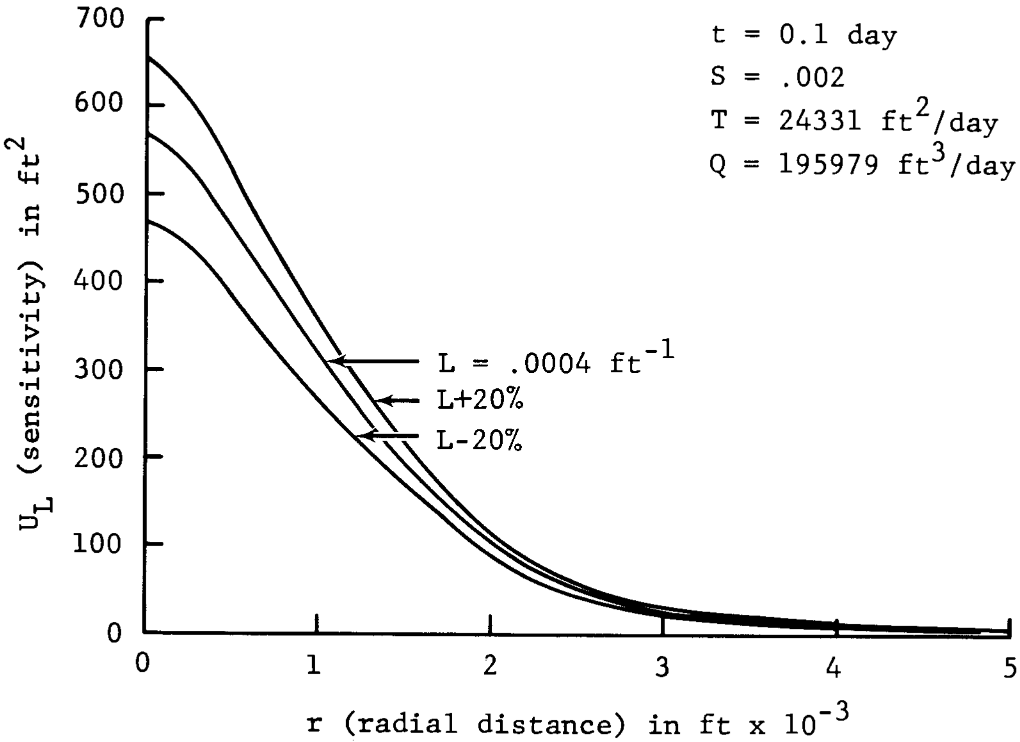


Figure 6. Effects of changes in L on the radial dependence of U_L.

To apply the parametric sensitivity method to the fitting problem, it is necessary to define the squared error function

$$E = \sum_i [s_e(t_i) - s^*(t_i)]^2$$

where E is the summation over i discrete samples of the squared difference between the experimental drawdown s_e and the updated drawdown s^*, which is computed from the truncated Taylor's Series

$$s^* = s + U_T \Delta T + U_S \Delta S + U_L \Delta L.$$

The argument t_i represents the i^{th} value of time. Expanding the squared error function, taking partial derivatives with respect to the perturbed parameters, and setting the partial derivatives equal to zero yields a set of three simultaneous linear equations that must be satisfied to obtain the best fit. More specifically, for minimizing E, it is required that

$$\frac{\partial E}{\partial \Delta T} = \frac{\partial E}{\partial \Delta S} = \frac{\partial E}{\partial \Delta L} = 0.$$

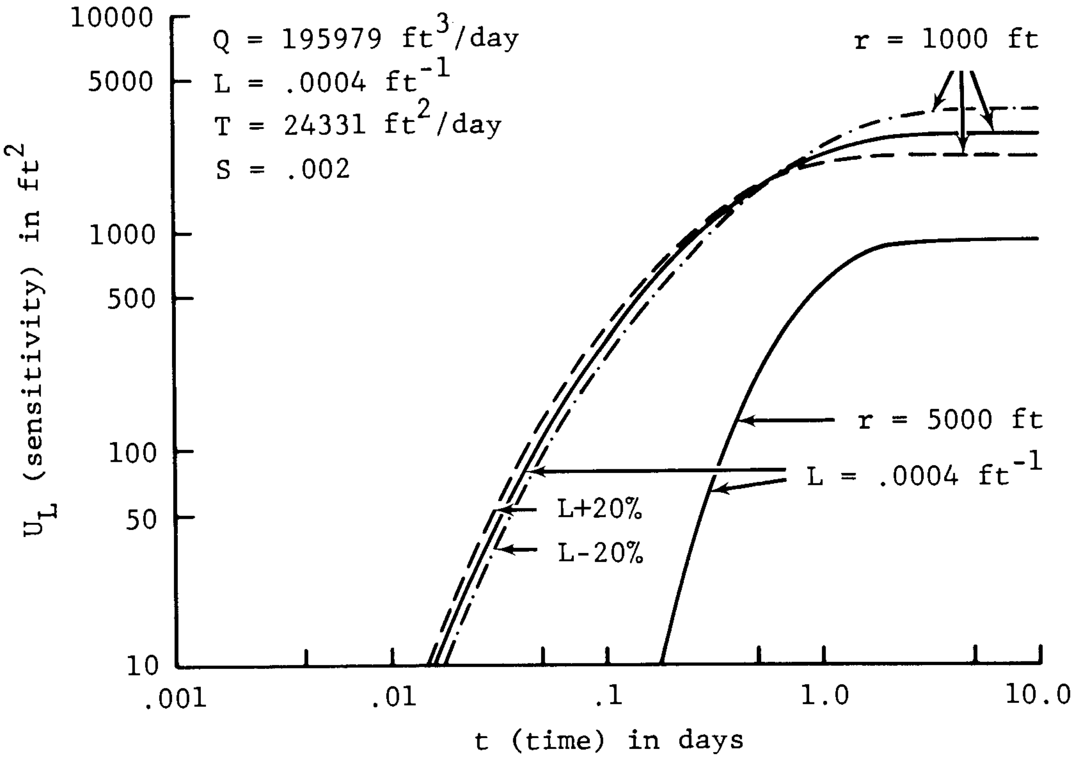


Figure 7. Effect of changes in L and R on the time dependence of U_L.

The linear system of equations that results is

$$\begin{bmatrix} \sum_i U_L^2 & \sum_i U_L U_S & \sum_i U_L U_T \\ \sum_i U_S U_L & \sum_i U_S^2 & \sum_i U_S U_T \\ \sum_i U_T U_L & \sum_i U_T U_S & \sum_i U_T^2 \end{bmatrix} \begin{bmatrix} \Delta L \\ \Delta S \\ \Delta T \end{bmatrix} = \begin{bmatrix} \sum_i U_L(s-s_e) \\ \sum_i U_S(s-s_e) \\ \sum_i U_T(s-s_e) \end{bmatrix}$$

and can be solved explicitly for ΔL, ΔS, and ΔT. The term s is the theoretical drawdown at time t calculated from the previously updated values of L, S, and T. The new values of the parameters are simply

$$L_{i+1} = L_i + \Delta L_i$$

$$S_{i+1} = S_i + \Delta S_i$$

$$T_{i+1} = T_i + \Delta T_i.$$

This process continues until the values of ΔL_i, ΔS_i, and ΔT_i simultaneously satisfy a specified convergence criteria. The goodness of fit obtained at the termination of the last iteration is indicated by the value of the root mean squared error

$$\frac{\sum_i (s - s_e)^2}{n}$$

where n is the number of discrete samples of s.

The success of this methodology is dependent to a degree upon the initial guesses of the parameters S, T, and L. However, numerical experiments conducted with the most recent version of the program indicate that the initial guesses may be as much as three orders of magnitude above or below the converged solution values and still obtain convergence.

To satisfy physical reality and to improve numerical stability, the parameters S, T, and L must always be positive. Furthermore, the increments ΔT, ΔS, and ΔL are never allowed to exceed 0.5 or be less than -0.2 of their respective parameters. These values were derived experimentally. This mechanism insures that the algorithm executes in a convergent fashion to the local and possibly global minimum.

CONVERGENCE PROPERTIES OF THE FITTING ALGORITHM

To achieve a converged solution, the typical regression model involving sensitivity analysis requires initial estimates of the aquifer parameters. If these estimates vary greatly from the actual values, convergence may not occur. A desirable property of this type of algorithm would be the existence of a large "window" in which estimates can be made and convergence can be achieved in a reasonable number of iterations.

The algorithm presented in this discussion (see Appendices I, II, IV) has consistently proven its ability to converge to the correct set of aquifer parameters for a given data set. These "best fit" values are comparable to values of the parameters obtained by curve-matching methods, and are achieved over a spectrum of initial estimates ranging from three orders of magnitude above to three orders of magnitude below the converged values (see Table 1). The sources for these data are tabulated in Appendix V. Typical data sets, with initial estimates in this range, converge in fewer than 40 iterations. Numerical experiments indicate that convergence tends to occur more rapidly for underestimated parameter sets. Iterations are reduced as the estimated parameter values approach the true values. For typical data sets the rms error tends to be only a few tenths of a foot, while for fairly idealized sets of data, the rms error is a few hundredths of a foot (see Table 2). Iterations can be reduced by increasing the size of the acceptable error criteria, but only at the cost of increased rms error.

We have extensively tested the algorithm on some synthetic data formulated by Cooper (1963) for an infinite leaky confined aquifer. Typical results appear in Table 1. Note that, for all permutations of three orders of magnitude above and below the correct values, the regression results are virtually identical, showing that the algorithm's convergence properties are not parameter specific. The rms error for these data is shown in Table 2 (data set 2a) to be .038 feet.

If the data diverge too much from ideal data, convergence may not occur. If convergence does occur the rms error may be unacceptable. Although this algorithm gives a unique solution to any data set for which it can achieve a converged set of values, it cannot distinguish absolutely between different types of aquifers. Since the three analytical degrees of freedom give the algorithm considerable latitude in achieving convergence, an imperfect data set from a confined aquifer may be successfully run and a set of values for transmissivity, storage, and leakage produced. This fact points to several cautions. First, only the best data available should be analyzed. Second, experienced personnel should carefully examine the geohydrology to aid in classifying the aquifer type. Third, if doubt exists about the validity of the converged values, the rms error value should be noted, but the individual "best fit" drawdowns should be compared to the field data for gross divergences.

APPLICATION OF THE ALGORITHM TO DATA SETS

For testing the leaky-aquifer regression-analysis program we had access to only a limited number of data sets. The sources of these data are listed in Appendix V. These data fall into three categories. The first category is synthetic data, that which is generated from the closed-form integral expres-

Table 1. "Best fit" parameters for various combinations of overestimation and underestimation of initial values. Values from Cooper (1963).

Type Curve Values: SC = .0001 KB = 13,300 ft^2/day LC = .0005 ft^{-1}

Initial-Guess Values			Converged Values r = 100 ft		
SC	KB (ft^2/day)	LC (1/ft)	SC	KB (ft^2/day)	LC (1/ft)
1×10^{-7}	13.3	4.98×10^{-7}	9.789×10^{-5}	13338.	4.9401×10^{-4}
1×10^{-1}	13.3×10^{6}	4.98×10^{-1}	9.789×10^{-5}	13338.	4.9404×10^{-4}
1×10^{-7}	13.3	4.98×10^{-1}	9.789×10^{-5}	13338.	4.9404×10^{-4}
1×10^{-7}	13.3×10^{6}	4.98×10^{-1}	9.789×10^{-5}	13338.	4.9404×10^{-4}
1×10^{-7}	13.3×10^{6}	4.98×10^{-7}	9.789×10^{-5}	13338.	4.9403×10^{-4}
1×10^{-1}	13.3	4.98×10^{-7}	9.789×10^{-5}	13338.	4.9402×10^{-4}
1×10^{-1}	13.3	4.98×10^{-1}	9.789×10^{-5}	13338.	4.9403×10^{-4}
1×10^{-1}	13.3×10^{6}	4.98×10^{-7}	9.789×10^{-5}	13338.	4.9403×10^{-4}

Table 2. Comparison of aquifer parameters for typical data sets obtained by graphical analysis and by automated analysis.

Data Source Code	Graphical-Analysis Values	Automated Analysis Values	Automated rms error
1	T = 182000 gpd/ft S = .002 B = 2500 ft	202000 gpd/ft .002 3300 ft	.007 ft
2* a		99000 .00097 20000	.038
2* b	T = 99400 S = .0001 B = 2000	100000 .000097 1980	.016
2* c		97800 .0001 1950	.010
2**	T = 99400 S = .0001 B = 2000	99026 .000099 1967	.046
3	T = 1500 S = .00020 B = 430	1800 .00017 650	.125
4 a	T = 49000 S = .000090 B = 4100	44000 .000086 3900	.378
4	T = 41000 S = .000080 B = 4000	46000 .000084 4800	.030

T = Transmissivity
S = Storage coefficient
B = Leakage coefficient

*The values obtained by graphical analysis represent an average of three sets of data taken for different values of radius. Each data set was independently analyzed and tabulated for the automated analysis.

**Same conditions as (*) except that the automated values represent a simultaneous solution of the three data sets.

sion for the leaky aquifer drawdown (or a numerical model), using a hypothetical set of aquifer parameters. Second, there is actual pumping test data that matches the expected shape of the theoretical leaky aquifer type curve for a given set of parameters. A third type of data used is that in which the time-drawdown curve is not clearly of the leaky artesian category, but for which the site geology suggests that a leaky situation may be occurring. This last category has typically been analyzed by classical methods for confined aquifers.

Table 2 is a compilation of data sets of the first and second types. The graphical analysis values are compared with corresponding automated-analysis values of the same data set. The automated rms error indicates the goodness of fit between the experimental data and the theoretical type curve. The "best" rms error indicated is for data set 1, while the worst is for set 4a. Data set 2 is interesting in that it has been analyzed in two ways. The graphical analysis simultaneously considered all data from three observation wells. Set 2* considered each observation well separately and produced three separate sets of automated values with associated rms errors. Note that the worst rms error (.038 ft) gives the closest numerical agreement with the graphic values. Next, at 2**, the three observation-well data sets were simultaneously analyzed by the automated routine. Numerical agreement between graphic and automated values is once again very close, but the rms error is greater yet at .046 ft. This is more than twice as much rms error as the average rms error of the three observation wells considered separately (≈.021 ft). This result tends to indicate that as more data are stacked together for simultaneous analysis, more "smear" is likely to appear in the automated analysis of that data. In effect the data are being averaged, but not in a strictly arithmetic fashion, as is indicated by the comparison of the rms average of 2* and the rms of 2**.

In summary, the principal feature of Table 2 is the quite good agreement between the "best fit" automated values and the type-curve values. All parameters have numerical values well within the same order of magnitude and in fact are not over 35 percent, most being in the 10-20 percent range. The fact that close numerical agreement between manual and automated values does not always produce the smallest rms error seems to be related to the sensitivities of the various parameters.

Table 3 is a comparison of parameter values derived from real data sets that were first evaluated by the confined artesian type-curve method and then by the automated, leaky artesian, regression algorithm. Although the rms errors are satisfactory, there are discrepancies of several orders of magni-

Table 3. "Best fit" leaky aquifer parameters compared with confined type curve analysis of data.

Data Source Code		Confined Aquifer Type Curve Values Obtained From Graphical Fit	Leaky Aquifer Values Obtained From Regression Fit	rms error
5		T = 44000 gpd/ft S = .00046 B = 0 ft	T = 42000 S = .00044 B = 8600	0.240
6	a	T = 42000 S = .000004 B = 0	T = 9800 S = .0045 B = 65	.036
6	b	T_1 = 48000, T_2 = 19500 S_1 = .0000065, S_2 = .002 B = 0	T = 25500 S = .00055 B = 1180	.032

T = Transmissivity
S = Storage coefficient
B = Leaky coefficient

tude in the storage values. This is especially true for examples 6a,b. Analysis 6b has two sets of S and T values derived from the same time-drawdown data, and the "best fit" S and T are the approximate averages of these values. These examples demonstrate the fact that non-ideal data can still achieve convergence in this algorithm. This points to the need to examine the geology of a site as well as the drawdown curve from an aquifer test. It also illustrates the uncertainty inherent in evaluating real field data. Some of this uncertainty is a direct result of inexactness in collection of the data. A second uncertainty rests in correctly understanding the physical situation of the local geology and hydrology. For example, besides being caused by leakage, steady-state conditions may arise from interception of a river-recharge boundary or from supplying of the total pump discharge by regional underflow. All three situations would yield a flattened curve, and each could be matched to some extent by this model, given some values of T, S, and L.

USING PROGRAM LEAKYFIT

LEAKYFIT is an algorithm that is run in batch mode. Data may be entered by cards or from a permfile when initiating the job from a remote interactive terminal. These files are in free-field format, where the data are separated by blanks or commas. In general, the algorithm is not in machine-dependent code. Only one statement (see Appendix I), CALL FXOPT (...), is specific to the Honeywell 66/60 now operating at The University of Kansas. This statement can be removed by a user implementing this algorithm on a "foreign" machine. The purpose of this statement is to circumvent program termination caused by exponential overflows, exponential underflows, and division checks.

There are some pre-assigned parameter values in the algorithm that the user may need to redefine from time to time. The size of arrays SE, T, SGS, R, and RB has been set, so as to handle a maximum of 100 time-drawdown pairs from a maximum of eight observation wells. If the number of observation wells (NOW) is greater than B and the number of time-drawdown data pairs (NTDP) is greater than 100, then the size of arrays needs to be readjusted. The parameter ITMAX controls the maximum number of iterations performed by the program. It is set at 50, since our experiments show that all the data sets tested converge in fewer than 50 iterations. However, if convergence does not occur, this parameter may be increased. The value of ERROR checks for convergence of the parameter updating quantities ΔU_T, ΔU_S, and ΔU_L. This value gives good fits with acceptable numbers of iterations. Changing this parameter is not recommended. Making it larger reduces number of iterations while decreasing the accuracy of the fit. Reducing its value has the opposite effect.

However, the number of time-drawdown data pairs or the data quality may not be sufficient to achieve convergence. At this stage, if one wishes to force the solution of the data, the converge criteria may be relaxed (i.e., the value of the ERROR may be increased anywhere from .001 to .01). This change in ERROR would force the data to converge with a high rms error, and the converged values of SC, KB, and LC would be the best approximate values of the limited data given. This strategy is probably advisable only if the user is convinced that the aquifer is really of the leaky artesian type.

Data input is straightforward. The first free-field format entry consists of the following parameters:

SC = Elastic storage coefficient of confined aquifer

KB = Transmissivity of confined aquifer

LC = Inverse leakage coefficient of the confining bed

NOW = Number of observation wells

R = Radial distance from the pumping well to the observation well

Q = Pumping rate

NTDP = Number of time-drawdown pairs

TCL = Thickness of the confining bed if known.

These parameters may have any consistent set of units associated with them. If TCL is known, the program can compute K', the vertical hydraulic conductivity of the confining bed. If TCL is unknown, entering zero will cause this calculation to be skipped.

The second and succeeding free-format fields contain the time-drawdown pairs, the number of which must be NTDP. Entering one pair at a time reduces possibilities of mistakes in punching or keying in data. All data read by the program are echoed as the first part of the output.

As iterations of the fitting algorithm are completed, the standard deviation (rms error) is printed along with the values of LC, SC, and KB. If the routine completes ITMAX iterations without converging, a message appears announcing the fact and the program is abnormally terminated. Successful convergence is similarly announced at termination. The best fit time-drawdown pairs are printed, followed by the converged values for LC, SC, and KB. If TCL has been given a positive value, the value of K' is also printed. The program then terminates.

One other feature that may be occasionally useful is the option IGENDATA (see Appendix I for a listing of LEAKYFIT). When this statement is set equal to zero (/0/), LEAKYFIT operates in a normal regression-type mode. Setting IGENDATA equal to one (/1/) causes the program to terminate after the first iteration and produces a set of values that define a portion of the theoretical leaky aquifer type curve. This is useful if the user wishes to check point by point the deviation of a converged set of values against the theoretical values.

A TYPICAL BATCH RUN OF PROGRAM LEAKYFIT

```
 SC=          0.00150000
KB=          1.00000000
LC=          0.00150000
NOW=               4
Q=          1.28399999
NTDP  =    4
ERROR=            0.00100000
ITMAX=         50

  R =
  0.30500000E 02  0.61000000E 02  0.12200000E 03  0.24000000E 03

              T                    SE

    0.100000E 01      1.1450    0.7450    0.3950    0.1350

    0.600000E 01      1.6000    1.1900    0.7900    0.4400

    0.435000E 02      2.1950    1.7700    1.3400    0.9600

    0.340000E 03      2.4850    2.0100    1.6000    1.1600

 THE STANDARD DEVIATION FOR ITERATION NUMBER,  1IS    2.32944140

 LC=      0.12000000E-02
SC=      0.12000000E-02
KB=      0.80000000E 00


 THE STANDARD DEVIATION FOR ITERATION NUMBER,  2IS    2.19945490

 LC=      0.18000000E-02
SC=      0.96000000E-03
KB=      0.64000000E 00


 THE STANDARD DEVIATION FOR ITERATION NUMBER,  3IS    2.08265403

 LC=      0.14400000E-02
SC=      0.76800000E-03
KB=      0.51200000E 00

                .                              .
                .                              .
                .                              .
```

THE STANDARD DEVIATION FOR ITERATION NUMBER, 21IS 0.06501416

LC= 0.63876567E-03
SC= 0.19800827E-04
KB= 0.33877176E 00

THE STANDARD DEVIATION FOR ITERATION NUMBER, 22IS 0.06500053

THE PARAMETERS CONVERGED IN 22 ITERATIONS.

THE BEST FIT TIME-DRAWDOWN PAIRS FOR THE CONVERGED VALUES OF S, T, AND L ARE

T		SE		
0.100000E 01	1.1244	0.7186	0.3471	0.0865
0.600000E 01	1.6511	1.2354	0.8265	0.4510
0.435000E 02	2.1758	1.7585	1.3432	0.9443
0.340000E 03	2.4370	2.0196	1.6038	1.2029

LEAKAGE COEFFICIENT = 0.00063880

STORAGE COEFFICIENT = 0.00001980

TRANSMISSIVITY = 0.33876759

PROGRAM TSSLEAK AND ITS USE

This program is simply an interactive time-sharing version of LEAKYFIT. A listing appears in Appendix II. Prompting statements query the user for values of storage (SC), transmissivity (KB), leakage (LC), pumping rate (Q), number of observation wells (NOW), radial observation distance(s) (R), and number of time-drawdown pairs (NTDP). All data are echo printed, as well as ITMAX and ERROR, and the user is given an opportunity to correct the input data. Next, the algorithm asks for time-drawdown pairs. Only NTDP pairs may be entered. After entry is complete, all pairs are echo printed and the user is again asked to examine for errors and correct any that are found. Correction is done by entering the sequential number of the erroneous time-drawdown pair and the correct values. After reading the corrections, the terminal again asks if there are any errors in the time-drawdown data. As long as affirmative responses are given by the user, the program will ask for data corrections. As soon as a negative response is given, the program proceeds.

At this point, output is identical to that printed by LEAKYFIT (as may be seen by comparing the sample runs). Values of the rms error, iteration number, and LC, SC, and KB are printed at the end of each iterative step. Final convergence status is announced. The successful "best fit" time-drawdown pairs are printed and the converged values of LC, SC, and KB are printed. The program then asks if the thickness of the semiconfining bed is known. If an affirmative response is made, the value of that parameter is asked, and the value of the leaky permeability K' is printed. The program then terminates.

A TYPICAL INTERACTIVE TIMESHARING SESSION WITH TSSLEAK

```
ESTIMATE FOR STORAGE ?
=0.0015
ESTIMATE FOR TRANSMISSIVITY? L**2/T
=1.0
ESTIMATE FOR LEAKAGE COEFFICIENT ? 1/L
=0.0015
CONSTANT PUMPAGE RATE? L**3/T
=1.284
NUMBER OF OBSERVATION WELLS ?
=4
OBSERVATION DISTANCE FROM PUMPING WELL? L
=30.5 61.0 122.0 240.0
NUMBER OF TIME-DRAWDOWN PAIRS TO BE READ?
=4
ECHO THE INITIAL DATA
SC=         0.00150000
KB=         1.00000000
LC=         0.00150000
Q=         1.28399999
NO. OF OBS. WELLS(NOW) =   4
NTDP=        4
ITMAX=       60
ERROR=         0.00100000
RADIAL DISTANCES OF OBSERVATION WELLS ARE :
  30.5000      61.0000      122.0000      240.0000
ARE THERE ANY ERRORS IN DATA INPUT?
ANSWER YES IF ANY ERROR, OTHERWISE NO
=NO
TYPE IN TIME-DRAWDOWN PAIRS IN ORDER OF INCREASING TIME.
=1 1.1450 0.7450 0.3950 0.1350
=6 1.6000 1.1900 0.7900 0.4400
=43.5 2.1950 1.7700 1.3400 0.9600
=340 2.4850 2.0100 1.6000 1.1600
THE PUMP TEST DATA IN  TIME DRAWDOWN PAIRS
  0.100000E 01      1.1450     0.7450     0.3950     0.1350
  0.600000E 01      1.6000     1.1900     0.7900     0.4400
  0.435000E 02      2.1950     1.7700     1.3400     0.9600
  0.340000E 03      2.4850     2.0100     1.6000     1.1600
```

```
ARE THERE ANY ERRORS IN TIME-DRAWDOWN PAIRS?
ANSWER YES OR NO
=NO

THE STANDARD DEVIATION FOR ITERATION NUMBER,    1IS           2.32944140

LC=       0.12000000E-02
SC=       0.12000000E-02
KB=       0.80000000E 00


THE STANDARD DEVIATION FOR ITERATION NUMBER,    2IS           2.19945490

LC=       0.18000000E-02
SC=       0.96000000E-03
KB=       0.64000000E 00

              . . .

THE STANDARD DEVIATION FOR ITERATION NUMBER,   21IS           0.06501416

LC=       0.63876567E-03
SC=       0.19800827E-04
KB=       0.33877176E 00


THE STANDARD DEVIATION FOR ITERATION NUMBER,   22IS           0.06500053
THE PARAMETERS CONVERGED IN   22 ITERATIONS.

THE BEST FIT TIME-DRAWDOWN PAIRS FOR THE CONVERGED VALUES OF S, T, AND
L ARE

            T                    SE
  0.100000E 01      1.1244    0.7186    0.3471    0.0865
  0.600000E 01      1.6511    1.2354    0.8265    0.4510
  0.435000E 02      2.1758    1.7585    1.3432    0.9443
  0.340000E 03      2.4370    2.0196    1.6038    1.2029

LEAKAGE COEFFICIENT =           0.00063880

STORAGE COEFFICIENT =           0.00001980

TRANSMISSIVITY =           0.33876759
DO YOU WANT TO COMPUTE AQUITARD PERMEABILITY ?
ANSWER YES IF TCL IS KNOWN OTHERWISE NO
=YES
THICKNESS OF CONFINING LAYER ?
=30
AQUITARD PERMEABILITY =          0.00000415
```

PROGRAM HANTUSH AND ITS USE

The algorithm titled HANTUSH is an interactive time-sharing program that solves the drawdown equation for a leaky confined aquifer as formulated by Jacob and Hantush in 1955. Figure 1 shows the definition of the appropriate aquifer system. A complete listing of this program appears in Appendix III. The input parameters for this program are the discharge (Q), the inverse leakage coefficient (LC), the confined aquifer transmissivity (KB), the elastic storage coefficient (SC), the unit length designation (LU), the unit time designator (TU), the observation radial distance (R), and the pumping period (T). The parameter LC is simply the inverse of B as defined by Jacob and Hantush (1955).

The program introduces itself with a short explanation of its function and queries the user for the data indicated above. At critical points in the program all important data are echo printed and the user is given an opportunity to correct any errors. Corrections may be made in several ways. At points of direct questioning, appropriate responses (YES or NO) will determine whether or not corrections will be made. In other locations, data may simply be re-entered when the user recognizes an error. The program may be formally exited by entering zeros for R and T at any read statement, or simply by using the BREAK key.

An example run is included here as a tutorial. This run was made in TSS FORTRAN as compiled on the Honeywell 66/60 at the University of Kansas Computing Center. Access and run commands shown here may or may not be meaningful on other systems. All queries (data-reading locations) are indicated by "equals" signs (=). The user enters the appropriate data string in free format following the prompt symbol. If the user fails to enter all the neccessary data in the first string, the prompting symbol will continue to appear until all requested data have been read. Free format means that data fields are not of specific size or location, but are entered in a sequential fashion, each field separated by a blank space or a comma. The responses of the user are underlined in the example.

The obvious advantage of this program is the accuracy of the value of W(U,r/B) as compared to interpolating values from tables. This is especially true if the table is sparse in its value range of U and r/B. Table 4 shows drawdowns computed from exact and interpolated values of W(U,r/B). These values are compared to drawdowns computed by the program. Note that for the exact table values (Table 5), the hand-computed and algorithm-computed values are virtually identical, while there are considerable differences in the

Table 4. Tables of drawdowns for interpolated* values U, r/B, and W(U,r/B).

Drawdown Interpolated from Table				Drawdown Computed by Program			
U	r/B	W(U,r/B)	s	U	r/B	W(U,r/B)	s
4.41	0.9392	0.0288	0.1146	4.4100	0.9392	.0022	0.009
0.7350	0.9392	0.3194	1.2708	.7350	0.9392	0.2896	1.152
0.5625	0.3354	0.5007	1.9922	0.5625	0.3354	0.4760	1.894
1.3496	0.3354	0.1965	0.7818	1.3496	0.3354	0.1236	0.492

*A linear interpolation scheme is used.

Table 5. Tables of drawdowns for exact table values of U, r/B, and W(U,r/B).

Drawdown From Exact Table Values				Drawdown From Program			
U	r/B	W(U,r/B)	s	U	r/B	W(U,r/B)	s
0.05	0.2	2.3110	69.883	.05	.20	2.3110	69.884
0.01	0.6	1.5550	9.405	.01	.5999	1.5551	9.405
0.0001	0.03	7.2122	0.436	.00010	.0300	7.2123	0.435

interpolated values. The table used is from Walton (1970) and a linear interpolation scheme is used. A more sophistocated interpolation scheme might reduce the discrepancy in computed drawdown.

Another application of this algorithm, which has not been initiated as of this writing, is its use as the core of a leaky confined well-field simulator. Well-field simulators for confined aquifers are common, but no such simulator is known to exist for the leaky artesian case.

A TYPICAL INTERACTIVE TIMESHARING SESSION WITH HANTUSH

```
THIS PROGRAM CALCULATES THE DRAWDOWN IN A
LEAKY ARTESIAN AQUIFER.  THE RADIAL DISTANCE FOR THE
OBSERVATION WELL AND THE PUMPING PERIOD MAY BE CONSTANT
OR VARIABLE.  ANY CONSISTENT SET OF UNITS MAY BE USED.

ENTER THE FOLLOWING DATA IN A FREE FORMAT FIELD:
Q = PUMPING RATE (L**3/T)
LC = INVERSE LEAKAGE COEFFICIENT (1/L)
KB = AQUIFER TRANSMISSIVITY (L**2/T)
SC = STORAGE COEFFICIENT (UNITLESS)
LU = UNIT OF LENGTH (3 CHARACTERS MAX)
TU = UNIT OF TIME (3 CHARACTERS MAX)
=5.E4
=750
=8.E3
=.003
=FT
=MIN

ARE THERE ANY ERRORS IN THE ABOVE ENTRIES?
IF NOT, ANSWER NO.
=YES

ENTER THE FOLLOWING DATA IN A FREE FORMAT FIELD:
Q = PUMPING RATE (L**3/T)
LC = INVERSE LEAKAGE COEFFICIENT (1/L)
KB = AQUIFER TRANSMISSIVITY (L**2/T)
SC = STORAGE COEFFICIENT (UNITLESS)
LU = UNIT OF LENGTH (3 CHARACTERS MAX)
TU = UNIT OF TIME (3 CHARACTERS MAX)
=5.E4
=.00133333
=8.E3
=.003
=FT
=DAY

ARE THERE ANY ERRORS IN THE ABOVE ENTRIES?
IF NOT, ANSWER NO.
=NO

THE FOLLOWING PARAMETERS ARE USED IN THIS SOLUTION:

Q=  50000.00

LC=0.00133333

KB=   8000.00

SC=0.00300000

LU =FT
```

```
TU =DAY

ENTER THE FOLLOWING DATA:
R,T
R = DISTANCE TO OBSERVATION WELL (L)
T = LENGTH OF PUMPING PERIOD (T)
TERMINATE BY  RETURNING BLANK FIELDS.

ENTER R,T
=100.,.05

U=  0.018750

R/B=   0.13333

W(U,R/B)=  3.213409

THE DRAWDOWN   100. FT
FROM THE PUMPING WELL IS    1.59822 FT
AFTER       0.05 DAY OF PUMPING.

ENTER R,T
=50.,.5

U=  0.000469

R/B=   0.06667

W(U,R/B)=  5.626895

THE DRAWDOWN    50. FT
FROM THE PUMPING WELL IS    2.79859 FT
AFTER       0.50 DAY OF PUMPING.

ENTER R,T
=100.,.5

U=  0.001875

R/B=   0.13333

W(U,R/B)=  4.259999

THE DRAWDOWN   100. FT
FROM THE PUMPING WELL IS    2.11875 FT
AFTER       0.50 DAY OF PUMPING.

ENTER R,T
=0.,0.

YOU HAVE TERMINATED THE PROGRAM.
IS THIS CORRECT?
=YES
```

DISCUSSION AND SUMMARY

In this paper we have discussed the development and application of an automated-fitting routine to determine the parameters of a simple leaky artesian aquifer using sensitivity theory to implement a least-squares-fitting procedure. This has allowed the fitting of a theoretical curve to experimental data, yielding not only the aquifer parameters, but also a meaningful estimate of the "goodness of fit." This is typically a quick and economical process, usually producing results in less than 40 iterations at a cost of only a few dollars.

It should be reiterated here that, although the algorithm is very reliable for smooth data that are a good approximation to typical leaky-type data, it can also be used on data that are less than perfect. A converged solution can sometimes be achieved, and may be meaningful in the sense that the data may represent a leaky situation not fully corresponding to the model assumed in this work. Thus, several cautions need to be restated here. First, only the most accurate and complete data available should be used. Second, the geohydrology should be carefully examined by experienced personnel to aid in classifying the aquifer type. Third, if any doubt exists about the validity of the converged values, the rms error value should be noted and individual best-fit drawdowns should be compared to the field data for gross deviations.

As data collection becomes more automated, programs such as this become more attractive for the simple reason that, except for possible preview of values by the hydrologist, the entire pumping test process may be done by computers with no intermediate "hard copy." Thus, the hydrologist may become more efficient and at the same time will not be replaced by the computer.

REFERENCES

Abramowitz, M., and Stegun, I. A., 1968, Handbook of mathematical functions: New York, Dover Publications, p. 231.

Cobb, P. M., McElwee, C. D., and Butt, M. A., 1978, Leaky aquifer parameter identification by sensitivity analysis: Transactions of the American Geophysical Union, v. 60, no. 6, p. 67.

Cooper, H. H., Jr., 1963, Type curves for nonsteady radial flow in an infinite leaky artesian aquifer, in Bentall, R., compiler, Short-cuts and special problems in aquifer tests: U.S. Geological Survey Water-Supply Paper 1545-C, p. C48-C55.

Hantush, M. S., 1960, Modification of the theory of leaky aquifers: Journal of Geophysical Research, v. 65, no. 11, p. 3713-3725.

Hantush, M. S., and Jacob, C. E., 1955, Non-steady radial flow in an infinite leaky aquifer: Transactions of the American Geophysical Union, v. 36, p. 95-100.

Hildebrand, F. B., 1962, Advanced calculus for applications: Englewood Cliffs, Prentice-Hall, p. 360.

Jacob, C. E., 1946, Radial flow in a leaky artesian aquifer: Transactions of the American Geophysical Union, v. 27, p. 198-208.

McCuen, R. H., 1973, Component sensitivity: a tool for the analysis of complex water resources systems: Water Resources Research, v. 9, no. 1, p. 243-246.

McElwee, C. D., 1980a, Theis parameter evaluation from pumping tests by sensitivity analysis: Ground Water, v. 18, p. 56-60.

McElwee, C. D., 1980b, The Theis equation: evaluation, sensitivity to storage and transmissivity, and automated fit of pump test data: Kansas Geological Survey Groundwater Series 3, 39 p.

McElwee, C. D., and Yukler, M. A., 1978, Sensitivity of groundwater models with respect to variations in transmissivity and storage: Water Resources Research, v. 14, no. 3, p. 451.

Neuman, S. P., and Witherspoon, P. A., 1969a, Theory of flow in a confined two aquifer system: Water Resources Research, v. 5, no. 4, p. 803-816.

Neuman, S. P., and Witherspoon, P. A., 1969b, Applicability of current theories of flow in leaky aquifers: Water Resources Research, v. 5, no. 4, p. 817-829.

Tomovic, R., 1962, Sensitivity analysis of dynamic systems: New York, McGraw-Hill, 141 p.

Vemuri, V., Dracup, J. A., Erdmann, R. C., and Vemuri, N., 1969, Sensitivity analysis method of system identification and its potential in hydrologic research: Water Resources Research, v. 5, no. 2, p. 341-349.

Walton, W. C., 1970, Groundwater resource evaluation: New York, McGraw-Hill, 664 p.

Yukler, M. A., 1976, Analysis of error in groundwater modeling, Ph.D. Dissertation, The University of Kansas, 182 p.

APPENDIX I. PROGRAM LEAKYFIT

```
C        PROGRAM  LEAKYFIT
      PARAMETER     DECRES = -.2
      PARAMETER     INCRES = .5
      COMMON        LC, ITERAT, LCOUNT, KB, KBCOUNT
      DIMENSION     X(3), SE(100, 8), T(100), SGS(100, 8), R(8), RB(8)
      REAL          KB, LC, INCRES
      PI = 3.1415926
      DATA          IGENDAT/0/
C     Q=PUMPAGE (L**3/T)
C     SC=STORAGE COEFF. (UNITLESS)
C     KB= TRANSMISSIVITY   (L**2/T)
C     LC=MODIFIED COEFFICIENT OF LEAKAGE (1/L)
C     ERROR= CONVERGENCE CRITERIA FOR MAIN DO LOOP (UNITLESS)
C     R= RADIAL DISTANCE FROM PUMPING WELL TO OBSERVATION WELL(L)
C     NOW=NUMBER OF OBSERVATION WELLS .
C     NTDP=NUMBER OF TIME DRAWDOWN PAIRS TO BE READ .
C     EPS=  CONVERGENCE CRITERIA FOR SUBROUTINE SIMUL (UNITLESS)
C     ITMAX= MAX NUMBER OF ITERATIONS (UNITLESS)
C     T=  TIME (T)
C     SE=  EXPERIMENTAL DRAWDOWN (L)
C      TCL=THICKNESS OF CONFINING LAYER (L)
C     INITIALIZE PROGRAM
      N = 3
      ERROR = 0.001
      ITMAX = 50
      ITERAT = 0
      CALL          FXOPT (89, 1, 1, 0)
      READ (5, 20) SC, KB, LC, NOW, (R(I), I = 1, NOW), Q, NTDP, TCL
      DO 10 I = 1, NTDP
        READ (5, 20) T(I), (SE(I, J), J = 1, NOW)
   10 CONTINUE
   20 FORMAT (V)
      WRITE (6, 30) SC, KB, LC, NOW, Q, NTDP, ERROR, ITMAX
   30 FORMAT (1H1,'SC=',F20.10/'KB=',F20.10/'LC=',F20.10/'NOW=
     & ',I5/'Q=', F20.10/ 'NTDP  =', I5/ 'ERROR=', F20.10/
     & 'ITMAX=', I10)
      WRITE (6, 40)
   40 FORMAT ( 1H0,4H R =)
      WRITE (6, 50)(R(I), I = 1, NOW)
   50 FORMAT (V)
      IF (TCL .GT. 0) WRITE (6, 60) TCL
   60 FORMAT ( 'TCL=', F20.10)
C     ECHO PRINT TIME - DRAWDOWN DATA PAIRS
      WRITE (6, 70)
   70 FORMAT (1H0,T15,1HT ,T35,2HSE)
      DO 80 I = 1, NTDP
        WRITE (6, 90) T(I), (SE(I, J), J = 1, NOW)
   80 CONTINUE
   90 FORMAT (1H0,E15.6,2X,8(F10.4))
      IF (IGENDAT .EQ. 1) GO TO 140
      LCOUNT = 0
      KBCOUNT = 0
```

```
  100 CONTINUE
  110 CONTINUE
      IF (KBCOUNT .GT. 0 .OR. LCOUNT .GT. 0) ITERAT = ITERAT-1
      LCOUNT = 0
      KBCOUNT = 0
C   INITIALIZE ITERATIONS
  120 ITERAT = ITERAT+1
C     ZERO OUT SUMMATIONS
      SDELS2 = 0.0
      SUSCDS = 0.0
      SURCDS = 0.0
      SUKBDS = 0.0
      SUKBUS = 0.0
      SURCUK = 0.0
      SUSCUR = 0.0
      SURC2 = 0.0
      SUSC2 = 0.0
      SUKB2 = 0.0
C     ZERO OUT  X MATRIX
      DO 130 K = 1, N
        X(K) = 0.0
  130 CONTINUE
C     COMPUTE RB
  140 DO 150 I = 1, NOW
        RB(I) = R(I)*LC
  150 CONTINUE
      DO 200 I = 1, NTDP
        DO 190 J = 1, NOW
C     COMPUTE U
          U = (R(J)*R(J)*SC)/(4*KB*T(I))
C     COMPUTE THEORETICAL DRAWDOWN
          SG = (Q/(4*PI*KB))*W(U, RB(J))
          IF (LCOUNT .EQ. 0 .AND. KBCOUNT .EQ. 0) GO TO 160
          GO TO 100
  160     SGS(I, J) = SG
          IF (IGENDAT .EQ. 1) GO TO 200
C     COMPUTE DIFFERENCE BETWEEN THEORETICAL AND EXPERIMENTAL DRAWDOWN
          DELS = SE(I, J)-SG
          IF (ABS(DELS) .LT. 1.0E-3) DELS = 0.0
          SDELS2 = SDELS2+DELS*DELS
C     COMPUTE DUMMY COEFFICIENT 'Z'
          Z = (U+(RB(J)*RB(J))/(4.*U))
C     COMPUTE SENSITIVITY COEFFICIENT AND SUMMATIONS
          USC = -(Q/(4.*PI*KB)*(1/U)*((R(J)*R(J))/(4.*KB*T(I)))*EXP(-
     & Z))
          UKB = -SG/KB+(Q/(4.*PI*KB))*((R(J)*R(J)*SC)/
     & (4.*KB*KB*T(I)))*(1/U)*EXP(-Z)
          IF (USC .EQ. 0 .OR. UKB .EQ. 0) ITERAT = ITERAT-1
          IF (USC .EQ. 0 .OR. UKB .EQ. 0) GO TO 230
          RBM = R(J)*.99*LC
          RBP = R(J)*1.01*LC
          WPLUS = W(U, RBP)
          IF (LCOUNT .EQ. 0 .AND. KBCOUNT .EQ. 0) GO TO 170
          GO TO 100
```

```
 170      WMINUS = W(U, RBM)
          IF (LCOUNT .EQ. 0 .AND. KBCOUNT .EQ. 0) GO TO 180
          GO TO 100
 180      URC = (Q/(4.0*PI*KB))*(WPLUS-WMINUS)/(.02*LC)
          IF (URC .EQ. 0) ITERAT = ITERAT-1
          IF (URC .EQ. 0) GO TO 230
          SUKB2 = SUKB2+UKB*UKB
          SUSC2 = SUSC2+USC*USC
          SURC2 = SURC2+URC*URC
          SUSCUR = SUSCUR+USC*URC
          SUKBUS = SUKBUS+UKB*USC
          SURCUK = SURCUK+URC*UKB
          SUSCDS = SUSCDS+USC*DELS
          SURCDS = SURCDS+URC*DELS
          SUKBDS = SUKBDS+UKB*DELS
 190    CONTINUE
 200  CONTINUE
      IF (IGENDAT .EQ. 0) GO TO 220
      WRITE (6, 70)
      DO 210 I = 1, NTDP
 210  WRITE (6, 90) T(I), (SGS(I, J), J = 1, NOW)
      IF (IGENDAT .EQ. 1) GO TO 380
C     COMPUTE MATRIX TO BE SOLVED FOR SENSITIVITY DELTAS
 220  U11 = SURC2*LC
      U12 = SUSCUR*SC
      U13 = SURCUK*KB
      U14 = SURCDS
      U21 = SUSCUR*LC
      U22 = SUSC2*SC
      U23 = SUKBUS*KB
      U24 = SUSCDS
      U31 = SURCUK*LC
      U32 = SUKBUS*SC
      U33 = SUKB2*KB
      U34 = SUKBDS
C      SOLVE MATRIX BY DIRECT GAUSS ELIMINATION
      X(3) = ((U14*U21-U24*U11)*(U12*U31-U32*U11)
     & -(U14*U31-U34*U11)*(U12*U21-U22*U11))/
     & ((U13*U21-U23*U11)*(U12*U31-U32*U11)
     & -(U13*U31-U33*U11)*(U12*U21-U22*U11))
      X(2) = ((U14*U21-U24*U11)-(U13*U21-U23*U11)*X(3))/
     & (U12*U21-U22*U11)
      X(1) = (U14-U13*X(3)-U12*X(2))/U11
      GO TO 240
 230  CONTINUE
C    UPDATE INITIAL GUESS VALUES OF  SC, KB, LC
      X(1) = INCRES
      LC = LC*(1.0+X(1))
      X(2) = INCRES
      SC = SC*(1.0+X(2))
      IF (SC.GE.1) SC = 1.0
      X(3) = INCRES
      KB = KB*(1.0+X(3))
      GO TO 260
```

```
  240 CONTINUE
C       COMPUTE STANDARD DEVIATION FOR EACH ITERATION
      SIGMA = SQRT(SDELS2/NTDP)
      WRITE (6, 250) ITERAT, SIGMA
  250 FORMAT (1H0,'THE STANDARD DEVIATION FOR ITERATION NUMBER,
     & 'I5,'IS', F20.10)
C      UP-DATE COEFFICIENTS
  260 CONTINUE
      IF (X(1) .LT. DECRES) X(1) = DECRES
      IF (X(1) .GT. INCRES) X(1) = INCRES
      LC = LC*(1.0+X(1))
      IF (X(2) .LT. DECRES) X(2) = DECRES
      IF (X(2) .GT. INCRES) X(2) = INCRES
      SC = SC*(1.0+X(2))
      IF (SC .GE.1) SC = 1
      IF (X(3) .LT. DECRES) X(3) = DECRES
      IF (X(3) .GT. INCRES) X(3) = INCRES
      KB = KB*(1.0+X(3))
C      CHECK FOR DELTA CONVERGENCE
      IF (ABS(X(1)) .GT. ERROR.
     & OR.ABS(X(2)) .GT. ERROR.
     & OR.ABS(X(3)) .GT. ERROR) GOTO 270
      GO TO 290
  270 IF (ITERAT.GE.ITMAX) GO TO 360
      IF (URC .EQ. 0 .OR. USC .EQ. 0 .OR. UKB .EQ. 0) GO TO 120
      WRITE (6, 280) LC, SC, KB
  280 FORMAT (1H0,3HLC=,E20.10/3HSC=,E20.10/3HKB=,E20.10//)
      GO TO 120
C       WRITE OUT FINAL PROGRAM STATUS
  290 WRITE (6, 300) ITERAT
  300 FORMAT (1H1,'THE PARAMETERS CONVERGED IN',I5,
     & 1X, 'ITERATIONS.')
      WRITE (6, 310)
  310 FORMAT (1H0,37HTHE BEST FIT TIME-DRAWDOWN PAIRS FOR ,
     & 39HTHE CONVERGED VALUES OF S, T, AND L ARE)
      WRITE (6, 70)
      DO 320 I = 1, NTDP
  320 WRITE (6, 90) T(I), (SGS(I, J), J = 1, NOW)
      WRITE (6, 330) LC, SC, KB
  330 FORMAT (1H0,22HLEAKAGE COEFFICIENT = ,F20.10,
     & //22HSTORAGE COEFFICIENT = ,F20.10,
     & //17HTRANSMISSIVITY = ,F20.10)
      IF (TCL .GT. 0) GO TO 340
      STOP
  340 AK = KB*TCL*LC**2
      WRITE (6, 350) AK
  350 FORMAT (1H0,'AQUITARD PERMEABILITY=',F20.10)
      STOP
  360 WRITE (6, 370) ITMAX
  370 FORMAT (1H0,'THE PROGRAM DID NOT CONVERGE IN ',I5,
     & 1X, 'ITERATIONS.')
  380 STOP
      END
C       FILE WURB
```

APPENDIX II.   PROGRAM TSSLEAK

```
C        PROGRAM  TSSLEAK
      PARAMETER    DECRES = -.2
      PARAMETER    INCRES = .5
      COMMON       LC, ITERAT, LCOUNT, KB, KBCOUNT
      DIMENSION    X(3), SE(100, 8), T(100), SGS(100, 8), R(8), RB(8)
      REAL         KB, LC
      CHARACTER    *3  CHEKDATA, WRITAQP
      PI = 3.1415926
      DATA         IGENDAT/0/
C     Q=PUMPAGE (L**3/T)
C     SC=STORAGE COEFF. (UNITLESS)
C     KB= TRANSMISSIVITY   (L**2/T)
C     LC=MODIFIED COEFFICIENT OF LEAKAGE (1/L)
C     ERROR= CONVERGENCE CRITERIA FOR MAIN DO LOOPE (UNITLESS)
C     R= RADIAL DISTANCE FROM PUMPING WELL TO OBSERVATION WELL(L)
C     NOW= NUMBER OF OBSERVATION WELLS .
C     NTDP= NUMBER OF TIME DRAWDOWN PAIRS TO BE READ .
C     EPS=  CONVERGENCE CRITERIA FOR SUBROUTINE SIMUL (UNITLESS)
C     ITMAX= MAX NUMBER OF ITERATIONS (UNITLESS)
C     T=  TIME
C     SE=  EXPERIMENTAL DRAWDOWN (L)
C     TCL=THICKNESS OF CONFINING LAYER (L)
C     INITIALIZE PROGRAM
      N = 3
      ERROR = 0.001
      ITMAX = 60
      ITERAT = 0
      CALL         FXOPT (89, 1, 1, 0)
C       READ IN THE INITIAL  DATA
   10 PRINT 20
   20 FORMAT ( 'ESTIMATE FOR STORAGE ?')
      READ:SC
      PRINT 30
   30 FORMAT ( 'ESTIMATE FOR TRANSMISSIVITY? L**2/T')
      READ:KB
      PRINT 40
   40 FORMAT ( 'ESTIMATE FOR LEAKAGE COEFFICIENT ? 1/L')
      READ:LC
      PRINT 50
   50 FORMAT ( 'CONSTANT PUMPAGE RATE? L**3/T ')
      READ:Q
      PRINT 60
   60 FORMAT ( 'NUMBER OF OBSERVATION WELLS ?')
      READ: NOW
      PRINT 70
   70 FORMAT ( 'OBSERVATION DISTANCE FROM PUMPING WELL? L')
      READ :(R(I), I = 1, NOW)
      PRINT 80
   80 FORMAT ( 'NUMBER OF TIME-DRAWDOWN PAIRS TO BE READ?')
      READ:NTDP
C      ECHO PRINT THE INITIAL DATA
      WRITE (6, 90) SC, KB, LC, Q, NOW, NTDP, ITMAX, ERROR
```

```
   90 FORMAT ( ' ECHO THE INITIAL DATA '/ 'SC=', F20.10/ 'KB='F20.10/
&        'LC=', F20.10/ 'Q=', F20.10/
&        ' NO. OF OBS. WELLS(NOW) = ', I3/ 'NTDP=', I10/
&        'ITMAX=', I10/ 'ERROR=', F20.10)
         PRINT 100
  100 FORMAT ( 'RADIAL DISTANCES OF OBSERVATION WELLS ARE : ')
         WRITE (6, 110)(R(I), I = 1, NOW)
  110 FORMAT (8(F10.4, 3X))
C        CHECK FOR THE DATA INPUT
         PRINT 120
  120 FORMAT ( 'ARE THERE ANY ERRORS IN DATA INPUT?', /,
&        'ANSWER YES IF ANY ERROR, OTHERWISE NO')
         READ:CHEKDATA
         IF (CHEKDATA .EQ. 3HYES) GO TO 10
C        TYPE IN DRAWDOWN-TIME PAIRS IN ORDER OF INCREASING TIME
         PRINT 130
  130 FORMAT ( 'TYPE IN TIME-DRAWDOWN PAIRS IN ORDER OF INCREASING TIME.')
         DO 140 I = 1, NTDP
  140 READ:T(I), (SE(I, J), J = 1, NOW)
C        ECHO PRINT THE TIME-DRAWDOWN PAIRS
         PRINT 150
  150 FORMAT ( 'THE PUMP TEST DATA IN  TIME DRAWDOWN PAIRS ')
         DO 160 I = 1, NTDP
  160 WRITE (6, 170) T(I), (SE(I, J), J = 1, NOW)
  170 FORMAT (E15.6, 2X, 8(F10.4))
C          CHECK FOR ANY ERRORS IN THE DATA INPUT
         NEWDATA = 0
  180 PRINT 190
  190 FORMAT ( 'ARE THERE ANY ERRORS IN TIME-DRAWDOWN PAIRS?', /,
&        'ANSWER YES OR NO')
         READ:CHEKDATA
         IF (CHEKDATA .EQ. 2HNO .AND. NEWDATA .EQ. 0) GO TO 250
         IF (CHEKDATA .EQ. 2HNO .AND. NEWDATA.GE.1) GO TO 230
         IF (CHEKDATA .EQ. 3HYES) GO TO 210
         IF (CHEKDATA .NE. 3HYES .OR. CHEKDATA .NE. 2HNO) PRINT 200
  200 FORMAT ( 'YOU HAVE AN ERROR IN DATA ENTRY')
  210 PRINT 220
  220 FORMAT ( 'ENTER THE LINE NUMBER (I),CORRECT TIME',
&      ',LOCATION (J) AND CORRECT DRAWDOWN')
         READ: I, T(I), J, SE(I, J)
         NEWDATA = NEWDATA+1
         GO TO 180
C          ECHO PRINT THE CORRECTED TIME DRAWDOWN PAIRS.
  230 DO 240 I = 1, NTDP
  240 WRITE (6, 170) T(I), (SE(I, J), J = 1, NOW)
  250 IF (IGENDAT .EQ. 1) GO TO 300
         LCOUNT = 0
         KBCOUNT = 0
  260 CONTINUE
  270 CONTINUE
         IF (KBCOUNT .GT. 0 .OR. LCOUNT .GT. 0) ITERAT = ITERAT-1
         LCOUNT = 0
         KBCOUNT = 0
C    INITIALIZE ITERATIONS
```

```
  280 ITERAT = ITERAT+1
C     ZERO OUT SUMMATIONS
      SDELS2 = 0.0
      SUSCDS = 0.0
      SURCDS = 0.0
      SUKBDS = 0.0
      SUKBUS = 0.0
      SURCUK = 0.0
      SUSCUR = 0.0
      SURC2 = 0.0
      SUSC2 = 0.0
      SUKB2 = 0.0
C     ZERO OUT  X MATRIX
      DO 290 K = 1, N
        X(K) = 0.0
  290 CONTINUE
C     COMPUTE RB
  300 DO 310 I = 1, NOW
  310 RB(I) = R(I)*LC
      DO 360 I = 1, NTDP
        DO 350 J = 1, NOW
C     COMPUTE U
          U = (R(J)*R(J)*SC)/(4*KB*T(I))
C     COMPUTE THEORETICAL DRAWDOWN
          SG = (Q/(4*PI*KB))*W(U, RB(J))
          IF (LCOUNT .EQ. 0 .AND. KBCOUNT .EQ. 0) GO TO 320
          GO TO 260
  320     SGS(I, J) = SG
          IF (IGENDAT .EQ. 1) GO TO 360
C     COMPUTE DIFFERENCE BETWEEN THEORETICAL AND EXPERIMENTAL DRAWDOWN
          DELS = SE(I, J)-SG
          IF (ABS(DELS) .LT. 1.0E-3) DELS = 0.0
          SDELS2 = SDELS2+DELS*DELS
C     COMPUTE DUMMY COEFFICIENT 'Z'
          Z = (U+(RB(J)*RB(J))/(4.*U))
C     COMPUTE SENSITIVITY COEFFICIENT AND SUMMATIONS
          USC =-(Q/(4.*PI*KB)*(1/U)*((R(J)*R(J))/(4.*KB*T(I)))*EXP(-Z))
          UKB = -SG/KB+(Q/(4.*PI*KB))*((R(J)*R(J)*SC)/
&     (4.*KB*KB*T(I)))*(1/U)*EXP(-Z)
          IF (ABS(USC) .LT. 1.E-30 .OR. ABS(UKB) .LT. 1.E-30) ITERAT =
&     ITERAT- 1
          IF (ABS(USC) .LT. 1.E-30 .OR. ABS(UKB) .LT. 1.E-30) GO TO 400
          RBM = R(J)*.99*LC
          RBP = R(J)*1.01*LC
          WPLUS = W(U, RBP)
          IF (LCOUNT .EQ. 0 .AND. KBCOUNT .EQ. 0) GO TO 330
          GO TO 260
  330     WMINUS = W(U, RBM)
          IF (LCOUNT .EQ. 0 .AND. KBCOUNT .EQ. 0) GO TO 340
          GO TO 260
  340     URC = (Q/(4.0*PI*KB))*(WPLUS-WMINUS)/(.02*LC)
          IF (ABS(URC) .LT. 1.E-30) ITERAT = ITERAT-1
          IF (ABS(URC) .LT. 1.E-30) GO TO 400
          SUKB2 = SUKB2+UKB*UKB
```

```
            SUSC2 = SUSC2+USC*USC
            SURC2 = SURC2+URC*URC
            SUSCUR = SUSCUR+USC*URC
            SUKBUS = SUKBUS+UKB*USC
            SURCUK = SURCUK+URC*UKB
            SUSCDS = SUSCDS+USC*DELS
            SURCDS = SURCDS+URC*DELS
            SUKBDS = SUKBDS+UKB*DELS
  350     CONTINUE
  360 CONTINUE
      IF (IGENDAT .EQ. 0) GO TO 390
      WRITE (6, 370)
  370 FORMAT (1H0,T15,1HT ,T35,2HSE)
      DO 380 I = 1, NTDP
  380 WRITE (6, 170) T(I), (SGS(I, J), J = 1, NOW)
      IF (IGENDAT .EQ. 1) STOP
C     COMPUTE MATRIX TO BE SOLVED FOR SENSITIVITY DELTAS
  390 U11 = SURC2*LC
      U12 = SUSCUR*SC
      U13 = SURCUK*KB
      U14 = SURCDS
      U21 = SUSCUR*LC
      U22 = SUSC2*SC
      U23 = SUKBUS*KB
      U24 = SUSCDS
      U31 = SURCUK*LC
      U32 = SUKBUS*SC
      U33 = SUKB2*KB
      U34 = SUKBDS
C      SOLVE MATRIX BY DIRECT GAUSS ELIMINATION
      X(3) = ((U14*U21-U24*U11)*(U12*U31-U32*U11)
&     -(U14*U31-U34*U11)*(U12*U21-U22*U11))/
&     ((U13*U21-U23*U11)*(U12*U31-U32*U11)
&     -(U13*U31-U33*U11)*(U12*U21-U22*U11))
      X(2) = ((U14*U21-U24*U11)-(U13*U21-U23*U11)*X(3))/
&     (U12*U21-U22*U11)
      X(1) = (U14-U13*X(3)-U12*X(2))/U11
      GO TO 410
  400 CONTINUE
C    UPDATE INITIAL GUESS VALUES OF  SC, KB, LC
      X(1) = INCRES
      LC = LC*(1.0+X(1))
      X(2) = INCRES
      SC = SC*(1.0+X(2))
      IF (SC.GE.1) SC = 1.0
      X(3) = INCRES
      KB = KB*(1.0+X(3))
      GO TO 430
  410 CONTINUE
C      COMPUTE STANDARD DEVIATION FOR EACH ITERATION
      SIGMA = SQRT(SDELS2/NTDP)
      WRITE (6, 420) ITERAT, SIGMA
  420 FORMAT (1H0,'THE STANDARD DEVIATION FOR ITERATION NUMBER,'I5,
&     'IS', F20.10)
```

```
C     UP-DATE COEFFICIENTS
  430 CONTINUE
      IF (X(1) .LT. DECRES) X(1) = DECRES
      IF (X(1) .GT. INCRES) X(1) = INCRES
      LC = LC*(1.0+X(1))
      IF (X(2) .LT. DECRES) X(2) = DECRES
      IF (X(2) .GT. INCRES) X(2) = INCRES
      SC = SC*(1.0+X(2))
      IF (SC .GE.1) SC = 1
      IF (X(3) .LT. DECRES) X(3) = DECRES
      IF (X(3) .GT. INCRES) X(3) = INCRES
      KB = KB*(1.0+X(3))
C     CHECK FOR DELTA CONVERGENCE
      IF (ABS(X(1)) .GT. ERROR .OR. ABS(X(2)) .GT. ERROR.
&     OR.ABS(X(3)) .GT. ERROR) GOTO 440
      GO TO 460
  440 IF (ITERAT.GE.ITMAX) GO TO 550
      IF (ABS(URC) .LT. 1.E-30 .OR. ABS(USC) .LT. 1.E-30 .OR. ABS(UKB)
&     .LT. 1.E-30) GO TO 280
      WRITE (6, 450) LC, SC, KB
  450 FORMAT (1H0,3HLC=,E20.10/3HSC=,E20.10/3HKB=,E20.10//)
      GO TO 280
C      WRITE OUT FINAL PROGRAM STATUS
  460 WRITE (6, 470) ITERAT
  470 FORMAT (1H1,'THE PARAMETERS CONVERGED IN',I5,1X,'ITERATIONS.')
      WRITE (6, 480)
  480 FORMAT (1H0,37HTHE BEST FIT TIME-DRAWDOWN PAIRS FOR ,
&     39HTHE CONVERGED VALUES OF S, T, AND L ARE)
      WRITE (6, 370)
      DO 490 I = 1, NTDP
  490 WRITE (6, 170) T(I), (SGS(I, J), J = 1, NOW)
      WRITE (6, 500) LC, SC, KB
  500 FORMAT (1H0,22HLEAKAGE COEFFICIENT = ,F20.10,
&     //22HSTORAGE COEFFICIENT = , F20.10,
&     //17HTRANSMISSIVITY = , F20.10)
      PRINT 510
  510 FORMAT ( ' DO YOU WANT TO COMPUTE AQUITARD PERMEABILITY ?', /,
&     'ANSWER YES IF TCL IS KNOWN OTHERWISE NO')
      READ:WRITAQP
      IF (WRITAQP .NE. 2HNO) GO TO 520
      IF (WRITAQP .NE. 3HYES) STOP
  520 PRINT 530
  530 FORMAT ( 'THICKNESS OF CONFINING LAYER ?')
      READ:TCL
      IF (TCL .LE. 0) STOP
      AQP = KB*TCL*LC**2
      WRITE (6, 540) AQP
  540 FORMAT ( 'AQUITARD PERMEABILITY =', F20.10)
      STOP
  550 WRITE (6, 560) ITMAX
  560 FORMAT (1H0,'THE PROGRAM DID NOT CONVERGE IN ',I5,1X,'ITERATIONS.')
      STOP
      END
C      FILE WURB
```

APPENDIX III.   PROGRAM HANTUSH

```
C       PROGRAM HANTUSH
        CHARACTER CHECK*3,LU*3,TU*3
        REAL KB,LC
        AA = 1.0
C       THIS PROGRAM COMPUTES THE DRAWDOWN IN A LEAKY AQUIFER AS
C       DEFINED BY JACOB AND HANTUSH, 1955. ALL INPUT IS IN CONSISTENT
C       UNITS.  A FULLY PENETRATING WELL IN AN ARTESIAN AQUIFER
C       AND NO WATER RELEASED FROM STORAGE IN THE AQUITARD,
C       WITH CONSTANT DISCHARGE CONDITIONS ARE THE PRINCIPLE
C       ASSUMPTIONS.
C       R= RADIUS OF OBSERVATION WELL FROM PUMPED WELL (L)
C       S= DRAWDOWN (L)
C       T= TIME (T)
C       Q= PUMPING RATE (L**3/T)
C       LC= INVERSE LEAKAGE COEFFICIENT OF SEMICONFINING BED (1/L)
C       LC= 1/B
C       KB= TRANSMISSIVITY OF AQUIFER (L**2/T)
C       SC= STORAGE COEFFICIENT OF AQUIFER (UNITLESS)
        PI= 3.1415926
        WRITE(6,1)
      1 FORMAT(1H0,'THIS PROGRAM CALCULATES THE DRAWDOWN IN A',
&       /,'LEAKY ARTESIAN AQUIFER.  THE RADIAL DISTANCE FOR THE',/,
&       'OBSERVATION WELL AND THE PUMPING PERIOD MAY BE CONSTANT',/,
&       'OR VARIABLE.  ANY CONSISTENT SET OF UNITS MAY BE USED.'//)
    100 WRITE (6,2)
      2 FORMAT(1H0,'ENTER THE FOLLOWING DATA IN A FREE FORMAT FIELD:',/
&       'Q = PUMPING RATE (L**3/T)',/,
&       'LC = INVERSE LEAKAGE COEFFICIENT (1/L)',/,
&       'KB = AQUIFER TRANSMISSIVITY (L**2/T)',/,
&       'SC = STORAGE COEFFICIENT (UNITLESS)',/,
&       'LU = UNIT OF LENGTH (3 CHARACTERS MAX)',/,
&       'TU = UNIT OF TIME (3 CHARACTERS MAX)'/)
        READ(5,3)Q,LC,KB,SC,LU,TU
      3 FORMAT(V)
        WRITE(6,4)
      4 FORMAT(1H0,'ARE THERE ANY ERRORS IN THE ABOVE ENTRIES?',/,
&       'IF NOT, ANSWER NO.')
        READ(5,3)CHECK
        IF(CHECK.NE.2HNO)GOTO100
        WRITE(6,10)
     10 FORMAT(1H0 ,'THE FOLLOWING PARAMETERS ARE USED IN THIS SOLUTION:')
        WRITE(6,11)Q,LC,KB,SC,LU,TU
     11 FORMAT(1H0,'Q=',F10.2,//,'LC=',F10.8,//,'KB=',F10.2,//,'SC=',F10.8,//
&       'LU =',A3,//,'TU =',A3,//)
        WRITE(6,7)
      7 FORMAT(1H0,'ENTER THE FOLLOWING DATA:',/,
&       'R,T',/,
&       'R = DISTANCE TO OBSERVATION WELL (L)',/,
&       'T = LENGTH OF PUMPING PERIOD (T)',/,
&       'TERMINATE BY  RETURNING BLANK FIELDS.')
    200 WRITE(6,13)
     13 FORMAT(1H0,'ENTER R,T')
```

```
      READ(5,3)R,T
      IF(R.EQ.0..OR.T.EQ.0.)GOTO300
      RB=R*LC
      U=(R*R*SC)/(4.*KB*T)
      WRITE(6,12) U,RB
   12 FORMAT(1H0,'U=',F10.6,1X,//,'R/B=',F10.5)
      S=(Q/(4.*PI*KB))*W(U,RB)
      WRITE(6,8)R,LU,S,LU,T,TU
    8 FORMAT(1H0,'THE DRAWDOWN',1X,F6.0,1X,A3,1X,/,
&     'FROM THE PUMPING WELL IS',1X,F10.5,1X,A3,1X,/,
&     'AFTER',1X,F10.2,1X,A3,1X,'OF PUMPING.',/)
      GOTO200
  300 WRITE(6,14)
   14 FORMAT(1H0,'YOU HAVE TERMINATED THE PROGRAM.',/,
&     'IS THIS CORRECT?')
      READ(5,3)CHECK
      IF(CHECK.NE.'YES')GOTO200
      STOP
      END
C       FILE WURB
```

APPENDIX IV. FILE WURB, A LIST OF EXPLICIT FUNCTIONS FOR SOLUTION OF W(U,RB).

```
C       FILE WURB
C       FILE WURB IS A LIST OF FUNCTIONS REQUIRED IN THE
C       SOLUTION OF THE LEAKY ARTESIAN WELL FUNCTION W (U,R/B).
      FUNCTION W (U, RB)
C       THIS FUNCTION DEFINES THE LEAKY ARTESIAN WELL FUNCTION.
C       THE THREE FORMS CORRESPOND TO THOSE OUTLINED IN
C       HANTUSH AND JACOB, 1955.
      COMMON        LC, ITERAT, LCOUNT, KB, KBCOUNT
      REAL          LC, KB
      IF (U.GE.1.0) GO TO 570
      IF (U .LT. 1.0 .AND. (RB*RB) .GT. U) GO TO 580
      IF (U .LT. 1.0 .AND. (RB*RB) .LE. U) GO TO 590
  570 W = SS(U, RB)
      GO TO 600
  580 W = (2*AK0(RB)-SS(U, RB))
      GO TO 600
  590 F1 = (RB*RB*.25/U)
      W = 2*AK0(RB)-AI0(RB)*(EI(F1))+EXP(-F1)*
&     (0.5772+ALOG(U)+EI(U)-U+U*
&     ((AI0(RB)-1)/(RB*RB*.25))-U*U*SUM(U, RB))
  600 RETURN
      END
      FUNCTION SS (U, RB)
C       THIS FUNCTION SOLVES THE INDEFINITE INTEGRAL OUTLINED
C       ON PAGE 231 OF THE TEXT.   THE METHOD USED IS
C       LAGUERRE INTEGRATION AS DISCUSSED IN A & S
C       (ABRAMOWITZ AND SEGUN, 1968), PAGE 923.
C       FOR AN EXPLANATION OF THE PARAMETER AA
C       IN THE CODE BELOW, SEE THE TEXT.
      COMMON        LC, ITERAT, LCOUNT, KB, KBCOUNT
      REAL          LC, KB
      DOUBLE PRECISION  Y(15), WF(15)
      DATA          AA/1./
      DATA          Y(1)/0.093307812017/
      DATA          Y(2)/0.492691740302/
      DATA          Y(3)/1.215595412071/
      DATA          Y(4)/2.269949526204/
      DATA          Y(5)/3.667622721751/
      DATA          Y(6)/5.425336627414/
      DATA          Y(7)/7.565916226613/
      DATA          Y(8)/10.120228568019/
      DATA          Y(9)/13.130282482176/
      DATA          Y(10)/16.654407708330/
      DATA          Y(11)/20.776478899449/
      DATA          Y(12)/25.623894226729/
      DATA          Y(13)/31.407519169754/
      DATA          Y(14)/38.530683306486/
      DATA          Y(15)/48.026085572686/
      DATA          WF(1)/0.239578170311/
      DATA          WF(2)/0.560100842793/
```

```
      DATA          WF(3)/0.887008262919/
      DATA          WF(4)/1.22366440215/
      DATA          WF(5)/1.57444872163/
      DATA          WF(6)/1.94475197653/
      DATA          WF(7)/2.34150205664/
      DATA          WF(8)/2.77404192683/
      DATA          WF(9)/3.25564334640/
      DATA          WF(10)/3.80631171423/
      DATA          WF(11)/4.45847775384/
      DATA          WF(12)/5.27001778443/
      DATA          WF(13)/6.35956346973/
      DATA          WF(14)/8.03178763212/
      DATA          WF(15)/11.5277721009/
      B = (RB)**2
      WU = 0
      DO 630 I = 1, 15
        IF (U .LT. 1.0) GO TO 610
        A = 1/(U+Y(I)/AA)
        F = A*EXP(-(U+B*A*.25+Y(I)/AA))
        FEW = F
        IF (FEW .LE. 0.) GO TO 640
        GO TO 620
  610   UM = 0.25*RB*RB*(1/U)
        A = 1/(UM+Y(I)/AA)
        F = A*EXP(-(UM+B*A*0.25+Y(I)/AA))
        FEW2 = F
        IF (FEW2 .LE. 0) GO TO 640
  620   FW = F*WF(I)
        WU = WU+FW
  630 CONTINUE
      SS = (1/AA)*WU
      RETURN
  640 CONTINUE
      TESTB = EXP(-(B*A*.25+Y(I)/AA))
      TESTU = EXP(-U)
      TESTUM = EXP(-UM)
      IF (TESTB .LE. 0 .OR. U .LT. 1 .AND. TESTUM .LE. 0) GO TO 650
      IF (U .GT. 1 .AND. TESTU .LE. 0) GO TO 660
  650 CONTINUE
      LC = LC*.05
      LCOUNT = LCOUNT+1
      RETURN
  660 CONTINUE
      KB = KB*10
      KBCOUNT = KBCOUNT+1
      RETURN
      END
      FUNCTION AK0 (RB)
C      THIS FUNCTION SOLVES THE MODIFIED BESSELS FUNCTION
C      OF THE SECOND KIND, ZERO ORDER, A & S, PAGE 379.
      IF (RB .GT. 2.) GO TO 670
      TJ = (RB/2.)
      AK0 = -ALOG(TJ)*AI0(RB)-.57721566
&     +0.42278420*(TJ)**2+0.23069756*(TJ)**4
```

```
&     +0.03488590*(TJ)**6+0.00262698*(TJ)**8
&     +0.00010750*(TJ)**10+0.00000740*(TJ)**12
      GO TO 680
  670 TM = (2./RB)
      AK0 = (1./SQRT(RB))*EXP(-RB)*(1.25331414
&     -0.07832358*TM+0.02189568*TM**2
&     -0.01062446*TM**3+0.00587872*TM**4
&     -0.00251540*TM**5+0.00053208*TM**6)
  680 RETURN
      END
      FUNCTION AI0 (RB)
C      THIS FUNCTION SOLVES THE MODIFIED BESSELS FUNCTION
C      OF THE FIRST KIND, ZERO ORDER, A & S, PAGE 378.
      TF = RB/3.75
      IF (RB .LE. 3.75) GO TO 690
      AI0 = (1/SQRT(RB))*EXP(RB)*(.39894228
&     +.01328592*TF**(-1)+.00225319*TF**(-2)
&     -.00157565*TF**(-3)+.00916281*TF**(-4)
&     -.02057706*TF**(-5)+.02635537*TF**(-6)
&     -.01647633*TF**(-7)+.00392377*TF**(-8))
      GO TO 700
  690 AI0 = 1.0+3.5156229*TF**2+3.0899424*TF**4
&     +1.2067492*TF**6+0.2659732*TF**8
&     +0.0360768*TF**10+0.0045813*TF**12
  700 RETURN
      END
C      FUNCTIONS SUM AND IFACT ARE SPECIAL FUNCTIONS
C      WHICH ARE USED TO SOLVE THE EQUATION BEGINNING
C      AT STATEMENT 3000 IN FUNCTION W.
      FUNCTION SUM (U, RB)
      DATA         LIMIT/5/
      SUM = 0.0
      RBF = RB*RB*0.25
      DO 720 N = 1, LIMIT
        DO 710 M = 1, N
          LF = (N-M+1)
          LH = (N+2)
          BFACT = IFACT(LH)*1.0
          AFACT = IFACT(LF)*1.0
          PSUM = ((-1)**(N+M))*(AFACT/(BFACT*BFACT))*(RBF**M)*
&     (U**(N-M))
          SUM = SUM+PSUM
          IF (PSUM .LT. 1.0E-8) GO TO 730
  710   CONTINUE
  720 CONTINUE
  730 RETURN
      END
      FUNCTION IFACT (L)
      IFACT = 1
      IF (L .EQ. 1) GO TO 750
      DO 740 I = 2, L
        MA = I
        IFACT = IFACT*MA
  740 CONTINUE
```

```
  750 RETURN
      END
      FUNCTION EI (U)
C      THIS FUNCTION SOLVES THE EXPONENTIAL INTEGRAL
C      DEFINED ON PAGE 231 OF A & S.
      IF (U .GT. 1.0) GO TO 760
      EI = -ALOG(U)-.5772156+.99999139*U-.24991055*U*U
&     +.05519968*U**3-.0097004*U**4+.00107857*U**5
      GO TO 770
  760 EI = (EXP(-U)/U)*(U*U+2.334733*U+.250621)/
&     (U*U+3.330657*U+1.681534)
  770 RETURN
      END
```

APPENDIX V. LIST OF TEST-DATA SOURCES

1. Walton, W. C., 1970, Groundwater resource evaluation: New York, McGraw-Hill, page 286, Problem 4.5.
2. Cooper, H. H., Jr., 1963, [see references], as cited in Lohman, S.W., 1972, Ground water hydraulics: U.S. Geological Survey Professional Paper 708, page 31, Table II.
3. Walton, W. C., 1962, Selected analytical methods for well and aquifer evaluation: Illinois State Water Survey Bulletin 49, Urbana, Department of Registration and Education, page 32, Table 5.
4. Gutentag, E. D., 1965, Aquifer test in the Ogallala Formation (26-37-21ddd): U.S. Geological Survey, Garden City, Kansas, open file data.
5. Gillespie, J. B., 1979, Results of aquifer tests in the Wellington aquifer near Salina, Kansas: Preliminary report, Lawrence, Kansas.
6. Burns and McDonnell Consultants, 1977, Pump test data at Test Well #1, _in_ Groundwater resources investigation of the Spratt Site for Sunflower Electric Cooperative: Kansas City, Burns and McDonnell.

APPENDIX VI. COMMENTS ON PROGRAM NOTATION

The variables appearing in the matrix equation are defined in the program by the following notation:

T : KB U_T : UKB

S : SC U_S : USC

L : LC U_L : URC

$\sum_i U_L (s_g - s_e)$: SURCDS

$\sum_i U_S (s_g - s_e)$: SUSCDS

$\sum_i U_T (s_g - s_e)$: SUKBDS

$\sum_i U_L^2$: SURC2

$\sum_i U_L U_S$: SUSCUR

$\sum_i U_L U_T$: SURCUK

$\sum_i U_S U_L$: SUSCUR

$\sum_i U_S^2$: SUSC2

$\sum_i U_S U_T$: SUKBUS

$\sum_i U_T U_L$: SURCUK

$\sum_i U_T U_S$: SUKBUS

$\sum_i U_T^2$: SUKB2