COMPUTER PROGRAM TO TABULATE THE PARTIAL PENETRATION LEAKY FUNCTION

Fred W. Hartman and John C. Halepaska

Open-File Report

Kansas Geological Survey

July 1971

COMPUTER PROGRAM TO TABULATE PARTIAL PENETRATION LEAKY FUNCTION

Purpose:

This program tabulates the solution of unsteady-state flow toward a partially penetrating well in an assumed infinitely thick, leaky aquifer (Halepaska, 1966).

$$s = \frac{Q}{8\pi KT'} \Big[2\int_u^\infty \frac{e^{-y}}{y} e^{\frac{z}{a}} e^{\frac{r^2}{4a^2y}} \operatorname{erfc}\left(\frac{r}{2a\sqrt{y}} + \frac{z\sqrt{y}}{r}\right) dy$$

$$- 2\int_u^\infty \frac{e^{-y}}{y} e^{\frac{l'+z}{a}} e^{\frac{r^2}{4a^2y}} \operatorname{erfc}\left(\frac{r}{2a\sqrt{y}} + \frac{(l'+z)\sqrt{y}}{r}\right) dy \qquad (1)$$

$$+ 2M(u,\frac{z}{r}) + M(u,\frac{l'-z}{r}) - M(u,\frac{l'+z}{r}) \Big]$$

where $u = \frac{r^2S}{4Tt}$ and $M(u,c)$ is defined by

$$M(u,c) = \int_u^\infty \frac{e^{-y}}{y} \operatorname{erf}(c\sqrt{y})\, dy$$

and where c is an argument in terms of l', z and r.

The parameters used in the above solution are defined as:

$\frac{1}{a}$ = leakage factor, $\frac{K'}{Kb'}$

b' = thickness of semipervious layer

K = hydraulic conductivity of aquifer

K' = hydraulic conductivity of semipervious bed

l' = length of penetration of the well

Q = discharge per unit of time

r = radial distance from axis of the pumping well to any point in space

s = drawdown at any point in the aquifer at any time t since pumping began

S = storage coefficient of aquifer

T = transmissibility of aquifer

z = depth from bottom of semipervious layer to any point in space.

The argument under the integral sign in Equation (1), for tabulation purposes, can be defined as follows:

$$f(x) = \frac{e^{-x}}{x} \left\{ e^{\frac{k_2^2}{x}} \left[k_1 \operatorname{erfc}\left(\frac{k_2}{\sqrt{x}} + k_3\sqrt{x}\right) + k_4 \operatorname{erfc}\left(\frac{k_2}{\sqrt{x}} + k_3\sqrt{x}\right)\right] + 2 \operatorname{erf}(k_3\sqrt{x}) + \operatorname{erf}(k_6\sqrt{x}) - \operatorname{erf}(k_5\sqrt{x}) \right\} \quad (2)$$

where k_1, k_2 . . . k_6 are constants and erf and erfc are the error function and complementary error function, respectively, and are defined by:

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt$$

$$\operatorname{erfc}(x) = 1 - \operatorname{erf}(x).$$

The integral is then in the form:

$$\int_a^\infty f(x)\, dx.$$

The evaluation over the semi-infinite interval is approximated by using a finite interval with the integral in the form:

$$\int_a^b f(x)\, dx$$

where b is sufficiently large to give good results to five significant digits. The quadrature formula used is the composite Newton-Coates closed formula with n = 2. This formula, commonly called Simpson's Rule, is of the form:

$$\int_a^b f(x)\,dx \cong \frac{\Delta x}{3}\,(f(a) + 2f(a + \Delta x) + 4f(a + 2\Delta x) + 2 \ldots$$

$$+ 4f(b - \Delta x) + f(b))$$

where $\Delta x = \frac{b - a}{m}$ and m is even (m is the number of subintervals).

A subroutine evaluates the error function using a table look-up for arguments, say y, such that

$$0 \leq y \leq 4.0.$$

For y > 4.0, an asymptotic expansion approximation is used of the form:

$$\mathrm{erf}\ (y) \cong 1 - \frac{e^{-y^2}}{\sqrt{\pi}}\,\left(\frac{1}{y} - \frac{1}{2y^3} + \frac{3}{4y^5}\right).$$

The table look-up method consists of comparing the argument value against the tabulated values until the first tabulated value which exceeds the argument is found. A 4-point Lagrangian interpolation polynomial is generated using the two tabulated values on each side of the argument to interpolate for the value of the error function.

Running instructions:

The program tabulates the solution given the three dimensionless groups:

$$\frac{z}{r},\ \frac{r}{T'},\ \text{and}\ \frac{z}{a}$$

The lower limit of the intergral, u, is equal to $\frac{r^2S}{4Tt}$ and an array of the form $\frac{1}{4u}$ can be read in to generate curves of $\frac{sKl'}{Q}$ vs $\frac{1}{4u}$. See the example type curves on page 8. Since $\frac{z}{r}$, $\frac{r}{l'}$, $\frac{z}{a}$, and $\frac{1}{4u}$ are dimensionless, any set of consistent units will give the proper result.

Data cards are fed into the program and output is in printed form only. Number of abscissas for all curves and number of curves are the only options in the program. A listed example input and corresponding printed ouput are shown on pages 6 and 7, respectively.

Input:

Card #1	N - number of abscissas to read Format (I5)
Cards #2 - #(N + 1)	Abscissas of curves ($\frac{1}{4u}$) (one per card) Format (F 14.9)
Card #(N + 2)	M - number of sets of $\frac{z}{a}$, $\frac{z}{r}$, $\frac{r}{l'}$ that are to be used (number of curves) Format (I5)
Cards #(N + 3) - #(M + N + 3)	$\frac{z}{a}$, $\frac{z}{r}$, $\frac{r}{l'}$ (3 values per card) Format (3F 12.5)

References:

Carnahan, B., Luther, H. A., and Wilkes, J. O., Applied numerical methods, John Wiley and Sons, Inc., 1969.

Halepaska, J. C., Drawdown distribution around a well penetrating a thick leaky aquifer: Master of Science thesis, New Mexico Institute of Mining and Technology, Socorro, 1966.

Handbook of mathematical functions: U.S. Government Printing Office, Washington, D. C., 1964.

Ingersoll, L. R., Zobel, O. J., and Ingersoll, A. C., Heat conduction with engineering, geological and other applications: The University of Wisconsin Press, Madison, 1954.

Kreyszig, E., Advanced engineering mathematics: John Wiley and Sons, Inc., 1964.

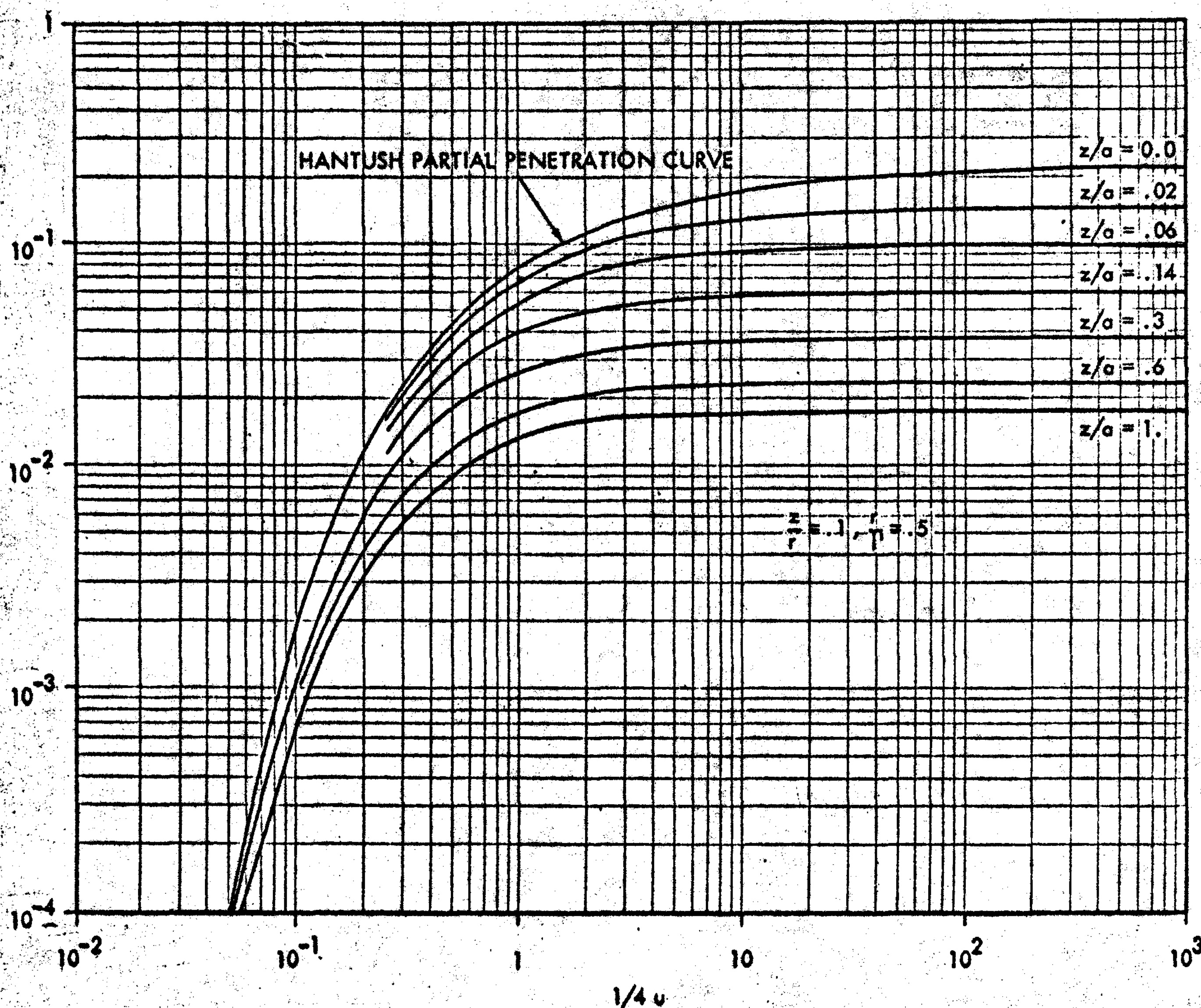
EXAMPLE INPUT:

```
*
  34
  .025
  .04
  .07
  .1
  .15
  .2
  .25
  .4
 .7
  1.
 1.5
  2.
 2.5
  3.
  3.6
 4.
  7.
  10.
  15.
  20.
  25.
  40.
  70.
 100.
 150.
 200.
 250.
 400.
 700.
1000.
1500.
2500.
25000.
250000.
   3
     .8          .2          5.
    1.4          .2          5.
    2.           .2          5.
```

EXAMPLE OUTPUT:

Z/A = 0.8000 Z/R = 0.2000 R/L = 5.0000

U	1/4U	SKL/Q
0.1000000004D 02	0.2499999991D-01	0.1223063775D-06
0.6249999994D 01	0.4000000004D-01	0.6398965919D-05
0.3571428556D 01	0.7000000030D-01	0.1081281793D-03
0.2500000009D 01	0.9999999963D-01	0.3311880159D-03
0.1666666663D 01	0.1500000004D 00	0.7777705139D-03
0.1250000005D 01	0.1999999993D 00	0.1177259244D-02
0.1000000000D 01	0.2500000000D 00	0.1498666207D-02
0.6250000023D 00	0.3999999985D 00	0.2114671190D-02
0.3571428556D 00	0.7000000030D 00	0.2645604162D-02
0.2500000000D 00	0.1000000000D 01	0.2860575766D-02
0.1666666667D 00	0.1500000000D 01	0.3011992296D-02
0.1250000000D 00	0.2000000000D 01	0.3088517067D-02
0.1000000000D 00	0.2500000000D 01	0.3138196106D-02
0.8333333333D-01	0.3000000000D 01	0.3163115979D-02
0.6944444456D-01	0.3599999994D 01	0.3182501894D-02
0.6250000000D-01	0.4000000000D 01	0.3191651428D-02
0.3571428571D-01	0.7000000000D 01	0.3222745389D-02
0.2500000000D-01	0.1000000000D 02	0.3232823703D-02
0.1666666667D-01	0.1500000000D 02	0.3239406705D-02
0.1250000000D-01	0.2000000000D 02	0.3242187751D-02
0.1000000000D-01	0.2500000000D 02	0.3243659914D-02
0.6250000000D-02	0.4000000000D 02	0.3245544617D-02
0.3571428571D-02	0.7000000000D 02	0.3246595297D-02
0.2500000000D-02	0.1000000000D 03	0.3246927287D-02
0.1666666667D-02	0.1500000000D 03	0.3247141397D-02
0.1250000000D-02	0.2000000000D 03	0.3247231088D-02
0.1000000000D-02	0.2500000000D 03	0.3247278356D-02
0.6250000000D-03	0.4000000000D 03	0.3247338618D-02
0.3571428571D-03	0.7000000000D 03	0.3247372069D-02
0.2500000000D-03	0.1000000000D 04	0.3247382612D-02
0.1666666667D-03	0.1500000000D 04	0.3247389403D-02
0.1000000000D-03	0.2500000000D 04	0.3247393743D-02
0.1000000000D-04	0.2500000000D 05	0.3247397397D-02
0.1000000000D-05	0.2500000000D 06	0.3247397512D-02



LEAKY PARTIAL PENETRATION TYPE CURVES

$W(u, \frac{z}{r}, \frac{r}{l}, \frac{z}{a})$

```
*
C          EVALUATION OF PARTIAL PENETRATION LEAKY FUNCTION
      DOUBLE PRECISION X1,X2,X3,X4,X7,X8,X9
      DOUBLE PRECISION Y,U,UX,XLO,SUM,SIM,DU,SUMX
      COMMON/E/X(250),Y(250)
      COMMON X1,X2,X3,X4,X7,X8,X9
      DIMENSION XLO(50),DU(20)
      DIMENSION SUM(20)
      SIM=SIM
      Y(1)=Y(1)-1.
   11 FORMAT(I5)
   42 FORMAT(14X,6HZ/A = ,F8.4,10X,6HZ/R = ,F8.4,10X,6HR/L = ,F8.4)
   43 FORMAT(18X,1HU,24X,4H1/4U,23X,5HSKL/Q)
   44 FORMAT(25X,34HPARTIAL PENETRATION LEAKY FUNCTION)
    2 FORMAT(6E12.5)
    5 FORMAT(10X,D17.10)
    6 FORMAT(3(10X,D17.10))
    7 FORMAT(1H )
   12 FORMAT(F14.9)
C     N = NUMBER OF LOWER LIMITS TO USE
C     READ IN ARRAY OF LOWER LIMITS OF INTEGRATION
      READ(5,11)N
      WRITE(6,44)
      DO 21 K=1,N
   21 READ(5,12)XLO(K)
C     INITIALIZE FINITE INTEGRATION INTERVALS
      DU(1)=1.D-06
      DU(2)=1.D-05
      DU(3)=1.D-04
      DU(4)=1.D-03
      DU(5)=1.D-02
      DU(6)=1.D-01
      DU(7)=1.D0
      DU(8)=60.D0
C     NT=NUMBER OF SETS OF CONSTANTS TO EVAUATE INTEGRAL
C     READ IN Z/A , Z/R , R/L PRIME
      READ(5,11)NT
      DO 20 L=1,NT
      READ(5,2)X1,X4,X7
      WRITE(6,7)
      WRITE(6,7)
      WRITE(6,7)
      WRITE(6,42)X1,X4,X7
      WRITE(6,7)
      WRITE(6,7)
      WRITE(6,43)
      WRITE(6,7)
      WRITE(6,7)
C     COMPUTE VALUES WHICH WILL BE CONSTANTS IN INTEGRATION
C     X1 = Z/A
C     X2 = R**2/4A**2
C     X3 = R/2A
C     X4 = Z/R
C     X7 = R/L
C     X8 = L/R
C     X9 = L/A
C     XX1 = EXP(L+Z)/A)
      X3=X1/(2.*X4)
      X2=X3**2
```

```
      X8=1./X7
      X9=X1/(X4*X7)
C     COMPUTE VALUE OF INTEGRAL OVER FINITE INTERVALS
      DO 30 I=1,6
      CALL SIMP(DU(I+1),DU(I),11,SIM)
      SUM(I)=SIM
   30 CONTINUE
      CALL SIMP(DU(8),DU(7),241,SIM)
      SUM(7)=SIM
      DO 20 K=1,N
C     COMPUTE LOWER LIMIT OF INTEGRATION
      U=1./(4.*XLO(K))
      UX=U
      I=1
C     TEST LOWER LIMIT AGAINST INITIALIZED VALUES
   34 EP=ABS(UX-DU(I))
      IF(EP-.00000001)32,32,39
   39 IF(UX-DU(I))31,32,33
   33 I=I+1
      GO TO 34
   32 SUMX=0.0
      GO TO 37
   38 CALL SIMP(DU(8),UX,241,SIM)
      SUMX=SIM
      GO TO 36
   31 IF(I.EQ.8)GO TO 38
C     INTERGRATE TO NEXT HIGHER INITIALIZED VALUE
      CALL SIMP(DU(I),UX,11,SIM)
      SUMX=SIM
   37 J=I
C     SUM UP VALUES OF INTEGRAL OVER REMAINING FINITE INTERVALS
      DO 35 I=J,7
      SUMX=SUMX+SUM(I)
   35 CONTINUE
   36 WRITE(6,6)UX,XLO(K),SUMX
   20 CONTINUE
      STOP
      END
```

```
*
      SUBROUTINE EVALF(A,B)
C     THIS SUBROUTINE EVALUATES THE FUNCTION UNDER THE INTEGRAL SIGN
C     A = ARGUEMENT OF THE FUNCTION
C     B = VALUE OF FUNCTION
      COMMON X1,X2,X3,X4,X7,X8,X9
      DOUBLE PRECISION X1,X2,X3,X4,X7,X8,X9
      DOUBLE PRECISION SK,A,B,B1,B2
      DOUBLE PRECISION PI,ED,FD
      DOUBLE PRECISION B3,B4,B5,C1,C2,C3,C4,C5,D1,D2,D3,D4,D5,E,F
      DOUBLE PRECISION EX,FX,PII
      C1=C1
      C2=C2
      C3=C3
      C4=C4
      C5=C5
      D1=D1
      D2=D2
      D3=D3
      D4 = D4
      D5=D5
      PI=3.1415926535897933
      PII = 1./(4.*PI)
      SK=DSQRT(A)
      B1=(X3/SK)+(X4*SK)
      B2=(X3/SK)+((X8+X4)*SK)
      B3=X4*SK
      B4=(X8-X4)*SK
      B5=(X8+X4)*SK
C     CALL SUBROUTINE THAT EVALUATES COMPLEMENTARY ERROR FUNCTION AND
C     ERROR FUNCTION
      IF((B1**2).GT.65.)GO TO 41
      IF((X2/A).GT.65.)GO TO 41
      IF(X1.GT.65.)GO TO 41
      IF((X1+X9).GT.65.)GO TO 41
      IF((B2**2).GT.65.)GO TO 41
      CALL EAFC(B1,C1,D1)
      ED=C1*DEXP(X2/A)*DEXP(X1)
      CALL EAFC(B2,C2,D2)
      FD=C2*DEXP(X2/A)*DEXP(X1+X9)
      E=ED-FD
      CALL EAFC(B3,C3,D3)
      CALL EAFC(B4,C4,D4)
      CALL EAFC(B5,C5,D5)
      F=(D3+(.5*D4)-(.5*D5))
      B=PII*(DEXP(-A)/A)*(E+F)
      GO TO 51
   41 EX=DEXP(-1.*B1**2+(X2/A)+X1-A)*(1./A)*((1./B1)-(1./(2.*B1**3))+
     1(3./(4.*B1**5)))*(1./DSQRT(PI))
      FX=DEXP(-1.*B2**2+(X2/A)+(X1+X9)-A)*(1./A)*((1./B2)-(1./(2.*B2**3
     1))+(3./(4.*B2**5)))*(1./DSQRT(PI))
      E = EX - FX
      CALL EAFC(B3,C3,D3)
      CALL EAFC(B4,C4,D4)
       CALL EAFC(B5,C5,D5)
      F=D3+(.5*D4)-(.5*D5)
      B = PII * (DEXP(-A)/A) * F + PII * E
   51 CONTINUE
      RETURN
      END
```

```
*
      SUBROUTINE EAFC(A,B,C)
C     THIS SUBROUTINE EVALUATES THE COMPLEMENTARY ERROR FUNCTION AND
C     ERROR FUNCTION
C     A = ARGUEMENT
C     B = VALUE OF ERFC(A)
C     C = VALUE OF ERF(A)
      COMMON/E/X(250),Y(250)
      DOUBLE PRECISION Y,SPI,PI
      DOUBLE PRECISION A,B,C,BZ,A2
      C=C
      Y(1)=0.0
      PI=3.1415926535897933
      SPI=1./DSQRT(PI)
C     CHECK TO SEE IF ARGUEMENT IS IN RANGE OF TABULAR VALUES
      IF(A-4.0)1,1,2
    2 A2=A**2
      BZ=SPI*DEXP(-1.*A2)
      B=((1./A)-(1./(2.*A**3))+(3./(4.*A**5)))*BZ
      C=1.-B
      GO TO 12
C     SEARCH TABLE FOR VALUE EXCEEDING ARGUEMENT
    1 I=1
    3 IF(A-X(I))4,5,6
    5 C=Y(I)
      GO TO 11
    4 IF(I-2)8,8,9
    8 I=1
      GO TO 10
    9 I=I-2
C     INTERPOLATE FOR ERROR FUNCTION USING 4 POINT INTERPOLATION
   10 CALL TERR(A,X(I),X(I+1),X(I+2),X(I+3),Y(I),Y(I+1),Y(I+2),Y(I+3),C)
      GO TO 11
    6 IF(I-248)13,9,9
   13 I=I+1
      GO TO 3
   11 CONTINUE
      B=1.-C
   12 CONTINUE
      RETURN
      END
```

```
*
      SUBROUTINE SIMP(UU,UL,NPT,SIM)
C     THIS SUBROUTINE PERFORMS NUMERICAL INTEGRATION BY SIMPSON=S RULE
C     UU = UPPER LIMIT
C     UL = LOWER LIMIT
C     NPT = NO. OF POINTS(MUST BE ODD)
C     SIM = VALUE OF INTEGRAL
      DOUBLE PRECISION SIM,YY,ZZ,SUM,XINC,UU,UL
      DOUBLE PRECISION XNP
      ZZ=ZZ
      YY=UL
      XNP=NPT-1
      XINC=(UU-UL)/XNP
      CALL EVALF(YY,ZZ)
      SUM=ZZ
      JKK=(NPT-3)/2
      DO 3 I=1,JKK
      YY=YY+XINC
      CALL EVALF(YY,ZZ)
      SUM=SUM+4.*ZZ
      YY=YY+XINC
      CALL EVALF(YY,ZZ)
      SUM=SUM+2.*ZZ
    3 CONTINUE
      YY=YY+XINC
      CALL EVALF(YY,ZZ)
      SUM=SUM+4.*ZZ
      YY=YY+XINC
      CALL EVALF(YY,ZZ)
      SUM=SUM+ZZ
      SIM=(XINC/3.)*SUM
      RETURN
      END
```

```
*
      SUBROUTINE TERR(XO,X1,X2,X3,X4,Y1,Y2,Y3,Y4,YO)
C     THIS SUBROUTINE INTERPOLATES EQUALLY OR NON EQUALLY SPACED DATA
C     POINTS USING LAGRANGIAN 4 POINT INTERPOLATION POLYNOMIAL
C     X=S = DATA POINTS
C     XO = POINT AT WHICH FUNCTION IS TO BE EVALUATED
C     Y=S = VALUES OF FUNCTION
C     YO = VALUE OF FUNCTION AT XO
      DOUBLE PRECISION XO,Y1,Y2,Y3,Y4,YO
      DOUBLE PRECISION DH,DH1,DH2,DH3,D1,D2,D3,D4,D5,D6,D7
      DOUBLE PRECISION Y5,Y6,Y7,Y8
    3 FORMAT(1X,D15.8)
      DH=XO-X1
      DH1=XO-X2
      DH2=XO-X3
      DH3=XO-X4
      D1=X2-X1
      D2=X2-X3
      D3=X2-X4
      D4=X3-X1
      D5=X3-X4
      D6=X1-X4
      D7=X2-X4
      Y5=((DH2*DH1*DH3)/(D1*D4*D6))*Y1
      Y6=((DH*DH2*DH3)/(D1*D2*D3))*Y2
      Y7=((DH*DH1*DH3)/(D4*D2*D5))*Y3*(-1.)
      Y8=((DH*DH1*DH2)/(D6*D7*D5))*Y4*(-1.)
      YO=Y5+Y6+Y7+Y8
      RETURN
      END
```

```
*
      BLOCK DATA
C     DATA TABLE OF ERROR FUNCTION FROM 0.0 TO 4.6
      COMMON/E/X(250),Y(250)
      DOUBLE PRECISION Y
C     DATA TABLE FROM ARGUEMENTS OF ERROR FUNCTION (X)
      DATA (X(I),I=1,51)/.00,.01,.02,.03,.04,.05,.06,.07,.08,.09,.1,
     1 .11,.12,.13,.14,.15,.16,.17,.18,.19,.2,.21,.22,.23,.24,.25,
     1.26,.27,.28,.29,.3,.31,.32,.33,.34,.35,.36,.37,.38,.39,.4,.41,
     1.42,.43,.44,.45,.46,.47,.48,.49,.5/
      DATA (X(I),I=52,101)/.51,.52,.53,.54,.55,.56,.57,.58,.59,.6,.61,
     1.62,.63,.64,.65,.66,.67,.68,.69,.7,.71,.72,.73,.74,.75,.76,.77,.78
     1,.79,.8,.81,.82,.83,.84,.85,.86,.87,.88,.89,.9,.91,.92,.93,.94,
     1.95,.96,.97,.98,.99,1./
      DATA (X(I),I=102,151)/1.01,1.02,1.03,1.04,1.05,1.06,1.07,1.08,
     11.09,1.1,1.11,1.12,1.13,1.14,1.15,1.16,1.17,1.18,1.19,1.2,1.21,
     11.22,1.23,1.24,1.25,1.26,1.27,1.28,1.29,1.3,1.31,1.32,1.33,1.34,
     11.35,1.36,1.37,1.38,1.39,1.4,1.41,1.42,1.43,1.44,1.45,1.46,1.47,
     11.48,1.49,1.5/
      DATA (X(I),I=152,201)/1.51,1.52,1.53,1.54,1.55,1.56,1.57,1.58,
     11.59,1.6,1.61,1.62,1.63,1.64,1.65,1.66,1.67,1.68,1.69,1.7,1.71,
     11.72,1.73,1.74,1.75,1.76,1.77,1.78,1.79,1.8,1.81,1.82,1.83,1.84,
     11.85,1.86,1.87,1.88,1.89,1.9,1.91,1.92,1.93,1.94,1.95,1.96,1.97,
     11.98,1.99,2./
      DATA(X(I),I=202,249)/2.02,2.04,2.06,2.08,2.1,2.12,2.14,2.16,2.18,
     12.2,2.22,2.24,2.26,2.28,2.3,2.32,2.34,2.36,2.38,2.4,2.42,2.44,
     12.46,2.48,2.5,2.55,2.6,2.65,2.7,2.75,2.8,2.85,2.9,2.95,3.,3.1,
     13.2,3.3,3.4,3.5,3.6,3.7,3.8,3.9,4.,4.2,4.4,4.6/
C     DATA TABLE FROM VALUES OF ERROR FUNCTION (Y)
      DATA(Y(I),I=1,51)/1.000000000000,.011283415600,.022564574700,
     1.033841222300,
     1.045111106100,.056371977800,.067621594400,.078857719800,
     1.090078125800,
     1.1012805939,.1124629160,.1236228962,.1347583518,.1458671148,
     1.1569470331,.1679959714,.1790118132,.1899924612,.2009358390,
     1.2118398922,.2227025892,.2335219230,.2442959116,.2550225996,
     1.2657000590,.2763263902,.2868997232,.2974182185,.3078800680,
     1.3182834959,.3286267595,.3389081503,.3491259948,.3592786550,
     1.3693645293,.3793820536,.3893297011,.3992059840,.4090094534,
     1.4187387001,.4283923550,.4379690902,.4474676184,.4568866945,
     1.4662251153,.4754817198,.4846553900,.4937450509,.5027496707,
     1.5116682612,.5204998778/
      DATA(Y(I),I=52,101)/.5292436198,.5378986305,.5464640969,
     1.5549392505,.5633233663,.5716157638,.5798158062,.5879229004,
     1.5959364972,.6038560908,.6116812189,.6194114619,.6270464433,
     1.6345858291,.6420293274,.6493766880,.6566277023,.6637822027,
     1.67084C0622,.6778011938,.6846655502,.6914331231,.6981039429,
     1.7046780779,.7111556337,.7175367528,.7238216140,.7300104313,
     1.7361034538,.7421009647,.7480032806,.7538107509,.7595237569,
     1.7651427115,.7706680576,.7761002683,.7814398455,.7866873192,
     1.7918432468,.7969082124,.8018828258,.8067677215,.8115635586,
     1.8162710190,.8208908073,.8254236496,.8298702930,.8342315043,
     1.8385080696,.8427007929/
      DATA (Y(I),I=102,151)/.8468104962,.8508380177,.8547842115,
     1.8586499465,.8624361061,.8661435866,.8697732972,.8733261584,
     1.8768031019,.8802050696,.8835330124,.8867878902,.8899706704,
     1.8930823276,.8961238429,.8990962029,.9020003990,.9048374269,
     1.9076082860,.9103139782,.9129555080,.9155338810,.9180501041,
     1.9205051843,.9229001283,.9252359418,.9275136293,.9297341930,
```

```
1.9318986327,.9340079449,.9360631228,.9380651551,.9400150262,
1.9419137153,.9437621961,.9455614366,.9473123980,.9490160353,
1.9506732958,.9522851198,.9538524394,.9553761786,.9568572531,
1.9582965696,.9596950256,.9610535095,.9623728999,.9636540654,
1.9648978648,.9661051465/
 DATA (Y(I),I=152,201)/.9672767481,.9684134969,.9695162091,
1.9705856899,.9716227333,.9726281220,.9736026275,.9745470093,
1.9754620158,.9763483833,.9772068366,.9780380884,.9788428397,
1.9796217795,.9803755850,.9811049213,.9818104416,.9824927820,
1.9831525869,.9837904586,.9844070075,.9850028274,.9855784998,
1.9861345950,.9866716712,.9871902752,.9876909422,.9881741959,
1.9886405487,.9890905016,.9895245446,.9899431565,.9903468051,
1.9907359476,.9911110301,.9914724883,.9918207476,.9921562228,
1.9924793184,.9927904292,.9930899398,.9933782251,.9936556502,
1.9939225709,.9941793336,.9944262755,.9946637246,.9948920004,
1.9951114132,.9953222650/
 DATA(Y(I),I=202,249)/.99571950000,.99608580000,.99642350000,
1.99673440000,.99702050000,.99728360000,.99752530000,
1.99774720000,.99795060000,.99813720000,.99830790000,
1.99846420000,.99860710000,.99873770000,.99885680000,
1.99896550000,.99906460000,.99915480000,.99923690000,
1.99931150000,.99937930000,.99944080000,.99949660000,
1.99954720000,.99959300000,.99968930000,
1.99976400000,.99982150000,.99986570000,.99989940000,
1.99992500000,.99994430000,.99995890000,.99996980000,
1.9999779095,.9999883513,.9999939742,
1.9999969423,.9999984780,.9999992569,.9999996441,.9999998328,
1.9999999230,.9999999652,.9999999846,.9999999971,.9999999995,
1.9999999999/
 END
```