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 13–20, in format E10.3; and a discharge index (a number from 1 through 5) coded in column 25, 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,

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

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

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

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),

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

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

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

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

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; DE-LTA, 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, \ldots, 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


Dudley, W. W., Jr., 1970, Nonsteady inflow to a chamber
### Figure 11.3: Example of output from program to compute the convolution integral for a leaky aquifer.

<table>
<thead>
<tr>
<th>$1/U$</th>
<th>$1/U^{*10**0}$</th>
<th>$1/U^{*10**1}$</th>
<th>$1/U^{*10**2}$</th>
<th>$1/U^{*10**3}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_0(T)$</td>
<td>$Q(T)/QR$</td>
<td>$S_0(T)$</td>
<td>$Q(T)/QR$</td>
<td>$S_0(T)$</td>
</tr>
<tr>
<td>$1.0$</td>
<td>$0.185$</td>
<td>$1.000E+00$</td>
<td>$0.819$</td>
<td>$1.000E+00$</td>
</tr>
<tr>
<td>$1.5$</td>
<td>$0.317$</td>
<td>$1.000E+00$</td>
<td>$0.837$</td>
<td>$1.000E+00$</td>
</tr>
<tr>
<td>$2.0$</td>
<td>$0.421$</td>
<td>$1.000E+00$</td>
<td>$0.841$</td>
<td>$1.000E+00$</td>
</tr>
<tr>
<td>$3.0$</td>
<td>$0.566$</td>
<td>$1.000E+00$</td>
<td>$0.842$</td>
<td>$1.000E+00$</td>
</tr>
<tr>
<td>$5.0$</td>
<td>$0.715$</td>
<td>$1.000E+00$</td>
<td>$0.842$</td>
<td>$1.000E+00$</td>
</tr>
<tr>
<td>$7.0$</td>
<td>$0.780$</td>
<td>$1.000E+00$</td>
<td>$0.842$</td>
<td>$1.000E+00$</td>
</tr>
</tbody>
</table>

---


---1961c, Tables of the function $M(u;8)$: New Mexico Inst. Mining and Technology Prof. Paper 102, 34 p.

---1961d, Tables of the function $H(u;8) = \int_0^\infty e^{-y/y} \text{erfc} \left( \frac{\beta \sqrt{u}}{\sqrt{V}y-y} \right) dy$: New Mexico Inst. Mining and Technology Prof. Paper 103, 12 p.

---1961e, Tables of the function $W(u;8) = \int_0^\infty e^{-y/y} \frac{\beta}{\sqrt{V}} dy$: New Mexico Inst. Mining and Technology Prof. Paper 104, 13 p.


SUPPLEMENTAL DATA
Table 2.1.—Listing of program for partial penetration in a nonleaky artesian aquifer

<table>
<thead>
<tr>
<th>Line No.</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Input data</td>
</tr>
<tr>
<td>3</td>
<td>1 card: format (3F5,1,15,ZE10,4)</td>
</tr>
<tr>
<td>4</td>
<td>B = Aquifer thickness</td>
</tr>
<tr>
<td>5</td>
<td>L = Depth, below top of aquifer, to bottom of pumping well screen</td>
</tr>
<tr>
<td>6</td>
<td>D = Depth, below top of aquifer, to top of pumping well screen</td>
</tr>
<tr>
<td>7</td>
<td>Num = Number of observation wells or piezometers times number of values of Kz/Kr,</td>
</tr>
<tr>
<td>8</td>
<td>Small = Smallest value of 1/U for which computation is desired</td>
</tr>
<tr>
<td>9</td>
<td>Large = Largest value of 1/U for which computation is desired</td>
</tr>
<tr>
<td>10</td>
<td>Num cards (one for each obs, well or piezometer and for each value of R*SQRT(Kz/Kr), format (3F5,1)</td>
</tr>
<tr>
<td>11</td>
<td>R = Radial distance from pumped well times sqrt(Kz/Kr),</td>
</tr>
<tr>
<td>12</td>
<td>Lprime = Depth, below top of aquifer, to bottom of obs well screen (zero for piezometer),</td>
</tr>
<tr>
<td>13</td>
<td>Dprime = Depth, below top of aquifer, to top of obs well screen (total depth for piezometer),</td>
</tr>
<tr>
<td>14</td>
<td>Subroutines and function subprograms required</td>
</tr>
<tr>
<td>15</td>
<td>DGL12, SERIES, BESK, FCT, L, F, EXPK</td>
</tr>
<tr>
<td>16</td>
<td>Real* U</td>
</tr>
<tr>
<td>17</td>
<td>Real*4 L, LB, LPB, Lprime, LARGE</td>
</tr>
<tr>
<td>18</td>
<td>Dimension array(13,12), IARG(12), ARG(13), A(12), C(12)</td>
</tr>
<tr>
<td>19</td>
<td>Data ARG/1,2,15,2,2,3,2,5,3,3,5,4,5,6,7,8,9/</td>
</tr>
<tr>
<td>20</td>
<td>Data A/12*! N*! C/12<em>10</em>!</td>
</tr>
<tr>
<td>21</td>
<td>IRD=5</td>
</tr>
<tr>
<td>22</td>
<td>IPT#6</td>
</tr>
<tr>
<td>23</td>
<td>READ (IRD,6) B,L,D,Num,Small,Large</td>
</tr>
<tr>
<td>24</td>
<td>LB=L/B</td>
</tr>
<tr>
<td>25</td>
<td>DB=D/B</td>
</tr>
<tr>
<td>26</td>
<td>IBEGIN=ALOG10(SMALL)</td>
</tr>
<tr>
<td>27</td>
<td>IEND=ALOG10(LARGE)+1</td>
</tr>
<tr>
<td>28</td>
<td>JLIMIT=IEND-IBEGIN</td>
</tr>
<tr>
<td>29</td>
<td>IF (JLIMIT.GT,12) JLIMIT=12</td>
</tr>
<tr>
<td>30</td>
<td>DO S K=1,Num</td>
</tr>
<tr>
<td>31</td>
<td>READ (IRD,6) R,Lprime,Dprime</td>
</tr>
<tr>
<td>32</td>
<td>RB=R/B</td>
</tr>
<tr>
<td>33</td>
<td>LPB=Lprime/B</td>
</tr>
<tr>
<td>34</td>
<td>DPB=Dprime/B</td>
</tr>
<tr>
<td>35</td>
<td>DO I IM=1,13</td>
</tr>
<tr>
<td>36</td>
<td>ARG1=ARG(I)</td>
</tr>
<tr>
<td>37</td>
<td>DO J JI=1,JLIMIT</td>
</tr>
<tr>
<td>38</td>
<td>IARG(J)IBEGIN+J=1</td>
</tr>
<tr>
<td>39</td>
<td>Y=ARG1*10,**(IBEGIN+J=1)</td>
</tr>
<tr>
<td>40</td>
<td>U1,Y</td>
</tr>
<tr>
<td>41</td>
<td>CALL EXPK(X, MU, DUMMY)</td>
</tr>
<tr>
<td>42</td>
<td>1 array(1,J)MU+F(U, RB, LB, DB, LPB, DPB)</td>
</tr>
<tr>
<td>43</td>
<td>IF (LPB=0.,) 2,2,3</td>
</tr>
</tbody>
</table>
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,DPB
4 WRITE (IPT,9) (A(I),C(I),IARG(I),I,JLIMIT)
DO 5 I=1,13
WRITE (IPT,10) ARG(I),(ARRAY(I,J),J=1,JLIMIT)
5 CONTINUE
STOP

6 FORMAT (3F5,1,15,2*10,4)
7 FORMAT (111,'W(U)*F(U,R/B,L/B,D/B,Z/B),Z/B=1,F5.2,1,3*SGRT(KZ/KR)*PPN
1R/B=1,F5.2,1,3*SGRT(KZ/KR)*R/B=1,F5.2,1,3*SGRT(KZ/KR)*R/B=1,F5.2,1,3*SGRT(KZ/KR)*PPN
2, U=1/N 1) PPN 72
8 FORMAT (111,'W(U)*F(U,R/B,L/B,D/B,L'/B)*B, L'/B=1,F5.2,1,3*SGRT(KZ/KR)*R/B=1,F5.2,1,3*SGRT(KZ/KR)*R/B=1,F5.2,1,3*SGRT(KZ/KR)*PPN
2, U=1/N 1) PPN 73
9 FORMAT (111,'W(U)*F(U,R/B,L/B,D/B,L'/B)*B, L'/B=1,F5.2,1,3*SGRT(KZ/KR)*R/B=1,F5.2,1,3*SGRT(KZ/KR)*R/B=1,F5.2,1,3*SGRT(KZ/KR)*PPN
2, U=1/N 1) PPN 74
10 FORMAT (111,'W(U)*F(U,R/B,L/B,D/B,L'/B)*B, L'/B=1,F5.2,1,3*SGRT(KZ/KR)*R/B=1,F5.2,1,3*SGRT(KZ/KR)*R/B=1,F5.2,1,3*SGRT(KZ/KR)*PPN
2, U=1/N 1) PPN 75
11 FORMAT (111,'W(U)*F(U,R/B,L/B,D/B,L'/B)*B, L'/B=1,F5.2,1,3*SGRT(KZ/KR)*R/B=1,F5.2,1,3*SGRT(KZ/KR)*R/B=1,F5.2,1,3*SGRT(KZ/KR)*PPN
2, U=1/N 1) PPN 76
END PPN 79

REAL FUNCTION F*4(U,RB,LB,DPB,DPB)

FUNCTION F

PURPOSE
TO COMPUTE DEPARTURES FROM THEIS CURVE CAUSED BY PARTIAL
PENETRATION OF PUMPED WELL.

USAGE
F(U,RB,LB,DPB,DPB)

DESCRIPTION OF PARAMETERS
ALL REAL, U DOUBLE PRECISION
U = H 2*4*TIME (RADIAL DISTANCE SQUARED * STORAGE
1 = TRANSMISSIVITY * TIME
2 = RADIAL DISTANCE / AQUIFER THICKNESS
LB = L/B (FRACTION OF AQUIFER PENETRATED BY PUMPED WELL)
DB = D/B (FRACTION OF AQUIFER ABOVE PUMPED WELL SCREEN)
LPS = L'/B (FRACTION OF AQUIFER PENETRATED BY OBS, WELL, ZERO
FOR PIEZOMETER)
DPS = D'/B (FRACTION OF AQUIFER ABOVE OBS, WELL SCREEN, TOTAL
DEPTH FOR PIEZOMETER)
SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
DQL12, SERIES, RISK, FCT, L

METHOD
SUMS THE SERIES THROUGH N*PI*R/B EQ 20

REAL*8 U,V
REAL*4 L,N,LB,DPB
SUM=0,
N=0,
PIRB=3.141593*RB
PILB=3.141593*LB
PDB=3.141593*DB
IF (LPB=0) 1,1,4
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

C
Table 2.1.—Listing of program for partial penetration in a nonleaky artesian aquifer—Continued

```
SUM=SUM+(SIN(N*PI*LB)-SIN(N*PI*DB))*COS(N*PI*PB)*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*PIPB/2,
IF (V,GT,10,) GO TO 6
TRUNCATES SERIES WHEN V>10
X=L(U,V)/N
SUM=SUM+(SIN(N*PI*LB)-SIN(N*PI*DB))*COS(N*PI*PB)*SIN(N*PI*PB))*X/N
GO TO 5
6 F=20264224*SUM/((LB*DB)*(LPB*DB))
7 RETURN
END

REAL FUNCTION L*Y(U,V)

**************Ct********2***************n*nnnn*nn*n*n*nnnnnn*n*nnnnn*nnn********

FUNCTION L

PURPOSE
TO COMPUTE THE INTEGRAL( EXP(-Y*V**2/Y)/Y) SUMMED OVER Y FROM
U TO INFINITY(WELL FUNCTION FOR LEAKY AQUIFERS),

DESCRIPTION OF PARAMETERS

DOUBLE PRECISION
U = R**2*3/4*rtrTIME (R4DIAL U1S SQUARED * STORAGE
COEFFICIENT / 4*TRANSMISSIVITY * TIME
V = R/t*SQRT(K1/(f*8')) - ONE*HALF RADIAL DISTANCE*SQUARE ROOT
(HYD, CONO, OF CONFINING BED/TRANSMISSIVITY*THICKNESS
OF CONFINING BED)

SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
DQL12, SERIES, BESK, FCT

METHOD
IN THE FOLLOWING F=EXP(-Y*V**2/Y)/Y
(1) U>=1, USES A GAUSSIAN-LAGUERRE QUADRATURE FORMULA TO
   EVALUATE INTEGRAL(F) FROM U TO INF,
(2) V**2<1, USES THE G-L QUADRATURE TO EVALUATE INTEGRAL(F)
   FROM ONE TO INF AND A SERIES EXPANSION TO EVALUATE INTEGRAL(F)
   FROM U TO ONE,
(3) U<1, USES THE REPRESENTATION INTEGRAL(F) FROM U
   TO INF, = 2*K0(2*V)=INTEGRAL(F) FROM V**2/U TO INF.
   EVALUATES THE ZERO ORDER MODIFIED BESSEL FUNCTION OF SECOND
   KIND WITH IBM SUBROUTINE.

**************n*****nn*nn*nn*nnnn*nnnn*n~nnn~nnnnnnn*nnnnnnn*nn**nnnnn

EXTERNAL FCT

REAL*8 U,V,Z,F,VV,Series
COMM0N /C1/ VV,Z
V=V
IF (U=1,) 1,2,2
C CHECKS IF UK1
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)
3 Z=1,
```

59 TYPE CURVES FOR FLOW TO WELLS IN CONFINED AQUIFERS

C
Table 2.1.—Listing of program for partial penetration in a nonleaky artesian aquifer—Continued

CALL DQL12(FCt,F)  
L=SERIES(U,V)  
C INTEGRAL 1 TO INF. BY G=L QUAD., INTEGRAL U TO 1 BY SERIES EXP.,  
GO TO 5  
4 TWOV=V*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)**  
**************************************************************************************************  
**FUNCTION SERIES**  
*******SERIES 1**********  
**PURPOSE**  
TO EVALUATE S(1)-S(U), WHERE S IS A SERIES EXPANSION OF  
INTEGRAL(3.13%/2/Y)DY/Y GIVEN BY S = SUM, M=0 TO INFINITY,  
(F(M)*SUM, N=0 TO INF,(V**((2*M)/(M+N))) WHERE F(M)=  
LUG(U) IF M=0 AND = ((M)**M/M)SUM, N=0 TO INF,(V**((2*M)/(M*N))) IF M>0,  
**DESCRIPTION OF PARAMETERS**  
U = R**2*3/4el*lIb'E (RADIAL DISTANCE Squared * STORAGE  
COEFFICIENT /4*TRANSMISSIVITY TIME  
V = R/2*SQR(K(T*81))--ONE-HALF RADIAL DISTANCE*SQUARE ROOT  
(HYD. COND. OF CONFINING BED/TRANSMISSIVITY*THICKNESS  
OF CONFINING BED)  
**SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED**  
NONE  
**METHOD**  
SUMMATION IS TERMINATED FOR THE INNER SERIES WHEN A TERM  
BECOMES LESS THAN 5,E-7/N AND FOR OUTER SERIES WHEN A TERM  
BECOMES LESS THAN 5,E-7  
**REAL*8 DLUG,DABS,S(2),VU,M,U**  
**REAL*8 TEST,U,UM,EM,EN,SUM1,SUM1,SIGN,V,V8G,VSGU,RMUL,TERM,TERM1**  
**TEST=5,D=07**  
**V8G=VU**  
**U=U**  
**D=1.12**  
**DO 6 I=1,2**  
**IF (I,E=0.2) U=1.**  
**UM1=1.**  
**EN=1.**  
**SUM1=0.**  
**SIGN=1.**  
**VUM1=1.**  
**VSGU=VSGU/U**  
**1 EM=EM+1.**  
**IF (EM>1) 2,3,3**  
**CHECKS FOR M=0**  
**2 RMUL=DLUG(U)**  
**TERM1=1.**  
**GO TO 4**  
**3 UM=UM+U**  
**IF (UM,LT,1.0=30) UM=0.**  
**VUM=VUM+V8G**  
**RMUL=(UM*VSGU/EM**  
**TERM1=TERM1/EM**
Table 2.1.—Listing of program for partial penetration in a nonleaky artesian aquifer—Continued

4 SIGN = SIGN
SUM = TERM
TERM = TERM
EN = 0

5 EN = EN + 1,
TERM = TERM * VSG / (EN * (EN * EN))
SUM = SUM + TERM
IF (TEST LE DB8 (RMUL + EN * TERM)) GO TO 5
TRUNCATES INNER SERIES IF OUTER TERM * INNER TERM < 5 * E-7
SUM = SUM + SIGN * RMUL * SUM
IF (EN LT 1) GO TO 1
IF (TEST LE DB8 (RMUL + SUM)) GO TO 1
TRUNCATES OUTER SERIES IF OUTER TERM * INNER SUM < 5 * E-7

6 S(I) = SUM
RETURN

REAL FUNCTION FCTB(X)

FUNCTION FCTB

PURPOSE
TO COMPUTE FCTB(X) = EXP(-Z * V**2 / (X + Z)) / (X + Z)

DESCRIPTION OF PARAMETERS
X = THE DOUBLE PRECISION VALUE OF X FOR WHICH FCTB IS COMPUTED
Y = THE RESULTING DOUBLE PRECISION VALUE OF FCTB

REAL FUNCTION FCTB(X)

FUNCTION FCTB

RETURN
END

SUBROUTINE DQL12(FCT, Y)

PURPOSE
TO COMPUTE INTEGRAL( EXP(-X) * FCT(X), SUMMED OVER X FROM 0 TO INFINITY).

CALL DQL12 (FCT, Y)
PARAMETER FCT REQUIRES AN EXTERNAL STATEMENT

DESCRIPTION OF PARAMETERS
FCT = THE NAME OF AN EXTERNAL DOUBLE PRECISION FUNCTION SUBPROGRAM USED.
Y = THE RESULTING DOUBLE PRECISION INTEGRAL VALUE.
Table 2.1—Listing of program for partial penetration in a nonleaky artesian aquifer—Continued

REMARKS
NONE

SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
THE EXTERNAL DOUBLE PRECISION FUNCTION SUBPROGRAM FCT(X)
MUST BE FURNISHED BY THE USER.

METHOD
EVALUATION IS DONE BY MEANS OF 12-POINT GAUSSIAN-LAGUERRE
QUADRATURE FORMULA, WHICH INTEGRATES EXACTLY,
WHENEVER FCT(X) IS A POLYNOMIAL UP TO DEGREE 23,
FOR REFERENCE, SEE
SHAO/CHEN/Frank, TABLES OF ZEROS AND GAUSSIAN WEIGHTS OF
CERTAIN ASSOCIATED LAGUERRE POLYNOMIALS AND THE RELATED
GENERALIZED HERMITE POLYNOMIALS, IBM TECHNICAL REPORT
TR00.1100 (MARCH 1964), PP.24-25.

***************************************************************

DOUBLE PRECISION X,Y,FCT

x= .3709912104446692 D2
y= .814807746742624 D=15*FCT(X)
x= .2648796725098400 D2
y= .3016601635035021 D=11*FCT(X)
x= .2215104037939701 D2
y= .134239103515004 D=8*FCT(X)
x= .171165518746226 D2
y= .1688493876540910 D=6*FCT(X)
x= .1300805499330635 D2
y= .8362055865661980 D=4*FCT(X)
x= .962131864245b87 D1
y= .2032813592669994 D=3*FCT(X)
x= .6844625453115177 D1
y= .2663973541655316 D=2*FCT(X)
x= .4599227639418348 D1
y= .10238115463410 D=1*FCT(X)
x= .2833751337743507 D1
y= .904922222116809 D=1*FCT(X)
x= .1512610269776419 D1
y= .2440820113198776 D0*FCT(X)
x= .6175746451511307 D0
y= .3777592758731380 D0*FCT(X)
x= .1157221173580207 D0
y= .2647313710544432 D0*FCT(X)
RETURN
END

SUBROUTINE BESK(N,BK,IER)

***************************************************************

SUBROUTINE BESK

COMPUTE THE K BESSEL FUNCTION FOR A GIVEN ARGUMENT AND ORDER

USAGE
CALL BESK(N,BK,IER)
Table 2.1.—Listing of program for partial penetration in a nonleaky artesian aquifer—Continued

DESCRIPTION OF PARAMETERS
X = THE ARGUMENT OF THE K BESSEL FUNCTION DESIRED
N = THE ORDER OF THE K BESSEL FUNCTION DESIRED
BK = THE RESULTANT K BESSEL FUNCTION
IER = RESULTANT ERROR CODE WHERE
IER = 0 NO ERROR
IER = 1 N IS NEGATIVE
IER = 2 X IS ZERO OR NEGATIVE
IER = 3 X GT 170, MACHINE RANGE EXCEEDED
IER = 4 BK GT 10**70

REMARKS
N MUST BE GREATER THAN OR EQUAL TO ZERO

SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
NONE

METHOD
COMPUTES ZERO ORDER AND FIRST ORDER BESSEL FUNCTIONS USING SERIES APPROXIMATIONS AND THEN COMPUTES NTH ORDER FUNCTION USING RECURRENCE RELATION.

******************************************************************************

DIMENSION T(12)
BK=0
IF(N)=1,11,11
10 IER=1
RETURN
11 IF(X)12,12,20
12 IER=2
RETURN
20 IF(X)170,022,22,21
21 IER=3
RETURN
22 IER=0
IF(X)36,36,25
25 A=EXP(-X)
B=1/X
C=SQR(T(8))
T(1)=B
DO 26 L=2,12
26 T(L)=T(L-1)*B
IF(N)127,29,27
27 G0=1.2533141,1566642*T(1),08811128*T(2),09139095*T(3)
2*1.14459*10(T(4),2298850*T(5),3792410*T(6),5247277*T(7)
3*5575368*T(8),4262633*T(9),2184518*T(10),96398977*T(11)
4*009189833*T(12))
IF(N)20,28,29
28 BK=G0
RETURN
Table 2.1—Listing of program for partial penetration in a nonleaky artesian aquifer—Continued

```plaintext
COMPUTE K1 USING POLYNOMIAL APPROXIMATION

29 G1 = A*(1, 2533141+4699927*T(1)+1468583*T(2)+1280427*T(3)
   +847510*T(4)+2847618*T(5)+45944342*T(6)+6283381*T(7)
   *6632295*T(8)+5050239*T(9)+2581304*T(10)+7880001*T(11)
   +1082418*T(12))*K
   IF(N=1) 20, 30, 31
30 BK=G1
   RETURN

FROM KG, K1 COMPUTE KN USING RECURRENCE RELATION

31 DO 35 J=2, N
   GJ=2.(FLOAT(J)=1.)*G1/X+G0
   IF(GJ=1.0E+0) 33, 33, 32
32 IFER=4
   G0 TO 34
33 G0=G1
35 G1=GJ
34 BK=GJ
   RETURN
36 BMX/2,
   A=5772157+ALOG(B)
   C=A*B
   IF(N=1) 37, 43, 37

COMPUTE KG USING SERIES EXPANSION

37 G0=A
   X2J=1,
   FACT=1,
   HJ=0
   DO 40 J=1, B
   RJ=1/FLOAT(J)
   IF(X2J, LT, 1.E-40) X2J=0,
   PREVIOUS STATEMENT ADDED TO IBM SUBROUTINE TO CORRECT UNDERFLOW
   PROBLEM ON MATFOR 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

COMPUTE K1 USING SERIES EXPANSION

43 X2J=B
   FACT=1,
   HJ=1
   G1=A/X+X2J*(.5+A=HJ)
   DO 50 J=1, B
   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 2.1.—Listing of program for partial penetration in a nonleaky artesian aquifer—Continued

SUBROUTINE EXPI(X,RES,AUX)

***************************************************************************
SUBROUTINE EXPI

PURPOSE
COMPUTES THE EXPONENTIAL INTEGRAL *Ei(X)

USAGE
CALL EXPI(X,RES)

DESCRIPTION OF PARAMETERS
X = ARGUMENT OF EXPONENTIAL INTEGRAL
RES = RESULT VALUE
AUX = RESULTANT AUXILIARY VALUE

REMARKS
* X GT 170 (X LT -174) MAY CAUSE UNDERFLOW (OVERFLOW)
* WITH THE EXPONENTIAL FUNCTION
* FOR X = 0 THE RESULT VALUE IS SET TO 1, E75

SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
NONE

METHOD
DEFINITION
RES*INTEGRAL(EXP(-T)/T, SUMMED OVER T FROM X TO INFINITY).
EVALUATION
THREE DIFFERENT RATIONAL APPROXIMATIONS ARE USED IN THE
RANGES 1 LE X, X LE -9 AND -9 LT X LE -3 RESPECTIVELY,
A POLYNOMIAL APPROXIMATION IS USED IN -3 LT X LT 1.

***************************************************************************

IF(X=1.,)2,1,1
1 Y=1./X
AUX=1.,Y=((((Y+3.377358E0)*Y+2.052156E0)*Y+2.709479E1)/(((Y*
11.072553E0+5.716943E0)*Y+6.945239E0)*Y+2.593886E0)*Y+2.709496E1)
RES=AUX*EXP(-X)
RETURN
2 IF(X+3.,)3,6,4
3 AUX=((((7.122452E7*X=1.766345E0)*X+2.928433E5)*X+2.335379E4)*X
1+1.66156E3)*X=1.041576E2)*X+5.55582E2)*X+2.500001E1)*X
2+.999999991
RES=1./X
IF(X)4,5,4
4 RES=AUX-ALOG(A9(X)=5.772157E1
5 RETURN
6 IF(X+9.,)9,8,7
7 AUX=((((5.176245E2)*X+3.061037E0)*X+1.243665E1)*X+2.242432E2)*X
1+2.486697E2)/(((X+3.995161E0)*X+3.893441E0)*X+2.263818E1)*X
2+1.80737E2)
GOTO 9
8 Y=9./X
9 AUX=((((Y+7.659824E0)*Y+7.271015E1)*Y+1.080693E0)/(((Y*
1+2.518750E0+1.122927E1)*Y+5.921405E0)*Y+8.666702E0)*Y=9.724216E0)
RES=AUX*EXP(-X)/X
RETURN
END
Table 4.3—Listing of program for radial flow in a leaky artesian aquifer

```
******n*********c**c**nn*n*nnnnnnnnnnnnnn**nn**nn*nnnnnn*n*nn*nn***wu0
PURPOSE
TO COMPUTE A TABLE OF VALUES OF THE LEAKY AQUIFER WELL
FUNCTION - W(U,R/B) = HANTUSH, M.S., AND JACOBI, C.E., 1955,
NON-STEADY RADIAL FLOW IN AN INFINITE LEAKY AQUIFER. AM,
GEOPHYS. UNION TRANS., V. 36, NO. 1, P. 95-100.

INPUT DATA
1 CARD - FORMAT(2E10,5)
USMAL = SMALLEST VALUE OF 1/U FOR WHICH COMPUTATION IS
DESIRED.
ULARGE = LARGEST VALUE OF 1/U FOR WHICH COMPUTATION IS
DESIRED.
2 CARDS - FORMAT(8E10,5)
BDAT = 12 VALUES OF R/B FOR TABLE.

SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
L, SERIES, FCT, BESK, DOL12

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) USMAL,ULARGE
READ (IRD,6) BDAT
IBEGIN=ALOG10(USMAL)
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=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
Um1=/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

6 FORMAT (8E10,5)
7 FORMAT ('11,'W(U,R/B)/1\'0,10X,'1 R/B/','1,6X,'1/U 1',12E10,2)
8 FORMAT ('11,E10.3,12F10.4)
END

REAL FUNCTION L*U(U,V)

FUNCTION L
```

---

**TECHNIQUES OF WATER-RESOURCES INVESTIGATIONS**

---

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

---

**Purpose:** To compute a table of values of the leaky aquifer well function - \( W(U, R/B) = HANTUSH, M.S., \) and Jacobi, C.E., 1955, non-steady radial flow in an infinite leaky aquifer. *Am.* Geophys. Union Trans., V. 36, No. 1, P. 95-100.

**Input Data:**
1 card - Format (2E10,5)
USMAL = Smallest value of 1/U for which computation is desired.
ULARGE = Largest value of 1/U for which computation is desired.
2 cards - Format (8E10,5)
BDAT = 12 values of R/B for table.

**Subroutines and Function Subprograms Required:**
L, SERIES, FCT, BESK, DOL12

---

**Program Code**

---

```plaintext
PURPOSE
TO COMPUTE A TABLE OF VALUES OF THE LEAKY AQUIFER WELL
FUNCTION - W(U,R/B) = HANTUSH, M.S., AND JACOBI, C.E., 1955,
NON-STEADY RADIAL FLOW IN AN INFINITE LEAKY AQUIFER. AM,
GEOPHYS. UNION TRANS., V. 36, NO. 1, P. 95-100.

INPUT DATA
1 CARD - FORMAT(2E10,5)
USMAL = SMALLEST VALUE OF 1/U FOR WHICH COMPUTATION IS
DESIRED.
ULARGE = LARGEST VALUE OF 1/U FOR WHICH COMPUTATION IS
DESIRED.
2 CARDS - FORMAT(8E10,5)
BDAT = 12 VALUES OF R/B FOR TABLE.

SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
L, SERIES, FCT, BESK, DOL12

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) USMAL,ULARGE
READ (IRD,6) BDAT
IBEGIN=ALOG10(USMAL)
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=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
Um1=/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

6 FORMAT (8E10,5)
7 FORMAT ('11,'W(U,R/B)/1\'0,10X,'1 R/B/','1,6X,'1/U 1',12E10,2)
8 FORMAT ('11,E10.3,12F10.4)
END

REAL FUNCTION L*U(U,V)

FUNCTION L
```
TYPE CURVES FOR FLOW TO WELLS IN CONFINED AQUIFERS

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

**PURPOSE**

TO COMPUTE THE INTEGRAL \( \exp(-y\sqrt{y})/y \) SUMMED OVER \( y \) FROM

U TO INFINITY (WELL FUNCTION FOR LEAKY AQUIFERS).

**DESCRIPTION OF PARAMETERS**

- BOTH DOUBLE PRECISION
  - \( u = R^2/4 \times \text{TIME} \) (RADIAL DISTANCE Squared * STORAGE
  - COEFFICIENT / TRANSMISSIVITY * TIME)
  - \( v = R^2/8 \times \sqrt{K/(T*B')} = \text{ONE} = \text{HALF RADIAL DISTANCE} \times \text{SQUARE ROOT}
    - (HYD, COND, OF CONFINING BED/TRANSMISSIVITY*THICKNESS
    - OF CONFINING BED)

**SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED**

- DQL12, SERIES, BESK, FCT

**METHOD**

IN THE FOLLOWING F=\( \exp(1+y^2/y) \)

1. \( u>1 \), USES A GAUSS-LAGUERRE QUADRATURE FORMULA TO
   EVALUATE INTEGRAL(F) FROM U TO INF.
2. \( v**2<u<1 \), USES THE G-L QUADRATURE TO EVALUATE INTEGRAL(F)
   FROM ONE TO INF AND A SERIES EXPANSION TO EVALUATE INTEGRAL(F)
   FROM U TO ONE.
3. \( u<1 \), \( u=v**2/u \), USES THE REPRESENTATION INTEGRAL(F) FROM U
   TO INF = 2*K0(2*v) INTEGRAL(F) FROM v**2/u TO INF.
   EVALUATES THE ZERO UNDER MODIFIED BESSEL FUNCTION OF SECOND
   KIND WITH IBM SUBROUTINE, EVALUATES INTEGRAL BY G-L QUAD.

***************

**EXTERNAL FCT**

REAL 8 U, V, S, F, VV, SERIES
COMMON /C1/ VV, VV
VV = V
IF (U=1.) 1, 2, 2
C CHECKS IF U<1
1 Z = V**2/U
IF (Z>1.) 3, 4, 4
C CHECKS IF V**2/U < 1
C 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 = SERIES(U, V)
C INTEGRAL 1 TO INF, BY G-L QUAD, INTEGRAL U TO 1 BY SERIES EXP,
GO TO 5
4 Z = V**2
CALL BESK(TMOV, 0, 8K, IER)
CALL DQL12(FCT, F)
L = SERIES(U, V)
C 2K0(2v) INTEGRAL V**2/u TO INF,
C 5 RETURN
END
REAL FUNCTION SERIES*8(U, V)

**FUNCTION SERIES**

TO EVALUATE S(1)=S(U), WHERE S IS A SERIES EXPANSION OF

INTEGRAL [\( \exp(-y\sqrt{y})/y \) DOY/Y] GIVEN BY S = SUM M TO INFINITY, S WHERE F(M) =

LOG(U) IF M = AND = (((1)M/M) * (U**2) (2)M/M) IF M > 0.

**DESCRIPTION OF PARAMETERS**

- BOTH DOUBLE PRECISION
  - \( u = R^2/4 \times \text{TIME} \) (RADIAL DISTANCE Squared * STORAGE
  - COEFFICIENT / TRANSMISSIVITY * TIME)
  - \( v = R^2/8 \times \sqrt{K/(T*B')} = \text{ONE} = \text{HALF RADIAL DISTANCE} \times \text{SQUARE ROOT}
    - (HYD, COND, OF CONFINING BED/TRANSMISSIVITY*THICKNESS
    - OF CONFINING BED)
Table 4.3—Listing of program for radial flow in a leaky artesian aquifer—Continued

SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED

NONE

REAL DLOG,DABS,S(2),VUM,UU
REAL TEST,U,UM,EM,EN,SM1,SM,SIGN,V,VSQ,VSQU,RMUL,TERM,TERM1
TEST=.5,D=.7
VSQ=V*V
UUM=U
DO 6 I=1,2
EVALUATES SERIES FOR LOWER LIMIT = U AND UPPER LIMIT = 1
IF (I.EQ.2) UM=1,
EN=I,
EM=EM+1,
SM1=0,
SIGN=1,
VUM=1,
VSQU=VSQU/U
1 EM=EM+1,
IF (EM.EQ.1) 2,3,3
CHECKS FOR M=0
2 RMUL=DLOG(U)
TERM=1,
GO TO 4
3 UUM=U
IF (VUM.LT.1, D=.30) VUM=0,
VUM=VUM*VSQU
RMUL=(UM*VUM)/EM
TERM=TERM1/EM
4 SIGN=SIGN
SUM=TERM1
TERM=TERM1
EN=0,
5 EM=EM+1,
TERM=TERM*VSQ/(EN*(EN+EM))
SUM=SUM+TERM
IF (TEST.EQ.9,DABS(RMUL*EN+TERM)) GO TO 5
C TRUNCATES INNER SERIES IF OUTER TERM*INNER TERM < .5,E=7
SUM=SUM1+SIGN*RMUL*SUM
IF (EM.LT.1) GO TO 1
IF (TEST.EQ.9,DABS(RMUL*SUM)) GO TO 1
C TRUNCATES OUTER SERIES IF OUTER TERM*INNER SUM < .5,E=7
6 S(1)=SUM1
UUM=U
SERIES=S(2)-S(1)
RETURN
END

REAL FUNCTION FCT*8(X)

FUNCTION FCT

PURPOSE
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

```fortran
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#S, D1) 3,3,1
3 FCT#DEXP(P)/(X+Z)
4 RETURN
END FCT

SUBROUTINE OQL12(FCT,Y)

PURPOSE
TO COMPUTE INTEGRAL(EXP(-X)*FCT(X), SUMMED OVER X
FROM 0 TO INFINITY),

USAGE
CALL OQL12 (FCT,Y)
PARAMETER FCT REQUIRES AN EXTERNAL STATEMENT

DESCRIPTION OF PARAMETERS
FCT  - THE NAME OF AN EXTERNAL DOUBLE PRECISION FUNCTION
SUBPROGRAM USED,
Y    - THE RESULTING DOUBLE PRECISION INTEGRAL VALUE.

REMARKS
NONE

SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
THE EXTERNAL DOUBLE PRECISION FUNCTION SUBPROGRAM FCT(X)
MUST BE FURNISHED BY THE USER.

METHOD
EVALUATION IS DONE BY MEANS OF 12-POINT GAUSSIAN-LAGUERRE
QUADRATURE FORMULA, WHICH INTEGRATES EXACTLY,
WHENEVER FCT(X) IS A POLYNOMIAL UP TO DEGREE 23.
FOR REFERENCE, SEE
SHAO/CHEN/Frank, TABLES OF ZEROS AND GAUSSIAN WEIGHTS OF
CERTAIN ASSOCIATED LAGUERRE POLYNOMIALS AND THE RELATED
GENERALIZED HERMITE POLYNOMIALS, IBM TECHNICAL REPORT
TR001100 (MARCH 1964), PP.24-25.

DOUBLE PRECISION X,Y,FCT

X=370991210446692 D2
Y=814807746742624 D0=15*FCT(X)
```
Table 4.3—Listing of program for radial flow in a leaky artesian aquifer—Continued

```
x(0) = 284.879675309840 12
y(0) = 300.1001635505021 11*FCT(X)
X(0) = 221510903739701 12
Y(0) = 1.14239030551504 6*FCT(X)
X(1) = 171166518746226 12
Y(1) = 1.16684367540910 6*FCT(X)
X(2) = 1.30600549933055 12
Y(2) = 0.5 6*FCT(X)
X(3) = 0.962131684265687 11
Y(3) = 0.203215926629904 3*FCT(X)
X(4) = 0.684452545311577 11
Y(4) = 0.266397351835316 2*FCT(X)
X(5) = 0.4599227639018348 11
Y(5) = 0.2010238115463410 1*FCT(X)
X(6) = 0.2033751337474507 11
Y(6) = 0.150449222116809 1*FCT(X)
X(7) = 0.1302610269776419 11
Y(7) = 0.117574845151307 0
X(8) = 0.0777592758731380 0
X(9) = 0.057221173580207 0
Y(10) = 0.1247313710554432 0
RETURN
END

SUBROUTINE BESK(X,N,BK,IER)

******************************************************************************

SUBROUTINE BESK

COMPUTE THE K BESSEL FUNCTION FOR A GIVEN ARGUMENT AND ORDER

USAGE

CALL BESK(X,N,BK,IER)

DESCRIPTION OF PARAMETERS

X = THE ARGUMENT OF THE K BESSEL FUNCTION DESIRED
N = THE ORDER OF THE K BESSEL FUNCTION DESIRED
BK = THE RESULTANT K BESSEL FUNCTION
IER = RESULTANT ERROR CODE WHERE
IER = 0  NO ERROR
IER = 1  N IS NEGATIVE
IER = 2  X IS ZERO OR NEGATIVE
IER = 3  X > GT  170, MACHINE RANGE EXCEEDED
IER = 4  BK > GT  10**70

REMARKS

N MUST BE GREATER THAN OR EQUAL TO ZERO

SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED

NONE

METHOD

COMPUTES ZERO ORDER AND FIRST ORDER BESSEL FUNCTIONS USING SERIES APPROXIMATIONS AND THEN COMPUTES NTH ORDER FUNCTION USING RECURRENCE RELATION. 

RECURRENCE RELATION AND POLYNOMIAL APPROXIMATION TECHNIQUE AS DESCRIBED BY A.J.W. HITCHCOCK, POLYNOMIAL APPROXIMATIONS TO BESSEL FUNCTIONS OF ORDER ZERO AND ONE AND TO RELATED FUNCTIONS, "A TREATISE ON THE THEORY OF BESSEL FUNCTIONS", CAMBRIDGE UNIVERSITY PRESS, 1958, P. 62

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

```
C DIMENSION T(12)
BK=0.0
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)127,29,27
27 G0=G0*(1,2533141*1566642*T(1)+08811128*T(2)=-09139095*T(3)
2*1.344569*T(4)=2299985*T(5)+3792410*T(6)+5247277*T(7)
3*1.555368*T(8)=4262633*T(9)+2184518*T(10)+06680977*T(11)
4*1.09189383*T(12)*C
IF(N=1)20,20,29
28 BK=G0
RETURN
29 G1=G1*(1,2533141*4699927*T(1)+1468583*T(2)+1280427*T(3)
2*1.7563432*T(4)+2847618*T(5)+0594342*T(6)+293381*T(7)
3*0632295*T(8)+5050239*T(9)+2581304*T(10)+7885601*T(11)
4*1.0182348*T(12)*C
IF(N=1)20,30,31
30 BK=G1
RETURN
31 FROM KO,K1 COMPUTE KN USING RECURRENCE RELATION
32 DO 35 J=2,N
GJ=2,*(FLOAT(J)-1,)*G1/X+G0
IF(GJ=1.0E70)33,33,32
33 G0=G1
35 G1=4GJ
34 BK=GJ
RETURN
36 B=X/2,
A=5772157*ALOG(B)
C=B*B
IF(N=1)37,43,37
37 COMPUTE KU USING SERIES EXPANSION
```

BESK 380
BESK 390
BESK 400
BESK 420
BESK 430
BESK 440
BESK 450
BESK 460
BESK 470
BESK 480
BESK 490
BESK 500
BESK 510
BESK 520
BESK 530
BESK 540
BESK 550
BESK 560
BESK 570
BESK 580
BESK 590
BESK 600
BESK 610
BESK 620
BESK 630
BESK 640
BESK 650
BESK 660
BESK 670
BESK 680
BESK 690
BESK 700
BESK 710
BESK 720
BESK 730
BESK 740
BESK 750
BESK 760
BESK 770
BESK 780
BESK 790
BESK 800
BESK 810
BESK 820
BESK 830
BESK 840
BESK 850
BESK 860
BESK 870
BESK 880
BESK 890
BESK 900
BESK 910
BESK 920
BESK 930
BESK 940
BESK 950
BESK 960
BESK 970
BESK 980
BESK 990
Table 4.3—Listing of program for radial flow in a leaky artesian aquifer—Continued

```c
37 GO TO A
X2J=J1
FACT=S
MJ=0
DO 40 J=1,6
  RJ=FLOAT(J)
  IF(ABS(LT.1,10),.EQ.0) X2J=0.
  PREVIOUS STATEMENT ADDED TO IBM SUBROUTINE TO CORRECT UNDERFLOW
  PROBLEM LIN MATFOR COMPILER
  X2J=X2J*FACT
  MJ=MJ+RJ
  GO TO 40
IF(N)43,42,43
42 BKG=0
RETURN
COMPUTE K1 USING SERIES EXPANSION
43 X2J=B
FACT=S
MJ=1
G1=1, X+X2J*(,.5+.A-MJ)
DO 50 J=2,9
  X2J=X2J*FACT
  MJ=MJ+RJ
  G1=G1+X2J*FACT*(.5+(A-MJ)*FLOAT(J))
  IF(N)31,55,31
50 G1=G1+X2J*FACT*(.5+(A-MJ)*FLOAT(J))
IF(N)31,52,31
52 BKG=I
RETURN
END
```

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

```c
C  **********************************************C
C  PURPOSE
C  TO COMPUTE TYPE CURVE FUNCTION VALUES FOR H(U,REIA) =
C  S, PAPADOPULUS.
C  INPUT DATA
C  1 CARD = FORMAT(2E10.5)
C  USMALL = SMALLEST(BEGINNING) VALUE OF 1/U,
C  USLARGE = LARGEST(ENDING) VALUE OF 1/U,
C  2 CARDS = FORMAT(2E10.5)
C  BDAT = 12 VALUES OF BETA (ZERO OR BLANK VALUES ARE
C  PERMITTED IF LESS THAN 12 DESIRED, WILL TERMINATE
C  AT FIRST ZERO OR BLANK VALUE),
C  SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C  M, D0G32, numeric must be included in deck,
C  USQRAT, DEXP, DERC, DLOG = Must be in computer library.
C  REAL*8 U, BETA
C  DIMENSION ARRAY(73,12), Y(73), BDAT(12), YNUM(6)
C  DATA YNUM/1,1,1,1,5,5,5,7,7/
C  THU=5
C  IPT=6
C  READ (IMD,6) USMALL, USLARGE
C  READ (IMD,6) BDAT
C  READ (IMD,6) BDAT, USMALL, USLARGE, IPT=6
C  TO BEGIN A CALCULATION WITH USMALL
C  TEND=USLARGE(USMALL)
C  IF (TENDEQBEGIN) Go to 2
C  IF (ILIMIT,GT,73) ILIMIT=73
C  DO 1 I=1,12
C  IF (BDAT(I),EQ,0,) Go to 2
C  CONTINUE
C  NB=12
```

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

```c
37 GO TO A
X2J=J1
FACT=S
MJ=0
DO 40 J=1,6
  RJ=FLOAT(J)
  IF(ABS(LT.1,10),.EQ.0) X2J=0.
  PREVIOUS STATEMENT ADDED TO IBM SUBROUTINE TO CORRECT UNDERFLOW
  PROBLEM LIN MATFOR COMPILER
  X2J=X2J*FACT
  MJ=MJ+RJ
  GO TO 40
IF(N)43,42,43
42 BKG=0
RETURN
COMPUTE K1 USING SERIES EXPANSION
43 X2J=B
FACT=S
MJ=1
G1=1, X+X2J*(,.5+.A-MJ)
DO 50 J=2,9
  X2J=X2J*FACT
  MJ=MJ+RJ
  G1=G1+X2J*FACT*(.5+(A-MJ)*FLOAT(J))
  IF(N)31,55,31
50 G1=G1+X2J*FACT*(.5+(A-MJ)*FLOAT(J))
IF(N)31,52,31
52 BKG=I
RETURN
END
```

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

```c
C  **********************************************C
C  PURPOSE
C  TO COMPUTE TYPE CURVE FUNCTION VALUES FOR H(U,REIA) =
C  S, PAPADOPULUS.
C  INPUT DATA
C  1 CARD = FORMAT(2E10.5)
C  USMALL = SMALLEST(BEGINNING) VALUE OF 1/U,
C  USLARGE = LARGEST(ENDING) VALUE OF 1/U,
C  2 CARDS = FORMAT(2E10.5)
C  BDAT = 12 VALUES OF BETA (ZERO OR BLANK VALUES ARE
C  PERMITTED IF LESS THAN 12 DESIRED, WILL TERMINATE
C  AT FIRST ZERO OR BLANK VALUE),
C  SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C  M, D0G32, numeric must be included in deck,
C  USQRAT, DEXP, DERC, DLOG = Must be in computer library.
C  REAL*8 U, BETA
C  DIMENSION ARRAY(73,12), Y(73), BDAT(12), YNUM(6)
C  DATA YNUM/1,1,1,1,5,5,5,7,7/
C  THU=5
C  IPT=6
C  READ (IMD,6) USMALL, USLARGE
C  READ (IMD,6) BDAT
C  READ (IMD,6) BDAT, USMALL, USLARGE, IPT=6
C  TO BEGIN A CALCULATION WITH USMALL
C  TEND=USLARGE(USMALL)
C  IF (TENDEQBEGIN) Go to 2
C  IF (ILIMIT,GT,73) ILIMIT=73
C  DO 1 I=1,12
C  IF (BDAT(I),EQ,0,) Go to 2
C  CONTINUE
C  NB=12
```
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 N=1
3 DO 4 =1,N
4 IF (N.GT.10) GO TO 5
5 WRITE (9,1) (ARRAY(I,J),I=1,N, J=1,N)
6 WRITE (9,1) (RH, I=1,N)
7 STOP

1 FORMAT (7E10.5)

DOUBLE PRECISION FUNCTION H(U,B)

FUNCTION H

PURPOSE

TO COMPUTE THE INTEGRAL OF

\[ \int_{V}^{\infty} \frac{X^{p-1}e^{-X}}{X^{2} + B^{2}} \, dX \]

FROM \( V \) TO INFINITY (FUNCTION \( H(U,B) \) OF \( U, B \)).

DESCRIPTION OF PARAMETERS

D0BLE PRECISION

\( U \) = R\(^2\)/4 \times \Delta T \times TIME \times STORAGE COEFFICIENT \times TIME, \( U \) MUST BE > L.D-60,

\( B \) = (U\(^{2}\))/4 \times \Delta T \times STORAGE COEFFICIENT \times THICKNESS OF UPPER CONFINING BED.

\( K2,1,2 \) = HYD. COND., STORAGE COEFFICIENT OF LOWER CONFINING BED.

METHOD

1. FOR \( U < 1 \times 10^{-6} \), NO COMPUTATION IS MADE.

2. FOR \( B > 100 \), \( H(U,B) = 0 \).

3. \( H(U,B) = \int_{V}^{\infty} \frac{X^{p-1}e^{-X}}{X^{2} + B^{2}} \, dX \)

FOR \( U < \sqrt{B} \) WHERE \( \sqrt{B} \) IS THE \( U \) CORRESPONDING TO \( B = 2 \times 10^{-4} \).

4. \( H(U,B) = \) INTEGRAL FROM \( U \) TO \( U \) \( \times \sqrt{B} \).

5. \( H(U,B) = \) INTEGRAL FROM \( U \) TO \( \infty \).

END

IMPLICIT REAL*8(A-H,O-Z)
COMMON U,W,BB,B3,B2,BU,UL,UM,UMW
EXTERNAL H

B208=B
IF (U.GT.1,U.EQ.0) GO TO 1
\*RITE (9,7)
STOP
1 IF (U.EQ.0,U.EQ.0) GO TO 5
IF (U.EQ.10,U.EQ.0) GO TO 6
H208=B
IF (B.GT.1,0,AND.BU.GE.0) GO TO 6
H208=B
UP=10,0
UB0,5=UB(1,0,UBOR(T1,0+0.025*0.B8/U))
UB0,5=UBOR(T1,0+1.025*0.B8/U))
IF (UB0,5>UB) GO TO 2
H208=UB
UP=UB
2 H208=B
3 X=UB10,X
IF (X.U.EQ.1) X=UB1
CALL UG23(X,UL,UM,ARE)
H208=ARE
X=UB1
IF (X.EQ.1) GO TO 3
4 H208=ARE
RETURN
5 H208=UB
RETURN
Table 5.2—Listing of program for radial flow in a leaky artesian aquifer with storage of water in the confining beds—Continued

SUBROUTINE DOG32(XL, XU, FCT, Y)

SUBROUTINE DOG32

PURPOSE

TO COMPUTE INTEGRALS(FCT(X), SUMMED OVER X FROM XL TO XU)

USAGE

CALL DOG32 (XL, XU, FCT, Y)

PARAMETER FCT REQUIRES AN EXTERNAL STATEMENT

DESCRIPTION OF PARAMETERS

XL = DOUBLE PRECISION LOWER BOUND OF THE INTERVAL,
XU = DOUBLE PRECISION UPPER BOUND OF THE INTERVAL,
FCT = THE NAME OF AN EXTERNAL DOUBLE PRECISION FUNCTION SUBPROGRAM USED,
Y = THE RESULTING DOUBLE PRECISION INTEGRAL VALUE.

REMARKS

NONE

SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED

THE EXTERNAL DOUBLE PRECISION FUNCTION SUBPROGRAM FCT(X) MUST BE FURNISHED BY THE USER,

METHOD

EVALUATION IS DONE BY MEANS OF 32-POINT GAUSS QUADRATURE FORMULA, WHICH INTEGRATES POLYNOMIALS UP TO DEGREE 63 EXACTLY, FOR REFERENCE, SEE V.L. KRYLOV, APPROXIMATE CALCULATION OF INTEGRALS, MACMILLAN, NEW YORK/LONDON, 1962, PP.100-111 AND 337-340.

DOUBLE PRECISION XL, XU, Y, A, B, C, FCT

FUNCTION HUB

PURPOSE

TO COMPUTE VALUES OF THE INTEGRAND OF H(U, B)

DESCRIPTION OF PARAMETER

X = DOUBLE PRECISION, POINT AT WHICH INTEGRAND IS EVALUATED.

DOUBLE PRECISION FUNCTION HUB(X)
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
        FUNCTION EVALUATION OF FUNCTION
        IMPLICIT REAL(A,M,D,Z)
        COMMON UU,DD
        A = 0.95910524 + 2.91991999 + 0.00796909
        M = 0.007
        N = 0.005
        END
        DOUBLE PRECISION FUNCTION MLJ
        C

        FUNCTION MLJ
        C
        PURPOSE
        TO EVALUATE THE WELL FUNCTION OF THIS
        DESCRIPTION OF PARAMETERS
        "U" = DOUBLE PRECISION, ARGUMENT FOR WELL FUNCTION,

```

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

```
C       ********************************************************************
PPL  1
C       PURPOSE
        TO COMPUTE TYPE CURVE FUNCTION VALUES FOR PARTIAL PENETRATION
        IN A LEAKY AQUIFER USING Eqs. 73 OF HANTUSH, M. S., 1964.
        HYDRAULICS OF WELLS IN CMR. VEN TE, ADVANCES IN HYDRUSCIENCE,
        VOL. 11 ACADEMIC PRESS, NEW YORK, P. 281-442.
C       INPUT DATA
        1 CARD = FORMAT (3F5.1, IS, 2E10, 4)
        B = AQUIFER THICKNESS
        E = DEPTH, BELOW TOP OF AQUIFER, TO BOTTOM OF PUMPING WELL SCREEN
        D = DEPTH, BELOW TOP OF AQUIFER, TO TOP OF PUMPING WELL SCREEN
        NUM = NUMBER OF OBSERVATION WELLS OR PIEZOMETERS TIMES
        NUMBER OF VALUES OF Kz/KR
        SMALL = SMALLEST VALUE OF 1/U FOR WHICH COMPUTATION IS DESIRED
        LARGE = LARGEST VALUE OF 1/U FOR WHICH COMPUTATION IS DESIRED
        2 CARDS = FORMAT (8E10.5)
        BDAT = 12 VALUES OF R/BR, NON ZERO VALUES SHOULD BE
        FIRST, WILL TERMINATE AT FIRST ZERO (OR BLANK) VALUE.
        NUM CARDS (ONE FOR EACH OBS, WELL OR PIEZOMETER AND FOR EACH
        VALUE OF RSORT(KZ/KR) = FORMAT (3F5.1)
        R = RADIAL DISTANCE FROM PUMPED WELL TIMES SQRT(KZ/KR)
        LPRIME = DEPTH, BELOW TOP OF AQUIFER, TO TOP OF OBS, WELL
        SPF = 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

SCREEN (TOTAL DEPTH FOR PIEZOMETER)                           PPL 30
SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED                  PPL 31
DG12, SERIES, BESK, FCT, LFL                                    PPL 32

******************************************************************************PPL 34
REAL*6 U, V                                                      PPL 35
REAL*4 L, LB, LPB, LPrime, LARGE                               PPL 36
DIMENSION ARRAY(55, 12), ARG(6), BDAT(12), Y(55)               PPL 37
DATA ARG/1, 1, 5, 2, 3, 5, 7,/                                  PPL 38
DATA ARRAY/660*0, 1, Y/55*0,/                                   PPL 39
IRD=5                                                           PPL 40
IPT=6                                                           PPL 41
READ (IRD, 9) R, E, D, NUM, SMALL, LARGE                       PPL 42
READ (IRD, 14) BDAT                                            PPL 43
DO 1 I=1, 12                                                     PPL 44
IF (BDAT(I), EQ, 0, ) GO TO 2                                    PPL 45
1 CONTINUE                                                      PPL 46
NB=12                                                           PPL 47
GO TO 3                                                         PPL 48
2 NB=1=1                                                        PPL 49
3 LB=EB/8                                                       PPL 50
DB=DB/B                                                         PPL 51
IBEGIN=ALOG10(SMALL)                                           PPL 52
IEND=ALOG10(LARGE)+1                                           PPL 53
JLIMIT=IEND-IBEGIN                                            PPL 54
IF (JLIMIT.GT, 9) JLIMIT=9                                     PPL 55
ILIMIT=ILIMIT+1                                                 PPL 56
DO 8 K=1, NUM                                                   PPL 57
READ (IRD, 9) R, LPrime, DPRIME                                 PPL 58
RB=RB/8                                                         PPL 59
LPB=LPrime/H                                                    PPL 60
DPB=DPrime/H                                                    PPL 61
DO 4 I=1, ILIMIT                                                 PPL 62
INDEX=(I-1)/6                                                   PPL 63
IEXP=IBEGIN+INDEX                                              PPL 64
LIM=INDEX**b                                                    PPL 65
Y(I)=ARG(I)*10, **IEXP                                         PPL 66
U=1, Y(I)                                                      PPL 67
DO 4 J=1, NB                                                    PPL 68
BETA=BDAT(J)                                                    PPL 69
V=BETA/2,                                                       PPL 70
4 ARRAY(I, J)=L(U, V)+FL(U, KB, BETA, LB, DB, LPB, DPB)         PPL 71
IF (LPB=0 ) 5, 5, 6                                             PPL 72
5 WRITE (IPT, 10) DB, KB, LB, DB                                 PPL 73
GO TO 7                                                         PPL 74
6 WRITE (IPT, 11) LPB, DPB, RB, LB, DB                          PPL 75
7 WRITE (IPT, 12) (BDAT(I), I=1, NB)                            PPL 76
DO 8 I=1, JLIMIT                                                 PPL 77
WRITE (IPT, 13) Y(I), (ARRAY(I, J), J=1, NB)                   PPL 78
8 CONTINUE                                                      PPL 79
STOP                                                            PPL 80

9 FORMAT (3F5, 15, 2E10, 4)                                      PPL 81
10 FORMAT (15F1, 17, 15F10, 4)                                   PPL 82
11 FORMAT (15F1, 17, 15F10, 4, 9X, D, 8X, 8X, 8X, D, B=1, F5, 2, 1) PPL 83
12 FORMAT (15F1, 17, 15F10, 4, D, 8X, 8X, 8X, D, B=1, F5, 2, 1, D, B=1, F5, 2, 1) PPL 84
13 FORMAT (15F1, 17, 15F10, 4, D, 8X, 8X, 8X, D, B=1, F5, 2, 1, D, B=1, F5, 2, 1) PPL 85
14 FORMAT (4F10, 8)                                              PPL 86
END                                                             PPL 87
REAL FUNCTION FL*4(U,RB,BETA,LB,DB,LPB,DPB)

FUNCTION FL

PURPOSE

TO COMPUTE DEPARTURES FROM HANTUSH-JACOB LEAKY AQUIFER CURVE

CAUSED BY PARTIAL PENETRATION OF PUMPED WELL.

USAGE

FL(U,RB,BETA,LB,DB,LPB,DPB)

DESCRIPTION OF PARAMETERS

ALL REAL, U DOUBLE PRECISION

U = H**2*S/Y*T*TIME (RADIAL DISTANCE SQUARED * STORAGE

COEFFICIENT / 4*TRANSMISSIVITY * TIME

RB = R/B (RADIAL DISTANCE / AQUIFER THICKNESS)

BETA = R*SQRT(K/BlT) . (RADIAL DISTANCE * SQUARE ROUT

(HYD. COND. OF CONFINING BED/THICKNESS OF CONFINING

BED * TRANSMISSIVITY OF AQUIFER))

LB = L/B (FRACTION OF AQUIFER PENETRATED BY PUMPED WELL)

DB = D/B (FRACTION OF AQUIFER ABOVE PUMPED WELL SCREEN)

LPB = L'/B (FRACTION OF AQUIFER PENETRATED BY OBS. WELL, ZERO

FOR PIEZOMETER)

DPB = D'/B (FRACTION OF AQUIFER ABOVE OBS. WELL SCREEN, TOTAL

DEPTH FOR PIEZOMETER)

SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED

DUL12, SERIES, BESK, FCT, L

METHOD

SUMS THE SERIES THROUGH N*PI*R/B EQ 20

REAL*8 U,V,DSQ RT

REAL*4 L,N,LB,LPB

SUM=0.,

N=0.,

BETSA=BETA*BETA

PIB=3.141593*LB

PIID=3.141593*DB

IF (LPB=0.) 1,1,4

C CHECKS FOR WELL OR PIEZOMETER

1 PI ZB=3.141593*DPB

2 N=N+1.

V=SQRT(BETA**2*L*PIB**2)/(REL*LB**2/2.

IF (V,GT,10.) GO TO 3

C TRUNCATES SERIES WHEN V>10

X=L(N,V)/N

SUM=SUM+(SIN(N*PIB)*SIN(N*PIDR)*COS(N*PIZB)*X)

GO TO 2

3 FL=6366198*SUM/(LB=DB)

GO TO 7

4 PIB=3.141593*LB

PIID=3.141593*DB

5 N=N+1

V=SQRT(BETA**2*L*PIB**2)/(REL*LB**2/2.

IF (V,GT,10.) GO TO 6

C TRUNCATES SERIES WHEN V>10

X=L(N,V)/N

SUM=SUM+(SIN(N*PIB)*SIN(N*PIDR)*SIN(N*PIZB)*X/N)

GO TO 5

6 FL=2026424*SUM/(LB=DB)*(LPB=DPB)

7 RETURN

END
REAL FUNCTION L*4(U,V)

FUNCTION L

PURPOSE
TO COMPUTE THE INTEGRAL( EXP(-Y-V**2/Y)/Y) SUMMED OVER Y FROM
U TO INFINITY(WELL FUNCTION FOR LEAKY AQUIFERS).

DESCRIPTION OF PARAMETERS

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>U</td>
<td>R**2<em>5/4</em>T</td>
</tr>
<tr>
<td>V</td>
<td>R/2<em>SQRT(K1/(T</em>B1))</td>
</tr>
</tbody>
</table>

SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED:
DGL12, SERIES, BESK, FCT

METHOD

IN THE FOLLOWING F = EXP(-Y-V**2/Y)/Y

1. U>1, USES A GAUSSIAN-LAGUERRE QUADRATURE FORMULA TO
   EVALUATE INTEGRAL(F) FROM U TO INF,

2. V**2<=U<=1, USES THE G-L QUADRATURE TO EVALUATE INTEGRAL(F)
   FROM ONE TO INF AND A SERIES EXPANSION TO EVALUATE INTEGRAL(F)
   FROM U TO ONE,

3. U<1, V**2<=U, USES THE REPRESENTATION INTEGRAL(F) FROM U
   TO INF = 2*K0(2*V)*INTEGRAL(F) FROM V**2/U TO INF,
   EVALUATES THE ZERO ORDER MODIFIED BESSEL FUNCTION OF SECOND
   KIND WITH IBM SUBROUTINE, EVALUATES INTEGRAL BY G-L QUAD,

C

EXTERNAL FCT
REAL U, V, Z, F, VV, SERIES
COMMON /C1/ VV, Z

IF (U<1) 1,2,2
CHECKS IF U<1
C
1 Z=V*V/U
IF (Z<1) 3,4,4
CHECKS IF V**2/U < 1
C
2 Z=U
CALL DGL12(FCT,F)
L=F
C
INTEGRAL U TO INF, EVALUATED BY GAUSS-LAGUERRE QUADRATURE
GO TO 5
C
3 Z=1,
CALL DGL12(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
C
4 T=V**2.*V
CALL BESK(T,K1,0,BK,IER)
CALL DGL12(FCT,F)
L=2.*BK-F
C
2*K0(2*V)=INTEGRAL V**2/U TO INF,
C
RETURN

END

REAL FUNCTION SERIES*8(U,V)

FUNCTION SERIES

PURPOSE

C


TO EVALUATE $S(1) = 8(U)$, WHERE $S$ IS A SERIES EXPANSION OF
INTEGRAL $\int \exp{-\pi y^2 / 2} dy / y$ GIVEN BY
$F(M) = \sum, m = 0 \text{ to } \infty, (\pi y^2 / (1 + \pi y^2)) \text{ WHERE } F(M) = \log(U) \text{ IF } M = 0 \text{ AND } M = \left(\frac{\pi}{\gamma} + 1\right)$, $U = \left(\frac{V}{2} + \sqrt{\frac{V}{2} + \frac{1}{4} \gamma^2}ight) - \frac{1}{2}$, $V = \left(\frac{1}{2} \gamma \frac{u}{u^2}ight)$, ONE = HALF RADIAL DISTANCE SQUARED * SQUARE ROOT
HYD. COND. OF CONFINING BED / TRANSMISSIVITY * THICKNESS
OF CONFINING BED

SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
NONE

METHOD
SUMMATION IS TERMINATED FOR THE INNER SERIES WHEN A TERM
BECOMES LESS THAN $5,10^{-7}$ AND FOR OUTER SERIES WHEN A TERM
BECOMES LESS THAN $5,10^{-7}$

**********************************************************************
REAL*H DLOGDASS, S(2), VUM, UVU
REAL*8 TEBT, U, UM, EN, SUM1, SUM, SIGN, V, VSQ, VSQU, RMUL, TERM, TERM1
TEST=5, U=0.7
VSQ=V*V
UIM=1, EM=-1,
SUM=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)
TERM=1,
GO TO 4
3 UM=UM+1,
IF (UM, LT, 1, D=30) UM=0,
VUM=VUM+VSQU
RMUL=(UM+UM)/EM
TERM1=TERM/EM
4 SIGN=SIGN
TERM=TERM1
SUM=SUM1
EN=1
5 EN=EN+1,
TERM=TERM1*VSQ/(EN*(EN+EM))
SUM=SUM+TERM
IF (TEST, LE, DABS(RMUL*EN+TERM)) GO TO 5
C TRUNCATES INNER SERIES IF OUTER TERM*INNER TERM < $5,10^{-7}$
SUM1=SUM1+SIGN*RMUL*SUM
IF (EM, LT, 1) GO TO 1
IF (TEST, LE, DABS(SUM1)) GO TO 1
C TRUNCATES OUTER SERIES IF OUTER TERM*INNER SUM < $5,10^{-7}$
6 S(1)=SUM1
UM=UVU
SERIES=S(2)+S(1)
RETURN
END
<table>
<thead>
<tr>
<th>Purpose</th>
<th>Description of Parameters</th>
<th>Remarks</th>
<th>Subroutines and Function Subprograms Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>To compute integral of $\exp(-x)FCT(x)$, summed over $x$ from 0 to infinity.</td>
<td>$FCT$ = the name of an external double precision function subprogram used.</td>
<td>None</td>
<td>The external double precision function subprogram $FCT(x)$ must be furnished by the user.</td>
</tr>
<tr>
<td></td>
<td>$V$ = the resulting double precision integral value.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Example Subroutine DQL12

```c
REAL FUNCTION FCT#8(X)

FUNCTION FCT

PURPOSE
To compute $FCT(X)=\exp(-Z**2/(X+Z))/(X+Z)$

DESCRIPTION OF PARAMETERS
$X$ = the double precision value of $X$ for which $FCT$ is computed

SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED

NONE

METHOD
FORTRAN EVALUATION OF FUNCTION

REAL*8 X,V,Z,P,DEXP
COMMON /C/ V,Z
IF (X) 12,2
1 FCT=0.
GO TO 4
2 P=Z+V**2/(X+Z)
IF (P<0.0) 3,3,1
3 FCT=DEXP(-P)/(X+Z)
4 RETURN
END

SUBROUTINE DQL12(FCT,Y)

SUBROUTINE DQL12

PURPOSE
To compute integral of $\exp(-X)*FCT(X)$, summed over $X$ from 0 to infinity.

USAGE
CALL DQL12 (FCT,Y)
PARAMETER FCT requires an external statement

DESCRIPTION OF PARAMETERS
$FCT$ = the name of an external double precision function subprogram used.
$V$ = the resulting double precision integral value.

REMARKS
None

SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED

The external double precision function subprogram $FCT(X)$ must be furnished by the user.

METHOD
Evaluation is done by means of 12-point Gaussian-Laguerre quadrature formula, which integrates exactly when $FCT(X)$ is a polynomial up to degree 23.

For reference, see
SHAO/CHEN/FRANK, Tables of zeros and Gaussian weights of certain associated Laguerre polynomials and the related Generalized Hermite polynomials, IBM technical report TROO.1100 (March 1964), pp. 24-25.

```

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

```
DOUBLE PRECISION X,Y,FCT

C
X=3709912104446692 D2
Y=814867746742624 D=15*FCT(X)
X=2848796725019840 D2
Y=301601635035021 D=11*FCT(X)
X=2215109037439701 D2
Y=1342391030515004 D=6*FCT(X)
X=171488518746226 D2
Y=166849307540910 D=6*FCT(X)
X=130060549930635 D2
Y=836505585681980 D=5*FCT(X)
X=962131684265687 D1
Y=2032315926629994 D=3*FCT(X)
X=6844525453115177 D1
Y=2663973541865316 D=2*FCT(X)
X=459927639418348 D1
Y=2010238115463410 D=1*FCT(X)
X=2833751377743507 D1
Y=904492222116609 D=1*FCT(X)
X=1512610269776419 D1
Y=2440820113198776 D=0*FCT(X)
X=611757465151307 D0
Y=3777592756731380 D=0*FCT(X)
X=1157221173580207 D0
Y=2647313710554432 D=0*FCT(X)
RETURN
END
SUBROUTINE BESK(X,N,BK,IER)

*****************************************************************************

SUBROUTINE BESK

COMPUTE THE K BESSEL FUNCTION FOR A GIVEN ARGUMENT AND ORDER

USAGE

CALL BESK(X,N,BK,IER)

DESCRIPTION OF PARAMETERS

X  = THE ARGUMENT OF THE K BESSEL FUNCTION DESIRED
N  = THE ORDER OF THE K BESSEL FUNCTION DESIRED
BK = THE RESULTANT K BESSEL FUNCTION
IER=RESULTANT ERROR CODE WHERE
  IER=0  NO ERROR
  IER=1  N IS NEGATIVE
  IER=2  X IS ZERO OR NEGATIVE
  IER=3  X .GT. 170, MACHINE RANGE EXCEEDED
  IER=4  BK .GT. 10**70

REMARKS

N MUST BE GREATER THAN OR EQUAL TO ZERO

SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED

NONE

METHOD

COMPUTES ZERO ORDER AND FIRST ORDER BESSEL FUNCTIONS USING
SERIES APPROXIMATIONS AND THEN COMPUTES N TH ORDER FUNCTION
USING RECURRANCE RELATION.
```

(Continued on the next page)
Table 6.1.—Listing of program for partial penetration in a leaky artesian aquifer—Continued


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=1700)22,22,21
21 IER=3
RETURN
22 IER=0
IF(X=1)36,36,25
25 A=EXP(X)
B=1/X
C=SIN(T)
T=0
DO 26 N=2,12
26 T(N)=T(N-1)*X
IF(N=1)27,29,27

COMPUTE KO USING POLYNOMIAL APPROXIMATION

27 G0=EXP(125331411.566642*T(1)+.08611128*T(2)+.09139095*T(3)
2+.1344569*T(4)+.2299985*T(5)+.0792410*T(6)+.5247277*T(7)
3+.5075368*T(8)+.4262633*T(9)+.2184518*T(10)+.0680977*T(11)
4+.009189381*T(12))C
IF(N=20)20,20,29
28 BK=GO
RETURN

COMPUTE K1 USING POLYNOMIAL APPROXIMATION

29 G1=EXP(12533141+4699927*T(1)+.1499853*T(2)+.1280427*T(3)
2+.1739332*T(4)+.2687618*T(5)+.4594832*T(6)+.628381*T(7)
3+.6633295*T(8)+.5050239*T(9)+.2881304*T(10)+.0758001*T(11)
4+.01082418*T(12))C
IF(N=120)120,30,31
30 BK=G1
RETURN

FROM KO,K1 COMPUTE K USING RECURRANCE RELATION

31 DO 35 J=2,N
32 IER=4
GO TO 34
33 G0=G1
35 G1=G0
34 BK=G0
RETURN
36 B=x/2.
Table 6.1.—Listing of program for partial penetration in a leaky artesian aquifer—Continued

```
A=$772157+A\log(B)
C=B*B
IF(N=1)37,43,37

C
COMPUTE KO USING SERIES EXPANSION
37 G0=A
X2J=1,
FACT=1,
HJ=0
DO 40 J=1,6
RJ=J/FLOAT(J)
IF(X2J,LT,1,E=40) X2J=0,
PREVIOUS STATEMENT ADDED TO IBM SUBROUTINE TO CORRECT UNDERFLOW
40 GO=G0+X2J*FACT*(HJ=A)
IF(N)43,42,31
RETURN

C
COMPUTE K1 USING SERIES EXPANSION
43 X2J=8
FACT=1,
HJ=1,
G1=1./X+X2J*(,5+A=HJ)
DO 50 J=8,8
X2J=X2J+1,
FACT=FACT*RJ*RJ
HJ=HJ+RJ
50 G1=G1+X2J*FACT*(,5+(A=MJ)*FLOAT(J))
IF(N=1)31,52,31
RETURN
END
```

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

```
****************************************************************************************
PURPOSE
TO COMPUTE A TABLE OF FUNCTION VALUES FOR DRAWDOWN IN A LEAKY ARTESIAN AQUIFER IN RESPONSE TO A STEP CHANGE IN WATER LEVEL IN THE CONTROL WELL, FUNCTION VALUES ARE EXPRESSED AS A FRACTION OF DRAWDOWN IN CONTROL WELL (S/SM). REFERENCE = HANTUSH, M. S., 1959, NONSTEADY FLOW TO FLOWING WELLS IN LEAKY AQUIFERS: JOUR. GEOPHYS. RESEARCH; V, 64, NO. 8, P. 1043-1052.

INPUT DATA
1 CARD = FORMAT(2E10.5)
2 CARD = FORMAT(13F5.0)
3 SMALL = SMALLEST VALUE OF ALPHA FOR WHICH COMPUTATION IS DESIRED,
4 LARGE = LARGEST VALUE OF ALPHA FOR WHICH COMPUTATION IS DESIRED.
5 NM = 13 VALUES OF Rm/B, NON ZERO VALUES SHOULD BE GE 1 AND LT 10, FIRST ZERO (OR BLANK) WILL TERMINATE THE LIST, AT LEAST ONE NON ZERO VALUE MUST BE CODED, INPUT VALUES ARE MULTIPLIED BY POWER OF TEN DETERMINED BY 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 NH = 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 (UR BLANK) VALUE WILL TERMINATE COMPUTATION
C
C METHOD
C EVALUATES EQ. 13 OF HANTUSH, EVALUATION OF BESSEL FUNCTIONS
C BY SUBROUTINES BESK AND BESY AND FUNCTION JO, EVALUATES
C INTEGRAL BY SUM, IM1 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
C REMARKS
C SMALLEST RW/B GE .01/SQRT(SMALLEST ALPHA)
C SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C BESK,BESY,JO
C
C******************************************************************************
C REAL*8 SUM1,SUM2
C REAL*4 KBP,K0B,J0,JOU,Y(8000),J(8000),F(8000),FT(8000),
C 1 FB(8000),RDAT(9),TDAT(6),BOAT(13),ARRAY(25,9,13),S(13),T(25)
C DATA FT/8000*0.0/,FB/8000*0.0/,DATA HDAT//*1,DATA ARHY/0.0/,TDAT/1,1,5,2,3,5,7,/
C IRD=5 IRT=6
C
C READ (IRD,24) TSMALL,TLARGE
C READ (IRD,23) BOAT
C READ (IRD,22) RW,RDAT
C IBEGIN=ALOG10(TSMALL)
C IEND=ALOG10(TLARGE)*99999
C IF ((IBEGIN/2)2,LT,IBEGIN) IBEGIN=IBEGIN-1
C ISPAN=IEND-IBEGIN
C MLIMIT=(ISPAN+1)/2
C
C COMPUTES INITIAL DELTA U (DU) = .001/SQRT(SMALLEST ALPHA)
C DU=.001/SQRT(TDAT(1)*10.**IBEGIN)
C
C EXPONENT (JBEGIN) OF SMALLEST RW/B IS COMPUTED FROM EXPONENT
C (JBEGIN) OF SMALLEST ALPHA
C JBEGIN=JBEGIN/2=2
C DU 1 I=1,13
C IF (BOAT(1),EQ,0.) GO TO 2
C 1 CONTINUE
C NB=13
C GO TO 3
C 2 NB=1
C 3 CONTINUE
C DO 4 I=1,9
C IF (RDAT(1),EQ,0.) GO TO 5
C 4 RDAT(1)=RDAT(1)/RW
C NR=9
C GO TO 6
C 5 NR=1
C 6 DO 21 NR=1,MLIMIT
C NUM=8000
C START=DU/2,
C U=START
C DO 7 I=1,NUM
C U+DU
C CALL BESY(U,0,Y(I),IDUMY)
C 7 J(I)=JO(U)
C DO 19 I=1,NR
```

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

```
RHO0=DAT(IR)
UMSTART
DO 8 I=1,NUM
U*U+DU
CALL BESY(RHO*U,0,YOPU,IDUMY)
JOPU=JO(RHO*U)
JUUM(J)
YUUM(Y(J))
8 F(I)=(JOPU*YOU*YOPU*JUU)/(JOU*JOU+YOU*YOU)
DO 19 IT=1,25
INDEX=(IT=1)/6
IEXP=IBEGIN+INDEX
II=I-INDEX+6
TAU=DAT(II)*10.**IEXP
T(IT)=TAU
UMSTART
NUMT=NUM
DO 9 I=1,NUMT
U*U+DU
FTEST=F(I)
IF (ABS(FTEST),LT,1,E=30) GU 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
INDEX=(IB=1)/NB
JEXP=JBEGIN+INDEX
JJ=IB-JINDEX+NB
BETA=BDAT(JJ)*10.**JEXP
B(IB)=BETA
UMSTART
BSQ=BETA*BETA
NUM=RNUM
DO 12 I=1,NUM
U*U+DU
FTEST=F(I)
IF (ABS(FTEST),LT,1,E=30) GO TO 13
12 FT(I)=FTEST/(U*BSQ/U)
GO TO 14
13 NUM=I+1
FB(I)=0.
14 SUM1=0.
SUM2=0.
DO 15 I=1,NUM,2
SUM1=SUM1+FB(I)
15 SUM2=SUM2+FB(I+1)
XINT=(SUM1+SUM2)*DU
CALL BESK(KMU*BETA,0,KUBP,IDUMY)
CALL BESK(BETA,0,KOB,IDUMY)
RATIO0.
IF (KUBP,GT,0) RATIO=KOB/KPB
XTEST=TAU*BSQ
IF (XTEST+30.) 16,17,17
16 XPT=0.
GO TO 18
17 XPT=EXP(XTEST)
18 Z=XPT(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

```plaintext
      DO 20 K=1,NH
      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
      JBEGIN=JBEGIN+2

      C DELTA U (DU) IS DIVIDED BY 10 EACH TIME THROUGH THE LOOP
      21 DU=1.0DU
      STOP
      C
      22 FORMAT (10F8,2)
      23 FORMAT (13F5,0)
      24 FORMAT (2E10,5)
      25 FORMAT ('(ALPHA,R/Rw,Rw/B) ', R/Rw,1,9X,11 RW/B'/11
      13x,'ALPHA l',14E9.2))
      26 FORMAT (' ',10F9.3,13F9.3)
      END

      REAL FUNCTION JOU(X)
      PURPOSE
      TO COMPUTE THE ZER0 UNDER J BESSEL FUNCTION FOR A GIVEN
      ARGUMENT.
      USAGE
      JOU(X)
      DESCRIPTION OF PARAMETER
      X = REAL*4, ARGUMENT OF JO BESSEL FUNCTION DESIRED.
      SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
      NONE.
      METHOD
      POLYNOMIAL APPROXIMATION FOR X<4 AND ASYMPTOTIC SERIES fur
      X GE 4, THE POLYNOMIAL APPROXIMATION IS THE FIRST 10 TERMS OF
      THE POWER SERIES FOR JO(X) (MILLER, K. S., 1957.
      ENGINEERING MATHEMATICS; RINEHART AND CO., INC., NEW YORK,
      P. 120), THE ASYMPTOTIC EXPANSION OF JO(X) IS GIVEN ON P. 82
      OF BUCKMAN, FRANK, 1958, INTRODUCTION TO BESSEL FUNCTIONS;
      DOVER PUBLICATIONS INC., NEW YORK, THE TERMS P('/A*PO1') AND
      Q ('B*QQ1) OF THE ASYMPTOTIC EXPANSION ARE COMPUTED BY AN
      ALGORITHM FROM IBM SUBROUTINE BESY.
```

```
      IF (X<4.) 1,3,3
  1 A=X*X/4.,
     B=1.,
     DO 2 I=1,10
     C=11./I
     2 B=B*A/(C*C))
     JOU=JOU+B
     GO TO 4
  3 T=4./X
     T2=T*T
     P0=(0., 0.000037043*10.2, 0.000173565)*T2+(0.000487613)*T2+(0.00017343)*
     1T2+(0.001757622)*T2+(0.3989423)
     Q0=(0., 0.000032312*10.2, 0.000142078)*T2+(0.000342468)*T2+(0.000669791)
     1T2+(0.000564324)*T2+(0.124694)
     A2=E0/25RT(X)
```
Table 7.2. Listing of program for constant drawdown in a well in an infinite leaky aquifer—Continued

```
R=4+1
C=7.753982
JO=AJ*PO+COS(C)-8*Q0*SIN(C)
4 RETURN
END
SUBROUTINE BESY(X,N,BY,IER)

SUBROUTINE BESY
PURPOSE
COMPUTE THE Y BESSEL FUNCTION FOR A GIVEN ARGUMENT AND ORDER
USAGE
CALL BESY(X,N,BY,IER)
DESCRIPTION OF PARAMETERS
X = THE ARGUMENT OF THE Y BESSEL FUNCTION DESIRED
N = THE ORDER OF THE Y BESSEL FUNCTION DESIRED
BY = THE RESULTANT Y BESSEL FUNCTION
IER = RESULTANT ERROR CODE WHERE
IER = 0 NO ERROR
IER = 1 N IS NEGATIVE
IER = 2 X IS NEGATIVE OR ZERO
IER = 3 BY HAS EXCEEDED MAGNITUDE OF 10**70
REMARKS
VERY SMALL VALUES OF X MAY CAUSE THE RANGE OF THE LIBRARY
FUNCTION ALOG TO BE EXCEEDED
X MUST BE GREATER THAN ZERO
N MUST BE GREATER THAN OR EQUAL TO ZERO
SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
NONE
METHOD
RECURRENCE RELATION AND POLYNOMIAL APPROXIMATION TECHNIQUE
AS DESCRIBED BY A.J.M.HITCHCOCK, 'POLYNOMIAL APPROXIMATIONS
TO BESSEL FUNCTIONS OF ORDER ZERO AND ONE AND TO RELATED
FUNCTIONS!', M.T.A.C., V.11, 1957, P.88-89, AND G.N. WATSON,
'I A TREATISE ON THE THEORY OF BESSEL FUNCTIONS', CAMBRIDGE
UNIVERSITY PRESS, 1958, P. 62

CHECK FOR ERRORS IN N AND X

10 IER=0
IF(N)180,10,10
10 IER=0
IF(X)190,190,20
BRANCH IF X LESS THAN OR EQUAL 4
20 IF(X<4)40,40,30
COMPUTE Y0 AND Y1 FOR X GREATER THAN 4
30 T1=X0,0
T2=T1*T1
PO=((1.000037043*T2+0.000173565)*T2+0.000487613)*T2
```

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

```plaintext
1 + 0.0017343*F2 = 0.0017530621*F2 + 3.398423  
Q0 = (((((1.0000323121*F2 = 0.000142076)*F2 + 0.0000342468)*F2  
1 = 0.0000697911*F2 + 0.0004546324  
P1 = (((((1.00004214*F2 = 0.000200920)*F2 + 0.000580759)*F2  
1 = 0.00223203*F2 + 0.02921826  
Q1 = (((((1.000036594*F2 + 0.0001622)*F2 + 0.000398708)*F2  
1 + 0.00164741*F2 + 0.0003940040)*F2 + 0.03740084  
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)  
GOTO 90  

COMPUTE Y0 AND Y1 FOR X LESS THAN OR EQUAL TO 4

40 XX = X/2  
X2 = XX*XX  
T = ALOG(XX) + 5772157  
SUM = 0  
TERM = T  
Y0 = SUM  
DO 70 L = 1, 15  
IF (L = 1) SUM = SUM  
TERM = TERM + T  
GO TO 70  
50 SUM = SUM + 1/FLOAT(L = 1)  
60 FL = L  
TSM = SUM  
IF (ABST(TERM) + 1.0E-40) TERM = 0  
TERM = TERM + ((X2)/(FL + 2.0)*(1.0 - 1.0/(FL + 8.0)))  
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  
TSM = SUM  
IF (ABST(TERM) + 1.0E-40) TERM = 0  
TERM = TERM + ((X2)/(FL + 2.0)*(1.0 - 1.0/(FL + 8.0)))  
80 Y1 = Y1 + TERM  
P1Z = 5368198  
Y0 = Y0 + P1Z  
Y1 = P1Z + X + P1Z + Y1  
C CHECK IF ONLY Y0 OR Y1 IS DESIRED  
90 IF (N = 1) 100, 100, 130  
C RETURN EITHER Y0 OR Y1 AS REQUIRED  
100 IF (N = 1) 110, 120, 110  
110 BY = Y1  
GO TO 170  
120 BY = Y0  
GO TO 170  
C PERFORM RECURRANCE OPERATIONS TO FIND VY*(X)  
130 Y0 = 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
141 IER = 3
RETURN
145 K = 1
IF (K = N) 150, 160, 150
150 Y = Y
155 GOC TO 140
160 B = Y
170 RETURN
180 IER = 1
RETURN
190 IER = 2
RETURN
SUBROUTINE BESK(X, N, BK, IER)
                        BESY1160
                        BESY1190
                        BESY1200
                        BESY1210
                        BESY1220
                        BESY1230
                        BESY1240
                        BESY1250
                        BESY1260
                        BESY1270
                        BESY1280
                        BESY1290
                        BESY1300
                        BESY1310
                        BESY1320
                        BESY1330
                        BESY1340
                        BESK 410
                        BESK 10
                        BESK 20
                        BESK 30
                        BESK 40
                        BESK 50
                        BESK 110
                        BESK 120
                        BESK 130
                        BESK 140
                        BESK 150
                        BESK 160
                        BESK 170
                        BESK 180
                        BESK 190
                        BESK 200
                        BESK 210
                        BESK 220
                        BESK 230
                        BESK 240
                        BESK 250
                        BESK 260
                        BESK 270
                        BESK 280
                        BESK 290
                        BESK 300
                        BESK 310
                        BESK 320
                        BESK 330
                        BESK 340
                        BESK 350
                        BESK 360
                        BESK 370
                        BESK 380
                        BESK 390
                        BESK 400
                        BESK 410
                        BESK 420
                        BESK 430
                        BESK 440
                        BESK 450
```

**SUBROUTINE BESK**

**USAGE**

CALL BESK(X, N, BK, IER)

**DESCRIPTION OF PARAMETERS**

- **X** — THE ARGUMENT OF THE K BESSEL FUNCTION DESIRED
- **N** — THE ORDER OF THE K BESSEL FUNCTION DESIRED
- **BK** — THE RESULTANT K BESSEL FUNCTION
- **IER** — RESULTANT ERROR CODE WHERE
  - IER = 0 NO ERROR
  - IER = 1 N IS NEGATIVE
  - IER = 2 X IS ZERO OR NEGATIVE
  - IER = 3 X GT, 170, MACHINE RANGE EXCEEDED
  - IER = 4 BK GT, 10**70

**REMARKS**

- N MUST BE GREATER THAN OR EQUAL TO ZERO
- SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
  - NONE

**METHOD**

- COMPUTES ZERO ORDER AND FIRST ORDER BESSEL FUNCTIONS USING
  - SERIES APPROXIMATIONS AND THEN COMPUTES N TH ORDER FUNCTION
- USING RECURRENCE RELATION
- RECURRENCE RELATION AND POLYNOMIAL APPROXIMATION TECHNIQUE
- AS DESCRIBED BY A. J. M. MITCHELL, POLYNOMIAL APPROXIMATIONS
- TO BESSEL FUNCTIONS OF ORDER ZERO AND ONE AND TO RELATED
  - FUNCTIONS', M. T. A. C., V. 11, 1957, PP. 86-98, AND G. N. WATSON,
- 'A TREATISE ON THE THEORY OF BESSEL FUNCTIONS', CAMBRIDGE
- UNIVERSITY PRESS, 1958, P. 82
RETURN
11 IF(X112,12,20
12 IER=2
RETURN
20 IF(X1170,0)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 COMPUTE KO USING POLYNOMIAL APPROXIMATION
27 G0=A*(1.2533141+.1566642*T(1)+1.0811128*T(2)+.0913995*T(3)
2+.1349568*T(4)+.2299850*T(5)+.3792410*T(6)+.5247277*T(7)
3+.5575366*T(8)+.4262633*T(9)+.2184518*T(10)+.06680977*T(11)
4+.001809383*T(12))X C
IF(N)20,28,29
28 B=G0
RETURN
C COMPUTE K1 USING POLYNOMIAL APPROXIMATION
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+.5575366*T(8)+.4262633*T(9)+.2184518*T(10)+.06680977*T(11)
4+.001809383*T(12))X C
IF(N=1)30,31,30
30 B=G1
RETURN
C FROM KO,K1 COMPUTE KN USING RECURRENCE RELATION
31 DO 35 J=2,N
GJ=G1*(FLOAT(J)=1.)*G1/X+G0
IF(GJ=1.0)33,33,32
32 IER=4
GO TO 34
33 G0=G1
35 G1=GJ
34 B=GJ
RETURN
36 B=X/2.
A=.5772157+ALOG(8)
B=8
IF(N=1)37,43,37
C COMPUTE KO USING SERIES EXPANSION
37 GO=A
X2J=1.
FACT=1.
HM=0
DO 40 J=1,6
RJ=1/FLOAT(J)
IF(X2J>LT,1,E40)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
C PROBLEM ON WATFOR COMPILER
X2JX2JXC
FACT1FACT*RHJ
HJMJRHJ
40 G0=G0+X2JXFACT*(HJ=A)
IF(N)43,42,43
42 BK=G0
RETURN
C COMPUTE K1 USING SERIES EXPANSION
C
43 X2JX8
FACT1,
HJ1,
G1=X*X2J*(S*A=HJ)
DO 50 J=2,8
X2JX2JX
RHJ1/FLOAT(J)
FACT1FACT*RHJ
HJMJRHJ
50 G1=G1+X2JXFACT*(S+(A=HJ)*FLOAT(J))
IF(N)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,HU) FOR RHO > 1 - FAR
A WELL OF LARGE DIAMETER, WATER RESOURCES RESEARCH, V, 3,
NO 1, P, 241-244, FAR
C PROGRAM BY S,S,PAPADUPULUS,
C INPUT DATA = ONE OR MORE GROUPS, EACH GROUP CODED AS FOLLOWS
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
C 1 CARD = FORMAT(16E5,0)
**TABLE 8.2.—Listing of programs for constant discharge from a fully penetrating well of finite diameter—Continued**

<table>
<thead>
<tr>
<th>Line</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>WRITE (6,17) ALPHA, RHO</td>
<td>33</td>
</tr>
<tr>
<td>3</td>
<td>WRITE (6,18)</td>
<td>34</td>
</tr>
<tr>
<td>4</td>
<td>READ (5,19) U</td>
<td>35</td>
</tr>
<tr>
<td>5</td>
<td>DO 14 II=1,16</td>
<td>36</td>
</tr>
<tr>
<td></td>
<td>IF (U(II)) 1,1,4</td>
<td>37</td>
</tr>
<tr>
<td>6</td>
<td>WRITE (6,20) XPK, U</td>
<td>38</td>
</tr>
<tr>
<td>8</td>
<td>EPS=0.000001</td>
<td>39</td>
</tr>
<tr>
<td></td>
<td>CALL SIMPS(0,0,XPK, EPS, HBAR, SUM, DEL, EXBSL1)</td>
<td>40</td>
</tr>
<tr>
<td></td>
<td>XM1=((3,14159265<em>7.0)/3,0</em>(RHO=1.0)) + 1,5<em>6</em>RHO/2</td>
<td>41</td>
</tr>
<tr>
<td></td>
<td>DX1=X1=1,0E=6*RHO</td>
<td>42</td>
</tr>
<tr>
<td></td>
<td>DXN=(2,0<em>3,14159265</em>RHO)/5.5*(RHO=1.0)</td>
<td>43</td>
</tr>
<tr>
<td></td>
<td>DLX3,14159265*RHO/(RHO=1.0)</td>
<td>44</td>
</tr>
<tr>
<td></td>
<td>CALL ROOTS(XM1,DX1, RT1, EXBSL1)</td>
<td>45</td>
</tr>
<tr>
<td></td>
<td>HBAR=0.007 RT1=XP</td>
<td>46</td>
</tr>
<tr>
<td></td>
<td>CALL SIMPS(XPK, RT1, EPS, HBAR, TRM1, ERR1, EXBSL1)</td>
<td>47</td>
</tr>
<tr>
<td></td>
<td>SUM=SUM+TRM1</td>
<td>48</td>
</tr>
<tr>
<td></td>
<td>X1=RT1</td>
<td>49</td>
</tr>
<tr>
<td></td>
<td>I=1</td>
<td>50</td>
</tr>
<tr>
<td></td>
<td>CALL ROOTS(XM, DXN, X2, EXBSL1)</td>
<td>51</td>
</tr>
<tr>
<td></td>
<td>HBAR=0.007*(X2=X1)</td>
<td>52</td>
</tr>
<tr>
<td></td>
<td>CALL SIMPS(X1, X2, EPS, HBAR, TRM, ERR, EXBSL1)</td>
<td>53</td>
</tr>
<tr>
<td></td>
<td>V(I,J)=ABS(TRM)</td>
<td>54</td>
</tr>
<tr>
<td></td>
<td>DEL=DEL+ERR</td>
<td>55</td>
</tr>
<tr>
<td></td>
<td>I=I+1</td>
<td>56</td>
</tr>
<tr>
<td></td>
<td>IF (I=40) 10,10,11</td>
<td>57</td>
</tr>
<tr>
<td></td>
<td>X1=X2</td>
<td>58</td>
</tr>
<tr>
<td></td>
<td>GO TO 9</td>
<td>59</td>
</tr>
<tr>
<td></td>
<td>E=0.0</td>
<td>60</td>
</tr>
<tr>
<td></td>
<td>DO 12 K=2,40</td>
<td>61</td>
</tr>
<tr>
<td></td>
<td>M=I+1</td>
<td>62</td>
</tr>
<tr>
<td></td>
<td>DO 12 J=1, M</td>
<td>63</td>
</tr>
<tr>
<td></td>
<td>V(K,J)=V(K,J-1)+V(K-1,J)</td>
<td>64</td>
</tr>
<tr>
<td></td>
<td>DO 13 N=1,4</td>
<td>65</td>
</tr>
<tr>
<td></td>
<td>L=N=1</td>
<td>66</td>
</tr>
<tr>
<td></td>
<td>DEL=(0.5)<strong>L</strong>(L*(N+1))</td>
<td>67</td>
</tr>
<tr>
<td></td>
<td>EST=EST+(0.5)*DELV</td>
<td>68</td>
</tr>
<tr>
<td></td>
<td>SUM=SUM+EST</td>
<td>69</td>
</tr>
<tr>
<td></td>
<td>PUAR=Q+4<em>RHO</em>SUM/3*14159265</td>
<td>70</td>
</tr>
<tr>
<td></td>
<td>WRITE (6,22) U(I,II), SUM, DEL, PUAR</td>
<td>71</td>
</tr>
<tr>
<td></td>
<td>CONTINUE</td>
<td>72</td>
</tr>
<tr>
<td></td>
<td>GO TO 1</td>
<td>73</td>
</tr>
<tr>
<td></td>
<td>STUP</td>
<td>74</td>
</tr>
</tbody>
</table>

**C**

<table>
<thead>
<tr>
<th>Line</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>FORMAT (2E10.5)</td>
<td>75</td>
</tr>
<tr>
<td>17</td>
<td>FORMAT ('111', IF(U, ALPHA, RHO))</td>
<td>76</td>
</tr>
<tr>
<td>18</td>
<td>FORMAT (1H0,12X, 1HU, 16X, 9HINTEGRAL, 9X, 14HINTEGRAL ERRUR, 6X, 14HF(U, 1H, 1ALPHA, 1HU))/1M</td>
<td>77</td>
</tr>
<tr>
<td>19</td>
<td>FORMAT (16E5, 0)</td>
<td>78</td>
</tr>
</tbody>
</table>
Table 8.2.—Listing of programs for constant discharge from a fully penetrating well of finite diameter—Continued

```fortran
20 FORMAT (5H XP*,E15.8,3X,16HT00 SMALL FOR U*,E15.8) FAR 93
21 FORMAT (5H XP*,E15.8,3X,16HT00 LARGE FOR U*,E15.8) FAR 94
22 FORMAT (1H,1PE20.8) FAR 95
END FAR 96

FUNCTION EXS1L1(X) EB1 1
C******************************************************E1B1 2
C PURPOSE EB1 3
C COMPUTES VALUES OF THE INTEGRAND FOR F(U,ALPHA,RHO) EB1 4
C DESCRIPTION OF PARAMETER EB1 5
X= REAL= ARGUMENT OF INTEGRAND EB1 6
C
C******************************************************E1B1 7
C
COMMON/PBLK/A,R,R EB1 8

IF (X) 1,2 IF 9
1 EXS1L1=X EB1 9
GO TO 8 EB1 10
2 X*R IF 10
3 IF (X*1.E7) 4,4,3 EB1 11
4 FNU=A*COS(X*1.0)-A*SIN(X*1.0)) EB1 12
DEN=A*SQRT(R)*A*A EB1 13
EXS1L1=FNU/DE EB1 14
GO TO 8 EB1 15
5 Y*_X*X IF 16
6 EXP(Y=(1.0=0.5)*(1.0/6.0)*Y*(1.0/24.0))) EB1 17
GO TO 7 EB1 18
7 CALL JY0(X,JY0) EB1 19
CALL JY1(X,JY1) EB1 20
AW=X*Y0=AW*Y1 EB1 21
BQ=X*J0=AW*J1 EB1 22
CALL JY0(X,BJ0,BY0) EB1 23
FNUM=EXPO*(AW*BJ0=AW*BY0) EB1 24
DEN=x*x*(AW*BJ0=AW*BY0) EB1 25
EXS1L1=FNUM/DEN EB1 26
8 RETURN EB1 27
END EB1 28

SUBROUTINE ROOTS(XM,DX,ROOT,F) R00 1
C******************************************************R00 2
C PURPOSE R00 3
C SEARCHES FOR ROOT OF F IN THE INTERVAL XM+DX TO XM+DX, R00 4
C DESCRIPTION OF PARAMETERS = ALL REAL R00 5
XM = CENTER OF INTERVAL SEARCHED, R00 6
DX = HALF WIDTH OF INTERVAL SEARCHED, R00 7
ROOT = RETURNED ROOT LOCATION, R00 8
F = FUNCTION REFERENCE, R00 9

XL=XM+DX R00 10
XR=XM+DX R00 11
YL=F(XL) R00 12
YR=F(XR) R00 13
EP=0.000001*AABS(YL) R00 14
DO 9 IF=1,200 R00 15
YM=F(XM) R00 16
UP=AABS(YM) R00 17
IF (UP, LT, EP) GO TO 1 R00 18
IF (YM) 2,1,2 R00 19
9 CONTINUE R00 20
```

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

<table>
<thead>
<tr>
<th>Line</th>
<th>Code</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ROU=XM</td>
<td>START program</td>
</tr>
<tr>
<td>2</td>
<td>IF (YM*YL) 7,3,4</td>
<td>Check conditions</td>
</tr>
<tr>
<td>3</td>
<td>ROU=XL</td>
<td>Go to 10</td>
</tr>
<tr>
<td>4</td>
<td>IF (YM*YR) 8,5,6</td>
<td>Check conditions</td>
</tr>
<tr>
<td>5</td>
<td>ROU=XR</td>
<td>Go to 10</td>
</tr>
<tr>
<td>6</td>
<td>WRITE (6,11) XL, XR</td>
<td>Output data</td>
</tr>
<tr>
<td>7</td>
<td>STOP</td>
<td>End program</td>
</tr>
<tr>
<td>8</td>
<td>IF (YM*YR) 9,12,10</td>
<td>Check conditions</td>
</tr>
<tr>
<td>9</td>
<td>WRITE (6,11) XM, XR</td>
<td>Output data</td>
</tr>
<tr>
<td>10</td>
<td>RETURN</td>
<td>Return to calling routine</td>
</tr>
</tbody>
</table>

**Purpose:***

Subroutine APEKE(Ex88l)

- **Purpose:** Gets first approximation to peak position

```fortran
COMMON XPK, YPK
XPK=0.0
YPK=0.0
DO 1 L=1,17
  XM=10.0**(1.09)
  Y=EX88L(X)
  IF (Y=YPK) 3,3,1
1 XPK=X
  YPK=Y
2 CONTINUE
3 RETURN
END
```

Subroutine PEAK(Ex88l)

- **Purpose:** Attempts to find position of maximum for integrand

```fortran
COMMON XPK, YPK
YPK=EX88L(XPK)
DO 13 I=1,200
  DX=0.01*XPK
  XM=XM+DX
  YL=EX88L(XL)
  XM=XM+DX
  YR=EX88L(XR)
  DEN=YR+YL-YPK
  IF (DEN) 1,9,1
1 XM=XM+0.5*(YR+YL)*DX/DEN
2 IF (X) 3,4,4
```

Note: The code snippet above is incomplete and may require additional context or correction for proper execution.
Table 8.2.—Listing of programs for constant discharge from a fully penetrating well of finite diameter—Continued

```
3  X=0.0  X
4  Y=EXBSL(X)  PEA 20
   IF (YH=Y) 6,6,5  PEA 21
5  Y=YH  PEA 22
   X=XH  PEA 23
6  IF (YL=Y) 8,8,7  PEA 24
7  Y=YL  PEA 25
   X=XL  PEA 26
8  IF (Y=YPK) 14,14,12  PEA 27
9  IF (Y=YPK) 11,10,10  PEA 28
10 X=XPK+DX+DX  PEA 29
GO TO 2  PEA 30
11 X=XPK+DX+DX  PEA 31
GO TO 2  PEA 32
12 YPK=Y  PEA 33
   XPK=XPK  PEA 34
13 CONTINUE  PEA 35
   XPK=XPK  PEA 36
14 RETURN  PEA 37
   END  PEA 38*
```

SUBROUTINE SIMPS(Q,R,EPs,HBAR,AREA,DEL,F)

**Purpose**

To determine the integral of a function, \( f \), from \( Q \) to \( R \), using Simpson’s rule.

**Description of Parameters**

- **Q** - Lower limit of integral
- **R** - Upper limit of integral
- **EPs** - Desired accuracy
- **HBAR** - Minimum division of the interval
- **AREA** - Computed value of integral between \( Q \) and \( R \)
- **DEL** - Computed estimate of error
- \( f \) - The integrand (function reference)

**Method**

Uses Simpson’s rule to compute a sum approximating the integral. Uses an initial \( h=(R-Q)/2 \), computes a sequence of sums by halving \( h \) each time, computes estimate of error (\( \text{DEL} \)) as \( (\text{PREVIOUS SUM}-\text{CURRENT SUM})/15 \), computation stops when \( |\text{ABS(DEL)}|<\text{ABS(EPs*CURRENT SUM)} \), if \( \text{HBAR} \leq 0 \), then \( \text{HBAR}=0.007*(R-Q) \).

```
   **H=Q**  SIM  1
   IF (H) 1,1,2  SIM  2
   1  AREA=0.0  SIM  3
   DEL=0.0  SIM  4
   GO TO 10  SIM  5
   C R MUST BE GREATER THAN Q  SIM  6
   2  SP=1.0E-35  SIM  7
   SIM  8
   SIM  9
   SIM 10
   SIM 11
   SIM 12
   HBAR=0.007*H  SIM 13
   3  HBAR=0.007*H  SIM 14
   4  S2=0.0  SIM 15
   X=W+0.5*H  SIM 16
   5  S2=S2+4.0*F(X)  SIM 17
   X=X+H  SIM 18
   IF (X=R) 5,5,6  SIM 19
   6  SC=(S1+S2+S3)*H*0.1666667  SIM 20
```
TABLE 8.2.—Listing of programs for constant discharge from a fully penetrating well of finite diameter—Continued

```
DEL=0.06666667*(SP=SC)
IF (ABS(DEL)=ABS(EPS*SC)) 7,0,8
7 AREA=DEL
GO TO 10
8 33=3+0.5*32
H=0.5*H
IF (H=0.5BAR) 7,9,9
9 SP=SC
GO TO 4
10 RETURN
END

SUBROUTINE JYO(X,J0,YO)

C***********************************************************************
C PURPOSE
C COMPUTES BESSEL FUNCTIONS OF THE FIRST AND SECOND KIND,
C FOR POSITIVE ARGUMENTS, SEE NBS AMS 55, P. 369-370.
C DESCRIPTION OF PARAMETERS — ALL REAL
X = ARGUMENT, MUST BE > 0
J0 = RETURNED FUNCTION VALUE, J0(X)
Y0 = RETURNED FUNCTION VALUE, Y0(X)
C********************************************************************************
REAL J0, Y0
1 IF (X=3.0) 1,2,3
2 IF (X=4.42) 1,2,3
2 Z=X*(0.33333333*X)**2
J0=0.0,1.0*Z*(2.2499997=Z*(1.2656206=Z*(0.3163866=Z*(0.0444479=Z*(0.0000021=Z)))
Y0=0.63661977*ALOG(0.5*X)*J0*0.36787961*Z*(0.6059366=Z*(0.7435036=Z)
14=Z*(0.2530117=Z*(0.4261214=Z*(0.00024846*Z)))
RETURN
3 Z=X=0.0/X
F=0.7978456-Z*(0.0,77E-6+Z*(0.005327+Z*(0.0000912=Z*(0.0137237=Z)
1*0.00072805=0.00014476*Z)))
P=0.78539168*Z*(0.0166397+Z*(0.00003954=Z*(0.00262573=Z*(0.000541=Z)
125=Z*(0.00029333=0.0013558*Z)))
G=SRT(1.0/X)
J0=J0=F*G*(X=P)
Y0=J0*F*SIN((X=P)
4 RETURN
END

SUBROUTINE JY1(X,J1,Y1)

C***********************************************************************
C PURPOSE
C COMPUTES BESSEL FUNCTIONS OF THE FIRST AND SECOND KIND,
C FOR POSITIVE ARGUMENTS, SEE NBS AMS 55, P. 370.
C DESCRIPTION OF PARAMETERS — ALL REAL
X = ARGUMENT, MUST BE > 0
J1 = RETURNED FUNCTION VALUE, J1(X)
Y1 = RETURNED FUNCTION VALUE, Y1(X)
C********************************************************************************
REAL J1, Y1
1 IF (X=3.0) 1,2,3
2 IF (X=4.42) 1,2,3
2 Z=X*(0.33333333*X)**2
```

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

<table>
<thead>
<tr>
<th>Statement</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| J1* = (0.5*Z*(0.56249985*Z*(0.21093573*Z*(0.03954289*Z*(0.0433319=Y1+18)
| 17*(0.00331761=0.00000109+Z))))) | Computes function values of F(UW,ALPHA) |
| Y10.0,63661977+ALOG0.5*X)*J1+ =0.6366198+Z*(0.2212091+Z*2,1682709Y1 | |
| 1=Z*(1.0,3164627*Z*(0.3123591*Z*(0.0409976+0.0027873*Z))))/X | |
| RETURN | |
| J1=Y1+21 | |
| 3 Z=5.0/X | |
| F=0.7978856*Z*(0.156E+6*Z*(0.01659667*Z*(0.00017105*Z*(0.00249511+Y1 | |
| 1*Z*(0.0013653=0.00020033+Z)))) | |
| P=0.76519816*Z*(0.12499612+Z*(0.000565*Z*(0.00637879*Z*(0.0007434+Y1 | |
| 16*Z*(0.00079824+0.00029166+Z)))) | |
| Q=SQRT(1.0/X) | |
| J1=Q*F*8IN(1=X=P) | |
| Y1=Q*F*COS(1=X=P) | |
| RETURN | |
| END | |

C**T********************~~~~~~~~~~~~~~~~~*~*---------~~~~~~~~~*~*-------*~*~ FuA 1 |
C**T********************~~~~~~~~~~~~~~~~~*~*---------~~~~~~~~~*~*-------*~*~ FuA 2 |
C**T********************~~~~~~~~~~~~~~~~~*~*---------~~~~~~~~~*~*-------*~*~ FuA 3 |
C**T********************~~~~~~~~~~~~~~~~~*~*---------~~~~~~~~~*~*-------*~*~ FuA 4 |
C**T********************~~~~~~~~~~~~~~~~~*~*---------~~~~~~~~~*~*-------*~*~ FuA 5 |
C**T********************~~~~~~~~~~~~~~~~~*~*---------~~~~~~~~~*~*-------*~*~ FuA 6 |
C**T********************~~~~~~~~~~~~~~~~~*~*---------~~~~~~~~~*~*-------*~*~ FuA 7 |
C**T********************~~~~~~~~~~~~~~~~~*~*---------~~~~~~~~~*~*-------*~*~ FuA 8 |
C**T********************~~~~~~~~~~~~~~~~~*~*---------~~~~~~~~~*~*-------*~*~ FuA 9 |
C**T********************~~~~~~~~~~~~~~~~~*~*---------~~~~~~~~~*~*-------*~*~ FuA 10 |
C**T********************~~~~~~~~~~~~~~~~~*~*---------~~~~~~~~~*~*-------*~*~ FuA 11 |
C**T********************~~~~~~~~~~~~~~~~~*~*---------~~~~~~~~~*~*-------*~*~ FuA 12 |
C**T********************~~~~~~~~~~~~~~~~~*~*---------~~~~~~~~~*~*-------*~*~ FuA 13 |
C**T********************~~~~~~~~~~~~~~~~~*~*---------~~~~~~~~~*~*-------*~*~ FuA 14 |
C**T********************~~~~~~~~~~~~~~~~~*~*---------~~~~~~~~~*~*-------*~*~ FuA 15 |
C**T********************~~~~~~~~~~~~~~~~~*~*---------~~~~~~~~~*~*-------*~*~ FuA 16 |
C**T********************~~~~~~~~~~~~~~~~~*~*---------~~~~~~~~~*~*-------*~*~ FuA 17 |
C**T********************~~~~~~~~~~~~~~~~~*~*---------~~~~~~~~~*~*-------*~*~ FuA 18 |
C**T********************~~~~~~~~~~~~~~~~~*~*---------~~~~~~~~~*~*-------*~*~ FuA 19 |
C**T********************~~~~~~~~~~~~~~~~~*~*---------~~~~~~~~~*~*-------*~*~ FuA 20 |
C**T********************~~~~~~~~~~~~~~~~~*~*---------~~~~~~~~~*~*-------*~*~ FuA 21 |
C**T********************~~~~~~~~~~~~~~~~~*~*---------~~~~~~~~~*~*-------*~*~ FuA 22 |
C**T********************~~~~~~~~~~~~~~~~~*~*---------~~~~~~~~~*~*-------*~*~ FuA 23 |
C**T********************~~~~~~~~~~~~~~~~~*~*---------~~~~~~~~~*~*-------*~*~ FuA 24 |
C**T********************~~~~~~~~~~~~~~~~~*~*---------~~~~~~~~~*~*-------*~*~ FuA 25 |
C**T********************~~~~~~~~~~~~~~~~~*~*---------~~~~~~~~~*~*-------*~*~ FuA 26 |
C**T********************~~~~~~~~~~~~~~~~~*~*---------~~~~~~~~~*~*-------*~*~ FuA 27 |
C**T********************~~~~~~~~~~~~~~~~~*~*---------~~~~~~~~~*~*-------*~*~ FuA 28 |
C**T********************~~~~~~~~~~~~~~~~~*~*---------~~~~~~~~~*~*-------*~*~ FuA 29 |
C**T********************~~~~~~~~~~~~~~~~~*~*---------~~~~~~~~~*~*-------*~*~ FuA 30 |
C**T********************~~~~~~~~~~~~~~~~~*~*---------~~~~~~~~~*~*-------*~*~ FuA 31 |
C**T********************~~~~~~~~~~~~~~~~~*~*---------~~~~~~~~~*~*-------*~*~ FuA 32 |
C**T********************~~~~~~~~~~~~~~~~~*~*---------~~~~~~~~~*~*-------*~*~ FuA 33 |
C**T********************~~~~~~~~~~~~~~~~~*~*---------~~~~~~~~~*~*-------*~*~ FuA 34 |
C**T********************~~~~~~~~~~~~~~~~~*~*---------~~~~~~~~~*~*-------*~*~ FuA 35 |
C**T********************~~~~~~~~~~~~~~~~~*~*---------~~~~~~~~~*~*-------*~*~ FuA 36 |
C**T********************~~~~~~~~~~~~~~~~~*~*---------~~~~~~~~~*~*-------*~*~ FuA 37 |
C**T********************~~~~~~~~~~~~~~~~~*~*---------~~~~~~~~~*~*-------*~*~ FuA 38 |
C**T********************~~~~~~~~~~~~~~~~~*~*---------~~~~~~~~~*~*-------*~*~ FuA 39 |
C**T********************~~~~~~~~~~~~~~~~~*~*---------~~~~~~~~~*~*-------*~*~ FuA 40 |
C**T********************~~~~~~~~~~~~~~~~~*~*---------~~~~~~~~~*~*-------*~*~ FuA 41 |
C**T********************~~~~~~~~~~~~~~~~~*~*---------~~~~~~~~~*~*-------*~*~ FuA 42 |
C**T********************~~~~~~~~~~~~~~~~~*~*---------~~~~~~~~~*~*-------*~*~ FuA 43 |
C**T********************~~~~~~~~~~~~~~~~~*~*---------~~~~~~~~~*~*-------*~*~ FuA 44 |
C**T********************~~~~~~~~~~~~~~~~~*~*---------~~~~~~~~~*~*-------*~*~ FuA 45 |
C**T********************~~~~~~~~~~~~~~~~~*~*---------~~~~~~~~~*~*-------*~*~ FuA 46 |
Table 8.2.—Listing of programs for constant discharge from a fully penetrating well of finite diameter—Continued

```fortran
CALL SIMPS(0.0, XPK, EPS, HBAR, SUM, DEL, EXBSL2) FUA 47
X2=XPK FUA 48
DX=XPK FUA 49
8 DX=10.0*DX FUA 50
X1=X2 FUA 51
X2=X1+DX FUA 52
Y=EXBSL2(X2) FUA 53
HBAR=0.007*DX FUA 54
CALL SIMPS(X1, X2, EPS, HBAR, TRM, ERR, EXBSL2) FUA 55
SUM=SUM+TRM FUA 56
DEL=DEL+ERR FUA 57
IF (X2=1.0E9) 9, 10, 10 FUA 58
9 Y=1.5707963/X2**4 FUA 59
IF (ABS(Y-YT)/YT<0.5E-6) 1808 FUA 60
10 EST=0.52359078/X2**3 FUA 61
SUM=SUM+EST FUA 62
FUX=S*2422779*SUM FUA 63
WHITE (6, 18) UW, SUM, DEL, FUX=SUM, XPK, YPK FUA 64
11 CONTINUE FUA 65
GO TO 1 FUA 66
12 STOP FUA 67
13 FORMAT (E10.5) FUA 68
14 FORMAT (16E5.0) FUA 69
15 FORMAT ('I1 ','F(UW,ALPHA) FOR ALPHA=', '1PE14.5/101,7X, '1UE1, 12X, '1INTEGRAL ERROR', '5X, '1F(UW,ALPHA)' '8X, '1X(PEAK)' '10X, '1Y(PEAF' FUA 70
2K)' 'I1 '1)
16 FORMAT (1H1, '1PE14.7, 9X, '1VALUES OF DUMMY VARIABLE TOO SMALL, '1PE25' FUA 71
1.7, '1PE17.7) FUA 72
17 FORMAT (1H1, '1PE14.7, 9X, '1VALUES OF DUMMY VARIABLE TOO LARGE, '1PE25' FUA 73
1.7, '1PE17.7) FUA 74
18 FORMAT (1H1, '1PE14.5, 1P5E17.5) FUA 75
END FUA 76

FUNCTION EXBSL2(X) FUA 77
C*********************************************************************:
C PURPOSE
C COMPUTES VALUES OF THE INTEGRAND FOR F(UW,ALPHA)
C DESCRIPTION OF PARAMETER
C X = REAL = ARGUMENT OF INTEGRAND
C*********************************************************************:
C COMMON/PBLK/A,B FUA 78
1 IF (X) 1, 2, 2 FUA 79
1 EXBSL2=0.0 FUA 80
GO TO 8 FUA 81
2 IF (X=1.0E7) 4, 4, 3 FUA 82
3 EXBSL2=1.5707963/X**4 FUA 83
GO TO 8 FUA 84
4 Y=3*X**X FUA 85
IF (Y=0.1) 5, 5, 6 FUA 86
5 FNUXY(1.0, Y*(5*XP(1.0/6, 0)=Y*(1.0/24, 0))) FUA 87
GO TO 7 FUA 88
6 FNUXY=EXP(Y) FUA 89
7 CALL JY0(X, BJ0, BY0) FUA 90
CALL JY1(X, BJ1, BY1) FUA 91
DEN=(X*BJ0=A*BJ1)**2+(X*BY0=A*BY1)**2)**3 FUA 92
EXBSL2=NUM/DEN FUA 93
8 RETURN FUA 94
END FUA 95
```

---

**Techniques of Water-Resources Investigations**
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*********************************************************************** FBA 1
C PURPOSE
C COMPUTES FUNCTION VALUES OF F(BETA,ALPHA) = THE SLUG TEST FBA 2
C FUNCTION = COOPER,H,R., JR., BREDEHOEFT,J.D., AND PAPADOPULOS, FBA 3
C L.S., 1967, RESPONSE OF A FINITE-DIAMETER WELL TO AN FBA 4
C INSTANTANEOUS CHARGE OF WATER: WATER RESOURCES RESEARCH, FBA 5
C Y., 3., NO. 1, P. 263=269, FBA 6
C PROGRAM BY S.S.PAPADOPULOS. FBA 7
C INPUT DATA
C 1 OR MORE CARDS = FORMAT(F16.5) FBA 8
C A = (ALPHA) = RW**2*S/KC**2 = RADIUS OF WELL (SCREEN OR FBA 9
C OPEN BORE IN AQUIFER) Squared = STORAGE COEFFICIENT FBA 10
C / RADIIU OF CASING (OVER INTERVAL OF WATER LEVEL FBA 11
C CHANGE) Squared. FBA 12
C SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED FBA 13
C PRX, DJY0, DJY1, DSIMPS = MUST BE INCLUDED IN DECK FBA 14
C METHOD
C THIS PROGRAM CALCULATES THE SLUG TEST FUNCTION, F(BETA,ALPHA), FBA 15
C FOR VALUES OF BETA RANGING FROM 0.001 TO 1000.0 BY INCREMENTS FBA 16
C OF 0.001. AVERAGE COMPUTATION TIME FBA 17
C IS ABOUT 30 SECONDS PER VALUE OF ALPHA ON IBM 360/155, FBA 18
C
C*********************************************************************** FBA 19
C DOUBLE PRECISION A,B,PI,ZZ, EPS,Y,X1,X2,TERM,FAB,DATAN,DEL,MBAR FBA 20
C DIMENSION ZZ(40), BB(39) FBA 21
C COMMON A,B,PI FBA 22
C EXTERNAL PRX FBA 23
C DATA ZZ(0), 0, 0, 1, 0, 1, 0, 5, 1, 0, 4, FBA 24
C 1, 0, 3, 1, 0, 2, 1, 0, 1, 3, 0, 1, 4, 0, 1, 5, 0, 1, 6, 0, FBA 25
C 1, 7, 0, 1, 8, 0, 1, 9, 0, 1, 0, 0, 0, FBA 26
C DATA 1, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, FBA 27
C 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, FBA 28
C 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, FBA 29
C 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, FBA 30
C DATA BB/0.001, 0.002, 0.004, 0.006, 0.008, 0.01, 0.02, 0.04, FBA 31
C 0.06, 0.08, 0.1, 0.2, 0.4, 0.6, 0.8, 1.0, 2.0, 3.0, 4.0, FBA 32
C 5.0, 6.0, 7.0, 8.0, 9.0, 10.0, 20.0, 30.0, 40.0, 50.0, FBA 33
C 60.0, 70.0, 80.0, 90.0, 100.0, FBA 34
C PI=3.14159265358979323846264338327950288419716939937, FBA 35
C EPS=0.0001 FBA 36
1 READ (5,b) A FBA 37
1 IF (A.LE.0.0) GO TO 5 FBA 38
WRITE (6,7) A FBA 39
WRITE (6,8) A FBA 40
WRITE (6,9) A FBA 41
WRITE (6,10) A FBA 42
WRITE (6,11) A FBA 43
WRITE (6,12) A FBA 44
WRITE (6,13) A FBA 45
WRITE (6,14) A FBA 46
WRITE (6,15) A FBA 47
WRITE (6,16) A FBA 48
WRITE (6,17) A FBA 49
WRITE (6,18) A FBA 50
WRITE (6,19) A FBA 51
WRITE (6,20) A FBA 52
WRITE (6,21) A FBA 53
WRITE (6,22) A FBA 54
WRITE (6,23) A FBA 55
WRITE (6,24) A FBA 56
WRITE (6,25) A FBA 57
WRITE (6,26) A FBA 58
WRITE (6,27) A FBA 59
WRITE (6,28) A FBA 60
WRITE (6,29) A FBA 61
WRITE (6,30) A FBA 62
WRITE (6,31) A FBA 63
WRITE (6,32) A FBA 64
END FBA 65
```

DOUBLE PRECISION FUNCTION PRX(X)

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******************************

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) 1,2
1 PRX=(PI*PI)/(lb,*A*S) GO TO 6
2 IF (XX,LT,150) GO TO 3
3 IF (XX,GT,O,0001) GO TO 4
C=DEXP(5,772156649D01)/2,
F1=PI***(1,4-A)
F2=XX*DEXP((C*C*A*X/B)+4*A)
PRX=(X*F1+F2*F2)/((X*(F1*F1+F2*F2))
GO TO 6
4 IF (XX,Lt,50) GO TO 5
PRX=DEXP(-W)/(X*(F1*F1+F2*F2))
GO TO 6
5 CALL DJYO(XX,J0,Y0)
CALL DJY1(XX,J1,Y1)
F1=XX*J0=2*A*J1
F2=XXX*Y0=2*A*Y1
PRX=DEXP(-W)/(X*(F1*F1+F2*F2))
RETURN
END

SUBROUTINE DJYO(X,X0,Y0)
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******************************

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) 4,412
2 Z=X**3/0=2
J0=1.0+Z*(2.2999997*)Z*(1.2656208=Z*(0,3163866=Z*(0,0444479=Z*(0,0000001=Z)
139444=Z*(0,0000002=Z*(0,0000000=Z)
W=(0,50000=Z0,3661977=DLOG(W)*J0+0,3674661+Z*(0,6059366=Z*(0,745303184=Z*DJO
1(0,25300117=Z*(0,0426121=Z*(0,00427916=Z*(0,0000000=Z))RETURN
3 Z=3,0/X
F=0,79788456=Z*(0,777=6+Z*(0,0055274=Z*(0,0000912=Z*(0,0137237=Z*DOJO
1+(0,000072605=0,000000004=Z=Z)*DOJO
P=0,78539816=Z*(0,04166297+Z*(0,00003954=Z*(0,00262573=Z*(0,0000541=DOJO
125+Z*(0,000029333=Z*(0,0000358=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=(1.000)/X
Q=DSQRT(U)
JO=Q*FDUS(X=P)
YO=Q*FDUS(X=P)
4 RETURN
END

SUBROUTINE DJY1(X,J1,Y1)

C*************t***t**********************~*~*****~*~*~~~*~~~**~**~~*~~~**~**~~~DJl
C
PURPOSE

COMPUTES BESSEL FUNCTIONS OF THE FIRST AND SECOND KIND, DJ1
FIRST ORDER, FOR POSITIVE ARGUMENTS,

DESCRIPTION OF PARAMETERS — ALL DOUBLE PRECISION
X — ARGUMENT, MUST BE >0
J1 — RETURNED FUNCTION VALUE, J1(X)
Y1 — RETURNED FUNCTION VALUE, Y1(X)

C**********L*****************C**********~**~**~~~~~~~~*~~~~~~***~~~***~~~DJ~
DOUBLE PRECISION X,J1,Y1,Z,DLOG,F,P,U,G,DSQ,D3QH,DCUS

IF (X=3.0) 1,2,3
1 IF (X) 4,4,2
2 Z=(X/3.0)**2
J1=X*(0.5-2*0.5624985-5*0.2109357-3*0.03954298-9*0.00443319)
Y1=0.63661977*DLOG(X)+J1*(0.63661977*0.5-0.312878990*Z)
RETURN

3 Z=3.0/X
F=0.79788456*Z*(0.0156+Z*(0.01659667+Z*(0.00017105+Z*(0.0024951)))
1=Z*(0.0013053+Z*(0.00020033+Z))
F=0.78539816*Z*(0.12498124*Z*(0.000637879*Z*(0.0007434)))
1=Z*(0.00079624+Z*(0.00029166+Z))
U=(1.000)/X
Q=DSQRT(U)
J1=Q*FDUS(X=P)
Y1=Q*FDUS(X=P)
4 RETURN
END

SUBROUTINE D3IMPS(A,B,EPS,HBAR,AREA,DEL,F)

C*************t***t**********************~*~*****~*~*~~~*~~~**~**~~*~~~**~**~~~DJl
C
PURPOSE

TO DETERMINE THE INTEGRAL OF A FUNCTION, F, FROM A TO B,
USING SIMPSON’S RULE,

DESCRIPTION OF PARAMETERS
ALL DOUBLE PRECISION
A = LOWER LIMIT OF INTEGRAL
B = UPPER LIMIT OF INTEGRAL
EPS = DESIRED ACCURACY
HBAR = MINIMUM DIVISION OF THE INTERVAL
AREA = COMPUTED VALUE OF INTEGRAL BETWEEN U AND R
DEL = COMPUTED ESTIMATE OF ERROR
F = THE INTEGRAND (FUNCTION REFERENCE)

METHOD

USES SIMPSON’S RULE TO COMPUTE A SUM APPROXIMATING THE INTEGRAL
USES INITIAL H=(B-A)/2, COMPUTES A SEQUENCE OF SUMS BY HALVING H EACH TIME, COMPUTES ESTIMATE OF ERROR (DEL) AS (PREVIOUS SUM - CURRENT SUM)/15, COMPUTATION STOPS WHEN 1) H<HBAR, 2) ABS(DEL)*ABS(EPS*CURRENT SUM), IF HBAR IS LE 0,
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******************************************************************************
C
C DOUBLE PRECISION M, MBAR, AREA, DEL, S1, S2, S3, SC, SP, X, A, B, EPS, EB
C
C AREA OF F FROM A TO B, EPS IS DESIRED ACCURACY, MBAR THE MINIMUM
C
C ALLOWABLE INTERVAL, DEL THE ESTIMATE OF THE ERROR
C
M = B - A
IF (M) 1, 1, 2
1 AREA = 0.0
DEL = 0.0
GO TO 10
2 SP = 1.0 D35
S3 = 0.0
S1 = F(A) + F(B)
IF (MBAR) 3, 3, 4
3 HBAR = 0.007 * H
4 S2 = 0.0
5 S2 = S2 + 0.0 * F(X)
X = X + 1
IF (X = B) 5, 5, 6
6 SC = (S1 + S2 + S3) * H * 0.16666666667
DEL = 0.00666666667 * (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 (M = MBAR) 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
C COMPUTES CHANGES IN WATER LEVEL, H(R, T), IN RESPONSE TO VARYING DISCHARGE USING THE
C CONVOLUTION INTEGRAL FOR LEAKY AQUIFERS—JOHN MOENCH, ALLEN, 1971, GROUND-WATER
C FLUCTUATIONS IN RESPONSE TO ARBITRARY PUMPAGE, GROUND WATER, VOL. 9, NO. 2, 1971.
C
C INPUT DATA—ONE OR MORE GROUPS, EACH GROUP CODED AS FOLLOWS
C
C 1 CARD = FORMAT (2E10.5, 4X, I1, 5X, E10.5)
C
C begin = Smallest value of time for output,
C
C end = Largest value of time for output,
C
C ig = Indicates form of discharge function, q(t),
C
C 1, 2, 3 REFER TO DISCHARGE FUNCTIONS IN HANTUSH, M. S., 1964, HYDRAULICS OF WELLS IN
C CHIMAYE, VEN TE, ED., ADVANCES IN HYDROSCIENCE, VOL. 11
C
C IG = 1, q(t) IS AN EXPONENTIAL FUNCTION, CASE A
C
C 343 OF HANTUSH
C
C IG = 2, q(t) IS A HYPERBOLIC FUNCTION, CASE B
C
C 344 OF HANTUSH
C
C IG = 3, q(t) IS AN INVERSE SQUARE ROOT FUNCTION, CASE C
C
C 345 OF HANTUSH.
Table 11.1.—Listing of program to compute the convolution integral for a leaky aquifer—Continued

```plaintext
C
C IG=4, Q(T) IS A FIFTH-DEGREE POLYNOMIAL, MRT 24
C IG=5, Q(T) IS A PIECEWISE LINEAR FUNCTION OF MRT 25
C TIME (EIGHT SEGMENTS). MRT 26
C OR = REFERENCE DISCHARGE, ZERO OR BLANK FOR PROJECTION. MRT 27
C 1 OR 4 CARDS, DEPENDING ON IG. MRT 28
C IF IG=1,2,3 = 1 CARD = FORMAT(3E10,3) MRT 29
C QST = EVENTUAL CONSTANT DISCHARGE. MRT 30
C DELTA = RATE PARAMETER. MRT 31
C TSTAR = TIME PARAMETER. MRT 32
C IF IG=4 = 1 CARD = FORMAT(6E10,3) MRT 33
C AQ(6) = 6 VALUES = THE POLYNOMIAL COEFFICIENTS MRT 34
C WITH A0 FIRST AND AS LAST. MRT 35
C IF IG=5 = 4 CARDS = FORMAT(6E10,3) MRT 36
C T(I),AI(I),BI(I),T(I+1),AI(I+1),BI(I+1),I=1,3,5,7 MRT 37
C PARAMETERS OF THE PIECEWISE LINEAR FUNCTION MRT 38
C (8 SEGMENTS), CODED 2 SEGMENTS PER CARD, FIRST MRT 39
C AND SECOND SEGMENTS ON FIRST CARD, THEN SEQUENTIALLY MRT 40
C ON SUCCESSING CARDS, EACH SEGMENT HAS THREE MRT 41
C PARAMETERS WHICH ARE IN CODING ORDER MRT 42
C T = ENDING TIME OF THE SEGMENT. MRT 43
C AI = DISCHARGE AT BEGINNING OF SEGMENT. MRT 44
C BI = RATE OF CHANGE IN DISCHARGE UNTIL SEGMENT MRT 45
C THE DISCHARGE FUNCTION IN EACH SEGMENT HAS THE MRT 46
C FORM Q(T) = AI(I)+BI(I)*(T-T(I-I)), IF LESS THAN 8 MRT 47
C SEGMENTS ARE NEEDED, BLANKS CAN BE CODED FOR MRT 48
C SUCCESSING SEGMENTS. MRT 49
C 2 OR MORE CARDS = FORMAT(8E10,3) MRT 50
C R = RADIAL DISTANCE FROM PUMPED WELL, BLANK OR ZERO MRT 51
C SIGNALS PROGRAM AS END TO GROUP OF DATA, MRT 52
C S = STORAGE COEFFICIENT MRT 53
C T = TRANSMISSIVITY MRT 54
C PH = (PI/M') = HYD. COND. OF CONFINING BED DIVIDED MRT 55
C BY THICKNESS OF CONFINING BED MRT 56
C SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED MRT 57
C CONVOL,Q = MUST BE INCLUDED IN DECK. MRT 58
C C*************************************************************************** C*************************************************************************** C***************************************************************************

DIMENSION D(12),IEX(12),X(6),H(12,6),QS(12,6),CP(12),CT(12) MRT 61
DIMENSION H1(12),H2(12),Q1(12),Q2(12) MRT 62
DIMENSION H3(12),H4(12),Q3(12),Q4(12) MRT 63
COMMON AQ(6),T(9),AI(9),BI(9),QST,DELTA,TSTAR MRT 64
DATA CP/12*! T = 1/CT/12!1/UT*/!D/12*10**1/ MRT 65
DATA H1/12*! R=H/12*1**R*/!Q1/12*! T = !/Q2/12*!Q(T)'/ MRT 66
DATA H3/12*! S1/H4/12*!Q(T)'/!Q3/12*!Q(T'/!Q4/12*!Q'/ MRT 67
DATA X/1,15,2,3,5,7/ MRT 68
T(1)*0. MRT 69
N=500 MRT 70

1 READ (5,18,END=17) TBEGIN,TEND,IO,OR MRT 71
IF (IQ,LT,4) READ (5,19) QST,DELTA,TSTAR MRT 72
IF (IQ,EQ,4) READ (5,19) AQ MRT 73
IF (IQ,EQ,5) READ (5,19) (T(I),AI(I),BI(I),I=2,9) MRT 74
WRITE (6,20) MRT 75

2 READ (5,19) R,S,T,PM MRT 76
IF (R,EQ,0.) GO TO 1 MRT 77
A=PM*R/(4.*T) MRT 78
B=PM/8 MRT 79
Y=ALOG10(TBEGIN) MRT 80
```
Table 11.1.—Listing of program to compute the convolution integral for a leaky aquifer—Continued

```plaintext
IF (Y) 3,5,4
3 Y=Y-.001
GO TO 5
4 Y=Y+.001
5 IBEGIN=Y
Y=ALUG10(TEND)
IF (Y) 6,8,7
6 Y=Y-.001
GO TO 8
7 Y=Y+.001
8 TEND=Y
H=TEND-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=TIME
CALL CONVOL(TIME,A,B,N,IO,SUM)
IF (QR,GT,0.) GO TO 9
H(I,J)=SUM/(1.5664*T)
QS(I,J)=Q(T-I,E,IQ)
GO TO 10
9 M(I,J)=SUM/QR
QS(I,J)=Q(TIME,IO)/QR
10 CONTINUE
K=M
IF (M,GT,6) M=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)
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
K=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)
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
18 FORMAT (2E10.5,4X,I1,5X,E10.5)
19 FORMAT (6E10.3)
20 FORMAT ('10,'R**2*8/(4*TRANS)#',1PE10.3,1', X'1/(3*B**1)**1',E10.3/10'1RMT 135
1,2X,'T',5X,6(2A4,12,9X))
21 FORMAT ('1 ',4X,6(2A4,2X,2A4,1X))
```

C
TYPE CURVES FOR FLOW TO WELLS IN CONFINED AQUIFERS

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

22 FORMAT ('1',F9.1,6('0PB,3,1PE11,3'))
23 FORMAT ('01',2X,'T',6,5X,'O,(2A4,12,9X))
24 FORMAT (1M1)
25 FORMAT ('01',1R**2*8/(4*TRANS)'1,1PE10,3',1,11/(S*B)'1,1,E10,3,1',
10=X,E10,3(1X,1/U'),6(2A4,12,9X))
26 FORMAT ('01',1X,'U',1/X,6(2A4,12,9X))

SUBROUTINE CONVOL (TIME,A,B,N,IU,BUM)

C**************************************************************
C PURPOSE
C COMPUTES VALUES OF THE CONVOLUTION INTEGRAL FOR LEAKY
C AQUIFERS, THE INTEGRAL IS, FROM 0 TO T, IF
C Q(T)/T = EXP(-1/2*A/T) - B*T
C DESCRIPTION OF PARAMETERS
C A,B, SUM ARE REAL; N,IU ARE INTEGER;
C A = R**2*8/(4*TRANS) = 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 IU = INDICATES FORM OF DISCHARGE FUNCTION;
C SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C W
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**************************************************************
C REAL*8 DSUM
C REAL*4 NEWT, NEWTP, NEWX, NEWF
C DOUBLE*0,4+B
C INTEGER*0,4+B
C
C INITIAL T COMPUTED FROM A,B
C AB=A*B
C IF (AB,GE,2500,) GO TO 7
C IF (B,GT,0,0) GO TO 2
C OLDT=U
C GO TO 3
C OLDX=(1-SQRT(1.+AB/2500,))*.50,0/B
C IF (ULDT,EQ.0,0) GO TO 1
C INITIAL T=T
C OLDTIME=OLDT
C OLDF=(OLDTP,1,)*EXP(ULDX)/ULDT
C END OF SUMMATION SEGMENT IS 10 TIMES THE BEGINNING
C ENDT=10,0+ULDT
C IF (ENDT,LT,TIME) GO TO 5
C IF (ULDT,GE,TIME) GO TO 7
C IS=1
C ENDTIME
C TECHNIQUES OF WATER RESOURCES INVESTIGATIONS

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

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'
6 NEW=OLDT+DEL T
NEWX=A/NEW+B*NEW
C TERMINATES SUMMATION WHEN EXP(-A/T'-B*T') < 1,37E-44
7 IF (NEWX,LT,101.) GO TO 7
NEW=TIME=NEW
NEWF=(NEWT,IP)*EXP(NEWX)/NEW
DSUM=DSUM(NEWF+OLDF)*DEL T
OLD=NEW
OLDF=NEW
C CONTINUE
8 IF (IS,GT,0) GO TO 7
C IF T' < T, BEGIN A NEW SEGMENT
9 GO TO 4
7 SUM=DSUM/2.D+0
RETURN
END

FUNCTION Q(TIME,IO)

******************************************************************************

C PURPOSE
C COMPUTES THE DISCHARGE FUNCTION, Q(T)
C
C DESCRIPTION OF PARAMETERS
C TIME = REAL = ELAPSED TIME SINCE BEGINNING OF DISCHARGE,
C IO = INTEGER = INDICATES FORM OF DISCHARGE FUNCTION,
C IG=1,2,3, CASES A,B,C, RESPECTIVELY, OF HANTUSH,M,S,
C 1964, HYDRAULICS OF WELLS IN CHUH, VEN TE, ED.,
C ADVANCES IN HYDROSCIENCE, VOL. 11 ACADEMIC PRESS,
C NEW YORK, P. 343,344,
C IG=4, DISCHARGE IS A FIFTH DEGREE POLYNOMIAL OF TIME,
C IG=5, DISCHARGE IS A PIECEWISE LINEAR FUNCTION OF UP TO
C 8 SEGMENTS,
C
C METHOD
C FORTRAN EVALUATION OF FUNCTIONS,
C
C******************************************************************************

COMMON AG(6),TI(9),AI(9),BI(9),QST,DELT A,TSTAR
GO TO (1,2,3,4,5), IO
1 Q=QST*(1.+DEL T/EXP(-TIME/TSTAR))
RETURN
2 Q=QST*(1.+DEL T/(1.+TIME/TSTAR))
RETURN
3 Q=QST*(1.+DEL T/SQRT(1.+TIME/TSTAR))
RETURN
4 Q=AG(1)+TIME*(AG(2)+TIME*(AG(3)+TIME*(AG(4)+TIME*(AG(5)+TIME*AG(6)
1))))
RETURN
5 DO 6 I=2,9
6 IF (TIME.LE.TI(I)) GO TO 7
7 IF (TIME.LE.TI(I)) GO TO 7
8 If (TIME.LE.TI(I)) GO TO 7
RETURN
END

GPO 689-143