Techniques of Water-Resources Investigations of the United States Geological Survey

Chapter B3

TYPE CURVES FOR SELECTED PROBLEMS OF FLOW TO WELLS IN CONFINED AQUIFERS

By J. E. Reed

Book 3
APPLICATIONS OF HYDRAULICS

starting point. The integral over this interval is approximated by a trapezoidal sum using 500 subdivisions of the interval. A new interval is then constructed using the previous end point as a new starting point and a new ending point equal to 10 times the new starting point. This new interval is again evaluated by a trapezoidal sum of 500 segments. This summation procedure over intervals that are successively an order of magnitude larger continues until either $t'=t$ or $(r^2S/4Tt')+(K't/Sb')>101$. Input to this program consists of cards coded in specific formats. Readers unfamiliar with FORTRAN formats should refer to a FORTRAN language manual. Input consists of one or more groups of data, each group consisting of the following. First, one card containing the beginning time of the period of analysis in columns 1–10, coded in format E10.3; the ending time coded in columns 1311–20, in format E10.3; and a discharge index (a number from 1 through 5) coded in column 25, in format I1; and a reference discharge, *QR*, coded in columns 31–40, in format E10.3. The discharge index, *IQ*, selects a discharge function, $Q(t)$, in the following manner. If $IQ=1$, the discharge function is exponentially decreasing,

$$Q(t)=Q_s\left[1+\delta\exp(-t/t^*)\right].$$

This is case (a) of Hantush (1964a, p. 343). If $IQ=2$, the discharge function is hyperbolically decreasing,

$$Q(t)=Q_s\left[1+\delta/(1+t/t^*)\right].$$

This is case (b) of Hantush (1964a, p. 344). If $IQ=3$, the discharge function is the same as case (c) of Hantush (1964a, p. 344),

$$Q(t)=Q_s\left[1+\delta/\sqrt{1+t/t^*}\right].$$

If $IQ=4$, the discharge function is a fifth-degree polynomial of time,

$$Q(t)=\sum_{i=0}^{5}a_it^i.$$

If $IQ=5$, the discharge function is a piecewise-linear function of time with eight or less segments,

$$Q(t)=a_j+b_j(t-t_{j-1})$$

for $t_{j-1}<t\leq t_j,\ j=1,2,\ldots,8.$

The reference discharge, *QR*, is used to determine the form of the output from the program: If *QR* is coded as zero (or blank), the output shows t, s (as defined by eq. 4), and $Q(t)$. If a value greater than zero is coded for *QR*, the output shows $1/u$, $SO(t)$ (as defined by eq. 6), and $Q(t)/QR$.

Second, there are one or more cards containing parameters of the discharge function. If $IQ=1$, 2, or 3, then it consists of one card containing: *QST*, the ultimate steady discharge, coded in columns 1–10, in format E10.3; DELTA, a rate parameter, coded in columns 11–20, in format E10.3; TSTAR, a time parameter, coded in columns 21–30, in format E10.3. If $IQ=4$, it is one card containing the six polynomial coefficients. They are coded in the order $a_0, a_1, \ldots, a_5$, in columns 1–10; 11–20, . . . , 51–60 all in format E10.3. If $IQ=5$, then the program requires four cards, each card containing t_j, a_j, b_j, t_{j+1}, a_{j+1}, b_{j+1}; the four cards representing $j=1, 3, 5, 7$. The last part of each set of data consists of two or more cards containing coded values for: distance from pumped well, in columns 1–10; storage coefficient, in columns 11–20; transmissivity, in columns 21–30; and ratio of hydraulic conductivity to thickness for the confining bed, in columns 31–40, all in format E10.3. A blank card is used to signal the end of each set of data. Output from this program is shown in figure 11.3.

References

Abu-Zied, M. A., and Scott, V. H., 1963, Nonsteady flow for wells with decreasing discharge: Hydraulics Div. Jour., Am. Soc. Civil Engineers Proc., v. 89, no. H-13, p. 119–132.

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

Cooper, H. H., Jr., Bredehoeft, J. D., and Papadopulos, I. S., 1967, Response of a finite-diameter well to an instantaneous charge of water: Water Resources Research, v. 3, no. 1, p. 263–269

Dudley, W. W., Jr., 1970, Nonsteady inflow to a chamber

R**2*S/(4*TRANS)= 1.000E-04, K'/(S*B')= 2.500E 03, QR= 1.257E 05

1/U	1/U*10** 0 SO(T)	Q(T)/QR	1/U*10** 1 SO(T)	Q(T)/QR	1/U*10** 2 SO(T)	Q(T)/QR	1/U*10** 3 SO(T)	Q(T)/QR
1.0	0.185	1.000E 00	0.819	1.000E 00	0.842	1.000E 00	0.842	1.000E 00
1.5	0.317	1.000E 00	0.837	1.000E 00	0.842	1.000E 00	0.842	1.000E 00
2.0	0.421	1.000E 00	0.841	1.000E 00	0.842	1.000E 00	0.842	1.000E 00
3.0	0.566	1.000E 00	0.842	1.000E 00	0.842	1.000E 00	0.842	1.000E 00
5.0	0.715	1.000E 00	0.842	1.000E 00	0.842	1.000E 00	0.842	1.000E 00
7.0	0.780	1.000E 00	0.842	1.000E 00	0.842	1.000E 00	0.842	1.000E 00

R**2*S/(4*TRANS)= 1.000E-04, K'/(S*B')= 2.500E 01, QR= 1.257E 05

1/U	1/U*10** 0 SO(T)	Q(T)/QR	1/U*10** 1 SO(T)	Q(T)/QR	1/U*10** 2 SO(T)	Q(T)/QR	1/U*10** 3 SO(T)	Q(T)/QR
1.0	0.219	1.000E 00	1.805	1.000E 00	3.815	1.000E 00	4.829	1.000E 00
1.5	0.397	1.000E 00	2.167	1.000E 00	4.111	1.000E 00	4.849	1.000E 00
2.0	0.558	1.000E 00	2.427	1.000E 00	4.296	1.000E 00	4.853	1.000E 00
3.0	0.826	1.000E 00	2.793	1.000E 00	4.515	1.000E 00	4.854	1.000E 00
5.0	1.216	1.000E 00	3.244	1.000E 00	4.708	1.000E 00	4.854	1.000E 00
7.0	1.495	1.000E 00	3.530	1.000E 00	4.785	1.000E 00	4.854	1.000E 00

R**2*S/(4*TRANS)= 1.000E-04, K'/(S*B')= 2.500E-01, QR= 1.257E 05

1/U	1/U*10** 0 SO(T)	Q(T)/QR	1/U*10** 1 SO(T)	Q(T)/QR	1/U*10** 2 SO(T)	Q(T)/QR	1/U*10** 3 SO(T)	Q(T)/QR
1.0	0.219	1.000E 00	1.823	1.000E 00	4.036	1.000E 00	6.307	1.000E 00
1.5	0.398	1.000E 00	2.196	1.000E 00	4.437	1.000E 00	6.700	1.000E 00
2.0	0.560	1.000E 00	2.468	1.000E 00	4.721	1.000E 00	6.975	1.000E 00
3.0	0.829	1.000E 00	2.857	1.000E 00	5.123	1.000E 00	7.356	1.000E 00
5.0	1.223	1.000E 00	3.354	1.000E 00	5.627	1.000E 00	7.820	1.000E 00
7.0	1.507	1.000E 00	3.684	1.000E 00	5.958	1.000E 00	8.110	1.000E 00

R**2*S/(4*TRANS)= 1.000E-04, K'/(S*B')= 2.500E-03, QR= 1.257E 05

1/U	1/U*10** 0 SO(T)	Q(T)/QR	1/U*10** 1 SO(T)	Q(T)/QR	1/U*10** 2 SO(T)	Q(T)/QR	1/U*10** 3 SO(T)	Q(T)/QR
1.0	0.219	1.000E 00	1.823	1.000E 00	4.038	1.000E 00	6.332	1.000E 00
1.5	0.398	1.000E 00	2.197	1.000E 00	4.440	1.000E 00	6.737	1.000E 00
2.0	0.560	1.000E 00	2.468	1.000E 00	4.726	1.000E 00	7.024	1.000E 00
3.0	0.829	1.000E 00	2.857	1.000E 00	5.130	1.000E 00	7.429	1.000E 00
5.0	1.223	1.000E 00	3.355	1.000E 00	5.639	1.000E 00	7.939	1.000E 00
7.0	1.507	1.000E 00	3.686	1.000E 00	5.975	1.000E 00	8.275	1.000E 00

FIGURE 11.3.—Example of output from program to compute the convolution integral for a leaky aquifer.

within a thick aquitard, *in* Geological Survey research 1970: U.S. Geol. Survey Prof. Paper 700–C, p. C206–C211.

Ferris, J. G., Knowles, D. B., Brown, R. H., and Stallman, R. W., 1962, Theory of aquifer tests: U.S. Geol. Survey Water-Supply Paper 1536–E, p. E69–E174.

Hantush, M. S., 1956, Analysis of data from pumping tests in leaky aquifers: Am. Geophys. Union Trans., v. 37, no. 6, p. 702–714.

———1959, Nonsteady flow to flowing wells in leaky aquifers: Jour Geophys Research, v. 64, no. 8, p. 1043–1052.

———1960, Modification of the theory of leaky aquifers: Jour. Geophys. Research, v. 65, no. 11, p. 3713–3725.

———1961a, Drawdown around a partially penetrating well: Hydraulics Div. Jour., Am Soc. Civil Engineers Proc., p. 83–98.

———1961b, Aquifer tests on partially penetrating wells: Hydraulics Div Jour , Am. Soc. Civil Engineers Proc., p. 171–195.

———1961c, Tables of the function $M(u,\beta)$: New Mexico Inst. Mining and Technology Prof. Paper 102, 34 p.

———1961d, Tables of the function $H(u,\beta)=\int_0^\infty \frac{e^{-y}}{y}\,\mathrm{erfc}\left[\frac{\beta\sqrt{u}}{\sqrt{y(y-u)}}\right]dy$: New Mexico Inst. Mining and Technology Prof. Paper 103, 12 p.

———1961e, Tables of the function $W(u,\beta)=\int_0^\infty \frac{e^{-y-\frac{\beta^2}{4y}}}{y}dy$: New Mexico Inst. Mining and Technology Prof. Paper 104, 13 p.

———1962, Flow of ground water in sands of nonuniform thickness, 3,Flow to wells: Jour. Geophys. Research, v. 67, no. 4, p. 1527–1534.

———1964a, Hydraulics of wells, *in* Chow, Ven Te, ed., Advances in hydroscience, Volume 1: New York, Academic Press, p. 281–442.

———1964b, Drawdown around wells of variable discharge: Jour. Geophys. Research, v. 69, no. 20, p. 4221–4235.

———1967, Flow of groundwater in relatively thick leaky aquifers: Water Resources Research, v. 3, no. 2, p. 583–590.

Hantush, M. S., and Jacob, C. E., 1954, Plane potential flow of ground water with linear leakage: Am. Geophys. Union Trans., v. 35, no. 6, p. 917–936.

———1955, Non-steady radial flow in an infinite leaky aquifer: Am. Geophys. Union Trans., v. 36, no. 1, p. 95–100.

Jacob, C. E., 1946, Radial flow in a leaky artesian aquifer: Am. Geophys. Union Trans., v. 27, no. 2, p. 198–205.

Jacob, C. E., and Lohman, S. W., 1952, Nonsteady flow to a well of constant drawdown in an extensive aquifer: Am. Geophys. Union Trans., v. 33, no. 4, p. 559–569.

Lohman, S. W., 1972, Ground-water hydraulics: U.S. Geol. Survey Prof. Paper 708, 70 p.

Lohman, S. W., and others, 1972, Definitions of selected ground-water terms—revisions and conceptual refinements: U.S. Geol. Survey Water-Supply Paper 1988, 21 p.

Moench, Allen, 1971, Ground-water fluctuations in response to arbitrary pumpage: Ground Water, v. 9, no. 2, p. 4–8.

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.

———1969b, Applicability of current theories of flow in leaky aquifers: Water Resources Research, v. 5, no. 4, p. 817–829.

———1971, Transient flow of ground water to wells in multiple-aquifer systems, *in* Sea-water intrusion: Aquitards in the coastal ground water basin of Oxnard Plain, Ventura County: California Department of Water Resources Bulletin 63–4, p. 159–359.

Papadopulos, I. S., 1965, Nonsteady flow to a well in an infinite anisotropic aquifer: Internat. Assoc. Sci. Hydrology Symposium, Dubrovnik [Yugoslavia], 1965, p. 21–31.

———1967, Drawdown distribution around a large-diameter well: Natl. Symposium on Ground-Water Hydrology, San Francisco, Calif., Nov. 6–8, 1967, Proc., p. 157–168.

Papadopulos, I. S., and Cooper, H. H., Jr., 1967, Drawdown in a well of large diameter: Water Resources Research, v. 3, no. 1, p. 241–244.

Papadopulos, S. S., Bredehoeft, J. D., and Cooper, H. H., Jr., 1973, On the analysis of slug test data: Water Resources Research, v. 9, no. 4, p. 1087–1089.

Rorabaugh, M. I., 1953, Graphical and theoretical analysis of step-drawdown test of artesian wells: Am. Soc. Civil Engineers Trans., v. 79, Separate 362, 23 p.

Stallman, R. W., 1971, Aquifer-test design, observation, and data analysis: U.S. Geol. Survey Techniques Water-Resources Inv., book 3, chap. B1, 26 p.

Theis, C. V., 1935, The relation between the lowering of the piezometric surface and the rate and duration of discharge of a well using ground-water storage: Am. Geophys. Union Trans., v. 16, p. 519–524.

Walton, W. C., 1962, Selected analytical methods for well and aquifer evaluation: Illinois State Water Survey Bull. 49, 81 p.

Weeks, E. P., 1964, Field methods for determining vertical permeability and aquifer anisotropy: U.S. Geol. Survey Prof. Paper 501–D, p. D193–D198.

———1969, Determining the ratio of horizontal to vertical permeability by aquifer-test analysis: Water Resources Research, v. 5, no. 1, p. 196–214.

Wenzel, L. K., 1942, Methods for determining permeability of water-bearing materials, with special reference to discharging-well methods: U.S. Geol. Survey Water-Supply Paper 887, 192 p

Werner, P. W., 1946, Notes on flow-time effects in the great artesian aquifers of the earth: Am Geophys. Union Trans., v. 27, no. 5, p. 687–708.

SUPPLEMENTAL DATA

TABLE 2.1.—*Listing of program for partial penetration in a nonleaky artesian aquifer*

```
C     *******************************************************************
C
C     PURPOSE
C        TO COMPUTE TYPE CURVE FUNCTION VALUES FOR PARTIAL PENETRATION
C        IN A NONLEAKY AQUIFER USING EQUATIONS 1 AND 9A OF HANTUSH,M.S.,
C        1961,DRAWDOWN AROUND A PARTIALLY PENETRATING WELL: HYDRAULIC
C        DIV. JOUR., AM. SOC. CIVIL ENGINEERS PROC., P. 83-98.
C     INPUT DATA
C        1 CARD  -  FORMAT (3F5.1,I5,2E10.4)
C              B - AQUIFER THICKNESS
C              L - DEPTH, BELOW TOP OF AQUIFER, TO BOTTOM OF PUMPING
C              WELL SCREEN
C              D - DEPTH, BELOW TOP OF AQUIFER, TO TOP OF PUMPING WELL
C              SCREEN
C              NUM - NUMBER OF OBSERVATION WELLS OR PIEZOMETERS TIMES
C              NUMBER OF VALUES OF KZ/KR.
C              SMALL - SMALLEST VALUE OF 1/U FOR WHICH COMPUTATION IS
C              DESIRED
C              LARGE - LARGEST  VALUE OF 1/U FOR WHICH COMPUTATION IS
C              DESIRED
C        NUM CARDS (ONE FOR EACH OBS. WELL OR PIEZOMETER AND FOR EACH  1
C              VALUE OF R*SQRT(KZ/KR), - FORMAT (3F5.1)
C              R - RADIAL DISTANCE FROM PUMPED WELL TIMES SQRT(KZ/KR).
C              LPRIME - DEPTH, BELOW TOP OF AQUIFER, TO BOTTOM OF OBS.
C              WELL SCREEN (ZERO FOR PIEZOMETER)
C              DPRIME - DEPTH, BELOW TOP OF AQUIFER, TO TOP OF OBS. WELL
C              SCREEN (TOTAL DEPTH FOR PIEZOMETER)
C     SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C        DQL12,SERIES,BESK,FCT,L,F,EXPI
C
C     *******************************************************************
      REAL*8 U
      REAL*4 L,LB,LPB,LPRIME,LARGE
      DIMENSION ARRAY(13,12), IARG(12), ARG(13), A(12), C(12)
      DATA ARG/1.,1.2,1.5,2.,2.5,3.,3.5,4.,5.,6.,7.,8.,9./
      DATA A/12*'  N*'/,C/12*'10**'/
      IRD=5
      IPT=6
      READ (IRD,6) B,L,D,NUM,SMALL,LARGE
      LB=L/B
      DB=D/B
      IBEGIN=ALOG10(SMALL)
      IEND=ALOG10(LARGE)+1.
      JLIMIT=IEND-IBEGIN
      IF (JLIMIT.GT.12) JLIMIT=12
      DO 5 K=1,NUM
      READ (IRD,6) R,LPRIME,DPRIME
      RB=R/B
      LPB=LPRIME/B
      DPB=DPRIME/B
      DO 1 I=1,13
      ARGI=ARG(I)
      DO 1 J=1,JLIMIT
      IARG(J)=IBEGIN+J-1
      Y=ARGI*10.**(IBEGIN+J-1)
      U=1./Y
      X=U
      CALL EXPI(X,WU,DUMMY)
    1 ARRAY(I,J)=WU+F(U,RB,LB,DB,LPB,DPB)
      IF (LPB-0.) 2,2,3
```

TABLE 2.1.—*Listing of program for partial penetration in a nonleaky artesian aquifer*—Continued

```
    2 WRITE (IPT,7) DPB,RB,LB,DB
      GO TO 4
    3 WRITE (IPT,8) LPB,DPB,RB,LB,DB
    4 WRITE (IPT,9) (A(I),C(I),IARG(I),I=1,JLIMIT)
      DO 5 I=1,13
      WRITE (IPT,10) ARG(I),(ARRAY(I,J),J=1,JLIMIT)
    5 CONTINUE
      STOP
C
C
    6 FORMAT (3F5.1,I5,2E10.4)
    7 FORMAT ('1','W(U)+F(U,R/B,L/B,D/B,Z/B), Z/B=',F5.2,', SQRT(KZ/KR)*
     1R/B=',F5.2,', L/B=',F5.2,', D/B=',F5.2,', U=1/N')
    8 FORMAT ('1','W(U)+F(U,R/B,L/B,D/B,L''/B,D''/B), L''/B=',F5.2,', D'
     1'/B=',F5.2,', SQRT(KZ/KR)*R/B=',F5.2,', L/B=',F5.2,', D/B=',F5.2,'
     2, U=1/N')
    9 FORMAT ('0',2X,'N',1X,12(2A4,I2))
   10 FORMAT ((' ',F4.1,12(F9.4,1X)))
      END
      REAL FUNCTION F*4(U,RB,LB,DB,LPB,DPB)
C     ******************************************************************
C
C     FUNCTION F
C
C     PURPOSE
C        TO COMPUTE DEPARTURES FROM THEIS CURVE CAUSED BY PARTIAL
C        PENETRATION OF PUMPED WELL.
C     USAGE
C        F(U,RB,LB,DB,LPB,DPB)
C     DESCRIPTION OF PARAMETERS
C        ALL REAL, U DOUBLE PRECISION
C        U = R**2*S/4*T*TIME (RADIAL DISTANCE SQUARED * STORAGE
C             COEFFICIENT / 4*TRANSMISSIVITY * TIME
C        RB = R/B ( RADIAL DISTANCE / AQUIFER THICKNESS )
C        LB = L/B ( FRACTION OF AQUIFER PENETRATED BY PUMPED WELL)
C        DB = D/B ( FRACTION OF AQUIFER ABOVE PUMPED WELL SCREEN)
C        LPB = L'/B (FRACTION OF AQUIFER PENETRATED BY OBS. WELL, ZERO
C            FOR PIEZOMETER)
C        DPB = D'/B (FRACTION OF AQUIFER ABOVE OBS. WELL SCREEN, TOTAL
C            DEPTH FOR PIEZOMETER)
C     SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C        DQL12,SERIES,BESK,FCT,L
C     METHOD
C        SUMS THE SERIES THROUGH N*PI*R/B EQ 20
C
C     ******************************************************************
      REAL*8 U,V
      REAL*4 L,N,LB,LPB
      SUM=0.
      N=0.
      PIRB=3.141593*RB
      PILB=3.141593*LB
      PIDB=3.141593*DB
      IF (LPB=0.) 1,1,4
C     CHECKS FOR WELL OR PIEZOMETER
    1 PIZB=3.141593*DPB
    2 N=N+1.
      V=N*PIRB/2.
      IF (V.GT.10.) GO TO 3
C     TRUNCATES SERIES WHEN V>10
      X=L(U,V)/N
```

TABLE 2.1.—*Listing of program for partial penetration in a nonleaky artesian aquifer*—Continued

```
      SUM=SUM+(SIN(N*PILB)-SIN(N*PIDB))*COS(N*PIZB)*X
      GO TO 2
    3 F=.6366198*SUM/(LB-DB)
      GO TO 7
    4 PILPB=3.141593*LPB
      PIDPB=3.141593*DPB
    5 N=N+1
      V=N*PIRB/2.
      IF (V.GT.10.) GO TO 6
C     TRUNCATES SERIES WHEN V>10
      X=L(U,V)/N
      SUM=SUM+(SIN(N*PILB)-SIN(N*PIDB))*(SIN(N*PILPB)-SIN(N*PIDPB))*X/N
      GO TO 5
    6 F=.2026424*SUM/((LB-DB)*(LPB-DPB))
    7 RETURN
      END

      REAL FUNCTION L*4(U,V)
C     ******************************************************************
C
C     FUNCTION L
C
C     PURPOSE
C        TO COMPUTE THE INTEGRAL( EXP(-Y-V**2/Y)/Y) SUMMED OVER Y FROM
C        U TO INFINITY(WELL FUNCTION FOR LEAKY AQUIFERS).
C     DESCRIPTION OF PARAMETERS
C        BOTH DOUBLE PRECISION
C        U = R**2*S/4*T*TIME (RADIAL DISTANCE SQUARED * STORAGE
C             COEFFICIENT / 4*TRANSMISSIVITY * TIME
C        V = R/2*SQRT(K'/(T*B'))--ONE-HALF RADIAL DISTANCE*SQUARE ROOT
C             (HYD. COND. OF CONFINING BED/TRANSMISSIVITY*THICKNESS
C             OF CONFINING BED)
C     SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C        DQL12,SERIES,BESK,FCT
C     METHOD
C        IN THE FOLLOWING F=EXP(-Y-V**2/Y)/Y
C      (1) U>=1, USES A GAUSSIAN-LAGUERRE QUADRATURE FORMULA TO
C        EVALUATE INTEGRAL(F) FROM U TO INF.
C      (2) V**2<U<1, USES THE G-L QUADRATURE TO EVALUATE INTEGRAL(F)
C        FROM ONE TO INF AND A SERIES EXPANSION TO EVALUATE INTEGRAL(F)
C        FROM U TO ONE.
C      (3) U<1, U<=V**2, USES THE REPRESENTATION INTEGRAL(F) FROM U
C        TO INF. = 2*K0(2*V)-INTEGRAL(F) FROM V**2/U TO INF.
C        EVALUATES THE ZERO ORDER MODIFIED BESSEL FUNCTION OF SECOND
C        KIND WITH IBM SUBROUTINE, EVALUATES INTEGRAL BY G-L QUAD.
C
C     ******************************************************************
      EXTERNAL FCT
      REAL*8 U,V,Z,F,VV,SERIES
      COMMON /C1/ VV,Z
      VV=V
      IF (U-1.) 1,2,2
C     CHECKS IF U<1
    1 Z=V*V/U
      IF (Z-1.) 3,4,4
C     CHECKS IF V**2/U < 1
    2 Z=U
      CALL DQL12(FCT,F)
      L=F
C     INTEGRAL U TO INF. EVALUATED BY GAUSS-LAGUERRE QUADRATURE
      GO TO 5
    3 Z=1.
```

TABLE 2.1.—*Listing of program for partial penetration in a nonleaky artesian aquifer*—Continued

```
      CALL DQL12(FCT,F)
      L=F+SERIES(U,V)
C     INTEGRAL 1 TO INF. BY G-L QUAD., INTEGRAL U TO 1 BY SERIES EXP.
      GO TO 5
    4 TWOV=2.*V
      CALL BESK(TWOV,0,BK,IER)
      CALL DQL12(FCT,F)
      L=2.*BK-F
C     2K0(2V)-INTEGRAL V**2/U TO INF.
    5 RETURN
      END

      REAL FUNCTION SERIES*8(U,V)
C     ******************************************************************
C
C     FUNCTION SERIES
C
C     PURPOSE
C        TO EVALUATE S(1)-S(U), WHERE S IS A SERIES EXPANSION OF
C        INTEGRAL(EXP(-Y-V**2/Y)DY/Y) GIVEN BY: S= SUM, M=0 TO INFINITY,
C        (F(M)*SUM, N=0 TO INF.,(V**(2*N)/((N!)*(M+N)!)) WHERE F(M)=
C        LOG(U) IF M=0 AND = ((-1)**M/M)*(U**M-(V**2/U)**M) IF M>0.
C     DESCRIPTION OF PARAMETERS
C        BOTH DOUBLE PRECISION
C        U - R**2*S/4*T*TIME (RADIAL DISTANCE SQUARED * STORAGE
C            COEFFICIENT / 4*TRANSMISSIVITY * TIME
C        V - R/2*SQRT(K'/(T*B'))--ONE-HALF RADIAL DISTANCE*SQUARE ROOT
C            (HYD. COND. OF CONFINING BED/TRANSMISSIVITY*THICKNESS
C            OF CONFINING BED)
C     SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C        NONE
C     METHOD
C        SUMMATION IS TERMINATED FOR THE INNER SERIES WHEN A TERM
C        BECOMES LESS THAN 5.E-7/N AND FOR OUTER SERIES WHEN A TERM
C        BECOMES LESS THAN 5.E-7
C
C     ******************************************************************
      REAL*8 DLOG,DABS,S(2),VUM,UU
      REAL*8 TEST,U,UM,EM,EN,SUM1,SUM,SIGN,V,VSQ,VSQU,RMUL,TERM,TERM1
      TEST=5.D-07
      VSQ=V*V
      UU=U
      DO 6 I=1,2
C     EVALUATES SERIES FOR LOWER LIMIT = U AND UPPER LIMIT = 1
      IF (I.EQ.2) U=1.
      UM=1.
      EM=-1.
      SUM1=0.
      SIGN=-1.
      VUM=1.
      VSQU=VSQ/U
    1 EM=EM+1.
      IF (EM-.1) 2,3,3
C     CHECKS FOR M=0
    2 RMUL=DLOG(U)
      TERM1=1.
      GO TO 4
    3 UM=UM*U
      IF (VUM.LT.1.D-30) VUM=0.
      VUM=VUM*VSQU
      RMUL=(UM-VUM)/EM
      TERM1=TERM1/EM
```

TABLE 2.1.—*Listing of program for partial penetration in a nonleaky artesian aquifer*—Continued

```
    4 SIGN=-SIGN
      SUM=TERM1
      TERM=TERM1
      EN=0.
    5 EN=EN+1.
      TERM=TERM*VSQ/(EN*(EN+EM))
      SUM=SUM+TERM
      IF (TEST.LE.DABS(RMUL*EN*TERM)) GO TO 5
C     TRUNCATES INNER SERIES IF OUTER TERM*N*INNER TERM < 5.E-7
      SUM1=SUM1+SIGN*RMUL*SUM
      IF (EM.LT..1) GO TO 1
      IF (TEST.LE.DABS(RMUL*SUM)) GO TO 1
C     TRUNCATES OUTER SERIES IF OUTER TERM*INNER SUM < 5.E-7
    6 S(I)=SUM1
      U=UU
      SERIES=S(2)-S(1)
      RETURN
      END
      REAL FUNCTION FCT*8(X)
C     *****************************************************************
C
C     FUNCTION FCT
C
C     PURPOSE
C        TO COMPUTE FCT(X)=EXP(-Z-V**2/(X+Z))/(X+Z)
C     DESCRIPTION OF PARAMETERS
C        X - THE DOUBLE PRECISION VALUE OF X FOR WHICH FCT IS COMPUTED
C     SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C        NONE
C     METHOD
C        FORTRAN EVALUATION OF FUNCTION
C
C     *****************************************************************
      REAL*8 X,V,Z,P,DEXP
      COMMON /C1/ V,Z
      IF (X) 1,2,2
    1 FCT=0.
      GO TO 4
    2 P=Z+V**2/(X+Z)
      IF (P-5.D1) 3,3,1
    3 FCT=DEXP(-P)/(X+Z)
    4 RETURN
      END
      SUBROUTINE DQL12(FCT,Y)
C
C     .................................................................
C
C        SUBROUTINE DQL12
C
C        PURPOSE
C           TO COMPUTE INTEGRAL(EXP(-X)*FCT(X), SUMMED OVER X
C                             FROM 0 TO INFINITY).
C
C        USAGE
C           CALL DQL12 (FCT,Y)
C           PARAMETER FCT REQUIRES AN EXTERNAL STATEMENT
C
C        DESCRIPTION OF PARAMETERS
C           FCT    - THE NAME OF AN EXTERNAL DOUBLE PRECISION FUNCTION
C                    SUBPROGRAM USED.
C           Y      - THE RESULTING DOUBLE PRECISION INTEGRAL VALUE.
```

TABLE 2.1.—*Listing of program for partial penetration in a nonleaky artesian aquifer*—Continued

```
C
C        REMARKS
C           NONE
C
C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C           THE EXTERNAL DOUBLE PRECISION FUNCTION SUBPROGRAM FCT(X)
C           MUST BE FURNISHED BY THE USER.
C
C        METHOD
C           EVALUATION IS DONE BY MEANS OF 12-POINT GAUSSIAN-LAGUERRE
C           QUADRATURE FORMULA, WHICH INTEGRATES EXACTLY,
C           WHENEVER FCT(X) IS A POLYNOMIAL UP TO DEGREE 23.
C           FOR REFERENCE, SEE
C           SHAO/CHEN/FRANK, TABLES OF ZEROS AND GAUSSIAN WEIGHTS OF
C           CERTAIN ASSOCIATED LAGUERRE POLYNOMIALS AND THE RELATED
C           GENERALIZED HERMITE POLYNOMIALS, IBM TECHNICAL REPORT
C           TR00.1100 (MARCH 1964), PP.24-25.
C
C     ..................................................................
C
C
C
      DOUBLE PRECISION X,Y,FCT
C
      X=.3709912104446692 D2
      Y=.8148077467426241 D-15*FCT(X)
      X=.2848796725098400 D2
      Y=Y+.3061601635035021 D-11*FCT(X)
      X=.2215109037939701 D2
      Y=Y+.1342391030515004 D-8*FCT(X)
      X=.1711685518746226 D2
      Y=Y+.1668493876540910 D-6*FCT(X)
      X=.1300605499330635 D2
      Y=Y+.8365055856819800 D-5*FCT(X)
      X=.9621316842456867 D1
      Y=Y+.2032315926629994 D-3*FCT(X)
      X=.6844525453115177 D1
      Y=Y+.2663973541865316 D-2*FCT(X)
      X=.4599227639418348 D1
      Y=Y+.2010238115463410 D-1*FCT(X)
      X=.2833751337743507 D1
      Y=Y+.9044922221168093 D-1*FCT(X)
      X=.1512610269776419 D1
      Y=Y+.2440820113198776 D0*FCT(X)
      X=.6117574845151307 D0
      Y=Y+.3777592758731380 D0*FCT(X)
      X=.1157221173580207 D0
      Y=Y+.2647313710554432 D0*FCT(X)
      RETURN
      END
      SUBROUTINE BESK(X,N,BK,IER)
C
C     ..................................................................
C
C        SUBROUTINE BESK
C
C           COMPUTE THE K BESSEL FUNCTION FOR A GIVEN ARGUMENT AND ORDER
C
C        USAGE
C           CALL BESK(X,N,BK,IER)
```

TABLE 2.1.—*Listing of program for partial penetration in a nonleaky artesian aquifer*—Continued

```
C
C       DESCRIPTION OF PARAMETERS
C          X   -THE ARGUMENT OF THE K BESSEL FUNCTION DESIRED
C          N   -THE ORDER OF THE K BESSEL FUNCTION DESIRED
C          BK  -THE RESULTANT K BESSEL FUNCTION
C          IER-RESULTANT ERROR CODE WHERE
C             IER=0  NO ERROR
C             IER=1  N IS NEGATIVE
C             IER=2  X IS ZERO OR NEGATIVE
C             IER=3  X .GT. 170, MACHINE RANGE EXCEEDED
C             IER=4  BK .GT. 10**70
C
C       REMARKS
C          N MUST BE GREATER THAN OR EQUAL TO ZERO
C
C       SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C          NONE
C
C       METHOD
C          COMPUTES ZERO ORDER AND FIRST ORDER BESSEL FUNCTIONS USING
C          SERIES APPROXIMATIONS AND THEN COMPUTES N TH ORDER FUNCTION
C          USING RECURRENCE RELATION.
C          RECURRENCE RELATION AND POLYNOMIAL APPROXIMATION TECHNIQUE
C          AS DESCRIBED BY A.J.M.HITCHCOCK,'POLYNOMIAL APPROXIMATIONS
C          TO BESSEL FUNCTIONS OF ORDER ZERO AND ONE AND TO RELATED
C          FUNCTIONS', M.T.A.C., V.11,1957,PP.86-88, AND G.N. WATSON,
C          'A TREATISE ON THE THEORY OF BESSEL FUNCTIONS', CAMBRIDGE
C          UNIVERSITY PRESS, 1958, P. 62
C
C     ..................................................................
C
      DIMENSION T(12)
      BK=.0
      IF(N)10,11,11
   10 IER=1
      RETURN
   11 IF(X)12,12,20
   12 IER=2
      RETURN
   20 IF(X-170.0)22,22,21
   21 IER=3
      RETURN
   22 IER=0
      IF(X-1.)36,36,25
   25 A=EXP(-X)
      B=1./X
      C=SQRT(B)
      T(1)=B
      DO 26 L=2,12
   26 T(L)=T(L-1)*B
      IF(N-1)27,29,27
C
C     COMPUTE K0 USING POLYNOMIAL APPROXIMATION
C
   27 G0=A*(1.2533141-.1566642*T(1)+.08811128*T(2)-.09139095*T(3)
     2+.1344596*T(4)-.2299850*T(5)+.3792410*T(6)-.5247277*T(7)
     3+.5575368*T(8)-.4262633*T(9)+.2184518*T(10)-.06680977*T(11)
     4+.009189383*T(12))*C
      IF(N)20,28,29
   28 BK=G0
      RETURN
```

TABLE 2.1.—*Listing of program for partial penetration in a nonleaky artesian aquifer*—Continued

```
C                                                                       BESK 720
C     COMPUTE K1 USING POLYNOMIAL APPROXIMATION                         BESK 730
C                                                                       BESK 740
   29 G1=A*(1.2533141+.4699927*T(1)-.1468583*T(2)+.1280427*T(3)         BESK 750
     2-.1736432*T(4)+.2847618*T(5)-.4594342*T(6)+.6283381*T(7)          BESK 760
     3-.6632295*T(8)+.5050239*T(9)-.2581304*T(10)+.07880001*T(11)       BESK 770
     4-.01082418*T(12))*C                                               BESK 780
      IF(N-1)20,30,31                                                   BESK 790
   30 BK=G1                                                             BESK 800
      RETURN                                                            BESK 810
C                                                                       BESK 820
C     FROM K0,K1 COMPUTE KN USING RECURRENCE RELATION                   BESK 830
C                                                                       BESK 840
   31 DO 35 J=2,N                                                       BESK 850
      GJ=2.*(FLOAT(J)-1.)*G1/X+G0                                       BESK 860
      IF(GJ-1.0E70)33,33,32                                             BESK 870
   32 IER=4                                                             BESK 880
      GO TO 34                                                          BESK 890
   33 G0=G1                                                             BESK 900
   35 G1=GJ                                                             BESK 910
   34 BK=GJ                                                             BESK 920
      RETURN                                                            BESK 930
   36 B=X/2.                                                            BESK 940
      A=.5772157+ALOG(B)                                                BESK 950
      C=B*B                                                             BESK 960
      IF(N-1)37,43,37                                                   BESK 970
C                                                                       BESK 980
C     COMPUTE K0 USING SERIES EXPANSION                                 BESK 990
C                                                                       BESK1000
   37 G0=-A                                                             BESK1010
      X2J=1.                                                            BESK1020
      FACT=1.                                                           BESK1030
      HJ=.0                                                             BESK1040
      DO 40 J=1,6                                                       BESK1050
      RJ=1./FLOAT(J)                                                    BESK1060
      IF(X2J.LT.1.E-40) X2J=0.                                          BESK1061
C     PREVIOUS STATEMENT ADDED TO IBM SUBROUTINE TO CORRECT UNDERFLOW   BESK1062
C     PROBLEM ON WATFOR COMPILER                                        BESK1063
      X2J=X2J*C                                                         BESK1070
      FACT=FACT*RJ*RJ                                                   BESK1080
      HJ=HJ+RJ                                                          BESK1090
   40 G0=G0+X2J*FACT*(HJ-A)                                             BESK1100
      IF(N)43,42,43                                                     BESK1110
   42 BK=G0                                                             BESK1120
      RETURN                                                            BESK1130
C                                                                       BESK1140
C     COMPUTE K1 USING SERIES EXPANSION                                 BESK1150
C                                                                       BESK1160
   43 X2J=B                                                             BESK1170
      FACT=1.                                                           BESK1180
      HJ=1.                                                             BESK1190
      G1=1./X+X2J*(.5+A-HJ)                                             BESK1200
      DO 50 J=2,8                                                       BESK1210
      X2J=X2J*C                                                         BESK1220
      RJ=1./FLOAT(J)                                                    BESK1230
      FACT=FACT*RJ*RJ                                                   BESK1240
      HJ=HJ+RJ                                                          BESK1250
   50 G1=G1+X2J*FACT*(.5+(A-HJ)*FLOAT(J))                               BESK1260
      IF(N-1)31,52,31                                                   BESK1270
   52 BK=G1                                                             BESK1280
      RETURN                                                            BESK1290
      END                                                               BESK1300
```

TABLE 2.1.—*Listing of program for partial penetration in a nonleaky artesian aquifer*—Continued

```
      SUBROUTINE EXPI(X,RES,AUX)
C
C     ..................................................................
C
C        SUBROUTINE EXPI
C
C        PURPOSE
C           COMPUTES THE EXPONENTIAL INTEGRAL -EI(-X)
C
C        USAGE
C           CALL EXPI(X,RES)
C
C        DESCRIPTION OF PARAMETERS
C           X      - ARGUMENT OF EXPONENTIAL INTEGRAL
C           RES    - RESULT VALUE
C           AUX    - RESULTANT AUXILIARY VALUE
C
C        REMARKS
C           X GT 170 (X LT -174) MAY CAUSE UNDERFLOW (OVERFLOW)
C           WITH THE EXPONENTIAL FUNCTION
C           FOR X = 0 THE RESULT VALUE IS SET TO -1.E75
C
C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C           NONE
C
C        METHOD
C           DEFINITION
C           RES=INTEGRAL(EXP(-T)/T, SUMMED OVER T FROM X TO INFINITY).
C           EVALUATION
C           THREE DIFFERENT RATIONAL APPROXIMATIONS ARE USED IN THE
C           RANGES 1 LE X, X LE -9 AND -9 LT X LE -3 RESPECTIVELY,
C           A POLYNOMIAL APPROXIMATION IS USED IN -3 LT X LT 1.
C
C     ..................................................................
C
      IF(X-1.)2,1,1
    1 Y=1./X
      AUX=1.-Y*(((Y+3.377358E0)*Y+2.052156E0)*Y+2.709479E-1)/((((Y*
     11.072553E0+5.716943E0)*Y+6.945239E0)*Y+2.593888E0)*Y+2.709496E-1)
      RES=AUX*Y*EXP(-X)
      RETURN
    2 IF(X+3.)6,6,3
    3 AUX=(((((((7.122452E-7*X-1.766345E-6)*X+2.928433E-5)*X-2.335379E-4
     1)*X+1.664156E-3)*X-1.041576E-2)*X+5.555682E-2)*X-2.500001E-1)*X
     2+9.999999E-1
      RES=-1.E75
      IF(X)4,5,4
    4 RES=X*AUX-ALOG(ABS(X))-5.772157E-1
    5 RETURN
    6 IF(X+9.)8,8,7
    7 AUX=1.-((((5.176245E-2*X+3.061037E0)*X+3.243665E1)*X+2.244234E2)*X
     1+2.486697E2)/((((X+3.995161E0)*X+3.893944E1)*X+2.263818E1)*X
     2+1.807837E2)
      GOTO 9
    8 Y=9./X
      AUX=1.-Y*(((Y+7.659824E-1)*Y-7.271015E-1)*Y-1.080693E0)/((((Y
     1*2.518750E0+1.122927E1)*Y+5.921405E0)*Y-8.666702E0)*Y-9.724216E0)
    9 RES=AUX*EXP(-X)/X
      RETURN
      END
```

TABLE 4.3—*Listing of program for radial flow in a leaky artesian aquifer*

```
C     *******************************************************************
C
C     PURPOSE
C        TO COMPUTE A TABLE OF VALUES OF THE LEAKY AQUIFER WELL
C        FUNCTION - W(U,R/B) - HANTUSH,M.S., AND JACOB,C.E., 1955,
C        NON-STEADY RADIAL FLOW IN AN INFINITE LEAKY AQUIFER: AM.
C        GEOPHYS. UNION TRANS., V. 36, NO. 1, P. 95-100.
C     INPUT DATA
C        1 CARD - FORMAT(2E10.5)
C             USMALL - SMALLEST VALUE OF 1/U FOR WHICH COMPUTATION IS
C             DESIRED.
C             ULARGE - LARGEST  VALUE OF 1/U FOR WHICH COMPUTATION IS
C             DESIRED.
C        2 CARDS - FORMAT(8E10.5)
C             BDAT - 12 VALUES OF R/B FOR TABLE.
C     SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C        L,SERIES,FCT,BESK,DQL12
C
C     *******************************************************************
      REAL*4 L
      REAL*8 U,V
      DIMENSION ARRAY(73,12), Y(73), BDAT(12), YNUM(6)
      DATA YNUM/1.,1.5,2.,3.,5.,7./
      IRD=5
      IPT=6
      READ (IRD,6) USMALL,ULARGE
      READ (IRD,6) BDAT
      IBEGIN=ALOG10(USMALL)
      IEND=ALOG10(ULARGE)+.99999
      ILIMIT=(IEND-IBEGIN)*6+1
      IF (ILIMIT.GT.73) ILIMIT=73
      DO 1 I=1,12
      IF (BDAT(I).EQ.0.) GO TO 2
    1 CONTINUE
      NB=12
      GO TO 3
    2 NB=I-1
    3 II=0
      DO 4 I=1,ILIMIT
      II=II+1
      IF (II.GT.6) II=1
      IEXP=IBEGIN+(I-1)/6
      Y(I)=YNUM(II)*10.**IEXP
      U=1./Y(I)
      DO 4 J=1,NB
      V=BDAT(J)/2.
    4 ARRAY(I,J)=L(U,V)
      WRITE (IPT,7) (BDAT(I),I=1,NB)
      DO 5 I=1,ILIMIT
    5 WRITE (IPT,8) Y(I),(ARRAY(I,J),J=1,NB)
      STOP
C
C
    6 FORMAT (8E10.5)
    7 FORMAT ('1','W(U,R/B)'/'0',10X,'| R/B'/' ',6X,'1/U |',12E10.2)
    8 FORMAT (' ',E10.3,12F10.4)
      END
      REAL FUNCTION L*4(U,V)
C     *******************************************************************
C
C     FUNCTION L
C
```

TABLE 4.3—*Listing of program for radial flow in a leaky artesian aquifer*—Continued

```
C      PURPOSE
C         TO COMPUTE THE INTEGRAL( EXP(-Y-V**2/Y)/Y) SUMMED OVER Y FROM
C         U TO INFINITY(WELL FUNCTION FOR LEAKY AQUIFERS).
C      DESCRIPTION OF PARAMETERS
C         BOTH DOUBLE PRECISION
C         U - R**2*S/4*T*TIME (RADIAL DISTANCE SQUARED * STORAGE
C              COEFFICIENT / 4*TRANSMISSIVITY * TIME
C         V - R/2*SQRT(K'/(T*B'))--ONE-HALF RADIAL DISTANCE*SQUARE ROOT
C              (HYD. COND. OF CONFINING BED/TRANSMISSIVITY*THICKNESS
C              OF CONFINING BED)
C      SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C         DQL12,SERIES,BESK,FCT
C      METHOD
C         IN THE FOLLOWING F=EXP(-Y-V**2/Y)/Y
C       (1) U>=1, USES A GAUSSIAN-LAGUERRE QUADRATURE FORMULA TO
C         EVALUATE INTEGRAL(F) FROM U TO INF.
C       (2) V**2<U<1, USES THE G-L QUADRATURE TO EVALUATE INTEGRAL(F)
C         FROM ONE TO INF AND A SERIES EXPANSION TO EVALUATE INTEGRAL(F)
C         FROM U TO ONE.
C       (3) U<1, U<=V**2, USES THE REPRESENTATION INTEGRAL(F) FROM U
C         TO INF. = 2*K0(2*V)-INTEGRAL(F) FROM V**2/U TO INF.
C         EVALUATES THE ZERO ORDER MODIFIED BESSEL FUNCTION OF SECOND
C         KIND WITH IBM SUBROUTINE, EVALUATES INTEGRAL BY G-L QUAD.
C
C      *****************************************************************
       EXTERNAL FCT
       REAL*8 U,V,Z,F,VV,SERIES
       COMMON /C1/ VV,Z
       VV=V
       IF (U-1.) 1,2,2
C      CHECKS IF U<1
     1 Z=V*V/U
       IF (Z-1.) 3,4,4
C      CHECKS IF V**2/U < 1
     2 Z=U
       CALL DQL12(FCT,F)
       L=F
C      INTEGRAL U TO INF. EVALUATED BY GAUSS-LAGUERRE QUADRATURE
       GO TO 5
     3 Z=1.
       CALL DQL12(FCT,F)
       L=F+SERIES(U,V)
C      INTEGRAL 1 TO INF. BY G-L QUAD., INTEGRAL U TO 1 BY SERIES EXP.
       GO TO 5
     4 TWOV=2.*V
       CALL BESK(TWOV,0,BK,IER)
       CALL DQL12(FCT,F)
       L=2.*BK-F
C      2K0(2V)-INTEGRAL V**2/U TO INF.
     5 RETURN
       END
       REAL FUNCTION SERIES*8(U,V)
C      *****************************************************************
C
C      FUNCTION SERIES
C
C      PURPOSE
C         TO EVALUATE S(1)-S(U), WHERE S IS A SERIES EXPANSION OF
C         INTEGRAL(EXP(-Y-V**2/Y)DY/Y) GIVEN BY: S= SUM, M=0 TO INFINITY,
C         (F(M)*SUM, N=0 TO INF.,(V**(2*N)/((N!)*(M+N)!)) WHERE F(M)=
C         LOG(U) IF M=0 AND = ((-1)**M/M)*(U**M-(V**2/U)**M) IF M>0.
C      DESCRIPTION OF PARAMETERS
C         BOTH DOUBLE PRECISION
C         U - R**2*S/4*T*TIME (RADIAL DISTANCE SQUARED * STORAGE
C              COEFFICIENT / 4*TRANSMISSIVITY * TIME
C         V - R/2*SQRT(K'/(T*B'))--ONE-HALF RADIAL DISTANCE*SQUARE ROOT
C              (HYD. COND. OF CONFINING BED/TRANSMISSIVITY*THICKNESS
C              OF CONFINING BED)
```

TABLE 4.3—*Listing of program for radial flow in a leaky artesian aquifer*—Continued

```
C     SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C        NONE
C     METHOD
C        SUMMATION IS TERMINATED FOR THE INNER SERIES WHEN A TERM
C        BECOMES LESS THAN 5.E-7/N AND FOR OUTER SERIES WHEN A TERM
C        BECOMES LESS THAN 5.E-7
C
C     ****************************************************************
      REAL*8 DLOG,DABS,S(2),VUM,UU
      REAL*8 TEST,U,UM,EM,EN,SUM1,SUM,SIGN,V,VSQ,VSQU,RMUL,TERM,TERM1
      TEST=5.D-07
      VSQ=V*V
      UU=U
      DO 6 I=1,2
C     EVALUATES SERIES FOR LOWER LIMIT = U AND UPPER LIMIT = 1
      IF (I.EQ.2) U=1.
      UM=1.
      EM=-1.
      SUM1=0.
      SIGN=-1.
      VUM=1.
      VSQU=VSQ/U
    1 EM=EM+1.
      IF (EM-.1) 2,3,3
C     CHECKS FOR M=0
    2 RMUL=DLOG(U)
      TERM1=1.
      GO TO 4
    3 UM=UM*U
      IF (VUM.LT.1.D-30) VUM=0.
      VUM=VUM*VSQU
      RMUL=(UM-VUM)/EM
      TERM1=TERM1/EM
    4 SIGN=-SIGN
      SUM=TERM1
      TERM=TERM1
      EN=0.
    5 EN=EN+1.
      TERM=TERM*VSQ/(EN*(EN+EM))
      SUM=SUM+TERM
      IF (TEST.LE.DABS(RMUL*EN*TERM)) GO TO 5
C     TRUNCATES INNER SERIES IF OUTER TERM*N*INNER TERM < 5.E-7
      SUM1=SUM1+SIGN*RMUL*SUM
      IF (EM.LT..1) GO TO 1
      IF (TEST.LE.DABS(RMUL*SUM)) GO TO 1
C     TRUNCATES OUTER SERIES IF OUTER TERM*INNER SUM < 5.E-7
    6 S(I)=SUM1
      U=UU
      SERIES=S(2)-S(1)
      RETURN
      END
      REAL FUNCTION FCT*8(X)
C     ****************************************************************
C
C     FUNCTION FCT
C
C     PURPOSE
C        TO COMPUTE FCT(X)=EXP(-Z-V**2/(X+Z))/(X+Z)
```

TABLE 4.3—*Listing of program for radial flow in a leaky artesian aquifer*—Continued

```
C      DESCRIPTION OF PARAMETERS
C         X - THE DOUBLE PRECISION VALUE OF X FOR WHICH FCT IS COMPUTED
C      SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C         NONE
C      METHOD
C         FORTRAN EVALUATION OF FUNCTION
C
C      ..................................................................
       REAL*8 X,V,Z,P,DEXP
       COMMON /C1/ V,Z
       IF (X) 1,2,2
     1 FCT=0.
       GO TO 4
     2 P=Z+V**2/(X+Z)
       IF (P-5.D1) 3,3,1
     3 FCT=DEXP(-P)/(X+Z)
     4 RETURN
       END
       SUBROUTINE DQL12(FCT,Y)
C
C      ..................................................................
C
C         SUBROUTINE DQL12
C
C         PURPOSE
C            TO COMPUTE INTEGRAL(EXP(-X)*FCT(X), SUMMED OVER X
C                                FROM 0 TO INFINITY).
C
C         USAGE
C            CALL DQL12 (FCT,Y)
C            PARAMETER FCT REQUIRES AN EXTERNAL STATEMENT
C
C         DESCRIPTION OF PARAMETERS
C            FCT    - THE NAME OF AN EXTERNAL DOUBLE PRECISION FUNCTION
C                     SUBPROGRAM USED.
C            Y      - THE RESULTING DOUBLE PRECISION INTEGRAL VALUE.
C
C         REMARKS
C            NONE
C
C         SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C            THE EXTERNAL DOUBLE PRECISION FUNCTION SUBPROGRAM FCT(X)
C            MUST BE FURNISHED BY THE USER.
C
C         METHOD
C            EVALUATION IS DONE BY MEANS OF 12-POINT GAUSSIAN-LAGUERRE
C            QUADRATURE FORMULA, WHICH INTEGRATES EXACTLY,
C            WHENEVER FCT(X) IS A POLYNOMIAL UP TO DEGREE 23.
C            FOR REFERENCE, SEE
C            SHAO/CHEN/FRANK, TABLES OF ZEROS AND GAUSSIAN WEIGHTS OF
C            CERTAIN ASSOCIATED LAGUERRE POLYNOMIALS AND THE RELATED
C            GENERALIZED HERMITE POLYNOMIALS, IBM TECHNICAL REPORT
C            TR00.1100 (MARCH 1964), PP.24-25.
C
C      ..................................................................
C
C
C
       DOUBLE PRECISION X,Y,FCT
C
       X=.3709912104446692 D2
       Y=.8148077467426242 D-15*FCT(X)
```

TABLE 4.3—*Listing of program for radial flow in a leaky artesian aquifer*—Continued

```
      X=,2848796725098400 D2
      Y=Y+,3061601635035021 D-11*FCT(X)
      X=,2215109037939701 D2
      Y=Y+,1342391030515004 D-8*FCT(X)
      X=,1711685518746226 D2
      Y=Y+,1668493876540910 D-6*FCT(X)
      X=,1300605499330635 D2
      Y=Y+,8365055856819680 D-5*FCT(X)
      X=,962131684245687 D1
      Y=Y+,2032315926629994 D-3*FCT(X)
      X=,6844525453115177 D1
      Y=Y+,2663973541865316 D-2*FCT(X)
      X=,4599227639418348 D1
      Y=Y+,2010238115463410 D-1*FCT(X)
      X=,2833751337743507 D1
      Y=Y+,9044922221116809 D-1*FCT(X)
      X=,1512610269776419 D1
      Y=Y+,2440820113198776 D0*FCT(X)
      X=,6117574845151307 D0
      Y=Y+,3777592758731380 D0*FCT(X)
      X=,1157221173580207 D0
      Y=Y+,2647313710554432 D0*FCT(X)
      RETURN
      END
      SUBROUTINE BESK(X,N,BK,IER)
C
C     ..................................................................
C
C        SUBROUTINE BESK
C
C           COMPUTE THE K BESSEL FUNCTION FOR A GIVEN ARGUMENT AND ORDER
C
C        USAGE
C           CALL BESK(X,N,BK,IER)
C
C        DESCRIPTION OF PARAMETERS
C           X  -THE ARGUMENT OF THE K BESSEL FUNCTION DESIRED
C           N  -THE ORDER OF THE K BESSEL FUNCTION DESIRED
C           BK -THE RESULTANT K BESSEL FUNCTION
C           IER-RESULTANT ERROR CODE WHERE
C              IER=0  NO ERROR
C              IER=1  N IS NEGATIVE
C              IER=2  X IS ZERO OR NEGATIVE
C              IER=3  X ,GT, 170, MACHINE RANGE EXCEEDED
C              IER=4  BK ,GT, 10**70
C
C        REMARKS
C           N MUST BE GREATER THAN OR EQUAL TO ZERO
C
C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C           NONE
C
C        METHOD
C           COMPUTES ZERO ORDER AND FIRST ORDER BESSEL FUNCTIONS USING
C           SERIES APPROXIMATIONS AND THEN COMPUTES N TH ORDER FUNCTION
C           USING RECURRENCE RELATION,
C           RECURRENCE RELATION AND POLYNOMIAL APPROXIMATION TECHNIQUE
C           AS DESCRIBED BY A,J,M,HITCHCOCK,'POLYNOMIAL APPROXIMATIONS
C           TO BESSEL FUNCTIONS OF ORDER ZERO AND ONE AND TO RELATED
C           FUNCTIONS', M,T,A,C,, V,11,1957,PP,86-88, AND G,N, WATSON,
C           'A TREATISE ON THE THEORY OF BESSEL FUNCTIONS', CAMBRIDGE
C           UNIVERSITY PRESS, 1958, P, 62
```

TABLE 4.3—*Listing of program for radial flow in a leaky artesian aquifer*—Continued

```
C
C     ...................................................................
C
      DIMENSION T(12)
      BK=.0
      IF(N)10,11,11
   10 IER=1
      RETURN
   11 IF(X)12,12,20
   12 IER=2
      RETURN
   20 IF(X-170.0)22,22,21
   21 IER=3
      RETURN
   22 IER=0
      IF(X-1.)36,36,25
   25 A=EXP(-X)
      B=1./X
      C=SQRT(B)
      T(1)=B
      DO 26 L=2,12
   26 T(L)=T(L-1)*B
      IF(N-1)27,29,27
C
C     COMPUTE KO USING POLYNOMIAL APPROXIMATION
C
   27 G0=A*(1.2533141-.1566642*T(1)+.08811128*T(2)-.09139095*T(3)
     2+.1344596*T(4)-.2299850*T(5)+.3792410*T(6)-.5247277*T(7)
     3+.5575368*T(8)-.4262633*T(9)+.2184518*T(10)-.06680977*T(11)
     4+.009189383*T(12))*C
      IF(N)20,28,29
   28 BK=G0
      RETURN
C
C     COMPUTE K1 USING POLYNOMIAL APPROXIMATION
C
   29 G1=A*(1.2533141+.4699927*T(1)-.1468583*T(2)+.1280427*T(3)
     2-.1736432*T(4)+.2847618*T(5)-.4594342*T(6)+.6283381*T(7)
     3-.6632295*T(8)+.5050239*T(9)-.2581304*T(10)+.07880001*T(11)
     4-.01082418*T(12))*C
      IF(N-1)20,30,31
   30 BK=G1
      RETURN
C
C     FROM KO,K1 COMPUTE KN USING RECURRENCE RELATION
C
   31 DO 35 J=2,N
      GJ=2.*(FLOAT(J)-1.)*G1/X+G0
      IF(GJ-1.0E70)33,33,32
   32 IER=4
      GO TO 34
   33 G0=G1
   35 G1=GJ
   34 BK=GJ
      RETURN
   36 B=X/2.
      A=.5772157+ALOG(B)
      C=B*B
      IF(N-1)37,43,37
C
C     COMPUTE KO USING SERIES EXPANSION
```

TABLE 4.3—*Listing of program for radial flow in a leaky artesian aquifer*—Continued

```
C                                                                      BESK1000
   37 G0=-A                                                            BESK1010
      X2J=1.                                                           BESK1020
      FACT=1.                                                          BESK1030
      HJ=.0                                                            BESK1040
      DO 40 J=1,6                                                      BESK1050
      RJ=1./FLOAT(J)                                                   BESK1060
      IF(X2J.LT.1.E-40) X2J=0.                                         BESK1061
C     PREVIOUS STATEMENT ADDED TO IBM SUBROUTINE TO CORRECT UNDERFLOW  BESK1062
C     PROBLEM ON WATFOR COMPILER                                       BESK1063
      X2J=X2J*C                                                        BESK1070
      FACT=FACT*RJ*RJ                                                  BESK1080
      HJ=HJ+RJ                                                         BESK1090
   40 G0=G0+X2J*FACT*(HJ-A)                                            BESK1100
      IF(N)43,42,43                                                    BESK1110
   42 BK=G0                                                            BESK1120
      RETURN                                                           BESK1130
C                                                                      BESK1140
C     COMPUTE K1 USING SERIES EXPANSION                                BESK1150
C                                                                      BESK1160
   43 X2J=B                                                            BESK1170
      FACT=1.                                                          BESK1180
      HJ=1.                                                            BESK1190
      G1=1./X+X2J*(.5+A-HJ)                                            BESK1200
      DO 50 J=2,8                                                      BESK1210
      X2J=X2J*C                                                        BESK1220
      RJ=1./FLOAT(J)                                                   BESK1230
      FACT=FACT*RJ*RJ                                                  BESK1240
      HJ=HJ+RJ                                                         BESK1250
   50 G1=G1+X2J*FACT*(.5+(A-HJ)*FLOAT(J))                              BESK1260
      IF(N-1)31,52,31                                                  BESK1270
   52 BK=G1                                                            BESK1280
      RETURN                                                           BESK1290
      END                                                              BESK130-
```

TABLE 5.2—*Listing of program for radial flow in a leaky artesian aquifer with storage of water in the confining beds*

```
C     *****************************************************************LST   1
C                                                                      LST   2
C     PURPOSE                                                          LST   3
C        TO COMPUTE TYPE CURVE FUNCTION VALUES FOR H(U,BETA) --        LST   4
C        HANTUSH,M.S.,1960, MODIFICATION OF THE THEORY OF LEAKY        LST   5
C        AQUIFERS: JOUR. GEOPHYS. RES., V. 65, NO. 11,  P. 3713-3725.  LST   6
C        THE COMPUTATIONAL ALGORITHM WAS DEVISED AND PROGRAMMED BY     LST   7
C        S.S.PAPADOPULOS.                                              LST   8
C     INPUT DATA                                                       LST   9
C        1 CARD - FORMAT(2E10.5)                                       LST  10
C             USMALL - SMALLEST(BEGINNING) VALUE OF 1/U.               LST  11
C             ULARGE - LARGEST(ENDING) VALUE OF 1/U.                   LST  12
C        2 CARDS - FORMAT(8E10.5)                                      LST  13
C             BDAT - 12 VALUES OF BETA (ZERO OR BLANK VALUES ARE       LST  14
C             PERMISSIBLE IF LESS THAN 12 DESIRED, WILL TERMINATE      LST  15
C             AT FIRST ZERO OR BLANK VALUE).                           LST  16
C     SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED                    LST  17
C        H,DQG32,MUH,W - MUST BE INCLUDED IN DECK.                     LST  18
C        DSQRT,DEXP,DERFC,DLOG - MUST BE IN COMPUTER LIBRARY.          LST  19
C                                                                      LST  20
C     *****************************************************************LST  21
      REAL*8 U,BETA,H                                                  LST  22
      DIMENSION ARRAY(73,12), Y(73), BDAT(12), YNUM(6)                 LST  23
      DATA YNUM/1.,1.5,2.,3.,5.,7./                                    LST  24
      IRD=5                                                            LST  25
      IPT=6                                                            LST  26
      READ (IRD,6) USMALL,ULARGE                                       LST  27
      READ (IRD,6) BDAT                                                LST  28
      IBEGIN=ALOG10(USMALL)                                            LST  29
      IEND=ALOG10(ULARGE)+.99999                                       LST  30
      ILIMIT=(IEND-IBEGIN)*6+1                                         LST  31
      IF (ILIMIT.GT.73) ILIMIT=73                                      LST  32
      DO 1 I=1,12                                                      LST  33
      IF (BDAT(I).EQ.0.) GO TO 2                                       LST  34
    1 CONTINUE                                                         LST  35
      NB=12                                                            LST  36
```

TABLE 5.2—*Listing of program for radial flow in a leaky artesian aquifer with storage of water in the confining beds*—Continued

```
      GO TO 3
    2 NB=I-1
      II=0
    3 DO 4 I=1,ILIMIT
      IEXP=IBEGIN+(I-1)/6
      II=II+1
      IF (II.GT.6) II=1
      Y(I)=YNUM(II)*10.**IEXP
      U=1./Y(I)
      DO 4 J=1,NB
      BETA=BDAT(J)
    4 ARRAY(I,J)=H(U,BETA)
      WRITE (IPT,7) (BDAT(I),I=1,NB)
      DO 5 I=1,ILIMIT
    5 WRITE (IPT,8) Y(I),(ARRAY(I,J),J=1,NB)
      STOP
C
    6 FORMAT (8E10.5)
    7 FORMAT ('1','H(U,BETA)'/'0',10X,'| BETA'/' ',6X,'1/U |',12E10.2)
    8 FORMAT (' ',E10.3,12F10.4)
      END
      DOUBLE PRECISION FUNCTION H(U,B)
C     ******************************************************************
C
C     FUNCTION H
C     PURPOSE
C        TO COMPUTE THE INTEGRAL OF
C        EXP(-Y)*ERFC(B*SQRT(U)/SQRT(Y*(Y-U)))/Y SUMMED OVER Y
C        FROM U TO INFINITY (FUNCTION H(U,BETA) OF HANTUSH).
C     DESCRIPTION OF PARAMETERS
C        BOTH DOUBLE PRECISION
C        U - R**2*S/(4*T*TIME), (RADIAL DISTANCE SQUARED * STORAGE
C        COEFFICIENT / (4 * TRANSMISSIVITY * TIME). U MUST BE > 1.D-60.
C        B - (R/4)*(SQRT(K'*S'/(B'*T*S)+K''*S''/(B''*T*S)),
C             K',S',B' - HYD. COND., STORAGE COEFF., THICKNESS OF
C                  UPPER CONFINING BED.
C             K'',S'',B'' - HYD. COND., STORAGE COEFF., THICKNESS OF
C                  LOWER CONFINING BED.
C     METHOD
C        I. FOR U < 1.D-60, NO COMPUTATION IS MADE.
C        II. FOR B=0, H(U,0)=W(U) (THEIS WELL FUNCTION).
C        III. H(U,B)=0 IF
C             1. U > 10.
C             2. B > 1 AND B**2*U > 300.
C        IV. ERFC(ARG)=0 FOR ARG > 40 AND H(U,B) = H(UB,B)
C             FOR U < Y < UB WHERE UB IS THE U CORRESPONDING TO ARG = 40
C             SINCE H(UB,B) < W(UB) THEN FOR UB > 10, H(U,B) = 0.
C             ERFC(ARG) = 1 FOR ARG < 2.E-10 AND H(UUB,B) = W(UUB)
C             WHERE UUB IS THE U CORRESPONDING TO ARG = 2.E-10.
C             IF UUB > 10, H(U,B) = INTEGRAL FROM UB TO 10.
C             IF UUB < 10, H(U,B) = INTEGRAL FROM UB TO UUB + W(UUB)
C
C     ******************************************************************
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON UUU,BBB
      EXTERNAL HUB
      UUU=U
      BBB=B
      IF (U.GT.1.D-60) GO TO 1
      WRITE (6,7)
      STOP
    1 IF (B.EQ.0.0) GO TO 5
      IF (U.GT.10.0) GO TO 6
      BU=B*B*U
      IF (B.GT.1.0.AND.BU.GE.300.0) GO TO 6
      H1=0.0
      UP=10.0
      UB=0.5*U*(1.0+DSQRT(1.0+0.025*B*B/U))
      IF (UB.GT.UP) GO TO 6
      UUB=0.5*U*(1.0+DSQRT(1.0+1.D20*B*B/U))
      IF (UUB.GT.UP) GO TO 2
      H1=W(UUB)
      UP=UUB
    2 H2=0.0
      XL=UB
    3 XU=10.*XL
      IF (XU.GE.UP) XU=UP
      CALL DQG32(XL,XU,HUB,AREA)
      H2=H2+AREA
      XL=XU
      IF (XL.EQ.UP) GO TO 4
      GO TO 3
    4 H=H1+H2
      RETURN
    5 H=W(U)
      RETURN
```

TABLE 5.2—*Listing of program for radial flow in a leaky artesian aquifer with storage of water in the confining beds—Continued*

```
    6 H=0.0
      RETURN
C
    7 FORMAT ('0','U TOO SMALL FOR COMPUTATION')
      END
      SUBROUTINE DQG32(XL,XU,FCT,Y)
C
C     ..................................................................
C
C        SUBROUTINE DQG32
C
C        PURPOSE
C           TO COMPUTE INTEGRAL(FCT(X), SUMMED OVER X FROM XL TO XU)
C
C        USAGE
C           CALL DQG32 (XL,XU,FCT,Y)
C           PARAMETER FCT REQUIRES AN EXTERNAL STATEMENT
C
C        DESCRIPTION OF PARAMETERS
C           XL     - DOUBLE PRECISION LOWER BOUND OF THE INTERVAL.
C           XU     - DOUBLE PRECISION UPPER BOUND OF THE INTERVAL.
C           FCT    - THE NAME OF AN EXTERNAL DOUBLE PRECISION FUNCTION
C                    SUBPROGRAM USED.
C           Y      - THE RESULTING DOUBLE PRECISION INTEGRAL VALUE.
C
C        REMARKS
C           NONE
C
C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C           THE EXTERNAL DOUBLE PRECISION FUNCTION SUBPROGRAM FCT(X)
C           MUST BE FURNISHED BY THE USER.
C
C        METHOD
C           EVALUATION IS DONE BY MEANS OF 32-POINT GAUSS QUADRATURE
C           FORMULA, WHICH INTEGRATES POLYNOMIALS UP TO DEGREE 63
C           EXACTLY. FOR REFERENCE, SEE
C           V.I.KRYLOV, APPROXIMATE CALCULATION OF INTEGRALS,
C           MACMILLAN, NEW YORK/LONDON, 1962, PP.100-111 AND 337-340.
C
C     ..................................................................
C
      DOUBLE PRECISION XL,XU,Y,A,B,C,FCT
      A=.5D0*(XU+XL)
      B=XU-XL
      C=.49863193092474080D0*B
      Y=.35093050047350480D-2*(FCT(A+C)+FCT(A-C))
      C=.49280575577263420D0*B
      Y=Y+.81371973654528840D-2*(FCT(A+C)+FCT(A-C))
      C=.48238112779375320D0*B
      Y=Y+.12696032654631030D-1*(FCT(A+C)+FCT(A-C))
      C=.46745303796886980D0*B
      Y=Y+.17136931456510720D-1*(FCT(A+C)+FCT(A-C))
      C=.44816057788302610D0*B
      Y=Y+.21417949011113340D-1*(FCT(A+C)+FCT(A-C))
      C=.42468380686628500D0*B
      Y=Y+.25499029631188090D-1*(FCT(A+C)+FCT(A-C))
      C=.39724189798397120D0*B
      Y=Y+.29342046739267770D-1*(FCT(A+C)+FCT(A-C))
      C=.36609105937014480D0*B
      Y=Y+.32911111388180920D-1*(FCT(A+C)+FCT(A-C))
      C=.33152213346510760D0*B
      Y=Y+.36172897054424250D-1*(FCT(A+C)+FCT(A-C))
      C=.29385787862038120D0*B
      Y=Y+.39096947893535150D-1*(FCT(A+C)+FCT(A-C))
      C=.25344995446611470D0*B
      Y=Y+.41655962113473380D-1*(FCT(A+C)+FCT(A-C))
      C=.21067563806531770D0*B
      Y=Y+.43826046502201910D-1*(FCT(A+C)+FCT(A-C))
      C=.16593430114106380D0*B
      Y=Y+.45586939347881940D-1*(FCT(A+C)+FCT(A-C))
      C=.11964368112606850D0*B
      Y=Y+.46922199540402280D-1*(FCT(A+C)+FCT(A-C))
      C=.72235980791398200-1*B
      Y=Y+.47819360039637430D-1*(FCT(A+C)+FCT(A-C))
      C=.24153832843869160D-1*B
      Y=B*(Y+.48270044257363900D-1*(FCT(A+C)+FCT(A-C)))
      RETURN
      END
      DOUBLE PRECISION FUNCTION HUB(X)
C     ******************************************************************
C
C     FUNCTION HUB
C     PURPOSE
C        TO COMPUTE VALUES OF THE INTEGRAND OF H(U,B)
C     DESCRIPTION OF PARAMETER
C        X - DOUBLE PRECISION, POINT AT WHICH INTEGRAND IS EVALUATED.
```

TABLE 5.2.—*Listing of program for radial flow in a leaky artesian aquifer with storage of water in the confining beds*—Continued

```
C       METHOD
C          FORTRAN EVALUATION OF FUNCTION.
C
C       ******************************************************************
        IMPLICIT REAL*8(A-H,O-Z)
        COMMON UUU,BBB
        ARG=DSQRT((BBB*BBB*UUU)/(X*X-X*UUU))
        HUB=DEXP(-X)*DERFC(ARG)/X
        RETURN
        END

        DOUBLE PRECISION FUNCTION W(U)
C       *****************************************************************
C
C       FUNCTION W
C       PURPOSE
C          TO EVALUATE THE WELL FUNCTION OF THEIS.
C       DESCRIPTION OF PARAMETER
C          U - DOUBLE PRECISION, ARGUMENT FOR WELL FUNCTION.
C
C       *****************************************************************
        IMPLICIT REAL*8 (A-H,O-Z)
        IF (U.LE.0.0) GO TO 2
        IF (U.GT.100.) GO TO 3
        IF (U.GE.1.0) GO TO 1
        W=-.57721566+U*(.99999193+U*(-.24991055+U*(.05519968+U*(-.00976004
       1+.00107857*U))))-DLOG(U)
        GO TO 4
      1 ENUM=DEXP(-U)*(.2677737343+U*(8.6347608925+U*(18.0590169730+U*(8.5
       1733287401+U))))
        DEN=U*(3.9584969228+U*(21.0996530827+U*(25.6329561486+U*(9.5733223
       1454+U))))
        W=ENUM/DEN
        GO TO 4
      2 WRITE (6,5) U
        STOP
      3 W=0.0
      4 RETURN
C
      5 FORMAT ('0',5X,'W(U) NOT DEFINED FOR U=',1PD15.8)
        END
```

TABLE 6.1.—*Listing of program for partial penetration in a leaky artesian aquifer*

```
C       *******************************************************************
C
C       PURPOSE
C          TO COMPUTE TYPE CURVE FUNCTION VALUES FOR PARTIAL PENETRATION
C          IN A LEAKY AQUIFER USING EQ. 73 OF HANTUSH,M.S., 1964,
C          HYDRAULICS OF WELLS IN CHOW, VEN TE, ADVANCES IN HYDROSCIENCE,
C          VOL. 1: ACADEMIC PRESS, NEW YORK, P. 281-442.
C       INPUT DATA
C          1 CARD  -  FORMAT (3F5.1,I5,2E10.4)
C                B - AQUIFER THICKNESS
C                E - DEPTH, BELOW TOP OF AQUIFER, TO BOTTOM OF PUMPING
C                WELL SCREEN
C                D - DEPTH, BELOW TOP OF AQUIFER, TO TOP OF PUMPING WELL
C                SCREEN
C                NUM - NUMBER OF OBSERVATION WELLS OR PIEZOMETERS TIMES
C                NUMBER OF VALUES OF KZ/KR.
C                SMALL - SMALLEST VALUE OF 1/U FOR WHICH COMPUTATION IS
C                DESIRED
C                LARGE - LARGEST  VALUE OF 1/U FOR WHICH COMPUTATION IS
C                DESIRED
C          2 CARDS - FORMAT(8E10.5)
C                BOAT - 12 VALUES OF R/BR, NON ZERO VALUES SHOULD BE
C                FIRST, WILL TERMINATE AT FIRST ZERO (OR BLANK) VALUE.
C          NUM CARDS (ONE FOR EACH OBS. WELL OR PIEZOMETER AND FOR EACH
C                VALUE OF R*SQRT(KZ/KR) - FORMAT (3F5.1)
C                R - RADIAL DISTANCE FROM PUMPED WELL TIMES SQRT(KZ/KR).
C                LPRIME - DEPTH, BELOW TOP OF AQUIFER, TO BOTTOM OF OBS.
C                WELL SCREEN (ZERO FOR PIEZOMETER)
C                DPRIME - DEPTH, BELOW TOP OF AQUIFER, TO TOP OF OBS. WELL
```

TABLE 6.1.—*Listing of program for partial penetration in a leaky artesian aquifer*—Continued

```
C             SCREEN (TOTAL DEPTH FOR PIEZOMETER)
C       SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C          DQL12,SERIES,BESK,FCT,L,FL
C
C       ****************************************************************
      REAL*8 U,V
      REAL*4 L,LB,LPB,LPRIME,LARGE
      DIMENSION ARRAY(55,12), ARG(6), BDAT(12), Y(55)
      DATA ARG/1.,1.5,2.,3.,5.,7./
      DATA ARRAY/660*0./,Y/55*0./
      IRD=5
      IPT=6
      READ (IRD,9) B,E,D,NUM,SMALL,LARGE
      READ (IRD,14) BDAT
      DO 1 I=1,12
      IF (BDAT(I).EQ.0.) GO TO 2
    1 CONTINUE
      NB=12
      GO TO 3
    2 NB=I-1
    3 LB=E/B
      DB=D/B
      IBEGIN=ALOG10(SMALL)
      IEND=ALOG10(LARGE)+.1
      JLIMIT=IEND-IBEGIN
      IF (JLIMIT.GT.9) JLIMIT=9
      ILIMIT=6*JLIMIT+1
      DO 8 K=1,NUM
      READ (IRD,9) R,LPRIME,DPRIME
      RB=R/B
      LPB=LPRIME/B
      DPB=DPRIME/B
      DO 4 I=1,ILIMIT
      INDEX=(I-1)/6
      IEXP=IBEGIN+INDEX
      II=I-INDEX*6
      Y(I)=ARG(II)*10.**IEXP
      U=1./Y(I)
      DO 4 J=1,NB
      BETA=BDAT(J)
      V=BETA/2.
    4 ARRAY(I,J)=L(U,V)+FL(U,RB,BETA,LB,DB,LPB,DPB)
      IF (LPB-0.) 5,5,6
    5 WRITE (IPT,10) DPB,RB,LB,DB
      GO TO 7
    6 WRITE (IPT,11) LPB,DPB,RB,LB,DB
    7 WRITE (IPT,12) (BDAT(I),I=1,NB)
      DO 8 I=1,ILIMIT
      WRITE (IPT,13) Y(I),(ARRAY(I,J),J=1,NB)
    8 CONTINUE
      STOP
C
C
    9 FORMAT (3F5.1,I5,2E10.4)
   10 FORMAT ('1','W(U,R/BR)+F(U,R/B,R/BR,L/B,D/B,Z/B), Z/B=',F5.2,', SQ
     1RT(KZ/KR)*R/B=',F5.2,', L/B=',F5.2,', D/B=',F5.2)
   11 FORMAT ('1','W(U,R/BR)+F(U,R/B,R/BR,L/B,D/B,L''/B,D''/B), L''/B=',
     1F5.2,', D''/B=',F5.2,', SQRT(KZ/KR)*R/B=',F5.2,', L/B=',F5.2,', D/
     2B=',F5.2)
   12 FORMAT ('0',9X,'| R/BR'/' ',5X,'1/U |',12E10.2)
   13 FORMAT (' ',E10.3,12F10.4)
   14 FORMAT (8E10.5)
      END
```

TABLE 6.1.—*Listing of program for partial penetration in a leaky artesian aquifer*—Continued

```
      REAL FUNCTION FL*4(U,RB,BETA,LB,DB,LPB,DPB)
C     ******************************************************************
C
C     FUNCTION FL
C
C     PURPOSE
C        TO COMPUTE DEPARTURES FROM HANTUSH-JACOB LEAKY AQUIFER CURVE
C        CAUSED BY PARTIAL PENETRATION OF PUMPED WELL.
C     USAGE
C        FL(U,RB,BETA,LB,DB,LPB,DPB)
C     DESCRIPTION OF PARAMETERS
C        ALL REAL, U DOUBLE PRECISION
C        U = R**2*S/4*T*TIME (RADIAL DISTANCE SQUARED * STORAGE
C            COEFFICIENT / 4*TRANSMISSIVITY * TIME
C        RB = R/B ( RADIAL DISTANCE / AQUIFER THICKNESS )
C        BETA = R*SQRT(K'/B'T) = (RADIAL DISTANCE * SQUARE ROOT
C               (HYD. COND. OF CONFINING BED/THICKNESS OF CONFINING
C               BED * TRANSMISSIVITY OF AQUIFER))
C        LB = L/B ( FRACTION OF AQUIFER PENETRATED BY PUMPED WELL)
C        DB = D/B ( FRACTION OF AQUIFER ABOVE PUMPED WELL SCREEN)
C        LPB = L'/B (FRACTION OF AQUIFER PENETRATED BY OBS. WELL, ZERO
C              FOR PIEZOMETER)
C        DPB = D'/B (FRACTION OF AQUIFER ABOVE OBS. WELL SCREEN, TOTAL
C              DEPTH FOR PIEZOMETER)
C     SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C        DQL12,SERIES,BESK,FCT,L
C     METHOD
C        SUMS THE SERIES THROUGH N*PI*R/B EQ 20
C
C     ******************************************************************
      REAL*8 U,V,DSQRT
      REAL*4 L,N,LB,LPB
      SUM=0.
      N=0.
      BETSQ=BETA*BETA
      PIRBSQ=9.869604*RB*RB
      PILB=3.141593*LB
      PIDB=3.141593*DB
      IF (LPB-0.) 1,1,4
C     CHECKS FOR WELL OR PIEZOMETER
    1 PIZB=3.141593*DPB
    2 N=N+1.
      V=SQRT(BETSQ+N*N*PIRBSQ)/2.
      IF (V.GT.10.) GO TO 3
C     TRUNCATES SERIES WHEN V>10
      X=L(U,V)/N
      SUM=SUM+(SIN(N*PILB)-SIN(N*PIDB))*COS(N*PIZB)*X
      GO TO 2
    3 FL=.6366198*SUM/(LB-DB)
      GO TO 7
    4 PILPB=3.141593*LPB
      PIDPB=3.141593*DPB
    5 N=N+1
      V=SQRT(BETSQ+N*N*PIRBSQ)/2.
      IF (V.GT.10.) GO TO 6
C     TRUNCATES SERIES WHEN V>10
      X=L(U,V)/N
      SUM=SUM+(SIN(N*PILB)-SIN(N*PIDB))*(SIN(N*PILPB)-SIN(N*PIDPB))*X/N
      GO TO 5
    6 FL=.2026424*SUM/((LB-DB)*(LPB-DPB))
    7 RETURN
      END
```

TABLE 6.1.—*Listing of program for partial penetration in a leaky artesian aquifer*—Continued

```
      REAL FUNCTION L*4(U,V)
C     ******************************************************************
C
C     FUNCTION L
C
C     PURPOSE
C        TO COMPUTE THE INTEGRAL( EXP(-Y-V**2/Y)/Y) SUMMED OVER Y FROM
C        U TO INFINITY(WELL FUNCTION FOR LEAKY AQUIFERS).
C     DESCRIPTION OF PARAMETERS
C        BOTH DOUBLE PRECISION
C        U = R**2*S/4*T*TIME (RADIAL DISTANCE SQUARED * STORAGE
C            COEFFICIENT / 4*TRANSMISSIVITY * TIME
C        V = R/2*SQRT(K'/(T*B'))--ONE-HALF RADIAL DISTANCE*SQUARE ROOT
C            (HYD. COND. OF CONFINING BED/TRANSMISSIVITY*THICKNESS
C            OF CONFINING BED)
C     SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED\
C        DQL12,SERIES,BESK,FCT
C     METHOD
C        IN THE FOLLOWING F=EXP(-Y-V**2/Y)/Y
C      (1) U>=1, USES A GAUSSIAN-LAGUERRE QUADRATURE FORMULA TO
C        EVALUATE INTEGRAL(F) FROM U TO INF.
C      (2) V**2<U<1, USES THE G-L QUADRATURE TO EVALUATE INTEGRAL(F)
C        FROM ONE TO INF AND A SERIES EXPANSION TO EVALUATE INTEGRAL(F)
C        FROM U TO ONE.
C      (3) U<1, U<=V**2, USES THE REPRESENTATION INTEGRAL(F) FROM U
C        TO INF. = 2*K0(2*V)-INTEGRAL(F) FROM V**2/U TO INF.
C        EVALUATES THE ZERO ORDER MODIFIED BESSEL FUNCTION OF SECOND
C        KIND WITH IBM SUBROUTINE, EVALUATES INTEGRAL BY G-L QUAD.
C
C     ******************************************************************
      EXTERNAL FCT
      REAL*8 U,V,Z,F,VV,SERIES
      COMMON /C1/ VV,Z
      VV=V
      IF (U-1.) 1,2,2
C     CHECKS IF U<1
    1 Z=V*V/U
      IF (Z-1.) 3,4,4
C     CHECKS IF V**2/U < 1
    2 Z=U
      CALL DQL12(FCT,F)
      L=F
C     INTEGRAL U TO INF. EVALUATED BY GAUSS-LAGUERRE QUADRATURE
      GO TO 5
    3 Z=1.
      CALL DQL12(FCT,F)
      L=F+SERIES(U,V)
C     INTEGRAL 1 TO INF. BY G-L QUAD., INTEGRAL U TO 1 BY SERIES EXP.
      GO TO 5
    4 TWOV=2.*V
      CALL BESK(TWOV,0,BK,IER)
      CALL DQL12(FCT,F)
      L=2.*BK-F
C     2K0(2V)-INTEGRAL V**2/U TO INF.
    5 RETURN
      END
      REAL FUNCTION SERIES*8(U,V)
C     ********************************************************************
C
C     FUNCTION SERIES
C
C     PURPOSE
```

TABLE 6.1.—*Listing of program for partial penetration in a leaky artesian aquifer*—Continued

```
C        TO EVALUATE S(1)=S(U), WHERE S IS A SERIES EXPANSION OF
C        INTEGRAL(EXP(-Y-V**2/Y)DY/Y) GIVEN BY; S= SUM, M=0 TO INFINITY,
C        (F(M)*SUM, N=0 TO INF,,(V**(2*N)/((N!)*(M+N)!)) WHERE F(M)=
C        LOG(U) IF M=0 AND = ((-1)**M/M)*(U**M-(V**2/U)**M) IF M>0,
C     DESCRIPTION OF PARAMETERS
C        BOTH DOUBLE PRECISION
C        U = R**2*S/4*T*TIME (RADIAL DISTANCE SQUARED * STORAGE
C            COEFFICIENT / 4*TRANSMISSIVITY * TIME
C        V = R/2*SQRT(K'/(T*B'))--ONE-HALF RADIAL DISTANCE*SQUARE ROOT
C            (HYD, COND, OF CONFINING BED/TRANSMISSIVITY*THICKNESS
C            OF CONFINING BED)
C     SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C        NONE
C     METHOD
C        SUMMATION IS TERMINATED FOR THE INNER SERIES WHEN A TERM
C        BECOMES LESS THAN 5,E-7/N AND FOR OUTER SERIES WHEN A TERM
C        BECOMES LESS THAN 5,E-7
C
C     *******************************************************************
      REAL*8 DLOG,DABS,S(2),VUM,UU
      REAL*8 TEST,U,UM,EM,EN,SUM1,SUM,SIGN,V,VSQ,VSQU,RMUL,TERM,TERM1
      TEST=5,D-07
      VSQ=V*V
      UU=U
      DO 6 I=1,2
C     EVALUATES SERIES FOR LOWER LIMIT = U AND UPPER LIMIT = 1
      IF (I,EQ,2) U=1,
      UM=1,
      EM=-1,
      SUM1=0,
      SIGN=-1,
      VUM=1,
      VSQU=VSQ/U
    1 EM=EM+1,
      IF (EM-,1) 2,3,3
C     CHECKS FOR M=0
    2 RMUL=DLOG(U)
      TERM1=1,
      GO TO 4
    3 UM=UM*U
      IF (VUM,LT,1,D-30) VUM=0,
      VUM=VUM*VSQU
      RMUL=(UM-VUM)/EM
      TERM1=TERM1/EM
    4 SIGN=-SIGN
      SUM=TERM1
      TERM=TERM1
      EN=0,
    5 EN=EN+1,
      TERM=TERM*VSQ/(EN*(EN+EM))
      SUM=SUM+TERM
      IF (TEST,LE,DABS(RMUL*EN*TERM)) GO TO 5
C     TRUNCATES INNER SERIES IF OUTER TERM*N*INNER TERM < 5,E-7
      SUM1=SUM1+SIGN*RMUL*SUM
      IF (EM,LT,,1) GO TO 1
      IF (TEST,LE,DABS(RMUL*SUM)) GO TO 1
C     TRUNCATES OUTER SERIES IF OUTER TERM*INNER SUM < 5,E-7
    6 S(I)=SUM1
      U=UU
      SERIES=S(2)-S(1)
      RETURN
      END
```

TABLE 6.1.—*Listing of program for partial penetration in a leaky artesian aquifer*—Continued

```
      REAL FUNCTION FCT*8(X)
C     ******************************************************************
C
C     FUNCTION FCT
C
C     PURPOSE
C        TO COMPUTE FCT(X)=EXP(-Z-V**2/(X+Z))/(X+Z)
C     DESCRIPTION OF PARAMETERS
C        X - THE DOUBLE PRECISION VALUE OF X FOR WHICH FCT IS COMPUTED
C     SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C        NONE
C     METHOD
C        FORTRAN EVALUATION OF FUNCTION
C
C     ******************************************************************
      REAL*8 X,V,Z,P,DEXP
      COMMON /C1/ V,Z
      IF (X) 1,2,2
    1 FCT=0.
      GO TO 4
    2 P=Z+V**2/(X+Z)
      IF (P-5.D1) 3,3,1
    3 FCT=DEXP(-P)/(X+Z)
    4 RETURN
      END
      SUBROUTINE DQL12(FCT,Y)
C
C     ..................................................................
C
C        SUBROUTINE DQL12
C
C        PURPOSE
C           TO COMPUTE INTEGRAL(EXP(-X)*FCT(X), SUMMED OVER X
C                               FROM 0 TO INFINITY).
C
C        USAGE
C           CALL DQL12 (FCT,Y)
C           PARAMETER FCT REQUIRES AN EXTERNAL STATEMENT
C
C        DESCRIPTION OF PARAMETERS
C           FCT    - THE NAME OF AN EXTERNAL DOUBLE PRECISION FUNCTION
C                    SUBPROGRAM USED.
C           Y      - THE RESULTING DOUBLE PRECISION INTEGRAL VALUE.
C
C        REMARKS
C           NONE
C
C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C           THE EXTERNAL DOUBLE PRECISION FUNCTION SUBPROGRAM FCT(X)
C           MUST BE FURNISHED BY THE USER.
C
C        METHOD
C           EVALUATION IS DONE BY MEANS OF 12-POINT GAUSSIAN-LAGUERRE
C           QUADRATURE FORMULA, WHICH INTEGRATES EXACTLY,
C           WHENEVER FCT(X) IS A POLYNOMIAL UP TO DEGREE 23.
C           FOR REFERENCE, SEE
C           SHAO/CHEN/FRANK, TABLES OF ZEROS AND GAUSSIAN WEIGHTS OF
C           CERTAIN ASSOCIATED LAGUERRE POLYNOMIALS AND THE RELATED
C           GENERALIZED HERMITE POLYNOMIALS, IBM TECHNICAL REPORT
C           TR00.1100 (MARCH 1964), PP.24-25.
C
C     ..................................................................
```

TABLE 6.1.—*Listing of program for partial penetration in a leaky artesian aquifer*—Continued

```
C
C
C
      DOUBLE PRECISION X,Y,FCT
C
      X=,3709912104446692 D2
      Y=,814807746742624 D-15*FCT(X)
      X=,2848796725098400 D2
      Y=Y+,3061601635035021 D-11*FCT(X)
      X=,2215109037939701 D2
      Y=Y+,1342391030515004 D-8*FCT(X)
      X=,1711685518746226 D2
      Y=Y+,1668493876540910 D-6*FCT(X)
      X=,1300605499330635 D2
      Y=Y+,836505585681980 D-5*FCT(X)
      X=,962131684245687 D1
      Y=Y+,2032315926629994 D-3*FCT(X)
      X=,6844525453115177 D1
      Y=Y+,2663973541865316 D-2*FCT(X)
      X=,4599227639418348 D1
      Y=Y+,2010238115463410 D-1*FCT(X)
      X=,2833751337743507 D1
      Y=Y+,904492222116809 D-1*FCT(X)
      X=,1512610269776419 D1
      Y=Y+,2440820113198776 D0*FCT(X)
      X=,6117574845151307 D0
      Y=Y+,3777592758731380 D0*FCT(X)
      X=,1157221173580207 D0
      Y=Y+,2647313710554432 D0*FCT(X)
      RETURN
      END
      SUBROUTINE BESK(X,N,BK,IER)
C
C     ..................................................................
C
C        SUBROUTINE BESK
C
C           COMPUTE THE K BESSEL FUNCTION FOR A GIVEN ARGUMENT AND ORDER
C
C        USAGE
C           CALL BESK(X,N,BK,IER)
C
C        DESCRIPTION OF PARAMETERS
C           X   =THE ARGUMENT OF THE K BESSEL FUNCTION DESIRED
C           N   =THE ORDER OF THE K BESSEL FUNCTION DESIRED
C           BK  =THE RESULTANT K BESSEL FUNCTION
C           IER=RESULTANT ERROR CODE WHERE
C              IER=0  NO ERROR
C              IER=1  N IS NEGATIVE
C              IER=2  X IS ZERO OR NEGATIVE
C              IER=3  X .GT. 170, MACHINE RANGE EXCEEDED
C              IER=4  BK .GT. 10**70
C
C        REMARKS
C           N MUST BE GREATER THAN OR EQUAL TO ZERO
C
C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C           NONE
C
C        METHOD
C           COMPUTES ZERO ORDER AND FIRST ORDER BESSEL FUNCTIONS USING
C           SERIES APPROXIMATIONS AND THEN COMPUTES N TH ORDER FUNCTION
C           USING RECURRENCE RELATION.
```

TABLE 6.1.—*Listing of program for partial penetration in a leaky artesian aquifer*—Continued

```
C           RECURRENCE RELATION AND POLYNOMIAL APPROXIMATION TECHNIQUE
C           AS DESCRIBED BY A.J.M.HITCHCOCK,'POLYNOMIAL APPROXIMATIONS
C           TO BESSEL FUNCTIONS OF ORDER ZERO AND ONE AND TO RELATED
C           FUNCTIONS', M.T.A.C., V.11,1957,PP.86-88, AND G.N. WATSON,
C           'A TREATISE ON THE THEORY OF BESSEL FUNCTIONS', CAMBRIDGE
C           UNIVERSITY PRESS, 1958, P. 62
C
C     ..................................................................
C
      DIMENSION T(12)
      BK=.0
      IF(N)10,11,11
   10 IER=1
      RETURN
   11 IF(X)12,12,20
   12 IER=2
      RETURN
   20 IF(X-170.0)22,22,21
   21 IER=3
      RETURN
   22 IER=0
      IF(X-1.)36,36,25
   25 A=EXP(-X)
      B=1./X
      C=SQRT(B)
      T(1)=B
      DO 26 L=2,12
   26 T(L)=T(L-1)*B
      IF(N-1)27,29,27
C
C     COMPUTE KO USING POLYNOMIAL APPROXIMATION
C
   27 G0=A*(1.2533141-.1566642*T(1)+.08811128*T(2)-.09139095*T(3)
     2+.1344596*T(4)-.2299850*T(5)+.3792410*T(6)-.5247277*T(7)
     3+.5575368*T(8)-.4262633*T(9)+.2184518*T(10)-.06680977*T(11)
     4+.009189383*T(12))*C
      IF(N)20,28,29
   28 BK=G0
      RETURN
C
C     COMPUTE K1 USING POLYNOMIAL APPROXIMATION
C
   29 G1=A*(1.2533141+.4699927*T(1)-.1468583*T(2)+.1280427*T(3)
     2-.1736432*T(4)+.2847618*T(5)-.4594342*T(6)+.6283381*T(7)
     3-.6632295*T(8)+.5050239*T(9)-.2581304*T(10)+.07880001*T(11)
     4-.01082418*T(12))*C
      IF(N-1)20,30,31
   30 BK=G1
      RETURN
C
C     FROM KO,K1 COMPUTE KN USING RECURRENCE RELATION
C
   31 DO 35 J=2,N
      GJ=2.*(FLOAT(J)-1.)*G1/X+G0
      IF(GJ-1.0E70)33,33,32
   32 IER=4
      GO TO 34
   33 G0=G1
   35 G1=GJ
   34 BK=GJ
      RETURN
   36 B=X/2.
```

TABLE 6.1.—*Listing of program for partial penetration in a leaky artesian aquifer*—Continued

```
      A=,5772157+ALOG(B)
      C=B*B
      IF(N-1)37,43,37
C
C     COMPUTE K0 USING SERIES EXPANSION
C
   37 G0=-A
      X2J=1,
      FACT=1,
      HJ=,0
      DO 40 J=1,6
      RJ=1,/FLOAT(J)
      IF(X2J,LT,1,E-40) X2J=0,
C     PREVIOUS STATEMENT ADDED TO IBM SUBROUTINE TO CORRECT UNDERFLOW
C     PROBLEM ON WATFOR COMPILER
      X2J=X2J*C
      FACT=FACT*RJ*RJ
      HJ=HJ+RJ
   40 G0=G0+X2J*FACT*(HJ-A)
      IF(N)43,42,43
   42 BK=G0
      RETURN
C
C     COMPUTE K1 USING SERIES EXPANSION
C
   43 X2J=B
      FACT=1,
      HJ=1,
      G1=1,/X+X2J*(,5+A-HJ)
      DO 50 J=2,8
      X2J=X2J*C
      RJ=1,/FLOAT(J)
      FACT=FACT*RJ*RJ
      HJ=HJ+RJ
   50 G1=G1+X2J*FACT*(,5+(A-HJ)*FLOAT(J))
      IF(N-1)31,52,31
   52 BK=G1
      RETURN
      END
```

TABLE 7.2.—*Listing of program for constant drawdown in a well in an infinite leaky aquifer*

```
C     ******************************************************************
C
C     PURPOSE
C        TO COMPUTE A TABLE OF FUNCTION VALUES FOR DRAWDOWN IN A
C        LEAKY ARTESIAN AQUIFER IN RESPONSE TO A STEP CHANGE IN
C        WATER LEVEL IN THE CONTROL WELL, FUNCTION VALUES ARE
C        EXPRESSED AS A FRACTION OF DRAWDOWN IN CONTROL WELL (S/SW),
C        REFERENCE - HANTUSH,M,S,, 1959, NONSTEADY FLOW TO FLOWING
C        WELLS IN LEAKY AQUIFERS: JOUR, GEOPHYS, RESEARCH, V, 64,
C        NO, 8, P, 1043-1052,
C     INPUT DATA
C        1 CARD - FORMAT(2E10,5)
C             TSMALL - SMALLEST VALUE OF ALPHA FOR WHICH COMPUTATION
C             IS DESIRED,
C             TLARGE -  LARGEST VALUE OF ALPHA FOR WHICH COMPUTATION
C             IS DESIRED,
C        1 CARD - FORMAT(13F5,0)
C             BDAT - 13 VALUES OF RW/B, NON ZERO VALUES SHOULD BE GE 1
C             AND LT 10, FIRST ZERO (OR BLANK) WILL TERMINATE THE
C             LIST, AT LEAST ONE NON ZERO VALUE MUST BE CODED, INPUT
C             VALUES ARE MULTIPLIED BY POWER OF TEN DETERMINED BY
C             PROGRAM FROM ALPHA,
```

TABLE 7.2.—*Listing of program for constant drawdown in a well in an infinite leaky aquifer*—Continued

```
C          1 CARD = FORMAT(10F8.2)
C               RW = RADIUS OF CONTROL WELL.
C               RDAT = 9 VALUES OF RADIAL DISTANCE OF OBSERVATION POINTS
C               FROM CONTROL WELL. SHOULD BE CODED WITH SMALLEST NUMBER
C               FIRST, THEN BY INCREASING DISTANCE. THE FIRST ZERO
C               (OR BLANK) VALUE WILL TERMINATE COMPUTATION.
C     METHOD
C        EVALUATES EQ. 13 OF HANTUSH. EVALUATION OF BESSEL FUNCTIONS
C        BY SUBROUTINES BESK AND BESY AND FUNCTION J0. EVALUATES
C        INTEGRAL BY SUM, I=1 TO 8000, F((DELTA U)*(I-.5))*(DELTA U),
C        CHOOSES INITIAL DELTA U = .001/SQRT(SMALLEST ALPHA) AND USES
C        THIS VALUE FOR ALL RW/B GE 10*(DELTA U). FOR SMALLER RW/B,
C        DIVIDES DELTA U BY 10 AND MULTIPLIES SMALLEST ALPHA BY 100.
C     REMARKS
C        SMALLEST RW/B GE .01/SQRT(SMALLEST ALPHA)
C     SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C        BESK,BESY,J0
C
C     ******************************************************************
      REAL*8 SUM1,SUM2
      REAL*4 KOBP,KOB,J0,J0PU,J0U,Y(8000),J(8000),F(8000),FT(8000),
     1   FB(8000),RDAT(9),TDAT(6),BDAT(13),ARRAY(25,9,13),B(13),T(25)
      DATA FT/8000*0./,FB/8000*0./
      DATA RDAT/9*1./
      DATA ARRAY/2925*0./,TDAT/1.,1.5,2.,3.,5.,7./
      IRD=5
      IPT=6
      READ (IRD,24) TSMALL,TLARGE
      READ (IRD,23) BDAT
      READ (IRD,22) RW,RDAT
      IBEGIN=ALOG10(TSMALL)
      IEND=ALOG10(TLARGE)+.99999
      IF ((IBEGIN/2*2).LT.IBEGIN) IBEGIN=IBEGIN-1
      ISPAN=IEND-IBEGIN
      MLIMIT=(ISPAN+1)/2
C     COMPUTES INITIAL DELTA U (DU) = .001/SQRT(SMALLEST ALPHA)
      DU=.001/SQRT(TDAT(1)*10.**IBEGIN)
C     EXPONENT (JBEGIN) OF SMALLEST RW/B IS COMPUTED FROM EXPONENT
C     (IBEGIN) OF SMALLEST ALPHA.
      JBEGIN=-IBEGIN/2-2
      DO 1 I=1,13
      IF (BDAT(I).EQ.0.) GO TO 2
    1 CONTINUE
      NB=13
      GO TO 3
    2 NB=I-1
    3 CONTINUE
      DO 4 I=1,9
      IF (RDAT(I).EQ.0.) GO TO 5
    4 RDAT(I)=RDAT(I)/RW
      NR=9
      GO TO 6
    5 NR=I-1
    6 DO 21 M=1,MLIMIT
      NUM=8000
      START=-DU/2.
      U=START
      DO 7 I=1,NUM
      U=U+DU
      CALL BESY(U,0,Y(I),IDUMY)
    7 J(I)=J0(U)
      DO 19 IR=1,NR
```

TABLE 7.2.—*Listing of program for constant drawdown in a well in an infinite leaky aquifer*—Continued

```
      RHO=RDAT(IR)
      U=START
      DO 8 I=1,NUM
      U=U+DU
      CALL BESY(RHO*U,0,YOPU,IDUMY)
      JOPU=J0(RHO*U)
      JOU=J(I)
      YOU=Y(I)
    8 F(I)=(JOPU*YOU-YOPU*JOU)/(JOU*JOU+YOU*YOU)
      DO 19 IT=1,25
      INDEX=(IT-1)/6
      IEXP=IBEGIN+INDEX
      II=IT-INDEX*6
      TAU=TDAT(II)*10.**IEXP
      T(IT)=TAU
      U=START
      NUMT=NUM
      DO 9 I=1,NUMT
      U=U+DU
      FTEST=F(I)
      IF (ABS(FTEST).LT.1.E-30) GO TO 10
      XTEST=-TAU*U*U
      IF (XTEST+69.) 10,10,9
    9 FT(I)=FTEST*EXP(XTEST)
      GO TO 11
   10 NUMT=I-1
      FT(I)=0.
   11 DO 19 IB=1,13
      JNDEX=(IB-1)/NB
      JEXP=JBEGIN+JNDEX
      JJ=IB-JNDEX*NB
      BETA=BDAT(JJ)*10.**JEXP
      B(IB)=BETA
      U=START
      BSQ=BETA*BETA
      NUMB=NUMT
      DO 12 I=1,NUMB
      U=U+DU
      FTEST=FT(I)
      IF (ABS(FTEST).LT.1.E-30) GO TO 13
   12 FB(I)=FTEST/(U+BSQ/U)
      GO TO 14
   13 NUMB=I-1
      FB(I)=0.
   14 SUM1=0.
      SUM2=0.
      DO 15 I=1,NUMB,2
      SUM1=SUM1+FB(I)
   15 SUM2=SUM2+FB(I+1)
      XINT=(SUM1+SUM2)*DU
      CALL BESK(RHO*BETA,0,KOBP,IDUMY)
      CALL BESK(BETA,0,KOB,IDUMY)
      RATIO=0.
      IF (KOBP.GT.0.) RATIO=KOBP/KOB
      XTEST=-TAU*BSQ
      IF (XTEST+30.) 16,17,17
   16 XPT=0.
      GO TO 18
   17 XPT=EXP(XTEST)
   18 Z=RATIO+.6366198*XPT*XINT
      IF ((Z.LT.0.).AND.(Z.GT.-5.E-5)) Z=0.E0
   19 ARRAY(IT,IR,IB)=Z
```

TABLE 7.2.—*Listing of program for constant drawdown in a well in an infinite leaky aquifer*—Continued

```
      DO 20 K=1,NR
      WRITE (IPT,25) RDAT(K),B
      WRITE (IPT,26) (T(I),(ARRAY(I,K,L),L=1,13),I=1,25)
   20 CONTINUE
C     EXPONENT OF SMALLEST RW/B DECREASED BY ONE EACH TIME THROUGH LOOP
      JBEGIN=JBEGIN-1
C     EXPONENT OF SMALLEST ALPHA INCREASED BY TWO EACH TIME THROUGH LOOP
      IBEGIN=IBEGIN+2
C     DELTA U (DU) IS DIVIDED BY 10 EACH TIME THROUGH THE LOOP
   21 DU=.1*DU
      STOP
C
   22 FORMAT (10F8.2)
   23 FORMAT (13F5.0)
   24 FORMAT (2E10.5)
   25 FORMAT ('1','Z(ALPHA,R/RW,RW/B), R/RW=',F6.0/'0',9X,'I RW/B'/(' ',
     13X,'ALPHA I',13E9.2))
   26 FORMAT (' ',E10.3,13F9.3)
      END
      REAL FUNCTION J0*4(X)
C     ******************************************************************
C
C     FUNCTION J0
C
C     PURPOSE
C        TO COMPUTE THE ZERO ORDER J BESSEL FUNCTION FOR A GIVEN
C        ARGUMENT.
C     USAGE
C        J0(X)
C     DESCRIPTION OF PARAMETER
C        X - REAL*4, ARGUMENT OF J0 BESSEL FUNCTION DESIRED.
C     SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C        NONE.
C     METHOD
C        POLYNOMIAL APPROXIMATION FOR X<4 AND ASYMPTOTIC SERIES FOR
C        X GE 4. THE POLYNOMIAL APPROXIMATION IS THE FIRST 10 TERMS OF
C        THE POWER SERIES FOR J0(X) (MILLER, K.S., 1957,
C        ENGINEERING MATHEMATICS: RINEHART AND CO., INC., NEW YORK,
C        P. 120). THE ASYMPTOTIC EXPANSION OF J0(X) IS GIVEN ON P. 82
C        OF BOWMAN, FRANK, 1958, INTRODUCTION  TO BESSEL FUNCTIONS:
C        DOVER PUBLICATIONS INC., NEW YORK. THE TERMS P ('A*P0') AND
C        Q ('-B*Q0') OF THE ASYMPTOTIC EXPANSION ARE COMPUTED BY AN
C        ALGORITHM FROM IBM SUBROUTINE BESY.
C
C     ******************************************************************
      IF (X-4.) 1,3,3
C     COMPUTE J0 BY FIRST 10 TERMS OF POWER SERIES
    1 A=-X*X/4.
      B=1.
      DO 2 I=1,10
      C=11.-I
    2 B=1.+B*(A/(C*C))
      J0=B
      GO TO 4
C     COMPUTE J0 BY ASYMPTOTIC SERIES
    3 T1=4./X
      T2=T1*T1
      P0=((((-.0000037043*T2+.0000173565)*T2-.0000487613)*T2+.00017343)*
     1T2-.001753062)*T2+.3989423
      Q0=((((.0000032312*T2-.0000142078)*T2+.0000342468)*T2-.0000869791)
     1*T2+.0004564324)*T2-.01246694
      A=2.0/SQRT(X)
```

TABLE 7.2.—*Listing of program for constant drawdown in a well in an infinite leaky aquifer*—Continued

```
      B=A*T1
      C=X-.7853982
      J0=A*P0*COS(C)-B*Q0*SIN(C)
    4 RETURN
      END
      SUBROUTINE BESY(X,N,BY,IER)
C
C     ..................................................................
C
C        SUBROUTINE BESY
C
C        PURPOSE
C           COMPUTE THE Y BESSEL FUNCTION FOR A GIVEN ARGUMENT AND ORDER
C
C        USAGE
C           CALL BESY(X,N,BY,IER)
C
C        DESCRIPTION OF PARAMETERS
C           X  -THE ARGUMENT OF THE Y BESSEL FUNCTION DESIRED
C           N  -THE ORDER OF THE Y BESSEL FUNCTION DESIRED
C           BY -THE RESULTANT Y BESSEL FUNCTION
C           IER-RESULTANT ERROR CODE WHERE
C              IER=0  NO ERROR
C              IER=1  N IS NEGATIVE
C              IER=2  X IS NEGATIVE OR ZERO
C              IER=3  BY HAS EXCEEDED MAGNITUDE OF 10**70
C
C        REMARKS
C           VERY SMALL VALUES OF X MAY CAUSE THE RANGE OF THE LIBRARY
C           FUNCTION ALOG TO BE EXCEEDED
C           X MUST BE GREATER THAN ZERO
C           N MUST BE GREATER THAN OR EQUAL TO ZERO
C
C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C           NONE
C
C        METHOD
C           RECURRENCE RELATION AND POLYNOMIAL APPROXIMATION TECHNIQUE
C           AS DESCRIBED BY A.J.M.HITCHCOCK,'POLYNOMIAL APPROXIMATIONS
C           TO BESSEL FUNCTIONS OF ORDER ZERO AND ONE AND TO RELATED
C           FUNCTIONS', M.T.A.C., V.11,1957,PP.86-88, AND G.N. WATSON,
C           'A TREATISE ON THE THEORY OF BESSEL FUNCTIONS', CAMBRIDGE
C           UNIVERSITY PRESS, 1958, P. 62
C
C     ..................................................................
C
C
C     CHECK FOR ERRORS IN N AND X
C
      IF(N)180,10,10
   10 IER=0
      IF(X)190,190,20
C
C     BRANCH IF X LESS THAN OR EQUAL 4
C
   20 IF(X-4.0)40,40,30
C
C        COMPUTE Y0 AND Y1 FOR X GREATER THAN 4
C
   30 T1=4.0/X
      T2=T1*T1
      P0=((((-.0000037043*T2+.0000173565)*T2-.0000487613)*T2
```

TABLE 7.2.—*Listing of program for constant drawdown in a well in an infinite leaky aquifer*—Continued

```
     1   +.00017343)*T2-.001753062)*T2+.3989423
      Q0=((((.0000032312*T2-.0000142078)*T2+.0000342468)*T2
     1   -.0000869791)*T2+.0004564324)*T2-.01246694
      P1=((((.0000042414*T2-.0000200920)*T2+.0000580759)*T2
     1   -.000223203)*T2+.002921826)*T2+.3989423
      Q1=((((-.0000036594*T2+.00001622)*T2-.0000398708)*T2
     1   +.0001064741)*T2-.0006390400)*T2+.03740084
      A=2.0/SQRT(X)
      B=A*T1
      C=X-.7853982
      Y0=A*P0*SIN(C)+B*Q0*COS(C)
      Y1=-A*P1*COS(C)+B*Q1*SIN(C)
      GO TO 90
C
C        COMPUTE Y0 AND Y1 FOR X LESS THAN OR EQUAL TO 4
C
   40 XX=X/2.
      X2=XX*XX
      T=ALOG(XX)+.5772157
      SUM=0.
      TERM=T
      Y0=T
      DO 70 L=1,15
      IF(L-1)50,60,50
   50 SUM=SUM+1./FLOAT(L-1)
   60 FL=L
      TS=T-SUM
      IF(ABS(TERM).LE.1.E-40) TERM=0.
      TERM=(TERM*(-X2)/FL**2)*(1.-1./(FL*TS))
   70 Y0=Y0+TERM
      TERM = XX*(T-.5)
      SUM=0.
      Y1=TERM
      DO 80 L=2,16
      SUM=SUM+1./FLOAT(L-1)
      FL=L
      FL1=FL-1.
      TS=T-SUM
      IF(ABS(TERM).LE.1.E-40) TERM=0.
      TERM=(TERM*(-X2)/(FL1*FL))*((TS-.5/FL)/(TS+.5/FL1))
   80 Y1=Y1+TERM
      PI2=.6366198
      Y0=PI2*Y0
      Y1=-PI2/X+PI2*Y1
C
C     CHECK IF ONLY Y0 OR Y1 IS DESIRED
C
   90 IF(N-1)100,100,130
C
C     RETURN EITHER Y0 OR Y1 AS REQUIRED
C
  100 IF(N)110,120,110
  110 BY=Y1
      GO TO 170
  120 BY=Y0
      GO TO 170
C
C    PERFORM RECURRENCE OPERATIONS TO FIND YN(X)
C
  130 YA=Y0
      YB=Y1
      K=1
```

TABLE 7.2.—*Listing of program for constant drawdown in a well in an infinite leaky aquifer*—Continued

```
 140 T=FLOAT(2*K)/X                                                   BESY1180
     YC=T*YB-YA                                                       BESY1190
     IF(ABS(YC)-1.0E70)145,145,141                                    BESY1200
 141 IER=3                                                            BESY1210
     RETURN                                                           BESY1220
 145 K=K+1                                                            BESY1230
     IF(K-N)150,160,150                                               BESY1240
 150 YA=YB                                                            BESY1250
     YB=YC                                                            BESY1260
     GO TO 140                                                        BESY1270
 160 BY=YC                                                            BESY1280
 170 RETURN                                                           BESY1290
 180 IER=1                                                            BESY1300
     RETURN                                                           BESY1310
 190 IER=2                                                            BESY1320
     RETURN                                                           BESY1330
     END                                                              BESY134-
     SUBROUTINE BESK(X,N,BK,IER)                                      BESK 410
C                                                                     BESK  10
C    .................................................................BESK  20
C                                                                     BESK  30
C       SUBROUTINE BESK                                               BESK  40
C                                                                     BESK  50
C          COMPUTE THE K BESSEL FUNCTION FOR A GIVEN ARGUMENT AND ORDERBESK  60
C                                                                     BESK  70
C       USAGE                                                         BESK  80
C          CALL BESK(X,N,BK,IER)                                      BESK  90
C                                                                     BESK 100
C       DESCRIPTION OF PARAMETERS                                     BESK 110
C          X  -THE ARGUMENT OF THE K BESSEL FUNCTION DESIRED          BESK 120
C          N  -THE ORDER OF THE K BESSEL FUNCTION DESIRED             BESK 130
C          BK -THE RESULTANT K BESSEL FUNCTION                        BESK 140
C          IER-RESULTANT ERROR CODE WHERE                             BESK 150
C             IER=0  NO ERROR                                         BESK 160
C             IER=1  N IS NEGATIVE                                    BESK 170
C             IER=2  X IS ZERO OR NEGATIVE                            BESK 180
C             IER=3  X .GT. 170, MACHINE RANGE EXCEEDED               BESK 190
C             IER=4  BK .GT. 10**70                                   BESK 200
C                                                                     BESK 210
C       REMARKS                                                       BESK 220
C          N MUST BE GREATER THAN OR EQUAL TO ZERO                    BESK 230
C                                                                     BESK 240
C       SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED                 BESK 250
C          NONE                                                       BESK 260
C                                                                     BESK 270
C       METHOD                                                        BESK 280
C          COMPUTES ZERO ORDER AND FIRST ORDER BESSEL FUNCTIONS USING BESK 290
C          SERIES APPROXIMATIONS AND THEN COMPUTES N TH ORDER FUNCTION BESK 300
C          USING RECURRENCE RELATION.                                 BESK 310
C          RECURRENCE RELATION AND POLYNOMIAL APPROXIMATION TECHNIQUE BESK 320
C          AS DESCRIBED BY A.J.M.HITCHCOCK,'POLYNOMIAL APPROXIMATIONS BESK 330
C          TO BESSEL FUNCTIONS OF ORDER ZERO AND ONE AND TO RELATED   BESK 340
C          FUNCTIONS', M.T.A.C., V.11,1957,PP.86-88, AND G.N. WATSON, BESK 350
C          'A TREATISE ON THE THEORY OF BESSEL FUNCTIONS', CAMBRIDGE  BESK 360
C          UNIVERSITY PRESS, 1958, P. 62                              BESK 370
C                                                                     BESK 380
C    .................................................................BESK 390
C                                                                     BESK 400
     DIMENSION T(12)                                                  BESK 420
     BK=.0                                                            BESK 430
     IF(N)10,11,11                                                    BESK 440
  10 IER=1                                                            BESK 450
```

TABLE 7.2.—*Listing of program for constant drawdown in a well in an infinite leaky aquifer*—Continued

```
      RETURN
   11 IF(X)12,12,20
   12 IER=2
      RETURN
   20 IF(X-170.0)22,22,21
   21 IER=3
      RETURN
   22 IER=0
      IF(X-1.)36,36,25
   25 A=EXP(-X)
      B=1./X
      C=SQRT(B)
      T(1)=B
      DO 26 L=2,12
   26 T(L)=T(L-1)*B
      IF(N-1)27,29,27
C
C     COMPUTE K0 USING POLYNOMIAL APPROXIMATION
C
   27 G0=A*(1.2533141-.1566642*T(1)+.08811128*T(2)-.09139095*T(3)
     2+.1344596*T(4)-.2299850*T(5)+.3792410*T(6)-.5247277*T(7)
     3+.5575368*T(8)-.4262633*T(9)+.2184518*T(10)-.06680977*T(11)
     4+.009189383*T(12))*C
      IF(N)20,28,29
   28 BK=G0
      RETURN
C
C     COMPUTE K1 USING POLYNOMIAL APPROXIMATION
C
   29 G1=A*(1.2533141+.4699927*T(1)-.1468583*T(2)+.1280427*T(3)
     2-.1736432*T(4)+.2847618*T(5)-.4594342*T(6)+.6283381*T(7)
     3-.6632295*T(8)+.5050239*T(9)-.2581304*T(10)+.07880001*T(11)
     4-.01082418*T(12))*C
      IF(N-1)20,30,31
   30 BK=G1
      RETURN
C
C     FROM K0,K1 COMPUTE KN USING RECURRENCE RELATION
C
   31 DO 35 J=2,N
      GJ=2.*(FLOAT(J)-1.)*G1/X+G0
      IF(GJ-1.0E70)33,33,32
   32 IER=4
      GO TO 34
   33 G0=G1
   35 G1=GJ
   34 BK=GJ
      RETURN
   36 B=X/2.
      A=.5772157+ALOG(B)
      C=B*B
      IF(N-1)37,43,37
C
C     COMPUTE K0 USING SERIES EXPANSION
C
   37 G0=-A
      X2J=1.
      FACT=1.
      HJ=.0
      DO 40 J=1,6
      RJ=1./FLOAT(J)
      IF(X2J.LT.1.E-40) X2J=0.
C     PREVIOUS STATEMENT ADDED TO IBM SUBROUTINE TO CORRECT UNDERFLOW
```

TABLE 7.2.—*Listing of program for constant drawdown in a well in an infinite leaky aquifer*—Continued

```
C     PROBLEM ON WATFOR COMPILER
      X2J=X2J*C
      FACT=FACT*RJ*RJ
      HJ=HJ+RJ
   40 G0=G0+X2J*FACT*(HJ-A)
      IF(N)43,42,43
   42 BK=G0
      RETURN
C
C     COMPUTE K1 USING SERIES EXPANSION
C
   43 X2J=B
      FACT=1.
      HJ=1.
      G1=1./X+X2J*(.5+A-HJ)
      DO 50 J=2,8
      X2J=X2J*C
      RJ=1./FLOAT(J)
      FACT=FACT*RJ*RJ
      HJ=HJ+RJ
   50 G1=G1+X2J*FACT*(.5+(A-HJ)*FLOAT(J))
      IF(N-1)31,52,31
   52 BK=G1
      RETURN
      END
```

TABLE 8.2.—*Listing of programs for constant discharge from a fully penetrating well of finite diameter*

```
C***********************************************************************
C
C     PURPOSE
C        COMPUTES FUNCTION VALUES OF F(U,ALPHA,RHO) FOR RHO > 1 -
C        PAPADOPULOS,I.S. AND COOPER,H.H.,JR., 1967, DRAWDOWN IN
C        A WELL OF LARGE DIAMETER: WATER RESOURCES RESEARCH, V. 3,
C        NO. 1, P. 241-244.
C        PROGRAM BY S.S.PAPADOPULOS.
C     INPUT DATA - ONE OR MORE GROUPS, EACH GROUP CODED AS FOLLOWS
C        1 CARD - FORMAT(2E10.5)
C             ALPHA  -      RW**2*S/RC**2 - RADIUS OF WELL (SCREEN
C                   OR OPEN BORE IN AQUIFER) SQUARED * STORAGE
C                   COEFFICIENT / RADIUS OF CASING (OVER INTERVAL OF
C                   WATER LEVEL CHANGE) SQUARED.
C             RHO - R/RW - DISTANCE FROM PUMPED WELL / RADIUS OF
C                   WELL (SCREEN OR OPEN BORE IN AQUIFER), MUST BE
C                   GREATER THAN ONE.
C        1 CARD - FORMAT(16E5.0)
C             U- 16 VALUES OF U - R**2*S/(4*T*TIME) - DISTANCE FROM
C                   PUMPED WELL SQUARED * STORAGE COEFFICIENT /
C                   4 * TRANSMISSIVITY * TIME. IF LESS THAN 16 DESIRED,
C                   BLANK OR ZERO VALUES MAY BE CODED FOR THE REST.
C     SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C        PEAK,SIMP,APEKE,EXBSL1,JY0,JY1,ROOTS - MUST BE IN DECK.
C
C***********************************************************************
      DIMENSION V(40,40),U(16)
      COMMON XPK,YPK
      COMMON/PBLK/A,B,RHO
      EXTERNAL EXBSL1
    1 READ (5,16,END=15) ALPHA,RHO
      IF (ALPHA) 15,15,2
```

TABLE 8.2.—*Listing of programs for constant discharge from a fully penetrating well of finite diameter*—Continued

```
    2 WRITE (6,17) ALPHA,RHO
      WRITE (6,18)
    3 READ (5,19) U
      DO 14 II=1,16
      IF (U(II)) 1,1,4
    4 A=ALPHA+ALPHA
      B=0.25/U(II)
      CALL APEKE(EXBSL1)
      CALL PEAK(EXBSL1)
      IF (XPK-1.0E-8) 5,6,6
    5 WRITE (6,20) XPK,U
      GO TO 3
    6 IF (XPK-3.0) 8,7,7
    7 WRITE (6,21) XPK,U
      GO TO 3
    8 EPS=0.000001
      HBAR=0.007*XPK
      CALL SIMPS(0.0,XPK,EPS,HBAR,SUM,DEL,EXBSL1)
      XM1=((3.14159265*7.0)/(8.0*(RHO-1.))+1.E-6)*RHO/2.
      DX1=XM1-(1.0E-6)*RHO
      DXN=(2.0*3.14159265*RHO)/(5.*(RHO-1.))
      DL=3.14159265*RHO/(RHO-1.)
      CALL ROOTS(XM1,DX1,RT1,EXBSL1)
      HBAR=0.007*(RT1-XPK)
      CALL SIMPS(XPK,RT1,EPS,HBAR,TRM1,ERR1,EXBSL1)
      SUM=SUM+TRM1
      DEL=DEL+ERR1
      X1=RT1
      I=1
    9 XM=X1+DL
      CALL ROOTS(XM,DXN,X2,EXBSL1)
      HBAR=0.007*(X2-X1)
      CALL SIMPS(X1,X2,EPS,HBAR,TRM,ERR,EXBSL1)
      V(1,I)=ABS(TRM)
      DEL=DEL+ERR
      I=I+1
      IF (I-40) 10,10,11
   10 X1=X2
      GO TO 9
   11 EST=0.0
      DO 12 K=2,40
      M=41-K
      DO 12 J=1,M
   12 V(K,J)=V(K-1,J+1)-V(K-1,J)
      DO 13 N=1,40
      L=N-1
      DELV=(-0.5)**L*V(N,1)
   13 EST=EST+(0.5)*DELV
      SUM=SUM-EST
      PUAR=4.0*A*RHO*SUM/3.14159265
      WRITE (6,22) U(II),SUM,DEL,PUAR
   14 CONTINUE
      GO TO 1
   15 STOP
C
   16 FORMAT (2E10.5)
   17 FORMAT ('1','F(U,ALPHA,RHO) FOR ALPHA=',1PE13.5,', RHO=',1E13.5)
   18 FORMAT (1H0,12X,1HU,16X,8HINTEGRAL,9X,14HINTEGRAL ERROR,6X,14HF(U,
     1ALPHA,RHO)/1H )
   19 FORMAT (16E5.0)
```

TABLE 8.2.—*Listing of programs for constant discharge from a fully penetrating well of finite diameter*—Continued

```
   20 FORMAT (5H XPK=,E15.8,3X,16HTOO SMALL FOR U=,E10.3)
   21 FORMAT (5H XPK=,E15.8,3X,16HTOO LARGE FOR U=,E10.3)
   22 FORMAT (1H ,1P4E20.8)
      END
      FUNCTION EXBSL1(X)
C***********************************************************************
C
C     PURPOSE
C        COMPUTES VALUES OF THE INTEGRAND FOR F(U,ALPHA,RHO)
C     DESCRIPTION OF PARAMETER
C        X- REAL - ARGUMENT OF INTEGRAND
C
C***********************************************************************
      COMMON/PBLK/A,B,R
      IF (X) 1,1,2
    1 EXBSL1=0.
      GO TO 8
    2 W=X/R
      IF (W-1.0E7) 4,4,3
    3 FNU=A*COS(W*(R-1.0))-W*SIN(W*(R-1.0))
      DE=(W*W*SQRT(R))*(W*W+A*A)
      EXBSL1=FNU/DE
      GO TO 8
    4 Y=B*X*X
      IF (Y-0.01) 5,5,6
    5 EXPO=Y*(1.0-Y*(0.5-Y*((1.0/6.0)-Y*(1.0/24.0))))
      GO TO 7
    6 EXPO=1.0-EXP(-Y)
    7 CALL JY0(W,WJ0,WY0)
      CALL JY1(W,WJ1,WY1)
      AW=W*WY0-A*WY1
      BW=W*WJ0-A*WJ1
      CALL JY0(X,BJ0,BY0)
      FNUM=EXPO*(AW*BJ0-BW*BY0)
      DEN=X*X*(AW*AW+BW*BW)
      EXBSL1=FNUM/DEN
    8 RETURN
      END
      SUBROUTINE ROOTS(XM,DX,ROOT,F)
C***********************************************************************
C
C     PURPOSE
C        SEARCHES FOR ROOT OF F IN THE INTERVAL XM-DX TO XM+DX.
C     DESCRIPTION OF PARAMETERS - ALL REAL
C        XM - CENTER OF INTERVAL SEARCHED.
C        DX - HALF WIDTH OF INTERVAL SEARCHED.
C        ROOT - RETURNED ROOT LOCATION.
C        F - FUNCTION REFERENCE.
C
C***********************************************************************
      XL=XM-DX
      XR=XM+DX
      YL=F(XL)
      YR=F(XR)
      EP=0.000001*ABS(YL)
      DO 9 I=1,200
      YM=F(XM)
      UP=ABS(YM)
      IF (UP.LT.EP.AND.UP.LT.1.0D-7) GO TO 1
      IF (YM) 2,1,2
```

TABLE 8.2.—*Listing of programs for constant discharge from a fully penetrating well of finite diameter*—Continued

```
    1 ROOT=XM
      GO TO 10
    2 IF (YM*YL) 7,3,4
    3 ROOT=XL
      GO TO 10
    4 IF (YM*YR) 8,5,6
    5 ROOT=XR
      GO TO 10
    6 WRITE (6,11) XL,XR
      STOP
    7 XR=XM
      YR=YM
      GO TO 9
    8 XL=XM
      YL=YM
    9 XM=(XL+XR)/2.0
      ROOT=XM
   10 RETURN
C
   11 FORMAT (1H ,10X,27HNO ROOT IN INTERVAL XM-DX =,1PE20.8,5X,11HAND X
     1M+DX =,1PE20.8/)
      END
      SUBROUTINE APEKE(EXBSL)
C**********************************************************************
C
C     PURPOSE
C        GETS FIRST APPROXIMATION TO PEAK POSITION
C
C**********************************************************************
      COMMON XPK,YPK
      XPK=0.0
      YPK=0.0
      DO 2 I=1,17
      X=10.0**(I-9)
      Y=EXBSL(X)
      IF (Y-YPK) 3,3,1
    1 XPK=X
      YPK=Y
    2 CONTINUE
    3 RETURN
      END
      SUBROUTINE PEAK(EXBSL)
C**********************************************************************
C
C     PURPOSE
C        ATTEMPTS TO FIND POSITION OF MAXIMUM FOR INTEGRAND
C
C**********************************************************************
      COMMON XPK,YPK
      YPK=EXBSL(XPK)
      DO 13 L=1,200
      DX=0.01*XPK
      XL=XPK-DX
      YL=EXBSL(XL)
      XR=XPK+DX
      YR=EXBSL(XR)
      DEN=YR+YL-YPK-YPK
      IF (DEN) 1,9,1
    1 X=XPK-0.5*(YR-YL)*DX/DEN
    2 IF (X) 3,4,4
```

TABLE 8.2.—*Listing of programs for constant discharge from a fully penetrating well of finite diameter*—Continued

```
    3 X=0.0
    4 Y=EXBSL(X)
      IF (YR-Y) 6,6,5
    5 Y=YR
      X=XR
    6 IF (YL-Y) 8,8,7
    7 Y=YL
      X=XL
    8 IF (Y-YPK) 14,14,12
    9 IF (YR-YPK) 11,10,10
   10 X=XPK+DX+DX
      GO TO 2
   11 X=XPK-DX-DX
      GO TO 2
   12 YPK=Y
      XPK=X
   13 CONTINUE
   14 RETURN
      END

      SUBROUTINE SIMPS(Q,R,EPS,HBAR,AREA,DEL,F)
C***********************************************************************
C
C     PURPOSE
C        TO DETERMINE THE INTEGRAL OF A FUNCTION, F, FROM Q TO R,
C        USING SIMPSON'S RULE.
C     DESCRIPTION OF PARAMETERS
C        ALL REAL
C        Q - LOWER LIMIT OF INTEGRAL
C        R - UPPER LIMIT OF INTEGRAL
C        EPS - DESIRED ACCURACY
C        HBAR - MINIMUM DIVISION OF THE INTERVAL
C        AREA - COMPUTED VALUE OF INTEGRAL BETWEEN Q AND R
C        DEL - COMPUTED ESTIMATE OF ERROR
C        F- THE INTEGRAND (FUNCTION REFERENCE)
C     METHOD
C        USES SIMPSON'S RULE TO COMPUTE A SUM APPROXIMATING THE INTEGRAL
C        USES INITIAL H=(R-Q)/2. COMPUTES A SEQUENCE OF SUMS BY HALVING
C        H EACH TIME. COMPUTES ESTIMATE OF ERROR (DEL) AS (PREVIOUS
C        SUM - CURRENT SUM)/15. COMPUTATION STOPS WHEN 1) H<HBAR,
C        2) ABS(DEL)<ABS(EPS*CURRENT SUM). IF HBAR IS LE 0,
C        THEN HBAR=.007*(R-Q).
C
C***********************************************************************
      H=R-Q
      IF (H) 1,1,2
    1 AREA=0.0
      DEL=0.0
      GO TO 10
C R MUST BE GREATER THAN Q
    2 SP=1.0E35
      S3=0.0
      S1=F(Q)+F(R)
      IF (HBAR) 3,3,4
    3 HBAR=0.007*H
    4 S2=0.0
      X=Q+0.5*H
    5 S2=S2+4.0*F(X)
      X=X+H
      IF (X-R) 5,5,6
    6 SC=(S1+S2+S3)*H*0.16666667
```

TABLE 8.2.—*Listing of programs for constant discharge from a fully penetrating well of finite diameter*—Continued

```
      DEL=0.066666667*(SP-SC)
      IF (ABS(DEL)-ABS(EPS*SC)) 7,8,8
    7 AREA=SC-DEL
      GO TO 10
    8 S3=S3+0.5*S2
      H=0.5*H
      IF (H-HBAR) 7,9,9
    9 SP=SC
      GO TO 4
   10 RETURN
      END
      SUBROUTINE JY0(X,J0,Y0)
C***********************************************************************
C
C     PURPOSE
C        COMPUTES BESSEL FUNCTIONS OF THE FIRST AND SECOND KIND,
C        ZERO ORDER, FOR POSITIVE ARGUMENTS.
C        SEE NBS AMS 55, P. 369-370.
C     DESCRIPTION OF PARAMETERS - ALL REAL
C        X- ARGUMENT, MUST BE >0
C        J0 - RETURNED FUNCTION VALUE, J0(X)
C        Y0 - RETURNED FUNCTION VALUE, Y0(X)
C
C***********************************************************************
      REAL J0
      IF (X-3.0) 1,2,3
    1 IF (X) 4,4,2
    2 Z=(0.33333333*X)**2
      J0=1.0-Z*(2.2499997-Z*(1.2656208-Z*(0.3163866-Z*(0.0444479-Z*(0.00
     139444-0.00021*Z)))))
      Y0=0.63661977*ALOG(0.5*X)*J0+0.36746691+Z*(0.60559366-Z*(0.7435038
     14-Z*(0.25300117-Z*(0.04261214-Z*(0.00427916-0.00024846*Z)))))
      RETURN
    3 Z=3.0/X
      F=0.79788456-Z*(0.77E-6+Z*(0.0055274+Z*(0.00009512-Z*(0.00137237-Z
     1*(0.00072805-0.00014476*Z)))))
      P=0.78539816+Z*(0.04166397+Z*(0.00003954-Z*(0.00262573-Z*(0.000541
     125+Z*(0.00029333-0.00013558*Z)))))
      Q=SQRT(1.0/X)
      J0=Q*F*COS(X-P)
      Y0=Q*F*SIN(X-P)
    4 RETURN
      END
      SUBROUTINE JY1(X,J1,Y1)
C***********************************************************************
C
C     PURPOSE
C        COMPUTES BESSEL FUNCTIONS OF THE FIRST AND SECOND KIND,
C        FIRST ORDER, FOR POSITIVE ARGUMENTS.
C        SEE NBS AMS 55, P. 370.
C     DESCRIPTION OF PARAMETERS - ALL REAL
C        X- ARGUMENT, MUST BE >0
C        J1 - RETURNED FUNCTION VALUE, J1(X)
C        Y1 - RETURNED FUNCTION VALUE, Y1(X)
C
C***********************************************************************
      REAL J1
      IF (X-3.0) 1,2,3
    1 IF (X) 4,4,2
    2 Z=(0.33333333*X)**2
```

TABLE 8.2.—*Listing of programs for constant discharge from a fully penetrating well of finite diameter*—Continued

```
      J1=X*(0,5-Z*(0,56249985-Z*(0,21093573-Z*(0,03954289-Z*(0,00443319-
     1Z*(0,00031761-0,00001109*Z))))))
      Y1=0,63661977*ALOG(0,5*X)*J1+(-0,6366198+Z*(0,2212091+Z*(2,1682709
     1-Z*(1,3164827-Z*(0,3123951-Z*(0,0400976-0,0027873*Z))))))/X
      RETURN
    3 Z=3,0/X
      F=0,79788456+Z*(0,156E-5+Z*(0,01659667+Z*(0,00017105-Z*(0,00249511
     1-Z*(0,00113653-0,00020033*Z)))))
      P=0,78539816-Z*(0,12499612+Z*(0,0000565-Z*(0,00637879-Z*(0,0007434
     18+Z*(0,00079824-0,00029166*Z)))))
      Q=SQRT(1,0/X)
      J1=Q*F*SIN(X-P)
      Y1=-Q*F*COS(X-P)
    4 RETURN
      END
C***********************************************************************
C
C     PURPOSE
C        COMPUTES FUNCTION VALUES OF F(UW,ALPHA)  -
C        PAPADOPULOS,I,S, AND COOPER,H,H,,JR,, 1967, DRAWDOWN IN
C        A WELL OF LARGE DIAMETER: WATER RESOURCES RESEARCH, V, 3,
C        NO, 1, P, 241-244,
C        PROGRAM BY S,S,PAPADOPULOS,
C     INPUT DATA - ONE OR MORE GROUPS, EACH GROUP CODED AS FOLLOWS
C        1 CARD - FORMAT (E10,5)
C              S - (ALPHA) - RW**2*S/RC**2 - RADIUS OF WELL (SCREEN
C                    OR OPEN BORE IN AQUIFER) SQUARED * STORAGE
C                    COEFFICIENT / RADIUS OF CASING (OVER INTERVAL OF
C                    WATER LEVEL CHANGE) SQUARED,
C        1 CARD - FORMAT(16E5,0)
C              U- 16 VALUES OF UW - RW**2*S/(4*T*TIME) - RADIUS OF
C                    PUMPED WELL SQUARED * STORAGE COEFFICIENT /
C                    4 * TRANSMISSIVITY * TIME, IF LESS THAN 16 DESIRED,
C                    BLANK OR ZERO VALUES MAY BE CODED FOR THE REST,
C     SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C        PEAK,SIMP,APEKE,EXBSL2,JY0,JY1 - MUST BE INCLUDED IN DECK,
C
C***********************************************************************
      COMMON XPK,YPK
      COMMON/PBLK/A,B
      EXTERNAL EXBSL2
      DIMENSION U(16)
      EPS=0,0001
    1 READ (5,13,END=12) S
      IF (S) 1,1,2
    2 READ (5,14) U
      WRITE (6,15) S
      DO 11 I=1,16
      UW=U(I)
      IF (UW) 1,1,3
    3 B=0,25/UW
      A=S+S
      CALL APEKE(EXBSL2)
      CALL PEAK(EXBSL2)
      IF (XPK-1,0E-8) 4,5,5
    4 WRITE (6,16) UW,S,XPK,YPK
      GO TO 11
    5 IF (XPK-1,0E8) 7,7,6
    6 WRITE (6,17) UW,S,XPK,YPK
      GO TO 11
    7 HBAR=0,007*XPK
```

TABLE 8.2.—*Listing of programs for constant discharge from a fully penetrating well of finite diameter*—Continued

```
      CALL SIMPS(0.,0.,XPK,EPS,HBAR,SUM,DEL,EXBSL2)                      FUA
      X2=XPK                                                             FUA
      DX=XPK                                                             FUA
    8 DX=10.0*DX                                                         FUA
      X1=X2                                                              FUA
      X2=X1+DX                                                           FUA
      Y=EXBSL2(X2)                                                       FUA
      HBAR=0.007*DX                                                      FUA
      CALL SIMPS(X1,X2,EPS,HBAR,TRM,ERR,EXBSL2)                          FUA
      SUM=SUM+TRM                                                        FUA
      DEL=DEL+ERR                                                        FUA
      IF (X2-1.0E9) 9,10,10                                              FUA
    9 YT=1.5707963/X2**4                                                 FUA
      IF (ABS(Y-YT)/YT-0.5E-6) 10,8,8                                    FUA
   10 EST=0.52359878/X2**3                                               FUA
      SUM=SUM+EST                                                        FUA
      FUWS=3.2422779*S*S*SUM                                             FUA
      WRITE (6,18) UW,SUM,DEL,FUWS,XPK,YPK                               FUA
   11 CONTINUE                                                           FUA
      GO TO 1                                                            FUA
   12 STOP                                                               FUA
C                                                                        FUA
   13 FORMAT (E10.5)                                                     FUA
   14 FORMAT (16E5.0)                                                    FUA
   15 FORMAT ('1','F(UW,ALPHA) FOR ALPHA=',1PE14.5/'0',7X,'UW',12X,'INTEFUA
     1GRAL',5X,'INTEGRAL ERROR',5X,'F(UW,ALPHA)',8X,'X(PEAK)',10X,'Y(PEAFUA
     2K)'/' ')                                                           FUA
   16 FORMAT (1H ,1PE14.7,9X,34HVALUES OF DUMMY VARIABLE TOO SMALL,1PE25FUA
     1.7,1PE17.7)                                                        FUA
   17 FORMAT (1H ,1PE14.7,9X,34HVALUES OF DUMMY VARIABLE TOO LARGE,1PE25FUA
     1.7,1PE17.7)                                                        FUA
   18 FORMAT (1H ,1PE14.5,1P5E17.5)                                      FUA
      END                                                                FUA

      FUNCTION EXBSL2(X)                                                 EB2
C***********************************************************************EB2
C                                                                        EB2
C     PURPOSE                                                            EB2
C        COMPUTES VALUES OF THE INTEGRAND FOR F(UW,ALPHA)                EB2
C     DESCRIPTION OF PARAMETER                                           EB2
C        X- REAL - ARGUMENT OF INTEGRAND                                 EB2
C                                                                        EB2
C***********************************************************************EB2
      COMMON/PBLK/A,B                                                    EB2
      IF (X) 1,1,2                                                       EB2
    1 EXBSL2=0.                                                          EB2
      GO TO 8                                                            EB2
    2 IF (X-1.E+7) 4,4,3                                                 EB2
    3 EXBSL2=1.5707963/X**4                                              EB2
      GO TO 8                                                            EB2
    4 Y=B*X*X                                                            EB2
      IF (Y-.01) 5,5,6                                                   EB2
    5 FNUM=Y*(1.-Y*(.5-Y*((1./6.)-Y*(1./24.))))                          EB2
      GO TO 7                                                            EB2
    6 FNUM=1.-EXP(-Y)                                                    EB2
    7 CALL JY0(X,BJ0,BY0)                                                EB2
      CALL JY1(X,BJ1,BY1)                                                EB2
      DEN=((X*BJ0-A*BJ1)**2+(X*BY0-A*BY1)**2)*X**3                       EB2
      EXBSL2=FNUM/DEN                                                    EB2
    8 RETURN                                                             EB2
      END                                                                EB2
```

TABLE 9.2.—*Listing of program to compute change in water level due to sudden injection of a slug of water into a well*

```
C***********************************************************************
C
C     PURPOSE
C        COMPUTES FUNCTION VALUES OF F(BETA,ALPHA) - THE SLUG TEST
C        FUNCTION - COOPER,H.H.,JR., BREDEHOEFT,J.D., AND PAPADOPULOS,
C        I.S., 1967, RESPONSE OF A FINITE-DIAMETER WELL TO AN
C        INSTANTANEOUS CHARGE OF WATER: WATER RESOURCES RESEARCH,
C        V. 3, NO. 1, P. 263-269.
C        PROGRAM BY S.S.PAPADOPULOS.
C     INPUT DATA
C        1 OR MORE CARDS - FORMAT(F16.5)
C              A - (ALPHA) - RW**2*S/RC**2 - RADIUS OF WELL (SCREEN OR
C                     OPEN BORE IN AQUIFER) SQUARED * STORAGE COEFFICIENT
C                     / RADIUS OF CASING (OVER INTERVAL OF WATER LEVEL
C                     CHANGE) SQUARED.
C     SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C        PRX,DJY0,DJY1,DSIMPS - MUST BE INCLUDED IN DECK
C     METHOD
C        THIS PROGRAM CALCULATES THE SLUG TEST FUNCTION, F(BETA,ALPHA),
C        FOR VALUES OF BETA RANGING FROM 0.001 TO 1000.0 BY INCREMENTING
C        BETA ACCORDING TO DATA ARRAY BB(I). AVERAGE COMPUTATION TIME
C        IS ABOUT 30 SECONDS PER VALUE OF ALPHA ON IBM 360/155.
C
C***********************************************************************
      DOUBLE PRECISION A,B,PI,ZZ,EPS,Y,X1,X2,TERM,FAB,DATAN,DEL,HBAR
      DIMENSION ZZ(40), BB(39)
      COMMON A,B,PI
      EXTERNAL PRX
      DATA ZZ/0.D+0,1.D-10,1.D-9,1.D-8,1.D-7,1.D-6,1.D-5,1.D-4,
     1   1.D-3,1.D-2,1.D-1,2.D-1,3.D-1,4.D-1,5.D-1,6.D-1,7.D-1,8.D-1,
     2   9.D-1,1.D+0,2.D+0,3.D+0,4.D+0,5.D+0,6.D+0,7.D+0,8.D+0,
     3   9.D+0,1.D+1,2.D+1,3.D+1,4.D+1,5.D+1,6.D+1,7.D+1,8.D+1,
     4   9.D+1,1.D+2,1.25D+2,1.5D+2/
      DATA BB/.001,.002,.004,.006,.008,.01,.02,.04,.06,.08,.1,.2,.4,.6,.
     18,1.,2.,3.,4.,5.,6.,7.,8.,9.,10.,20.,30.,40.,50.,60.,70.,80.,90.,1
     200.,200.,400.,600.,800.,1000./
      PI=4.*DATAN(1.0D+00)
      EPS=0.00001
    1 READ (5,6) A
      IF (A.LE.0.0) GO TO 5
      WRITE (6,7) A
      WRITE (6,8)
      DO 4 I=1,39
      B=BB(I)
      Y=0.0
      DO 2 L=1,39
      X1=ZZ(L)
      X2=ZZ(L+1)
      HBAR=0.
      CALL DSIMPS(X1,X2,EPS,HBAR,TERM,DEL,PRX)
      Y=Y+TERM
      IF (L.GT.20.AND.TERM.LT.EPS) GO TO 3
    2 CONTINUE
    3 FAB=4.*A*Y/(PI*PI)
    4 WRITE (6,9) B,FAB
      GO TO 1
    5 STOP
C
C
    6 FORMAT (F16.5)
    7 FORMAT ('1',41X,'F(BETA,ALPHA) FOR ALPHA=',1PD9.2/)
    8 FORMAT ('0',53X,'BETA',13X,'H/H0'/)
    9 FORMAT (' ',51X,1PD8.2,10X,0PF6.4)
      END
```

TABLE 9.2.—*Listing of program to compute change in water level due to sudden injection of a slug of water into a well*—Continued

```
      DOUBLE PRECISION FUNCTION PRX(X)
C*************************************************************************
C
C     PURPOSE
C        COMPUTE VALUES OF THE INTEGRAND FOR F(BETA,ALPHA)
C     DESCRIPTION OF PARAMETER
C        X - DOUBLE PRECISION - ARGUMENT OF INTEGRAND
C
C*************************************************************************
      DOUBLE PRECISION A,B,PI,XX,X,C,F1,F2,J0,Y0,J1,Y1
      DOUBLE PRECISION DLOG,DSQRT,DEXP
      COMMON A,B,PI
      XX=DSQRT(A*X/B)
      IF (X) 6,1,2
    1 PRX=(PI*PI)/(16.*A*B)
      GO TO 6
    2 IF (X.LT.150.) GO TO 3
      PRX=0.0
      GO TO 6
    3 IF (XX.GT.0.0001) GO TO 4
      C=DEXP(5.772156649D-01)/2.
      F1=PI*X*(1.-A)
      F2=X*DLOG(C*C*A*X/B)+4.*B
      PRX=(B*PI*PI*DEXP(-X))/(A*(F1*F1+F2*F2))
      GO TO 6
    4 IF (XX.LT.50.) GO TO 5
      PRX=(PI*DEXP(-X))/(2.*XX*(X+4.*A*B))
      GO TO 6
    5 CALL DJY0(XX,J0,Y0)
      CALL DJY1(XX,J1,Y1)
      F1=(XX*J0-2.*A*J1)
      F2=(XX*Y0-2.*A*Y1)
      PRX=DEXP(-X)/(X*(F1*F1+F2*F2))
    6 RETURN
      END

      SUBROUTINE DJY0(X,J0,Y0)
C***************************************************************************
C
C     PURPOSE
C        COMPUTES BESSEL FUNCTIONS OF THE FIRST AND SECOND KIND,
C        ZERO ORDER, FOR POSITIVE ARGUMENTS.
C     DESCRIPTION OF PARAMETERS - ALL DOUBLE PRECISION
C        X- ARGUMENT, MUST BE >0
C        J0 - RETURNED FUNCTION VALUE, J0(X)
C        Y0 - RETURNED FUNCTION VALUE, Y0(X)
C
C***************************************************************************
      DOUBLE PRECISION Z,J0,Y0,F,P,Q,U,W,X,DLOG,DCOS,DSIN,DSQRT
      IF (X-3.0) 1,2,3
    1 IF (X) 4,4,2
    2 Z=(X/3.0)**2
      J0=1.0-Z*(2.2499997-Z*(1.2656208-Z*(0.3163866-Z*(0.0444479-Z*(0.00
     139444-0.00021*Z)))))
      W=(0.500)*X
      Y0=0.63661977*DLOG(W)*J0+0.36746691+Z*(0.60559366-Z*(0.74350384-Z*
     1(0.25300117-Z*(0.04261214-Z*(0.00427916-0.00024846*Z)))))
      RETURN
    3 Z=3.0/X
      F=0.79788456-Z*(0.77D-6+Z*(0.0055274+Z*(0.00009512-Z*(0.00137237-Z
     1*(0.00072805-0.00014476*Z)))))
      P=0.78539816+Z*(0.04166397+Z*(0.00003954-Z*(0.00262573-Z*(0.000541
     125+Z*(0.00029333-0.00013558*Z)))))
```

TABLE 9.2.—*Listing of program to compute change in water level due to sudden injection of a slug of water into a well*—Continued

```
      U=(1.0D0)/X                                                       DJ0  28
      Q=DSQRT(U)                                                        DJ0  29
      J0=Q*F*DCOS(X-P)                                                  DJ0  30
      Y0=Q*F*DSIN(X-P)                                                  DJ0  31
    4 RETURN                                                            DJ0  32
      END                                                               DJ0  33-
      SUBROUTINE DJY1(X,J1,Y1)                                          DJ1   1
C***********************************************************************DJ1   2
C                                                                       DJ1   3
C     PURPOSE                                                           DJ1   4
C        COMPUTES BESSEL FUNCTIONS OF THE FIRST AND SECOND KIND,        DJ1   5
C        FIRST ORDER, FOR POSITIVE ARGUMENTS.                           DJ1   6
C     DESCRIPTION OF PARAMETERS - ALL DOUBLE PRECISION                  DJ1   7
C        X= ARGUMENT, MUST BE >0                                        DJ1   8
C        J1 - RETURNED FUNCTION VALUE, J1(X)                            DJ1   9
C        Y1 - RETURNED FUNCTION VALUE, Y1(X)                            DJ1  10
C                                                                       DJ1  11
C***********************************************************************DJ1  12
      DOUBLE PRECISION X,J1,Y1,Z,W,DLOG,F,P,U,Q,DSQRT,DSIN,DCOS         DJ1  13
      IF (X-3.0) 1,2,3                                                  DJ1  14
    1 IF (X) 4,4,2                                                      DJ1  15
    2 Z=(X/3.0)**2                                                      DJ1  16
      J1=X*(0.5-Z*(0.56249985-Z*(0.21093573-Z*(0.03954289-Z*(0.00443319-DJ1  17
     1Z*(0.00031761-0.00001109*Z))))))                                  DJ1  18
      W=(0.5D0)*X                                                       DJ1  19
      Y1=0.63661977*DLOG(W)*J1+(-0.6366198+Z*(0.2212091+Z*(2.1682709-Z*(DJ1  20
     11.3164827-Z*(0.3123951-Z*(0.0400976-0.0027873*Z))))))/X           DJ1  21
      RETURN                                                            DJ1  22
    3 Z=3.0/X                                                           DJ1  23
      F=0.79788456+Z*(0.156D-5+Z*(0.01659667+Z*(0.00017105-Z*(0.00249511DJ1  24
     1-Z*(0.00113653-0.00020033*Z))))))                                 DJ1  25
      P=0.78539816-Z*(0.12499612+Z*(0.0000565-Z*(0.00637879-Z*(0.0007434DJ1  26
     18+Z*(0.00079824-0.00029166*Z)))))                                 DJ1  27
      U=(1.0D0)/X                                                       DJ1  28
      Q=DSQRT(U)                                                        DJ1  29
      J1=Q*F*DSIN(X-P)                                                  DJ1  30
      Y1=-Q*F*DCOS(X-P)                                                 DJ1  31
    4 RETURN                                                            DJ1  32
      END                                                               DJ1  33-
      SUBROUTINE DSIMPS(A,B,EPS,HBAR,AREA,DEL,F)                        DSI   1
C***********************************************************************DSI   2
C                                                                       DSI   3
C     PURPOSE                                                           DSI   4
C        TO DETERMINE THE INTEGRAL OF A FUNCTION, F, FROM A TO B,       DSI   5
C        USING SIMPSON'S RULE.                                          DSI   6
C     DESCRIPTION OF PARAMETERS                                         DSI   7
C        ALL DOUBLE PRECISION                                           DSI   8
C        A - LOWER LIMIT OF INTEGRAL                                    DSI   9
C        B - UPPER LIMIT OF INTEGRAL                                    DSI  10
C        EPS - DESIRED ACCURACY                                         DSI  11
C        HBAR - MINIMUM DIVISION OF THE INTERVAL                        DSI  12
C        AREA - COMPUTED VALUE OF INTEGRAL BETWEEN Q AND R              DSI  13
C        DEL - COMPUTED ESTIMATE OF ERROR                               DSI  14
C        F= THE INTEGRAND (FUNCTION REFERENCE)                          DSI  15
C     METHOD                                                            DSI  16
C        USES SIMPSON'S RULE TO COMPUTE A SUM APPROXIMATING THE INTEGRALDSI  17
C        USES INITIAL H=(B-A)/2, COMPUTES A SEQUENCE OF SUMS BY HALVING DSI  18
C        H EACH TIME, COMPUTES ESTIMATE OF ERROR (DEL) AS (PREVIOUS     DSI  19
C        SUM - CURRENT SUM)/15, COMPUTATION STOPS WHEN 1) H<HBAR,       DSI  20
C        2) ABS(DEL)<ABS(EPS*CURRENT SUM), IF HBAR IS LE 0,             DSI  21
C        THEN HBAR=.007*(B-A),                                          DSI  22
C                                                                       DSI  23
```

TABLE 9.2.—*Listing of program to compute change in water level due to sudden injection of a slug of water into a well*—Continued

```
C************************************************************************
      DOUBLE PRECISION H,HBAR,AREA,DEL,S1,S2,S3,SC,SP,X,A,B,EPS,F,DABS
C  AREA OF F FROM A TO B,EPS IS DESIRED ACCURACY, HBAR THE MINIMUM
C  ALLOWABLE INTERVAL, DEL THE ESTIMATE OF THE ERROR
      H=B-A
      IF (H) 1,1,2
    1 AREA=0,0
      DEL=0,0
      GO TO 10
    2 SP=1,0D35
      S3=0,0
      S1=F(A)+F(B)
      IF (HBAR) 3,3,4
    3 HBAR=0,007*H
    4 S2=0,0
      X=A+0,5*H
    5 S2=S2+4,0*F(X)
      X=X+H
      IF (X-B) 5,5,6
    6 SC=(S1+S2+S3)*H*0,1666666667
      DEL=0,06666666667*(SP-SC)
      IF (DABS(DEL)-DABS(EPS*SC)) 7,8,8
    7 AREA=SC-DEL
      GO TO 10
    8 S3=S3+0,5*S2
      H=0,5*H
      IF (H-HBAR) 7,9,9
    9 SP=SC
      GO TO 4
   10 RETURN
      END
```

TABLE 11.1.—*Listing of program to compute the convolution integral for a leaky aquifer*

```
C************************************************************************
C
C     PURPOSE
C        COMPUTES CHANGES IN WATER LEVEL, H(R,T), IN RESPONSE TO
C        VARYING DISCHARGE USING THE CONVOLUTION INTEGRAL FOR
C        LEAKY AQUIFERS - EQ, 3 OF MOENCH,ALLEN,1971, GROUND-WATER
C        FLUCTUATIONS IN RESPONSE TO ARBITRARY PUMPAGE: GROUND WATER,
C        V,9, NO,2,P,4-8,
C     INPUT DATA - ONE OR MORE GROUPS, EACH GROUP CODED AS FOLLOWS
C        1 CARD - FORMAT(2E10,5,4X,I1,5X,E10,5)
C              TBEGIN - SMALLEST VALUE OF TIME FOR OUTPUT,
C              TEND - LARGEST VALUE OF TIME FOR OUTPUT,
C              IQ - INDICATES FORM OF DISCHARGE FUNCTION, Q(T),
C                   IQ=1,2,3 REFER TO DISCHARGE FUNCTIONS IN
C                   HANTUSH,M,S,, 1964, HYDRAULICS OF WELLS IN CHOW,
C                   VEN TE, ED,, ADVANCES IN HYDROSCIENCE, VOL, 1:
C                   ACADEMIC PRESS INC,, NEW YORK, P, 281-442,
C                   IQ=1, Q(T) IS AN EXPONENTIAL FUNCTION, CASE A,
C                         P, 343 OF HANTUSH,
C                   IQ=2, Q(T) IS A HYPERBOLIC FUNCTION, CASE B,
C                         P, 344 OF HANTUSH,
C                   IQ=3, Q(T) IS AN INVERSE SQUARE ROOT FUNCTION,
C                         CASE C, P, 344 OF HANTUSH,
```

TABLE 11.1.—*Listing of program to compute the convolution integral for a leaky aquifer*—Continued

```
C                 IQ=4, Q(T) IS A FIFTH-DEGREE POLYNOMIAL,
C                 IQ=5, Q(T) IS A PIECEWISE LINEAR FUNCTION OF
C                      TIME (EIGHT SEGMENTS),
C            QR = REFERENCE DISCHARGE, ZERO OR BLANK FOR PROJECTION,
C        1 OR 4 CARDS, DEPENDING ON IQ,
C            IF IQ=1,2,3 = 1 CARD = FORMAT(3E10,3)
C                 QST = EVENTUAL CONSTANT DISCHARGE,
C                 DELTA = RATE PARAMETER,
C                 TSTAR = TIME PARAMETER,
C            IF IQ=4 = 1 CARD = FORMAT(6E10,3)
C                 AQ(6) = 6 VALUES = THE POLYNOMIAL COEFFICIENTS
C                      WITH A0 FIRST AND A5 LAST,
C            IF IQ=5 = 4 CARDS = FORMAT(6E10,3)
C                 TI(I),AI(I),BI(I),TI(I+1),AI(I+1),BI(I+1),I=1,3,5,7
C                 PARAMETERS OF THE PIECEWISE LINEAR FUNCTION
C                 (8 SEGMENTS), CODED 2 SEGMENTS PER CARD, FIRST
C                 AND SECOND SEGMENTS ON FIRST CARD, THEN SEQUENTIALLY
C                 ON SUCCEEDING CARDS, EACH SEGMENT HAS THREE
C                 PARAMETERS WHICH ARE IN CODING ORDER
C                      TI = ENDING TIME OF THE SEGMENT,
C                      AI = DISCHARGE AT BEGINNING OF SEGMENT,
C                      BI = RATE OF CHANGE IN DISCHARGE DURING SEG,
C                 THE DISCHARGE FUNCTION IN EACH SEGMENT HAS THE
C                 FORM Q(T) = AI(I)+BI(I)*(T-TI(I-1)), IF LESS THAN 8
C                 SEGMENTS ARE NEEDED, BLANKS CAN BE CODED FOR
C                 SUCCEEDING SEGMENTS,
C        2 OR MORE CARDS = FORMAT(4E10,3)
C            R = RADIAL DISTANCE FROM PUMPED WELL, BLANK OR ZERO
C                 SIGNALS PROGRAM AS END TO GROUP OF DATA,
C            S = STORAGE COEFFICIENT
C            T = TRANSMISSIVITY
C            PM = (P'/M') = HYD, COND, OF CONFINING BED DIVIDED
C                 BY THICKNESS OF CONFINING BED,
C     SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C        CONVOL,Q = MUST BE INCLUDED IN DECK,
C
C***********************************************************************
      DIMENSION       D(12),IEX(12),X(6),H(12,6),QS(12,6),CP(12),CT(12)
      DIMENSION H1(12),H2(12),Q1(12),Q2(12)
      DIMENSION H3(12),H4(12),Q3(12),Q4(12)
      COMMON AQ(6),TI(9),AI(9),BI(9),QST,DELTA,TSTAR
      DATA CP/12*'  T*'/,CT/12*'1/U*'/,D/12*'10**'/
      DATA H1/12*'  S('/,H2/12*'R,T)'/,Q1/12*'    '/,Q2/12*'Q(T)'/
      DATA H3/12*'   S'/,H4/12*'D(T)'/,Q3/12*' Q(T'/,Q4/12*')/QR'/
      DATA X/1.,1.5,2.,3.,5.,7./
      TI(1)=0.
      N=500
    1 READ (5,18,END=17) TBEGIN,TEND,IQ,QR
      IF (IQ.LT.4) READ (5,19) QST,DELTA,TSTAR
      IF (IQ.EQ.4) READ (5,19) AQ
      IF (IQ.EQ.5) READ (5,19) (TI(I),AI(I),BI(I),I=2,9)
      WRITE (6,24)
    2 READ (5,19) R,S,T,PM
      IF (R.EQ.0.) GO TO 1
      A=R*R*S/(4.*T)
      B=PM/S
      Y=ALOG10(TBEGIN)
```

TABLE 11.1.—*Listing of program to compute the convolution integral for a leaky aquifer*—Continued

```
      IF (Y) 3,5,4
    3 Y=Y-.001
      GO TO 5
    4 Y=Y+.001
    5 IBEGIN=Y
      Y=ALOG10(TEND)
      IF (Y) 6,8,7
    6 Y=Y-.001
      GO TO 8
    7 Y=Y+.001
    8 IEND=Y
      M=IEND-IBEGIN+1
      IF (M.GT.12) M=12
      DO 10 I=1,M
      IEX(I)=IBEGIN+I-1
      Y=10.**(IBEGIN+I-1)
      DO 10 J=1,6
      TIME=X(J)*Y
      IF (QR.GT.0.) TIME=A*TIME
      CALL CONVOL(TIME,A,B,N,IQ,SUM)
      IF (QR.GT.0.) GO TO 9
      H(I,J)=SUM/(12.5664*T)
      QS(I,J)=Q(TIME,IQ)
      GO TO 10
    9 H(I,J)=SUM/QR
      QS(I,J)=Q(TIME,IQ)/QR
   10 CONTINUE
      K=M
      IF (M.GT.6) K=6
      IF (QR.GT.0.) GO TO 11
      WRITE (6,20) A,B,(CP(I),D(I),IEX(I),I=1,K)
      WRITE (6,21) (H1(I),H2(I),Q1(I),Q2(I),I=1,K)
      GO TO 12
   11 WRITE (6,25) A,B,QR,(CT(I),D(I),IEX(I),I=1,K)
      WRITE (6,21) (H3(I),H4(I),Q3(I),Q4(I),I=1,K)
   12 DO 13 J=1,6
      WRITE (6,22) X(J),(H(I,J),QS(I,J),I=1,K)
   13 CONTINUE
      IF (M.LE.6) GO TO 2
      K1=K+1
      IF (QR.GT.0.) GO TO 14
      WRITE (6,23) (CP(I),D(I),IEX(I),I=K1,M)
      WRITE (6,21) (H1(I),H2(I),Q1(I),Q2(I),I=K1,M)
      GO TO 15
   14 WRITE (6,26) (CT(I),D(I),IEX(I),I=K1,M)
      WRITE (6,21) (H3(I),H4(I),Q3(I),Q4(I),I=K1,M)
   15 DO 16 J=1,6
      WRITE (6,22) X(J),(H(I,J),QS(I,J),I=K1,M)
   16 CONTINUE
      GO TO 2
   17 STOP
C
   18 FORMAT (2E10.5,4X,I1,5X,E10.5)
   19 FORMAT (6E10.3)
   20 FORMAT ('0','R**2*S/(4*TRANS)=',1PE10.3,', K''/(S*B'')=',E10.3/'0'
     1,2X,'T',5X,6(2A4,I2,9X))
   21 FORMAT (' ',4X,6(2A4,2X,2A4,1X))
```

TABLE 11.1.—*Listing of program to compute the convolution integral for a leaky aquifer*—Continued

```
   22 FORMAT (' ',F4.1,6(0PF8.3,1PE11.3))
   23 FORMAT ('0',2X,'T',5X,6(2A4,I2,9X))
   24 FORMAT (1H1)
   25 FORMAT ('0','R**2*S/(4*TRANS)=',1PE10.3,', K''/(S*B'')=',E10.3,',
     1QR=',E10.3/'0',1X,'1/U',4X,6(2A4,I2,9X))
   26 FORMAT ('0',1X,'1/U',4X,6(2A4,I2,9X))
      END
      SUBROUTINE CONVOL(TIME,A,B,N,IQ,SUM)
C***********************************************************************
C
C     PURPOSE
C        COMPUTES VALUES OF THE CONVOLUTION INTEGRAL FOR LEAKY
C        AQUIFERS. THE INTEGRAL IS, FROM 0 TO T, OF
C        Q(T-T')/T'*EXP(-A/T'-B*T')*DT'.
C     DESCRIPTION OF PARAMETERS
C        A,B,SUM ARE REAL; N,IQ ARE INTEGER.
C        A - R**2*S/(4*T) - RADIAL DISTANCE SQUARED * STORAGE
C               COEFFICIENT / 4 * TRANSMISSIVITY.
C        B - P'/(S*M') - HYD. COND. OF CONFINING BED DIVIDED BY
C               AQUIFER STORAGE COEFFICIENT * THICKNESS OF CONF. BED.
C        N - NUMBER OF INCREMENTS FOR EACH INTERVAL OF THE SUM.
C        IQ - INDICATES FORM OF DISCHARGE FUNCTION.
C     SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C        Q
C     METHOD
C        APPROXIMATES INTEGRAL BY SUMMING THE TRAPEZOIDAL RULE APPLIED
C        TO A SEQUENCE OF SEGMENTS. LOWER LIMIT OF FIRST SEGMENT IS
C        PICKED AT POINT WHERE EXPONENT > -100 .
C        IF SUCH A POINT DOES NOT EXIST (A*B > 2500) A FUNCTION VALUE
C        OF 0 IS RETURNED. UPPER LIMIT = 10 * LOWER LIMIT FOR EACH
C        SEGMENT. USES INCREMENT OF DELTA T' = (U-L)/N WHERE N IS THE
C        NUMBER OF INCREMENTS IN THE CALL. CEASES SUMMATION WHEN
C        EXPONENT < -101 .
C
C***********************************************************************
      REAL*8 DSUM
      REAL*4 NEWT,NEWTP,NEWX,NEWF
      DSUM=0.D+0
      IS=0
C     INITIAL T' COMPUTED FROM A,B
      AB=A*B
      IF (AB.GE.2500.) GO TO 7
      IF (B.GT.0.) GO TO 2
    1 OLDT=.01*A
      GO TO 3
    2 OLDT=(1.-SQRT(1.-AB/2500.))*50./B
      IF (OLDT.EQ.0.) GO TO 1
C     INITIAL T-T'
    3 OLDTP=TIME-OLDT
      OLDX=-A/OLDT-B*OLDT
      OLDF=Q(OLDTP,IQ)*EXP(OLDX)/OLDT
C     END OF SUMMATION SEGMENT IS 10 TIMES THE BEGINNING
    4 ENDT=10.*OLDT
      IF (ENDT.LT.TIME) GO TO 5
      IF (OLDT.GE.TIME) GO TO 7
      IS=1
      ENDT=TIME
```

TABLE 11.1.—*Listing of program to compute the convolution integral for a leaky aquifer*—Continued

```
C     DELTA T' IS COMPUTED FROM LENGTH AND NUMBER OF INCREMENTS
    5 DELT=(ENDT-OLDT)/N
      DO 6 I=1,N
C     T' IS INCREMENTED BY DELTA T'
      NEWT=OLDT+DELT
      NEWX=-A/NEWT-B*NEWT
C     TERMINATES SUMMATION WHEN EXP(-A/T'-B*T') < 1.37E-44
      IF (NEWX.LT.-101.) GO TO 7
      NEWTP=TIME-NEWT
      NEWF=Q(NEWTP,IQ)*EXP(NEWX)/NEWT
      DSUM=DSUM+(NEWF+OLDF)*DELT
      OLDT=NEWT
      OLDF=NEWF
    6 CONTINUE
      IF (IS.GT.0) GO TO 7
C     IF T' < T, BEGINS A NEW SEGMENT
      GO TO 4
    7 SUM=DSUM/2.D+0
      RETURN
      END


      FUNCTION Q(TIME,IQ)
C**********************************************************************
C
C     PURPOSE
C        COMPUTES THE DISCHARGE FUNCTION, Q(T)
C     DESCRIPTION OF PARAMETERS
C        TIME - REAL - ELAPSED TIME SINCE BEGINNING OF DISCHARGE.
C        IQ - INTEGER - INDICATES FORM OF DISCHARGE FUNCTION.
C             IQ=1,2,3, CASES A,B,C, RESPECTIVELY, OF HANTUSH,M.S.,
C                  1964, HYDRAULICS OF WELLS IN CHOW, VEN TE, ED.,
C                  ADVANCES IN HYDROSCIENCE, VOL. 1: ACADEMIC PRESS,
C                  NEW YORK, P. 343,344.
C             IQ=4, DISCHARGE IS A FIFTH DEGREE POLYNOMIAL OF TIME.
C             IQ=5, DISCHARGE IS A PIECEWISE LINEAR FUNCTION OF UP TO
C                  8 SEGMENTS.
C     METHOD
C        FORTRAN EVALUATION OF FUNCTIONS.
C
C**********************************************************************
      COMMON AQ(6),TI(9),AI(9),BI(9),QST,DELTA,TSTAR
      GO TO (1,2,3,4,5), IQ
    1 Q=QST*(1.+DELTA*EXP(-TIME/TSTAR))
      RETURN
    2 Q=QST*(1.+DELTA/(1.+TIME/TSTAR))
      RETURN
    3 Q=QST*(1.+DELTA/SQRT(1.+TIME/TSTAR))
      RETURN
    4 Q=AQ(1)+TIME*(AQ(2)+TIME*(AQ(3)+TIME*(AQ(4)+TIME*(AQ(5)+TIME*AQ(6)
     1))))
      RETURN
    5 DO 6 I=2,9
      IF (TIME.LE.TI(I)) GO TO 7
    6 CONTINUE
      I=9
    7 Q=AI(I)+BI(I)*(TIME-TI(I-1))
      RETURN
      END
```