Almeida, C., T. Ribeiro e M. M. Oliveira (1992)

LARGWELL: a program to pumping test analysis in large diameter wells

Geolis, revista da Secção de Geologia Ec. e Aplicada, vol. VI(1, 2), p. 89-110.

Geolis, Vol. VI, Fasc. 1 e 2 (1992) 89-110

LARGWELL: A PROGRAM TO PUMPING TEST ANALYSIS IN LARGE DIAMETER WELLS

Carlos Almeida*, Teresa Ribeiro*, Manuel M. Oliveira**
*Departamento de Geologia, Faculdade de Ciências da Univ. de Lisboa
**Bolseiro da JNICT

RESUMO

Neste trabalho descreve-se um programa em QuickBASIC para interpretação de ensaios de bombagem realizados em poços de grande diâmetro. No programa é utilizado o método de Gauss--Newton para melhorar as estimativas iniciais dos parâmetros hidráulicos e a inversão numérica da transformada de Laplace para calcular a função de poço. Esta última técnica é utilizada a fim de evitar o cálculo de um integral impróprio presente naquela função o que permite uma notável economia no tempo de computação.

ABSTRACT

This paper presents a QuickBASIC program for analysing pumping tests carried out in large diameter wells in a confined aquifer. The program uses the Gauss-Newton method to improve an initial estimate of the hydraulic parameters and numerical inversion of the Laplace transform of the well function. This technique is used in order to avoid the calculation of an improper integral present in the well function whose computation is very time-consuming.

Introduction

Very often, pumping tests carried out in large diameter wells are the only means available to evaluate the hydraulic parameters of an aquifer. This is particularly true in undeveloped regions and in areas of rocks with low permeability (for example crystalline rocks), where that type of well is largely dominant.

The solution for the continuity equation in a confined aquifer, taking into account storage in the pumping well, was given by Papadopulos and Cooper (1967). Boulton and Streltsova (1975) gave an analytical model for a large diameter well in an unconfined aquifer. Rushton and Singh (1983) defined an alternative well function calculation by a numerical model including decreasing abstraction rates.

Analysis of pumping test data in large diameter wells is usually made by the classical method of curve matching. This method is a rather difficult one due to the similarity of curves in the early stages of pumping and the lack of intermediate curves. For that reason, many computer methods have been developed for automated or semi-automated estimation of aquifer parameters using data from large diameter wells. The solution given by Papadopulos and Cooper implies the computation of one improper integral involving Bessel functions. The numerical evaluation of this integral involves large computation and is time-consuming. In all the methods that have been proposed, other alternatives were used to avoid the computation of this integral.

Rushton and Chan (1976) estimated the hydraulic parameters by a numerical model from analysis of pump and recovery period in a confined aquifer. Rushton and Redshaw (1979) improved the Rushton and Chan model. Rushton and Holt (1981) and Herbert and Kitching (1981) applied this model on dug wells in an unconfined and anisotropic aquifer, respectively.

Patel and Mishra (1983) made an analysis of flow to a large diameter well in a confined aquifer by discrete kernel approach. Mishra and Chachadi (1985) extended the Patel and Mishra method to the recovery period and Rushton and Singh (1987) made another modification considering the existence of a seepage face in the case of unconfined aquifers. Singh and Gupta (1986) applied the convolution method and sensitivity analysis, to deal with variable abstraction rates in confined aquifers. Recently, Sakthivadivel and Rusthon (1989) used the Rushton and Holt (1981) model, modified in order to take into account the seepage face.

The present approach rather than to compute the well function from its analytical solution uses its Laplace transform. This technique is used in order to avoid the time-consuming computation of an improper integral present in the well function. The real space solution is obtained by numerical inversion of the Laplace transform using the Stehfest (1970) algorithm (Moench and Ogata, 1981, Moench and Ogata, 1984).

This approach has the advantage of using the exact solution and being totally automated. Furthermore it will be proved to be fast and accurate.

Analytical solution for the flow towards a large diameter well

As pointed out before, the analytical solution for the flow towards a large diameter well was given by Papadopulos and Cooper (1967). This solution is obtained solving the continuity equation:

$$\frac{\partial^2 s}{\partial r^2} + \frac{1}{r}\frac{\partial s}{\partial r} = \frac{S}{T}\frac{\partial s}{\partial t} \qquad (1)$$

subjected to the following boundary conditions and assuming that well losses are negligible:

$s(r_w, t) = s_w(t)$;

$s(\infty, t) = 0$;

$s(r, 0) = 0 \; r \geq r_w$;

$s_w(0) = 0$;

$$2\pi r_w T \frac{\partial s(r_w, t)}{\partial r} - \pi r_c^2 \frac{\partial s_w(t)}{\partial t} = -Q$$

where:

s = drawdown in the aquifer at distance r and time t;
s_w = drawdown in the pumping well at time t;
r = radial distance from the center of the well;
r_w = effective radius of well screen or open hole;
r_c = radius of well casing in the interval over which the water level declines;
t = time since well begins to discharge;
S = storage coefficient of the aquifer;
T = transmissivity of the aquifer;
Q = constant discharge of the well.

In order to solve this boundary value problem, Papadopulos and Cooper applied the Laplace transform with respect to time. The solution in the Laplace plane is:

$$\bar{s} = \frac{Q K_0(qr)}{\pi p\left[r_c^2 \, p K_0(qr_w) + 2 r_w T q K_1(qr_w)\right]} \qquad (2)$$

where $q = \sqrt{pS/T}$.

The inverse transform of this equation yields the drawdown solution anywhere in the aquifer as:

$$s_w = \frac{2Q\alpha}{\pi^2}\int_0^\infty (1 - e^{-\beta^2/4u_w}) \cdot \{J_0(\beta r/r_w)\left[\beta Y_0(\beta) - 2\alpha Y_1(\beta)\right] - Y_0(\beta r/r_w)\left[\beta J_0(\beta) - 2\alpha J_1(\beta)\right]\}\frac{d\beta}{\beta^3 \Delta(\beta)} \qquad (3)$$

$\alpha = r_c^2 S/r_c^2$; $u = r_w^2 S/4Tt$;

$\Delta(\beta) = \left[\beta J_0(\beta) - 2\alpha J_1(\beta)\right]^2 + \left[\beta Y_0(\beta) - 2\alpha Y_1(\beta)\right]^2$;

J_0, J_1, Y_0 and Y_1 are Bessel functions.

The drawdown inside the well is obtained by substituting $r = r_w$ in equation (3):

$$s_w = \frac{8Q\alpha^2}{T\pi^3}\int_0^{\infty}\frac{(1-e^{-\beta^2/4u_w})}{\beta^3\Delta(\beta)}d\beta \qquad (3a)$$

Outline of the program

In the program presented here, the inverse problem is solved by using the Gauss-Newton method. In this method, one needs a set of experimental data (drawdown values observed in the large diameter well or in an observation well) and a model that relates them with a set of parameters. Here the parameters are transmissivity and storage coefficient and the model is the Papadopulos and Cooper analytical model.

Furthermore, one needs a set of initial estimates of the hydraulic parameters. The estimate of the parameter is improved iteratively by a correction vector:

$$\mathbf{b}^{i+1} = \mathbf{b}^{i} + \Delta\mathbf{b} \qquad (4)$$

where $\mathbf{b}^{i+1}$ and $\mathbf{b}^{i}$ are the new and old estimate of the parameter vector, respectively, and $\Delta\mathbf{b}$ the correction vector.

The correction vector is obtained through the expression:

$$\Delta\mathbf{b} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T(\mathbf{s}^\circ - \mathbf{s}^*) \qquad (5)$$

where $\mathbf{s}^\circ$ and $\mathbf{s}^*$ are the vectors of observed and calculated drawdowns, respectively. The latter are obtained using estimated values of the parameters. A is the Jacobian matrix of the partial derivatives of **s** with respect to **b**, evaluated at all n points where, experimental observations are available. $\mathbf{A}^T$ is the transpose of **A**. In our case:

$$A = \begin{bmatrix} \frac{\partial s_1}{\partial T} & \frac{\partial s_1}{\partial S} \\ \vdots & \vdots \\ \frac{\partial s_n}{\partial T} & \frac{\partial s_n}{\partial S} \end{bmatrix}$$

The derivatives in **A** are calculated by a centered finite difference approximation:

$$\frac{\partial s}{\partial T} \cong \frac{s(T+\Delta T) - s(T-\Delta T)}{2\Delta T} \qquad (6)$$

$$\frac{\partial s}{\partial S} \cong \frac{s(S+\Delta S) - s(S-\Delta S)}{2\Delta S} \qquad (7)$$

The approximation becomes increasingly accurate as ΔT and ΔS approaches zero. Satisfactory evaluation of the partial derivatives occurred for an increment in the variable equal to .01 times the parameter (Cobb and others, 1982).

In order to calculate the theoretical values of the drawdown in the pumping well with the estimated parameters, the Laplace transform solution given by equation (2) was slightly modified to become:

$$\bar{s}_{wD} = \frac{Q\, K_0(r_D\sqrt{p})}{2\pi T\, p\left[W_D\, p\, K_0(\sqrt{p}) + \sqrt{p}\, K_1(\sqrt{p})\right]} \qquad (8)$$

where:

$\bar{s}_{wD}$ is the Laplace transform of the well function;

p is the Laplace transform variable concerning a parameter $\theta = Tt/r_w^2 S$

$$W_D = \frac{r_c^2}{2r_w^2 S}$$

K_0 and K_1 are modified Bessel functions.

As stated by Barker (1985) the numerical inversion of the Laplace transform is computationally more efficient than the evaluation of the well function from their analytical solutions in practically all cases, except the Theis well function. In the present case this method is five times faster than the evaluation of the well function by Gaussian quadrature with 8 points.

The numerical inversion of the Laplace transform was obtained by the algorithm developed by Stehfest (1970). Comparing the values calculated by this method with the well function table provided by Papadopulos and Cooper (1967) shows it to be accurate. The Bessel functions are evaluated by polynomial approximation (Abramowitz and Stegun, 1970).

In order to maintain physical reality and assure numerical stability, the conditions that T and S are always positive and S always less than one are imposed. Furthermore the variation of S in two consecutive iterations is not allowed to be greater than one order of magnitude.

During the development of the program, problems of convergence were observed for poor estimates of the transmissivity. Therefore a subroutine was included to estimate the transmissivity using the last drawdown and a halving procedure. With this method, convergence is obtained even for estimates of storage coefficient very far from the true value.

In order to enable the user to check the matching a graphic subroutine is included. This subroutine plots a log-log graphic of drawdowns versus time and the theoretical curve obtained with the calculated parameters.

Results and discussion

The program was tested both using theoretical values and using several field data. Two examples are given. In the first example theoretical values of drawdown were generated from the table provided by Papadopulos and Cooper (1967) by assigning different values of time (keeping

constant the other parameters: transmissivity = 150 m^2/day, storage coefficient = 10^{-4}, well casing radius, r_c = 1.5 m and well radius, r_w = 1.5 m). The time/drawdown values obtained are provided in table I. The program was run for different initial estimates of storage coefficient. The final results are given in table II. It can be seen that even for very different estimates of storage coefficient the same final results were achieved. These are coincident with those used to generate the time/drawdown values. A plot of the time/drawdown values together with the corresponding matching curve is shown in fig. 1.

In the second example a data set from a pumping test carried out in an unconfined aquifer located near Évora (South of Portugal) was used. Although the Papadopulos and Cooper (1967) model has been developed for confined aquifers it can also be used if drawdown values are submitted to the Jacob's correction. The observed and corrected values are shown in table III.

Once again the program was tested using different initial estimates of the storage coefficient; the resulting final values were also invariant (table IV). These results are more accurate when compared with those obtained by the standard curve matching technique (last row in table IV). A fitting curve to the field data is provided in fig. 2. The different values of transmissivity and storage coefficient as well as the root-mean-square deviation between the observed and the calculated drawdowns are also shown as a function of the iterations in table V.

It should be noted, as pointed out by Papadopulos and Cooper (1967), that when interpreting pumping tests carried out in large diameter wells the value of the storage coefficient is to be taken into account with caution. This is so because the low sensitivity of the well function to variations of the storage coefficient.

Notation

J_0 Bessel function of the first kind and order zero.
J_1 Bessel function of the first kind and order unity.
K_0 modified Bessel function of the second kind and order zero.
K_1 modified Bessel function of the second kind and order unity.
p Laplace transform variable, dimensionless.
Q well discharge rate, L^3T^{-1}.
r radial distance from the center of the well, L.
r_c radius of well casing in the interval over which the water level declines, L.
r_w effective radius of the well screen, L.
s drawdown, L.
s_w drawdown in the pumping well, L.
s_{wD} Laplace transform of the well function, dimensionless.
S storage coefficient, dimensionless.
t time since start of pumping, T.
T transmissivity, L^2T^{-1}.
u_w argument of well function, $Sr_w^2 / 4Tt$.
W_D dimensionless well bore storage coefficient, $r_c^2 / 2r_w^2S$
Y_0 Bessel function of the second kind and order zero.
Y_1 Bessel function of the second kind and order unity.
α dimensionless storage coefficient, r_w^2S / r_c^2 .

References

ABRAMOWITZ, M., STEGUN, I. A. (1970) - Handbook of mathematical functions. Dover Publications Inc., New York, 1046 p.
BOULTON, N. S., STRELTSOVA, T. D. (1975) - The drawdown near an abstraction well of large diameter under non-steady conditions in an unconfined aquifer. J. Hydrol., vol. 30, pp. 29-46.
COBB, P. M., McELWEE, C. D., BUTT, M. A. (1982) - Analysis of leaky aquifer pumping test data: an automated numerical solution using sensitivity analysis. Ground Water, vol. 20(3), pp. 325-333.
HERBERT, R., KITCHING, R. (1981) - Determination of aquifer parameters from large-diameter dug well pumping tests. Ground Water, vol. 19(6), pp. 593-599.
MISHRA, G. C., CHACHADI, A. G. (1985) - Analysis of flow to a large-diameter well during the recovery period. Ground Water, vol. 23(5), pp. 646-651.
MOENCH, A. F., OGATA, A. (1981) - A numerical inversion of the Laplace transform solution to radial dispersion in a porous medium. Water Resour. Res., vol. 17(1), pp. 250-252.
MOENCH, A. F., OGATA, A. (1984) - Analysis of constant discharge wells by numerical inversion of Laplace transform solutions, in Groundwater Hydraulics, Water Resources Monog. Ser., vol. 9, edited by J. Rosenshein & G. D. Bennett, pp. 146-170, AGU, Washington D. C.
PAPADOPULOS, I. S., COOPER Jr., H. H. (1967) - Drawdown in a well of large diameter. Water Resour. Res., vol. 3(1), pp. 241-244.
PATEL, S. C., MISHRA, G. C. (1983) - Analysis of flow to a large-diameter well by discrete Kernel approach. Ground Water, vol. 21(5), pp. 573-576.
RUSHTON, K. R. (1978) - Estimating transmissivity and storage coefficient from abstraction well data. Ground Water, vol. 16(2), pp. 81-85.
RUSHTON, K. R., REDSHAW, S. C. (1979) - Seepage and Groundwater Flow, Wiley, Chichester, 332 p.
RUSHTON, K. R., HOLT, S. M. (1981) - Estimating aquifer parameters for large-diameter wells. Ground Water, vol. 19(5), pp. 505-509.
RUSHTON, K. R., SINGH, V. S. (1983) - Drawdowns in large-diameter wells due to decreasing abstraction rates. Ground Water, vol. 21(6), pp. 670-677.
RUSHTON, K. R., SINGH, V. S. (1987) - Pumping test analysis in large diameter: wells with a seepage face by Kernel function technique. Ground Water, vol. 25(1), pp. 81-90.
SAKTHIVADIVEL, R., RUSHTON, K. R. (1989) - Numerical analysis of large diameter wells with a seepage face. J. Hydrol., vol. 107, pp. 43-55. -
SINGH, V. S., GUPTA, C. P. (1986) - Hydrogeological parameter estimation from pump tests on a large diameter well. J. Hydrol., vol. 87, pp. 223-232.
STEHFEST, H. (1970) - Numerical inversion of Laplace transforms. Commun. ACM, vol. 13(1), pp. 47-49.

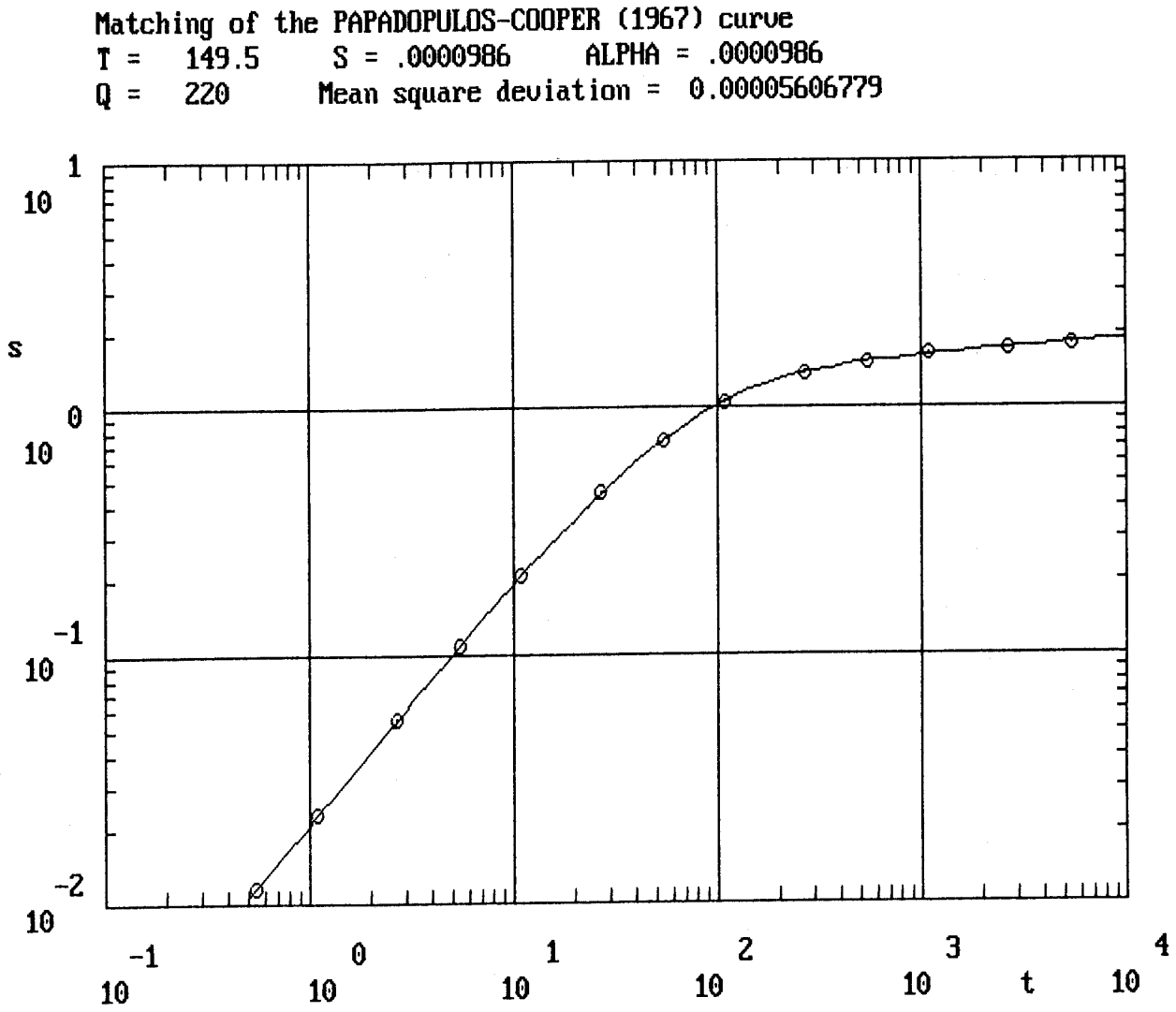

Fig. 1 – Matching of the Papadopulos-Cooper (1967) curve to data set given in Table I.

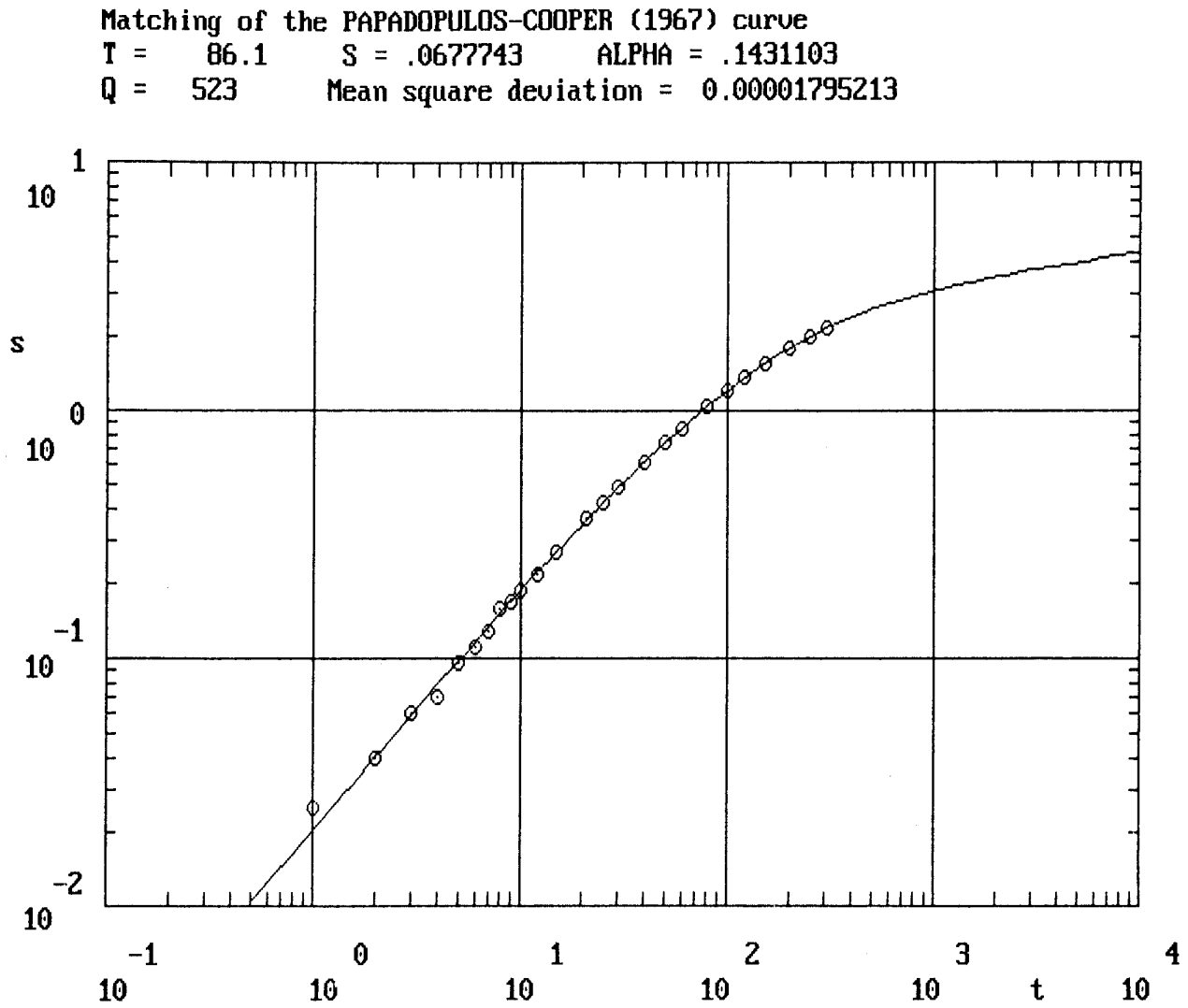

Fig. 2 – Matching of the Papadopulos-Cooper (1967) curve to data set given in Table III.

Appendix 1

TABLE I – Time-drawdown values used in example 1.
$Q = 220\ m^3/day$; $r_w = 1.5\ m$; $r_c = 1.5\ m$

Time (days)	Obs. drawd. (m)
3.75×10^{-4}	0.0116
7.5×10^{-4}	0.0229
1.875×10^{-3}	0.0562
3.75×10^{-3}	0.109
7.5×10^{-3}	0.2063
1.875×10^{-2}	0.4468
3.75×10^{-2}	0.7289
7.5×10^{-2}	1.0494
1.875×10^{-1}	1.3702
3.75×10^{-1}	1.5068
7.5×10^{-1}	1.6383
1.875	1.7262
3.75	1.8102

TABLE II – Final results obtained for different initial estimates of storage coefficient for the data set given in table I.

S est.	S final	T final	RMSD	No. iter.
1×10^{-1}	9.9×10^{-5}	149.5	5.6×10^{-5}	6
1×10^{-2}	9.9×10^{-5}	149.5	5.6×10^{-5}	6
1×10^{-3}	9.9×10^{-5}	149.5	5.6×10^{-5}	6
1×10^{-4}	9.9×10^{-5}	149.5	5.6×10^{-5}	6
1×10^{-5}	9.9×10^{-5}	149.5	5.6×10^{-5}	6
1×10^{-7}	9.9×10^{-5}	149.5	5.6×10^{-5}	6

TABLE III – Time-drawdown values used in example 2.
$Q = 523\ m^3/day$; $r_w = 1.6$ m; $r_c = 2.325$ m

1	2	3
1	0.025	0.025
2	0.04	0.04
3	0.06	0.06
4	0.07	0.07
5	0.095	0.095
6	0.11	0.11
7	0.13	0.129
8	0.16	0.159
9	0.17	0.169
10	0.19	0.189
12	0.22	0.218
15	0.27	0.268
21	0.37	0.365
25	0.43	0.424
30	0.5	0.492
40	0.63	0.617
50	0.76	0.741
60	0.87	0.845
80	1.08	1.041
100	1.26	1.207
120	1.43	1.362
152	1.65	1.559
200	1.92	1.797
250	2.14	1.987
305	2.34	2.157

1 – Time (min); 2 – Observed drawdown (m); 3 – Corrected drawdown (m)

TABLE IV – Final results obtained fior different initial estimates of storage coefficient for the data given in table III.

S est.	S final	T final	RMSD	No. iter.
1×10^{-1}	0.068	86.1	1.8×10^{-5}	4
1×10^{-3}	0.068	86.1	1.8×10^{-5}	6
1×10^{-5}	0.068	86.1	1.8×10^{-5}	8
1×10^{-7}	0.068	86.1	1.8×10^{-5}	10
Curve matching	0.022	111	2.2×10^{-4}	—

TABLE V – Variation of the values of the parameters in each iteration for an Initial estimate of $S = 10^{-5}$ for the data set given in table III.

iterat.	1	2	3	4	5	6	7	8
S	1E-05	1.1E-04	1.2E-03	9.9E-03	1.1E-01	6.1E-02	6.1E-02	6.1E-02
T	264.6	0.1	167	26	69.4	84.4	86	86.1
RMSD	3.6E-03	1.7	2.3E-03	5.4E-01	1.3E-03	4.8E-04	1.9E-05	1.8E-05

```
'              ****** PROGRAM LARGWELL ******
'
'  Written by C. Almeida, T. Ribeiro and M. M. Oliveira, January 1990
'
'  Program for  automated  interpretation of pumping tests carried
'  out in large  diameter  well  using the  Stehfest algorithm for
'  numeric inversion of the Laplace transform of the well function

DEFINT I-N:  DEFDBL D, U-Z

DIM DA(100, 2), DA0(16), DA1(16), DAT(2, 2), DB0(17), DB1(12), DV(10), DH(2 100)
DIM SC(100), SI(100), TD(100), TI(100), X(100, 5), Y(100), G(10), DH1(5)

'                    * Constants *

DA0 = -.57721566#: DA1 = .99999193#
DA0(1) = 3.5156229#: DA0(2) = 3.0899424#
DA0(3) = 1.2067492#: DA0(4) = .2659732#
DA0(5) = .0360768#: DA0(6) = .0045813#
DA0(7) = .39894228#: DA0(8) = .01328592#
DA0(9) = .00225319#: DA0(10) = -.00157565#
DA0(11) = .00916281#: DA0(12) = -.02057706#
DA0(13) = .02635537#: DA0(14) = -.01647633#
DA0(15) = .00392377#
DA1(1) = .87890594#: DA1(2) = .51498869#
DA1(3) = .15084934#: DA1(4) = .02658733#
DA1(5) = .00301532#: DA1(6) = .00032411#
DA1(7) = .39894228#: DA1(8) = -.03988024#
DA1(9) = -.00362018#: DA1(10) = .00163801#
DA1(11) = -.01031555#: DA1(12) = .02282967#
DA1(13) = -.02895312#: DA1(14) = .01787654#
DA1(15) = -.00420059#
DB0(1) = DA0: DB0(2) = .4227842#
DB0(3) = .23069756#: DB0(4) = .0348859#
DB0(5) = .00262698#: DB0(6) = .0001075#
DB0(7) = .0000074#: DB0(8) = 1.25331414#
DB0(9) = -.07832358#: DB0(10) = .02189568#
DB0(11) = -.01062446#: DB0(12) = .00587872#
DB0(13) = -.0025154#: DB0(14) = .00053208#
DB1(1) = .15443144#: DB1(2) = -.6727857900000001#
DB1(3) = -.18156897#: DB1(4) = -.01919402#
DB1(5) = -.00110404#: DB1(6) = -.00004686#
DB1(7) = .23498619#: DB1(8) = -.0365562#
DB1(9) = .01504268#: DB1(10) = -.00780353#
DB1(11) = .00325614#: DB1(12) = -.00068245#
```

```
DPI = 4# * ATN(1#)
DPIX4 = 4# * DPI
TOL = .005

DEF FNALD (X) = LOG(X) / LOG(10)    'LOG10

N = 10  'First part of the Stehfest algorithm
NH = 5
G(0) = 1
FOR I = 1 TO N
  G(I) = G(I - 1) * I
NEXT I
DH1(1) = 2# / CDBL(G(NH - 1))
FOR I = 2 TO NH
  DH1(I) = CDBL(CSNG(I ^ NH) * G(2 * I)) / CDBL(G(NH - I) * G(I) * G(I - 1))
NEXT I
SN = 1
FOR I = 1 TO N
  DV(I) = 0
  LI = INT((I + 1) / 2)
  IF I < NH THEN LS = I ELSE LS = NH
  FOR K = LI TO LS
    DV(I) = DV(I) + DH1(K) / CDBL(G(I - K) * G(2 * K - I))
  NEXT K
  DV(I) = CDBL(SN) * DV(I)
  SN = -SN
NEXT I

'           *** modified Bessel functions ***
'
DEF FNDBESSI0 (DX)     'modified Bessel function I0
STATIC DEN, DF, DI0, DNUM
STATIC DY, DY2, I

IF DX < 3.75 THEN
          DY = DX / 3.75#
          DY2 = DY * DY
          DI0 = 1#: DF = 1#
          FOR I = 1 TO 6
            DF = DF * DY2
            DI0 = DI0 + DF * DA0(I)
          NEXT I
ELSE
          DY = 3.75# / DX
          DNUM = DA0(7): DF = 1#
          FOR I = 8 TO 15
            DF = DF * DY
            DNUM = DNUM + DF * DA0(I)
          NEXT I
```

```
            DEN = SQR(DX) * EXP(-DX)
            DI0 = DNUM / DEN
END IF
FNDBESSI0 = DI0
END DEF

DEF FNDBESSI1 (DX)      'modified Bessel function I1
STATIC DEN, DF, DI1, DNUM
STATIC DY, DY2, I

IF DX < 3.75 THEN
            DY = DX / 3.75#
            DY2 = DY * DY
            DI1 = .5#: DF = 1#
            FOR I = 1 TO 6
              DF = DF * DY2
              DI1 = DI1 + DF * DA1(I)
            NEXT I
            DI1 = DI1 * DX
ELSE
             DY = 3.75# / DX
             DNUM = DA1(7): DF = 1#
             FOR I = 8 TO 15
               DF = DF * DY
               DNUM = DNUM + DF * DA1(I)
             NEXT I
             DEN = SQR(DX) * EXP(-DX)
             DI1 = DNUM / DEN
END IF
FNDBESSI1 = DI1
END DEF

DEF FNDBESSK0 (DX)     'modified Bessel function K0
STATIC DBX, DEN, DF, DI0, DK0
STATIC DNUM, DQXBD, DSOM, I

IF DX < 2 THEN
        DQXBD = DX * DX / 4#
        DI0 = FNDBESSI0(DX)
        DK0 = -LOG(DX / 2#) * DI0 + DB0(1)
        DSOM = 0: DF = 1#
        FOR I = 2 TO 7
          DF = DF * DQXBD
          DSOM = DSOM + DF * DB0(I)
        NEXT I
        DK0 = DK0 + DSOM
ELSE
         DBX = 2# / DX
         DNUM = DB0(8): DF = 1#
```

```
        FOR I = 9 TO 14
          DF = DF * DBX
          DNUM = DNUM + DF * DB0(I)
        NEXT I
        DEN = SQR(DX) * EXP(DX)
        DK0 = DNUM / DEN
END IF
FNDBESSK0 = DK0
END DEF

DEF FNDBESSK1 (DX)     'modified Bessel function K1
STATIC DBX, DEN, DF, DK1
STATIC DNUM, DQXBD, DSOM, I

IF DX < 2 THEN
        DQXBD = DX * DX / 4#
        DK1 = FNDBESSI1(DX) * LOG(DX / 2#) * DX + 1#
        DSOM = 0: DF = 1#
        FOR I = 1 TO 6
          DF = DF * DQXBD
          DSOM = DSOM + DF * DB1(I)
        NEXT I
        DK1 = (DK1 + DSOM) / DX
ELSE
        DBX = 2# / DX
        DNUM = 1.25331414#: DF = 1#
        FOR I = 7 TO 12
          DF = DF * DBX
          DNUM = DNUM + DF * DB1(I)
        NEXT I
        DEN = SQR(DX) * EXP(DX)
        DK1 = DNUM / DEN
END IF
FNDBESSK1 = DK1
END DEF

DEF FNDP (X)      'Laplace transform of the well function
STATIC DA, DARG, DB, DDEN, DNUM, DSQX, DX

DX = CDBL(X)
DSQX = SQR(DX)
DARG = DSQX
DNUM = FNDBESSK0(DARG)
DB = DNUM
DA = CDBL(CD) * DX * DB
DDEN = DX * (DA + DSQX * FNDBESSK1(DSQX))
FNDP = DNUM / DDEN
END DEF
```

```
DEF FNDWELLFUNC (X)      'Stehfest  algorithm  for numerical
'inversion of the Laplace transform
STATIC DA, DFA, DPIA, DS, DT, I

DT = CDBL(X)
DFA = 0: DA = LOG(2#) / DT
FOR I = 1 TO 10
  DS = CDBL(I) * DA
  DPIA = FNDP(DS)
  DFA = DFA + DV(I) * DPIA
NEXT I
DFA = DFA * DA
FNDWELLFUNC = 2# * DFA
END DEF

'                  *** MAIN PROGRAM ***
'                    * DATA INPUT *

CLS
PRINT TAB(27); "*** PROGRAM LARGWELL ***"
PRINT
PRINT TAB(8); "Program for automated interpretation of constant discharge pumping"
PRINT TAB(8); "tests carried  out in  large  diameter  wells in confined aquifers"
PRINT TAB(8); "under transient flow"
PRINT
PRINT TAB(8); : INPUT "File name ..... ", CFIC$
OPEN "I", 1, CFIC$
INPUT #1, N
FOR I = 1 TO N
  INPUT #1, TI(I), SI(I)
NEXT I
CLOSE

PRINT TAB(8); : INPUT "Discharge in m3/day (1), m3/min (2) or l/s (3) "; IUQ
IF IUQ = 1 THEN
FQ = 1
ELSEIF IUQ = 2 THEN
FQ = 1440
ELSE
FQ = 86.4
END IF
PRINT TAB(8); : INPUT "Time in min (1) or days (2) "; IUT
IF IUT = 1 THEN TD = 1440 ELSE TD = 1
FOR I = 1 TO N
  TD(I) = TI(I) / TD
NEXT I
PRINT
PRINT TAB(8); : INPUT "Enter the discharge ", Q
Q = Q * FQ
```

```
PRINT TAB(8); : INPUT "Enter the radius of the well casing (m)... ", RC
PRINT TAB(8); : INPUT "Enter the well radius (m)..... ", RW
PRINT TAB(8); : INPUT "Your guess for storage coefficient ..... ", S

'               * MINIMIZATION METHOD *

ITER = 0: RMS = 1: RMSOLD = 0
SDQ = 0: SQDOLD = 0
QT = Q / DPIX4
UX = RW * RW
CDF = RC * RC / (2# * UX)
SUL = SI(N)
40 IF RMS > .1 THEN GOSUB bissection
UDS = S / 100: UDT = T / 100
ITER = ITER + 1
CLS : PRINT STRING$(3, 10)
PRINT TAB(10); USING "Iteration number###"; ITER
PRINT TAB(10); USING "Transmissivity = #####.#     Storage coef.  = #.######"; T; S
PRINT
IF ITER = 1 THEN GOTO 45
PRINT TAB(10); "Mean square deviation:"
PRINT TAB(10); "   in this iteration: ..... "; RMS
IF ITER = 2 THEN GOTO 45
PRINT TAB(10); "   in former iteration: ..... "; RMSOLD
45 PRINT
SQDOLD = SQD: SQD = 0
RMSOLD = RMS: RMS = 0
PRINT TAB(10); " Obs. number   Obs. drawdown   Calcul. drawdown"
W$ = "   ###         ###.###         ###.###"
PRINT
FOR J = 1 TO N
    U = TD(J) * T / (UX * S)
    CD = CDF / S
    DW = FNDWELLFUNC(U)
    SC(J) = QT * DW / T
    U = TD(J) * (T + UDT) / (UX * S)
    DWP = FNDWELLFUNC(U)
    U = TD(J) * (T - UDT) / (UX * S)
    DWM = FNDWELLFUNC(U)
    UT = (-SC(J) + QT * (DWP - DWM) / (2# * UDT)) / T
    PRINT TAB(10); USING W$; J; SI(J); SC(J)
    U = TD(J) * T / (UX * (S + UDS))
    CD = CDF / (S + UDS)
    DWP = FNDWELLFUNC(U)
    U = TD(J) * T / (UX * (S - UDS))
    CD = CDF / (S - UDS)
    DWM = FNDWELLFUNC(U)
    US = QT * (DWP - DWM) / (2# * UDS) / T
```

```
    Y(J) = SI(J) - SC(J)
    SQD = SQD + Y(J) * Y(J)
    DA(J, 1) = US
    DA(J, 2) = UT
NEXT J
FOR I = 1 TO 2
  FOR J = 1 TO 2
    DAT(I, J) = 0
    FOR K = 1 TO N
          DAT(I, J) = DAT(I, J) + DA(K, I) * DA(K, J)
    NEXT K
  NEXT J
NEXT I
DET = DAT(1, 1) * DAT(2, 2) - DAT(1, 2) * DAT(2, 1)
DT11 = DAT(1, 1)
DAT(1, 1) = DAT(2, 2) / DET
DAT(2, 2) = DT11 / DET
DAT(1, 2) = -DAT(2, 1) / DET
DAT(2, 1) = DAT(1, 2)
FOR I = 1 TO 2
  FOR J = 1 TO N
    DH(I, J) = 0
    FOR K = 1 TO 2
          DH(I, J) = DH(I, J) + DAT(I, K) * DA(J, K)
    NEXT K
  NEXT J
NEXT I
DDS = 0: DDT = 0
FOR I = 1 TO N
   DDS = DDS + DH(1, I) * CDBL(Y(I))
   DDT = DDT + DH(2, I) * CDBL(Y(I))
NEXT I
IF DDS > 10 * S THEN DDS = 10 * S
IF DDS < -10 * S THEN DDS = -10 * S

'        * CORRECTION OF THE ESTIMATES *

S = S + DDS: T = T + DDT
IF S <= 0 THEN S = .00001
IF S > 1 THEN S = .2
IF T < 0 THEN T = .1
RMS = SQD / N
IF ABS(SQD - SQDOLD) > TOL THEN GOTO 40

CLS : LOCATE 3, 10
PRINT TAB(10); "Final results:"
PRINT
PRINT TAB(10); USING "Number of iterations ###"; ITER
PRINT TAB(10); USING "Transmissivity = #####.#     Storage coef. = #.######"; T; S
```

```
PRINT TAB(10); "Mean square deviation: "; RMS
50 PRINT TAB(10);
INPUT "Do you wish to see the matching (Y/N)? ", A$
IF A$ <> "y" AND A$ <> "n" AND A$ <> "Y" AND A$ <> "N" THEN GOTO 50
IF A$ = "y" OR A$ = "Y" THEN CALL graphic(N, Q, QT, RC, RMS, RW, S, SI(), T, TD(),
UX)
END

'                * BISSECTION METHOD *

bissection:
WOLD = 0
T2 = 10000: T1 = 1
U = TD(N) * T1 / (UX * S)
CD = CDF / S
FA = SUL - QT * FNDWELLFUNC(U) / T1
131 W = (T1 + T2) / 2
U = TD(N) * W / (UX * S)
FW = QT * FNDWELLFUNC(U) / W
FW = SUL - FW
IF FW * FA < 0 THEN T2 = W: GOTO 132
T1 = W: FA = FW
132 IF ABS(WOLD - W) < 10 THEN T = W: RETURN
WOLD = W
GOTO 131

SUB graphic (N, Q, QT, RC, RMS, RW, S, SI(), T, TD(), UX) STATIC

'              * SCREEN GRAPHIC CONSTRUCTION *

PRINT : PRINT TAB(10);
130 INPUT "Number of cycles in the xx axis [= time (maximum 5)] ..... ", NX
IF NX = 0 THEN NX = 5
IF NX > 5 OR NX < 1 THEN GOTO 130
PRINT TAB(10);
INPUT "  Enter the minimum time ..... ", XM
XP = 10 ^ (FNALD(XM) + NX)
PRINT : PRINT TAB(10);
140 INPUT "Number of cycles in the yy axis [= drawd. (maximum 3)] .... ", NY
IF NY = 0 THEN NY = 3
IF NY > 3 OR NY < 1 THEN GOTO 140
PRINT TAB(10);
INPUT "  Enter the minimum drawdown ..... ", YM
YP = 10 ^ (FNALD(YM) + NY)
IXP = FNALD(XP)
IYP = FNALD(YP)
IXM = FNALD(XM)
IYM = FNALD(YM)
```

```
AMP = 1#
IF NX = 2 AND NY = 1 THEN AMP = 2#
IF (NX = 2 OR NX = 3) AND NY = 2 THEN AMP = 1.5#
IF NX = 3 AND NY = 1 THEN AMP = 1.6#
IF NX = 4 AND (NY = 1 OR NY = 2) THEN AMP = 1.2#
CLS
SCREEN 2
X0 = 60
Y0 = 30
FOR I = 0 TO NY
  X1 = X0 + NX * 110 * AMP
  Y1 = Y0 + I * 45 * AMP
  LINE (X0, Y1)-(X1, Y1)
NEXT I
FOR I = 0 TO NX
  X1 = X0 + I * 110 * AMP
  Y1 = Y0 + NY * 45 * AMP
  LINE (X1, Y0)-(X1, Y1)
NEXT I
FOR J = 0 TO NX - 1
  AB = 10 ^ J
  FOR I = 1 TO 9
    X = X0 + 110 * FNALD(I * AB) * AMP
    LINE (X, 30)-(X, 35)
    LINE (X, 30 + 45 * NY * AMP)-(X, 25 + 45 * NY * AMP)
  NEXT I
NEXT J
FOR J = 0 TO NY - 1
  AB = 10 ^ J
  FOR I = 1 TO 9
    Y = -45 * FNALD(I * AB) * AMP + Y0 + 45 * NY * AMP
    LINE (60, Y)-(65, Y)
    LINE (60 + 110 * NX * AMP, Y)-(55 + 110 * NX * AMP, Y)
  NEXT I
NEXT J
FOR I = 0 TO NY
  LOCATE 5 + (5.3 * I * AMP), 3: PRINT "10"
NEXT I
FOR J = 0 TO NX
  LOCATE 7 + 5.3 * NY * AMP, 8 + 13.5 * J * AMP: PRINT "10"
NEXT J
FOR I = 0 TO NY
  LOCATE 4 + 5.3 * I * AMP, 5: PRINT IYP - I
NEXT I
FOR J = 0 TO NX
  LOCATE 6 + 5.3 * NY * AMP, 10 + 13.5 * J * AMP: PRINT IXP - NX + J
NEXT J
LOCATE 8, 2: PRINT "s"
LOCATE 7 + 5.3 * NY * AMP, 2 + 13.5 * NX * AMP: PRINT "t"
```

```
AXO = 60
AYO = 30 + NY * 45 * AMP

'* Data projection *

FOR I = 1 TO N
  TE = TD(I) * 1440: SF = SI(I)
  IF TE = 0 OR SF = 0 THEN GOTO 145
  IF TE < XM OR TE > XP THEN GOTO 145
  IF SF < YM OR SF > YP THEN GOTO 145
  AX = AXO + (FNALD(TE) - IXM) * 110 * AMP
  AY = AYO - (FNALD(SF) - IYM) * 45 * AMP
  PSET (AX, AY)
  CIRCLE (AX, AY), 3, 8
145 NEXT I

'* Curve Projection *

U = XM
IFL = 0
WHILE U < XP
  AX = AXO + (FNALD(U) - IXM) * 110 * AMP
  TU = T * U / (1440# * S * UX)
  WU = CSNG(FNDWELLFUNC(TU))
  REB = QT * WU / T
  IF REB < YM THEN GOTO 160
  IF REB > YP THEN GOTO 170
  AY = AYO - (FNALD(REB) - IYM) * 45 * AMP
  IF IFL = 0 THEN IFL = 1: GOTO 150
  LINE (AX1, AY1)-(AX, AY)
150 AX1 = AX: AY1 = AY
160 AG = U * .2: U = U + AG
WEND
U = XP
AX = AXO + (FNALD(U) - IXM) * 110 * AMP
TU = T * U / (1440# * S * UX)
WU = CSNG(FNDWELLFUNC(TU))
REB = QT * WU / T
AY = AYO - (FNALD(REB) - IYM) * 45 * AMP
LINE (AX1, AY1)-(AX, AY)
170 LOCATE 1, 8: PRINT "Matching of the PAPADOPULOS-COOPER (1967) curve"
LOCATE 2, 8: PRINT "T = "; USING "#####.#"; T
LOCATE 2, 24: PRINT "S = "; USING ".#######"; S
LOCATE 2, 41: PRINT "ALPHA = "; USING ".#######"; RC * RC * S / RW / RW
LOCATE 3, 8: PRINT "Q = "; USING "#####"; Q
LOCATE 3, 23: PRINT "Mean square deviation = "; USING "##.###########"; RMS

END SUB
```