GROUND WATER

COMPUTER NOTES

APPROXIMATE AND ANALYTICAL SOLUTIONS FOR SOLUTE TRANSPORT FROM AN INJECTION WELL INTO A SINGLE FRACTURE

by Chia-Shyun Chen[a] and S. R. Yates[b]

Introduction

In dealing with problems related to land-based nuclear waste management, a number of analytical and approximate solutions were developed to quantify radionuclide transport through fractures contained in the porous formation (e.g., Neretnieks, 1980; Rasmuson and Neretnieks, 1981; Tang *et al.*, 1981; Sudicky and Frind, 1982; Barker, 1982; Hodgkinson and Lever, 1983; Rasmuson, 1984; Neretnieks and Rasmuson, 1984; Chen, 1986). By treating the radioactive decay constant as the appropriate first-order rate constant, these solutions also can be used to study injection problems of a similar nature subject to first-order chemical or biological reactions. In these works, the fracture is idealized by a pair of parallel, smooth plates separated by an aperture of constant thickness. Using this macroscopic approach, Chen (1986) gave solutions to different cases regarding the injection of radioactive material into a fractured formation. The planar fracture was assumed to have a constant aperture thickness, 2b, and intersect the well with a radius r_0 (see Figure 1). Water containing radioactive constituents was discharged into the fracture through the well under a constant flow rate of Q. The injected radionuclides moved primarily through the fracture in a steady, radial flow field where the velocity as a function of radial distance, r, is described by

$$V(r) = A/r \qquad (1)$$

where $A = Q/(4\pi b)$ as the advection parameter.

Ground water was assumed to be immobile in the underlying and overlying porous formations due to their low permeabilities. However, the injected radionuclides were able to move from the fracture into the porous matrix by molecular diffusion (the matrix diffusion) due to possible concentration gradients across the interface between the fracture and the porous matrix (i.e., at z = 0). Two models (Models I and II) were studied by Chen (1986). Model I assumed advection and longitudinal dispersion as the transport mechanisms in the fracture, while Model II considered only advection. Both models included matrix diffusion. Solutions of these two models are different under transient conditions but converge to the same solution at steady state for commonly occurring conditions. Compared to the steady-state solutions of Model I, the steady-state solutions of Model II are mathematically simpler and thus are recommended for use when dealing with steady-state conditions of the stated problem. In addition to quantifying a "worst case" scenario, the steady-state solutions can be used to determine the maximum transport distance of the injected radionuclides in the fracture. For time-dependent conditions, however, the transient solutions of Model I are suggested because they are more generalized in the sense that the longitudinal dispersion process in the fracture is taken into account.

These transient and steady-state solutions have potential usefulness for quantitative study of problems where radioactive material is injected into a fractured formation for disposal or for tracer tests. They also can be employed to check the accuracy of portions of pertinent three-dimensional numerical codes; for axial symmetric systems the radial dimension is a combination of the horizontal x and y Cartesian dimension (i.e., $r^2 = x^2 + y^2$), and the matrix diffusion normal to the radial direc-

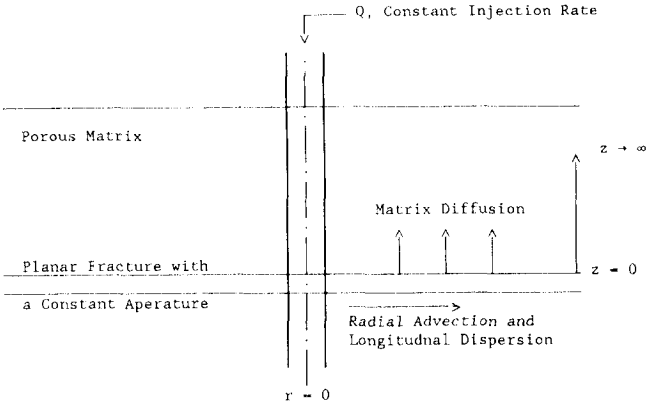

Fig. 1. Schematic of radionuclide transport from an injection well into a single, planar fracture situated in porous formation.

[a] Department of Geosciences, New Mexico Institute of Mining and Technology, Socorro, New Mexico 87801.
[b] U.S. Salinity Laboratory, USDA/ARS, Department of Soil and Environmental Science, University of California, Riverside, California 92521.

Received October 1987, revised April 1988, accepted April 1988.
Discussion open until July 1, 1989.

tion adds the third dimension, z. Consequently, these solutions could be used to check two-dimensional areal flow with matrix diffusions in the vertical direction.

By making use of the Stehfest method (Stehfest, 1970a, b), the transient solutions were determined by numerically inverting the solutions to Model I in the Laplace domain, which involve the transcendental Airy functions. Calculation of the transient solutions is not straightforward, and the purpose of this paper is to document a contained FORTRAN program, which computes the Stehfest inversion, the Airy functions, and gives the concentration distributions in the fracture as well as in the porous matrix for both transient and steady-state cases. A formula determining the maximum transport distance is given here.

Mathematical Model and Solutions

The mathematical model and its solutions are briefly discussed here. Detailed discussions of development of the model and derivation of the solutions are provided in Chen (1986).

The dispersion theory for solute transport in porous media is adopted, and the longitudinal dispersivity in the fracture is assumed to be constant. Hence, the longitudinal dispersion coefficient for the radial flow field neglecting molecular diffusion can be written as

$$D_r = \alpha_1 V \tag{2}$$

where V is the steady-state, radial ground-water velocity described by (1); and α_1 is the constant longitudinal dispersivity.

The governing equations of the model can be formulated as

$$D_m \frac{\partial^2 C_2}{\partial z^2} - \lambda R_2 C_2 = R_2 \frac{\partial C_2}{\partial t} \tag{3}$$

$$\frac{\alpha_1 A}{r} \frac{\partial^2 C_1}{\partial r^2} - \frac{A}{r} \frac{\partial C_1}{\partial r} + \frac{n_2 D_m}{b} \frac{\partial C_2}{\partial z}\bigg|_{z=0} - \lambda R_1 C_1 = R_1 \frac{\partial C_1}{\partial t} \tag{4}$$

where λ is the decay coefficient for the radionuclides (or the first-order rate constant for chemical or biological transformation);
C_1 and C_2 are concentrations in the fracture and in the porous matrix, respectively;
D_m, n_2, and R_2 are, respectively, the effective molecular diffusion coefficient, the porosity, and the retardation factor for the linear-isotherm adsorption in the porous matrix; and
α_1, b, and R_1 are, respectively, the dispersivity, half aperture thickness, and retardation factor in the fracture.

The initial condition for (3) and (4) is

$$C_1(r,0) = C_2(z,0) = 0 \tag{5}$$

which states that no contaminants exist in the system prior to injection.

The boundary condition at the interface of the fracture and the porous matrix is given by the continuity of concentrations as

$$C_1(r,t) = C_2(z,t); \qquad z = 0 \tag{6}$$

as $r \to \infty$ and $z \to \infty$, a bounded condition is prescribed for C_1 and C_2 as

$$C_1(\infty,t) = C_2(\infty,t) \text{ is bounded}; \quad r^2 + z^2 \to \infty \tag{7}$$

Two different boundary conditions for decay and nondecay sources are considered at the well bore. The decay boundary condition is

$$C_1(r_0,t) = C_0 e^{-\lambda t}/C_0 = e^{-\lambda t} \tag{8}$$

which may be relevant to injecting a radioactive substance with a short half-life. Due to the rapid decay, the concentration of the substance in the well bore cannot remain at a constant level but decreases with time following the exponential law as stated in (8).

The nondecay boundary condition, however, may be used if the concentrations at the injection well remain at a constant level because of the long half-life of the injected radioactive materials; that is,

$$C_1(r_0,t) = C_0/C_0 = 1 \tag{9}$$

In fact, if $\lambda t \leqslant 0.01$, the boundary conditions (8) and (9) are approximately equivalent since (8) yields a source concentration which like (9) is approximately equal to unity. Therefore, use of the decay or nondecay condition at the injection well does not cause significant difference in the calculated results provided $\lambda t \leqslant 0.01$.

Transient Solutions by Numerical Inversion

Analytical solutions to (3) and (4) subject to (5) through (8) or (9) can be determined by the Laplace transform technique. In appropriate dimensionless forms, the solutions for the decay boundary condition (8) in the Laplace domain is

$$G_1(\rho,p) = \frac{1}{p+\alpha_1} \exp[(\rho - \rho_0)/2] \frac{Ai[\beta^{1/3} y]}{Ai[\beta^{1/3} y_0]} \tag{10a}$$

$$G_2(\rho,p) = G_1 \cdot \exp[-\xi(p+\alpha_1)^{1/2}] \tag{10b}$$

where G_1 and G_2 denote the concentration distributions in the fracture; and within the porous matrix in the Laplace domain, respectively, p is the Laplace transform parameter of the dimensionless time τ defined by

$$\tau = At/R_1\alpha_1^2$$

and the symbol Ai(x) represents the Airy function. The dimensionless radial distance ρ, the dimensionless vertical distance ξ, and other dimensionless parameters are defined in the Nomenclature.

The analytical Laplace inversion of (10) gives closed form solutions of C_1 and C_2 for the problem. As shown by Chen (1986), however, approximate solutions determined by numerically inverting (10) with the Stehfest method (Stehfest, 1970a, b) yield accurate results for practical purposes. Specifically, C_1 and C_2 for the decay boundary condition are obtained by numerically inverting G_1 and G_2 given in (10) with the following finite series of N terms

$$C_1(\rho,\tau) \cong p \sum_{n=1}^{N} W_n G_1(\rho, np); \quad p = \ln(2)/\tau \quad (11a)$$

$$C_2(\rho,\tau) \cong p \sum_{n=1}^{N} W_n G_2(\rho, np); \quad p = \ln(2)/\tau \quad (11b)$$

During the inversion calculation, p is inversely related to τ, and N must be an even integer. The weighting factors, W_n, are determined with the rational function given by Stehfest (1970a, b). These weighting factors are only dependent on the value of N chosen; that is, they need to be determined only once for any numerical inversions so long as N is fixed. In the computer examples provided in the Appendix, 16 weighting factors (i.e., N = 16) are given. It was found that 16 weighting factors provided sufficiently accurate results on an IBM-AT compatible microcomputer or on a DEC-20 main frame. Double-precision calculations are suggested when using the program. It should be noted that the arguments in the Airy functions are also dependent on p and hence on N and τ (see Nomenclature).

The Airy functions in (10) are calculated using appropriate formulae given by Abramowitz and Stegun (1970). Arguments of the Airy functions in (10) are always positive. The first 16 terms of the power series given by Abramowitz and Stegun (1970, equation 10.4.2) are used to evaluate Ai(x) when $0 \leqslant x < 3$. For the condition, $3 \leqslant x \leqslant 5$, Ai(x) is determined using a two-step procedure. Firstly, the modified Bessel function of the second kind of order $\frac{1}{3}$, $K_{1/3}(x)$, is calculated by the integral formula of equation 9.6.24 in Abramowitz and Stegun (1970). Secondly, the calculated $K_{1/3}(x)$ is converted to Ai(x) using the mathematical identity of equation 10.4.14 in Abramowitz and Stegun (1970). This method of determining Ai(x) for $3 \leqslant x \leqslant 5$ increases the computational stability of the algorithm. For $x > 5$, the first 14 terms of the asymptotic expansion given by equation 10.4.59 in Abramowitz and Stegun (1970) are employed for evaluating Ai(x). If a computer with sufficient precision is available, Ai(x) can be calculated by using the power series in the range $0 \leqslant x \leqslant 5$, and by the asymptotic expansion for $x > 5$ as mentioned above. In this event, the two-step computation for $3 \leqslant x \leqslant 5$ is not required. When $x > 5$, Ai(x) becomes small and can cause exponential underflow problems. Therefore, Ai(x) is scaled by a multiplying factor, $x^{1/4} \exp[(2/3)x^{3/2}]$. To recover the actual value for the Airy function during the calculations, the result is multiplied by $x^{-1/4} \exp[-(2/3)x^{3/2}]$. This approach for evaluating Ai(x) was suggested by Hsieh (1986).

In a similar manner, C_1 for the nondecay boundary condition can be determined by replacing

$$G_1(\rho,p) = (1/p)\exp[(\rho - \rho_0)/2]\,\frac{Ai[\beta^{1/3} y]}{Ai[\beta^{1/3} y_0]} \quad (12)$$

in (11a), and C_2 can be obtained by introducing (12) to (10b) and (11b).

The effect of the nondecay boundary condition is to replace the term $1/(p + \alpha_1)$ in (10) by the term 1/p. The calculation for the nondecay case follows identical procedures as the decay case. Hence, determination of concentration distributions for both the decay and nondecay boundary conditions requires only a slightly different calculation in the program.

Exact Steady-State Solution

Under steady-state conditions (i.e., injection time approaches infinity), the decay boundary condition yields a zero source concentration at the injection well, leading to a trivial solution of zero concentration everywhere in the system. However, nontrivial steady-state solutions exist for $\lambda > 0$ and a nondecay boundary condition; that is,

$$C_1 = \exp[(-E_1\lambda - E_2\lambda^{1/2})\bar{r}] \quad (13a)$$

$$C_2 = C_1 \exp[-z(R_2\lambda/D_m)^{1/2}] \quad (13b)$$

The longitudinal dispersivity is absent in (13) because the longitudinal dispersion in the fracture was neglected. Although Chen (1986) noted that longitudinal dispersion in the fracture could be neglected for steady-state conditions without introducing noticeable error based on one problem, we

have verified that this conclusion is true for general conditions unless the parameter α is greater than approximately 10, which is unreasonably high and would rarely occur for practical problems. Therefore, (13) provides a useful steady-state solution for the stated problem.

The ultimate extent with which the concentration front can move in the fracture can be approximated with (13a). If the concentration front is taken as the location where x percent of the injected concentration takes place, then this ultimate moving distance is approximately equal to

$$r_X = \left[\frac{2 \ln(1/x)}{E_1 \lambda + E_2 \lambda^{1/2}} \right]^{1/2} \qquad (14)$$

which is derived from (13a) by setting C_1 to x and the well radius is neglected. For example, if the frontal concentration is taken as 0.05, then the associated ultimate moving distance is

$$r_{0.05} = 2.5 [E_1 \lambda + E_2 \lambda^{1/2}]^{-1/2} \qquad (15)$$

Examples

To illustrate the solutions contained herein, several hypothetical examples were created. To provide for the implementation of the computer program by future users, the data used to create the examples are reported in Appendix 2. To use the program, which is listed in Appendix 1, aquifer and chemical properties are required. The properties used for the following example are: half aperture thickness (b), well radius (r_0), flow rate into the fracture (Q), dispersivity (α_1), effective diffusion coefficient (D_m), and matrix porosity (n_2), respectively; 5.0×10^{-5} m, 0.1 m, 3.65 m^3/day, 0.1 m, 1.0×10^{-3} m^2/day, and 0.01 m^3/m^3. Other required parameters include the decay coefficient and retardation constant, which are 0.01 day^{-1} and 1.0, respectively. For each calculation, 16 Stehfest weighting coefficients and double precision were used.

Figure 2 shows the concentration distribution as a function of radial distance at several times and for two different boundary conditions at the well. The solid and dotted lines indicate, respectively, the concentration profiles based on the nondecay and decay boundary conditions. For the injection time equal to 0.01 day, the solutions determined by the two different boundary conditions are practically the same (see Figure 2) because the relationship $\lambda t \leq 0.01$ is satisfied. Under steady-state conditions, the solid line calculated by (12) with a large value of time is almost identical as the

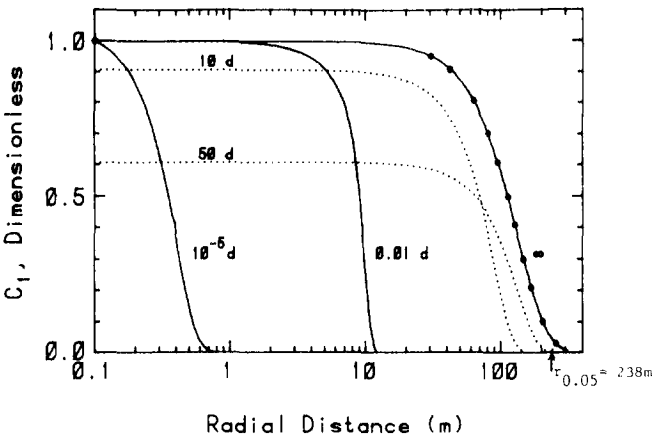

Fig. 2. Concentration with respect to time and radial distance in the fracture. The solid and dotted lines indicate the results from the nondecay and decay cases, respectively. The model coefficients are given in Appendix 2.

dots which resulted from the zero-dispersivity approximation, equation (13). This coincidence indicates that longitudinal dispersion in the fracture is not important for steady-state conditions. The ultimate moving distance, $r_{0.05}$, determined with equation (15), is about 238 m, which is found in Figure 2 by graphic interpolation.

Figure 3 is a diagram of the concentration distributions in the porous matrix for the example contained in Figure 2. In Figure 3a, the concentration profiles of C_2 at a radial distance of 1.0, 5.0, and 10.0 m and a time of 0.01 day is shown. In Figure 2b, the concentration profiles are for steady-state and radial distances of 1.0, 100.0, and 150.0 m. The dots indicate the results from the approximate solution. As was shown for the fracture, the zero-dispersivity approximation produces almost the same results as the more rigorous exact solution for this example.

Figure 4 contains a transient and steady-state contour diagram of the concentration in the fracture and porous matrix. For clarity, the fracture has been enlarged. The dotted line in Figure 4a indicates the position of the well bore. In Figure 4b, again it can be shown that equation (15) is a valid approximation for the ultimate moving distance, $r_{0.05}$.

Nomenclature

Dimensional Parameters

A	advection parameter equal to $Q/(4\pi b)$, m^2/s.
b	half fracture aperture, m.
C_0	concentration at the well bore, kg/m^3.
D_m	effective diffusion coefficient of porous matrix, m^2/s.

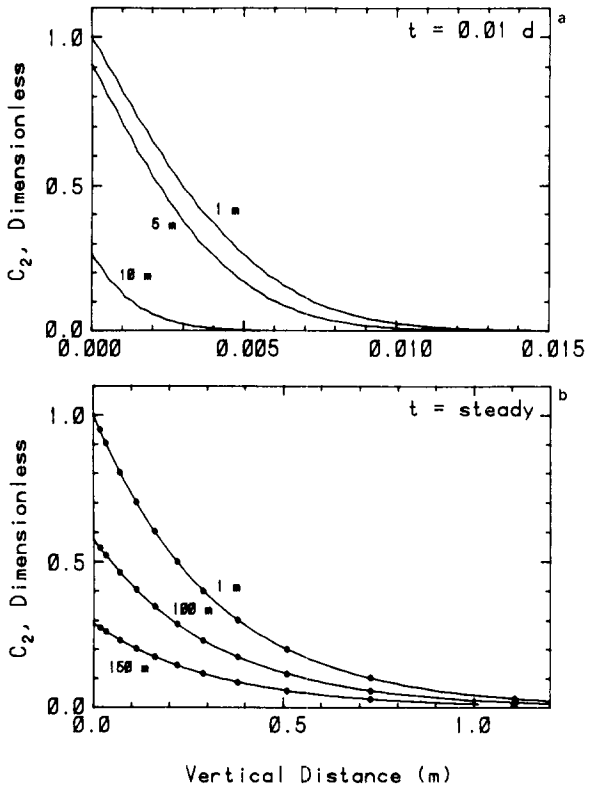

Fig. 3. Concentration in the porous media for times 0.01 d (a) and at steady-state (b). The dots in (b) indicate the results from equation (13). The model coefficients are given in Appendix 2.

D_r	longitudinal dispersion coefficient, m^2/s.
E_1	R_1/A, s/m^2.
E_2	$n_2(R_2D_m)^{1/2}/(bA)$, $s^{1/2}/m^2$.
Q	constant injection rate, m^3/s.
r	radial distance, m.
r_0	well radius, m.
$\bar{r}$	$(r^2 - r_0^2)/2$, m^2.
t	time, s.
V	ground water in fracture defined by (1), m/s.
z	vertical distance in the porous matrix, m.
α_1	dispersivity of fracture, m.
λ	radioactive decay constant or first-order rate constant for chemical or biological reactions, s^{-1}.

Dimensionless Parameters

C_1, C_2	normalized concentration in fracture and in porous matrix, respectively.
n_2	porosity of porous matrix.
R_1, R_2	retardation factors in fracture and in porous matrix.
p	Laplace transform parameter.
$y =$	$\rho + 1/(4\beta)$.
$y_0 =$	$\rho_0 + 1(4\beta)$.
$\alpha =$	$(n_2\alpha_1/b)(R_2D_m/R_1A)^{1/2}$.
$\alpha_1 =$	$R_1\lambda\alpha_1^2/A$.
$\beta =$	$p + \alpha_1 + \alpha(p + \alpha_1)^{1/2}$.
ξ	$(z/\alpha_1)(R_2A/R_1D_m)^{1/2}$, dimensionless vertical distance.
τ	$At/(R_1\alpha_1^2)$, dimensionless time.
ρ	r/α_1, dimensionless radial distance.
ρ_0	r_0/α_1, dimensionless well radius.

Function

Ai(x) Airy function.

Disclaimer

Although a portion of the research described in this article has been funded wholly or in part by the United States Environmental Protection Agency, it has not been subjected to the Agency's peer and administrative review and therefore may not necessarily reflect the views of the Agency, and no official endorsement should be inferred.

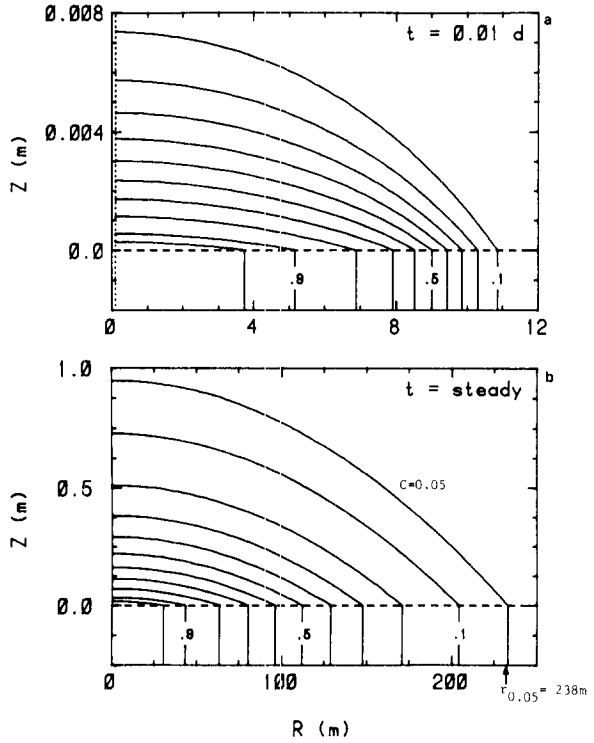

Fig. 4. Contour diagram of the concentration with respect to position and time. For t = 0.01 d (a), the contour levels are: 0.95, 0.9, 0.8, . . . , 0.1. For t = steady-state (b), the contour levels are: 0.95, 0.9, 0.8, . . . , 0.1, and 0.05. The dotted line in (a) indicates the position of the well bore.

```
C***********************************************************************
C
C      THIS PROGRAM COMPUTES THE LAPLACE INVERSION OF THE RADIAL
C      DISPERSION EQUATION FOR VELOCITY DEPENDENT FLOW AND RADIOACTIVE
C      DECAY GIVEN BY CHEN(1986) USING THE LAPLACE INVERTION METHOD OF
C      STEHFEST(1970)
C
C***********************************************************************
C
C   INPUT INFORMATION:
C   ==================
C
C        Input parameters can be provided to the program from either
C        a disk file or the keyboard.  In either case, the parameters
C        that must be supplied are:
C
C   INTERACTIVE INPUT (for opening files)
C   =====================================
C
C       IN          - Input file number.  IN=1 for disk, IN=5 for
C                     keyboard.
C
C       FILE        - IF IN=1, then give the input file name.
C
C
C       IO          - Output unit number.  IO=2 for disk, IO=5 for
C                     terminal, IO=6 for printer.
C
C       FILE        - IF IO=2, then give the output file name.
C
C
C   MODEL INPUT DATA (either from a disk file or interactively)
C   ===========================================================
C
C   RECORD 1:  (free format)
C
C       TITLE(3)    - Three lines of title or problem description.
C
C       IBC         - Steering parameter for the boundary condition at
C                     the well.  If IBC=0; then a decay boundary.
C                     If IBC=1; then a non-decay boundary.
C
C       N           - Number of Stehfest weighting coefficients.  For
C                     IBM-AT compatible computers use between 10 to 16.
C
C       R1          - Retardation coefficient for the fracture surface.
C
C       R2          - Retardation coefficient for the porous matrix.
C
C       D           - Dispersivity of the fracture.
C
C       B           - Fracture aperature thickness.
C
C       N2          - Porosity of the porous matrix.
C
C       Dm2         - Effective diffusion coefficient for the porous
C                     matrix.
C
C       LAM         - Radioactive decay coefficient.
C
C       Q           - Flow into the fracture.
C
C
C   RECORD 2:
C
C       NR          - Number of radial coordinates where a concentration
C                     is to be calculated.
C
C       Ro          - The radius of the wellbore.
C
C       R           - The radial distance where the first concentration
C                     is to be calculated.
C
C       DR          - The distance between consecutive radial distances.
C                     A concentration will be determined at R + (i-1)DR,
C                     for i=1,2,3,...,NR.
C
C
C   RECORD 3:
C
C       NT          - Number of times the concentration is to be calculated.
C
C       T(NT)       - The NT values of time.  The maximum size for this
C                     array is 10.
C
C
C   RECORD 4:
C
C       NZ          - Number of vertical coordinates (in the porous matrix)
C                     where a concentration is to be calculated.  Note: the
C                     total number of concentrations calculated will be:
C                     NR*NZ*NT.
C
C       DZ          - The distance between consecutive vertical distances.
C
C***********************************************************************
C
C   IMPORTANT VARIABLES
C
C       RHOO        - Dimensionless well radius
C       DRHO        - Incremental dimensionless radial distance
C       RHO         - Dimensionless radius
C       DXI         - Incremental dimensionless vertical distance
C       XI          - Dimensionless vertical distance
C       TAU         - Dimensionless time
C       A           - Advection parameter
C       ALF1        - Parameter relating to the radioactive decay
C       ALF         - Parameter relating to the diffusive leakage
C
C***********************************************************************
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION LAM,N2,T(15),V(30),G(30),H(15),XR(30)
     # ,AI0(30),Z0(30)
      CHARACTER FILE*20,TITLE(3)*70,MES1(2)*46,MES2(2)*48
      COMMON IN,IT,IO,IL
      DATA MES1(1)/'Vertical concentrations will not be calculated'/
     #,MES1(2)/'Vertical concentrations will be calculated'/
     #,MES2(1)/'A decay boundary condition exists at the well'/
     #,MES2(2)/'A constant boundary condition exists at the well'/
      DATA INN,IOO,ITT/1,2,5/,IA/30/,IB/15/
C
C     ----- read steering parameters -----
      IT=ITT
      WRITE(IT,900)
      READ(IT,*) IN
      IF(IN.EQ.0) CALL VT100
      IF(IN.EQ.1) CALL VT52
C
      CALL VTPOSI
      WRITE(IT,*)' GIVE INPUT  DEVICE NUMBER (1=dsk, 5=tty) '
      READ(IT,*) IN
      IF(IN.EQ.INN) THEN
      CALL VTPOSI
      WRITE(IT,*)' GIVE INPUT  FILE NAME'
      READ(IT,'(A)') FILE
      OPEN(UNIT=IN,FILE=FILE,STATUS='OLD',MODE='READ')
      ELSE
      ENDIF
C
      CALL VTPOSI
      WRITE(IT,*)' GIVE OUTPUT DEVICE NUMBER (2=dsk, 5=tty, 6=lp)'
      READ(IT,*) IO
      IL=55
      IF(IO.EQ.5) IL=20
      IF(IO.EQ.IOO) THEN
      CALL VTPOSI
      WRITE(IT,*)' GIVE OUTPUT FILE NAME'
      READ(IT,'(A)') FILE
      OPEN(UNIT=IO,FILE=FILE,STATUS='NEW')
      ELSE
      ENDIF
C
C     ----- read in input parameters of the fracture -----
      IF(IN.EQ.1) THEN
      READ(IN,'(A)') (TITLE(I),I=1,3)
      READ(IN,*) IBC
      READ(IN,*) N,R1,R2,D,B,N2,DM2,LAM,Q
      READ(IN,*) NR,RO,R,DR
      READ(IN,*) NT,(T(I),I=1,NT)
      READ(IN,*) NZ,DZ
      ELSE
C
C     ----- interactive input option -----
      CALL INTRAC(TITLE,IBC,N,R1,R2,D,B,N2,DM2,LAM,Q,NR,RO,R,DR,
     #NZ,DZ,NT,T)
      ENDIF
C
C     ----- write out input parameters -----
      IFZN=0
      IF(NZ.GT.0) IFZN=1
      IF(IO.EQ.IT) CALL VTPOS1
      WRITE(IO,905)
      WRITE(IO,910) (TITLE(I),I=1,3)
      WRITE(IO,915)
      WRITE(IO,920) MES1(IFZN+1),MES2(IBC+1)
      IF(D.NE.0.0D0) WRITE(IO,925) N
      IF(D.EQ.0.0D0) WRITE(IO,930)
      IF(IO.EQ.IT) CALL VTWAIT
      WRITE(IO,935) D,R1,B,DM2,R2,N2,LAM,Q
      IF(IO.EQ.IT) CALL VTWAIT
C
C     ----- go to appropriate analytical solution -----
C     -----   model1 if D > 0, otherwise model2   -----
      IF(D.NE.0.0D0) CALL MODEL1(IA,IB,IBC,N,R1,R2,D,B,N2,DM2,LAM,Q
     # ,NR,RO,R,DR,NZ,DZ,NT,T,V,G,H,XR,AI0,Z0)
      IF(D.EQ.0.0D0) CALL MODEL2(IB,IBC,R1,R2,B,N2,DM2,LAM,Q,NR,RO,R
     # ,DR,NZ,DZ,NT,T)
C
C     ----- format statements -----
  900 FORMAT(36(/),' GIVE THE TERMINAL TYPE:',//,5X,'0 = VT100',/,
     # 5X,'1 = VT52',T50,'==> ',$)
  905 FORMAT(/,1X,78(1H*),/,1X,'*',76X,'*')
  910 FORMAT(1X,'*',3X,A,3X,'*')
  915 FORMAT(1X,'*',76X,'*',/,1X,78(1H*))
  920 FORMAT(///,1X,'PROBLEM SPECIFICATIONS',/,1X,22(1H=),//,1X,
     #A46,/,1X,A48)
  925 FORMAT(1X,I2,' Stehfest weighting factors will be used to invert t
     #he Laplace transform')
  930 FORMAT(1X,'The dispersivity of the fracture is zero.  Will use the
     # approximate solution')
  935 FORMAT(///1X,'INPUT PARAMETERS'/,1X,16(1H=),//,
     #1X,'Dispersivity of the fracture [L].............',10(1H.)
     #,1PE13.6,/,
     #1X,'Retardation coefficient for fracture walls [0]',10(1H.)
     #,1PE13.6,/,
     #1X,'Half width of fracture aperature [L].........',10(1H.)
     #,1PE13.6,/,
     #1X,'Diffusion coefficient of porous matrix [L*L/T]',10(1H.)
     #,1PE13.6,/,
     #1X,'Retardation coefficient for porous matrix [0].',10(1H.)
     #,1PE13.6,/,
     #1X,'Porosity of the porous matrix [0]............',10(1H.)
     #,1PE13.6,/,
     #1X,'Radioactive decay constant [1/T].............',10(1H.)
     #,1PE13.6,/,
     #1X,'Constant injection rate [L*L*L/T]............',10(1H.)
     #,1PE13.6)
      END
C
C***********************************************************************
C
C      SUBROUTINE MODEL1 -- CALCULATES THE LAPLACE INVERSION SOLUTION
C                           OF Chen (1985) WHEN THE DISPERSIVITY IS
C                           GREATER THAN ZERO.
C
C***********************************************************************
      SUBROUTINE MODEL1(IA,IB,IBC,N,R1,R2,D,B,N2,DM2,LAM,Q,NR,RO
     # ,R,DR,NZ,DZ,NT,T,V,G,H,XR,AI0,Z0)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
```

```
      DOUBLE PRECISION LAM,N2,V(IA),XR(IA),T(IB),AI0(IA),Z0(IA),G(IA),
     #H(IB)
      COMMON IN,IT,IO,IL
      COMMON /ARGU/ A13,ALF,BETA,BETA3
C
C     ----- calculate the problem constants -----
      RHO0 = R0/D
      DRHO = DR/D
      RHOI = R/D
      PI   = 2.0D0*DATAN(1.D+30)
      A    = Q/(4.0D0*PI*B)
      ALF  = N2*D*DSQRT(R2*DM2/(R1*A))/B
      ALF1 = R1*LAM*D*D/A
      DXI  = DZ*DSQRT(R2*A/(R1*DM2))/D
      A13=1.D0/3.D0
      A23=2.0D0/3.0D0
      Aln2=DLOG(2.D0)
C
C     ----- print out calculated parameters -----
      WRITE(IO,900) A,RHO0,DRHO,ALF,ALF1
      IF(NZ.GT.0) WRITE(IO,905) DXI
C
C     ----- determine the Stehfest weighting coefficients -----
      IF(T(1).LT.0.0D0.AND.NT.EQ.1) GOTO 15
      CALL LINV(IA,IB,N,V,G,H)
      IF(IO.EQ.IT) CALL VTWAIT
      WRITE(IO,910)
      DO 10 I=1,N/2
      II = N/2 + I
      WRITE(IO,915) I,V(I),II,V(II)
   10 CONTINUE
      CALL VTWAIT
C
C     ----- calculate a concentration profile for each time -----
   15 IF(IO.EQ.IT) CALL VTPOSI
      WRITE(IO,920)
      IF(IO.EQ.IT) CALL VTWAIT
C
C     ----- new time -----
      DO 20 K=1,NT
      TAU = A*T(K)/(R1*D*D)
      RHO = RHOI
C
      IF(IO.EQ.IT) CALL VTPOS1
      IF(T(K).LT.0.0D0) WRITE(IO,925)
      IF(T(K).GE.0.0D0) WRITE(IO,930) T(K),TAU
      WRITE(IO,935)
C
C     ----- determine AiO and Z0 (only once) -----
      IF(T(K).LT.0.0D0) THEN
      IF(IBC.EQ.0) WRITE(5,940)
      IF(IBC.EQ.0) RETURN
      IF(LAM.EQ.0.0D0) WRITE(IO,945)
      IF(LAM.EQ.0.0D0) GOTO 20
C
C     ----- find values for steady state case -----
      Y0  = ARG(ALF1,RHO0)
      Z0(1) = A23*(Y0)**1.5D0
      IOPT=1
      IF(Y0.LT.3.0D0) IOPT=-1
      IF(Y0.GE.5.0D0) IOPT= 0
      AI0(1) = AI(Y0,IOPT)
      ELSE
C
C     ----- find values for time-dependent case -----
      YMN = ARG(DBLE(FLOAT(N))*Aln2/TAU+ALF1,RHO0)
      YMX = ARG(Aln2/TAU+ALF1,RHO0)
      IOPT=1
      IF(YMX.LT.3.0D0) IOPT=-1
      IF(YMN.GT.5.0D0) IOPT= 0
C
      DO 25 L=1,N
      PA1=DBLE(FLOAT(L))*Aln2/TAU+ALF1
      Y0 = ARG(PA1,RHO0)
      Z0(L) = A23*Y0**1.5D0
      AI0(L)= AI(Y0,IOPT)
   25 CONTINUE
      ENDIF
C
C     ----- calculate the concentrations in the fracture -----
      LO=0
      DO 30 IR=1,NR
      DD=0.5D0*(RHO-RHO0)
      XP=0.D0
      IF(T(K).LT.0.0D0) THEN
C
C     ----- calculate the steady-state values -----
      Y  = ARG(ALF1,RHO)
      IF(Y.LT.5.D0) THEN
      Z = A23*(Y)**1.5D0
      C1=DEXP(DD+Z0(1)-Z)*AI(Y,1)*(Y0/Y)**0.25D0/AI0(1)
      ELSE
      DXP=(Y0/Y)**.25D0
      XP = A23*(Y0**1.5D0-Y**1.5D0)
      C1=DXP*DEXP(DD+XP)
      ENDIF
      ELSE
C
C     ----- calculate the time-dependent values -----
      YMN = ARG(DBLE(FLOAT(N))*Aln2/TAU+ALF1,RHO)
      YMX = ARG(Aln2/TAU+ALF1,RHO)
      IOPT=1
      IF(YMX.LT.3.0D0) IOPT=-1
      IF(YMN.GT.5.0D0) IOPT= 0
C
C     ----- Stehfest numerical integration method -----
      DO 35 L=1,N
      PA1 = DBLE(FLOAT(L))*Aln2/TAU+ALF1
      Y   = ARG(PA1,RHO)
      Z   = A23*Y**1.5D0
      FACT = DEXP(DD+Z0(L)-Z)*(Z0(L)/Z)**.25
C
C     ----- if IBC=1, C=1.0 at the well bore -----
      IF(IBC.EQ.1) PA1=PA1-ALF1
      AIFN = AI(Y,IOPT)
      XR(L)=V(L)*((AIFN*FACT)/(AI0(L)*PA1))
      XP = XP + XR(L)
   35 CONTINUE
      C1=XP*Aln2/TAU
      ENDIF
C
C     ----- print out result -----
      R = D*RHO
      LO=LO+1
      IF(C1.LT.0.0D0) C1=0.0D0
      WRITE(IO,950) LO,R,0.0D0,C1
      IF(LO.EQ.NR .AND. NZ.EQ.0) GOTO 30
      IF(FLOAT(LO/IL).EQ.FLOAT(LO)/FLOAT(IL)) CALL VTWAIT
      IF(FLOAT(LO/IL).EQ.FLOAT(LO)/FLOAT(IL)) WRITE(IO,935)
C
C     ----- calculate concentration in porous matrix -----
      XI=DXI
      DO 40 IZ=1,NZ
      IF(T(K).LT.0.0D0) THEN
C
C     ----- calculate steady state concentraton -----
      C2=C1*DEXP(-XI*SQRT(ALF1))
      Z = D*XI/DSQRT(R2*A/(R1*DM2))
      ELSE
C
C     ----- calculate time-dependent concentration -----
      ZP=0.0D0
      DO 45 L=1,N
      PA1 = L*Aln2/TAU+ALF1
      Z2  = DEXP(-XI*DSQRT(PA1))
      ZP  = ZP + Z2*XR(L)
   45 CONTINUE
      C2=ZP*Aln2/TAU
      ENDIF



C     ----- print out results -----
      Z = D*XI/DSQRT(R2*A/(R1*DM2))
      LO=LO+1
      IF(C2.LT.0.0D0) C2=0.0D0
      WRITE(IO,950) LO,R,Z,C2
      IF(LO.EQ.NR*NZ) GOTO 40
      IF(FLOAT(LO/IL).EQ.FLOAT(LO)/FLOAT(IL)) CALL VTWAIT
      IF(FLOAT(LO/IL).EQ.FLOAT(LO)/FLOAT(IL)) WRITE(IO,935)
   40 XI =XI+DXI
   30 RHO=RHO+DRHO
   20 IF(K.NE.NT) CALL VTWAIT
C
C     ----- format statements -----
  900 FORMAT(///1X,'CALCULATED PARAMETERS'/1X,22(1H=)//
     #1X,'Advection parameter (A).....................',10(1H.),
     #1PE13.6,/,
     #1X,'Dimensionless radius of the well (RHO0).....',10(1H.)
     #,1PE13.6,/,
     #1X,'Dimensionless distance between radii (DRHO)...',10(1H.)
     #,1PE13.6,/,
     #1X,'Ratio of diffusive loss to injection (ALPHA)..',10(1H.)
     #,1PE13.6,/,
     #1X,'Dimensionless radioactive decay constant (ALPHA1)',7(1H.)
     #,1PE13.6)
  905 FORMAT(1X,'Dimensionless vertical spacing (DXI).........'
     # ,10(1H.),1PE13.6)
  910 FORMAT(///1X,'STEHFEST WEIGHTING FACTORS'/1X,27(1H=)//
     #5X,'I',12X,'V(I)',27X,'II',11X,'V(II)')
  915 FORMAT(1X,I5,1PE20.7,20X,I5,1PE20.7)
  920 FORMAT(16X,'CONCENTRATION DISTRIBUTION'
     #,/,16X,26(1H=),//)
  925 FORMAT(5X,'Time = Steady State',/)
  930 FORMAT(5X,'Time = ',1PE15.5,10X,'Tau = ',1PE15.5,/)
  935 FORMAT(5X,'I',14X,'R',16X,'Z',15X,'C/Co')
  940 FORMAT(5X,'ERROR: IBC must = 1 for a steady state solution')
  945 FORMAT(5X,'ERROR: Lambda cannot be zero for a steady state
     # solution.',/,12X,'The concentration is 1.0 for X < infinity',/)
  950 FORMAT(1X,I5,5X,F12.3,5(5X,F12.4))
      RETURN
      END
C
C*****************************************************************
C
C       SUBROUTINE NAME: MODEL2 -- THIS PROGRAM CALCULATES THE SOLUTION
C                                  OF Chen (1986) WHEN THE DISPERSIVITY
C                                  IS ZERO.
C
C*****************************************************************
      SUBROUTINE MODEL2(IB,IBC,R1,R2,B,N2,DM2,LAM,Q,NR,R0,R
     # ,DR,NZ,DZ,NT,T)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION LAM,N2
      DIMENSION T(IB)
      COMMON IN,IT,IO,IL
C
C     ----- calculate problem constants -----
      PI = 2.0D0*DATAN(1.D+30)
      A  = Q/(4.0D0*PI*B)
      E1 = R1/A
      E2 = N2*DSQRT(R2*DM2)/A/B
      RI = R
      WRITE(IO,900) A,E1,E2
      CALL VTWAIT
      IF(IO.EQ.IT) CALL VTPOSI
      WRITE(IO,905)
      IF(IO.EQ.IT) CALL VTWAIT
C
C     ----- calculate concentration for each time -----
      DO 10 K=1,NT
      IF(IO.EQ.IT) CALL VTPOS1
      IF(T(K).LT.0.0D0) WRITE(IO,910)
      IF(T(K).GE.0.0D0) WRITE(IO,915) T(K)
      WRITE(IO,920)
C
C     ----- calculate the concentration in the fracture -----
      Z=0.0D0
      R=RI
      LO=0
C
      DO 15 IR=1,NR
      RR=R*R/2.-R0*R0/2.
      Z=0.0D0
      IF(T(K).LT.0.0D0) THEN
C
```

```
C     ----- steady-state solution -----
      ARG1 = -E1*LAM*RR - E2*DSQRT(LAM)*RR
      IF(IBC.EQ.0.AND.LAM.NE.0.0D0) C1 = 0.0D0
      IF(IBC.NE.0.AND.LAM.NE.0.0D0) C1=DEXP(ARG1)
      IF(IBC.NE.0.AND.LAM.EQ.0.0D0) C1=1.0D0
      ELSE
C
C     ----- time-dependent solution -----
      T1=T(K)-E1*RR
      IF(T1.LE.0.0) GOTO 25
      ARG1=E2*RR/DSQRT(T1)/2.0
      ARG2=DSQRT(LAM*T1)
      EXP1=E2*RR*DSQRT(LAM)
      EXP2=E1*RR*LAM
C
C     ----- calculation for a decay boundary condition -----
      IF(IBC.EQ.1) GOTO 20
      C1=DEXF(-LAM*T(K),ARG1)
      GOTO 30
C
C     ----- calculation for a non-decay boundary condition -----
   20 C1=0.5D0*(DEXF(-EXP1-EXP2,ARG1-ARG2)
     # +DEXF(EXP1-EXP2,ARG1+ARG2))
      GOTO 30
   25 C1=0.0
   30 CONTINUE
      ENDIF
C
C     ----- print out results -----
      LO=LO+1
      WRITE(IO,925) LO,R,Z,C1
      IF(LO.EQ.NR .AND. NZ.EQ.0) GOTO 35
      IF(FLOAT(LO/IL).EQ.FLOAT(LO)/FLOAT(IL)) CALL VTWAIT
      IF(FLOAT(LO/IL).EQ.FLOAT(LO)/FLOAT(IL)) WRITE(IO,920)
C
C     ----- calculate concentration in porous media -----
   35 Z=DZ
      DO 40 N=1,NZ
      IF(T(K).LT.0.0D0) THEN
C
C     ----- steady-state solution (only non-decay boundary allowed) -----
      IF(IBC.NE.0) C2 = DEXP(ARG1 - Z*DSQRT(R2*LAM/DM2))
      ELSE
C
C     ----- time-dependent solution -----
      IF (T1.LE.0.0) GOTO 50
      ZZ=Z*DSQRT(R2/DM2)
      ARG1=(E2*RR+ZZ)/DSQRT(T1)/2.0
      IF(IBC.EQ.1) GOTO 45
C
C     ----- decay boundary condition -----
      C2=DEXF(-LAM*T(K),ARG1)
      GOTO 55
C
C     ----- non-decay boundary condition -----
   45 C2=0.5D0*(DEXF(-EXP1-EXP2-DSQRT(LAM)*ZZ,ARG1-ARG2)
     #     +DEXF(EXP1-EXP2-DSQRT(LAM)*ZZ,ARG1+ARG2))
      GOTO 55
   50 C2=0.0
   55 CONTINUE
      ENDIF
C
C     ----- print out results -----
      LO=LO+1
      WRITE(IO,925) LO,R,Z,C2
      IF(LO.EQ.NR*NZ) GOTO 40
      IF(FLOAT(LO/IL).EQ.FLOAT(LO)/FLOAT(IL)) CALL VTWAIT
      IF(FLOAT(LO/IL).EQ.FLOAT(LO)/FLOAT(IL)) WRITE(IO,920)
   40 Z=Z+DZ
   15 R=R+DR
   10 IF(K.NE.NT) CALL VTWAIT
      RETURN
C
C     ----- format statements -----
  900 FORMAT(///1X,'CALCULATED PARAMETERS'/1X,22(1H=)//
     #1X,'Advection parameter (A).....................',10(1H.),
     #1PE13.6,/,
     #1X,'Ratio of retardation in fracture to "A" (E1)..',10(1H.)
     #,1PE13.6,/,
     #1X,'Factor E2...................................',10(1H.)
     #,1PE13.6)
  905 FORMAT(16X,'CONCENTRATION DISTRIBUTION'
     #,/,16X,26(1H=),//)
  910 FORMAT(5X,'Time = Steady State',/)
  915 FORMAT(5X,'Time = ',1PE15.5,/)
  920 FORMAT(5X,'I',14X,'R',16X,'Z',15X,'C/Co')
  925 FORMAT(1X,I5,5X,F12.3,5(5X,F12.4))
      END
C
C*****************************************************************************
C
C       FUNCTION ARG -- CALCULATES THE ARGUMENT FOR THE AIRY FUNCTION
C
C*****************************************************************************
      DOUBLE PRECISION FUNCTION ARG(P,R)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON /ARGU/ A13,ALF,BETA,BETA3
C
      BETA   = P+ALF*DSQRT(P)
      BETA3  = BETA**A13
      ARG = BETA3*(R + 0.25D0/BETA)
      RETURN
      END
C
C*****************************************************************************
C
C     SUBROUTINE DEXF -- EVALUATES EXP(A)ERFC(B) IN DOUBLE PRECISION
C
C*****************************************************************************
      DOUBLE PRECISION FUNCTION DEXF(A,B)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DATA P/.3275911D0/,A1/.254829592D0/,A2/.28449673600/
     # ,A3/1.421413741D0/,A4/1.453152027D0/,A5/1.061405429D0/
C
      DEXF=0.0D0
      IF((DABS(A).GT.82.D0).AND.(B.LE.0.0D0)) RETURN
      IF(B.NE.0.0) GOTO 10
      DEXF=DEXP(A)
      RETURN
   10 C=A-B*B
      IF((DABS(C).GT.82.D0).AND.(B.GT.0.D0)) RETURN
      IF(C.LT.-82.0D0) GOTO 25
      X=DABS(B)
      IF(X.GT.3.0D0) GOTO 15
      T = 1.D0/(1.0D0+P*X)
      Y = T*(A1-T*(A2-T*(A3-T*(A4-A5*T))))
      GOTO 20
   15 Y = .564189600/(X+.5D0/(X+1.D0/(X+1.5D0/(X+2.D0/(X+2.5D0/(X+1.D0
     #))))))
   20 DEXF = Y*DEXP(C)
   25 IF(B.LT.0.0D0) DEXF = 2.D0*DEXP(A)-DEXF
      RETURN
      END
C
C*****************************************************************************
C
C     FUNCTION AI(ZA,IOPT) -- THIS FUNCTION SUBROUTINE COMPUTES THE
C                             AIRY FUNCTION FOR POSITIVE ARGUMENTS.
C
C     IF IOPT = -1, USE THE SMALL ARGUMENT SERIES SOLUTION
C     IF IOPT =  0, USE THE LARGE ARGUMENT SERIES SOLUTION
C     IF IOPT =  1, USE THE INTEGRAL SOLUTION METHOD
C
C     THE AIRY FUNCTION IS SCALED (MULTIPLIED) BY:
C
C        (Z**0.25)*EXP(U), WHERE U=(2./3.)*(Z**1.5)
C
C*****************************************************************************
      DOUBLE PRECISION FUNCTION AI(ZA,IOPT)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION XG(10),WG(10)
      COMMON IN,IT,IO,IL
      DATA C1,C2/.3550280538878D0,.2588194037928D0/
      DATA COEF1,COEF2,COEF3,COEF4,COEF5,COEF6/9.555526226877D-29,
     1   4.2356055970200-32,1.661021802753D-35,1.0135782122940-29,
     2   4.3094311747180-33,1.6249740477820-36/
      DATA V/.3333333333333333D0/,PISQR3/5.441398093D0/,
     # P1RT2/3.5449077018D0/,P1/3.1415926535900D0/,P1D4/7.853981634D-1/
     #,A23/.6666666666666667D0/
      DATA NG/10/,XG/.7652652113349733D-1,.2277858511416450D0,
     #.3737060887154195D0,.5108670019508270D0,.6360536807265150D0,
     #.7463319064601507D0,.8391169718222188D0,.9122344282513259D0,
     #.9639719272779137D0,.9931285991850949D0/
      DATA WG/.1527533871307258D0,.1491729864726037D0,
     #.1420961093183820D0,.1316886384491766D0,.1181945319615184D0,
     #.1019301198172404D0,.8327674157670474D-1,.6267204833410906D-1,
     #.4060142980038694D-1,.1761400713915211D-1/
C
C     ----- function statements -----
      FN(Y) = DEXP(-ZK*DCOSH(Y))*DCOSH(V*Y)
      DACOSH(Y) = DLOG(Y+DSQRT(Y*Y-1.0D0))
C
      IF(ZA.LT.0.D0) WRITE(IO,900)
      IF(ZA.LT.0.D0) STOP
      IF(IOPT) 10,20,30
C
C     ----- series expansion for Ai(ZA)  (for 0.0 =< ZA < 3.0) -----
   10 P=ZA**3
      F=1.D0+P*( 1.666666666667D-01+P*( 5.555555555560-03+
     1       P*( 7.716049382716D-05+P*( 5.8454919566030-07+
     2       P*( 2.783567598382D-09+P*( 9.096626138505D-12+
     3       P*( 2.165863366311D-14+P*( 3.923665518679D-17+
     4       P*( 5.589267120625D-20+P*( 6.424444966235D-23+
     5       P*( 6.083754702874D-26+P*( 4.828376748313D-29+
     6       P*( 3.258014000211D-32+P*( 1.891994192922D-35+
     7       P*(      1.0D-10*COEF1+P*(      1.0D-10*COEF2+
     8       P*(      1.0D-10* COEF3)))))))))))))))
      G=ZA*(1.D0+P*( 8.333333333333D-02+P*( 1.9841269841270-03+
     1          P*( 2.2045855379190-05+P*( 1.4131958576400-07+
     2          P*( 5.8883160735010-10+P*( 1.7217298460530-12+
     3          P*( 3.7266879784700-15+P*( 6.2111466307830-18+
     4          P*( 8.2158024216700-21+P*( 8.8341961523330-24+
     5          P*( 7.8736151090320-27+P*( 5.9111224542280-30+
     6          P*( 3.7891810604030-33+P*( 2.0981068994480-36+
     7          P*(      1.0D-10*COEF4+P*(      1.0D-10*COEF5+
     8          P*(      1.0D-10*COEF6))))))))))))))))
      AI=C1*F-C2*G
      U=A23*ZA**1.5D0
      AI=(ZA**0.25D0)*DEXP(U)*AI
      RETURN
C
C     ----- asymptotic expansion for Ai(ZA)  (for ZA > 5.0D0) -----
   20 ZK=A23*ZA**1.5D0
      P=1.D0/ZK
      A=1.D0+P*(-6.944444444444D-02+P*( 3.713348765432D-02+
     1       P*(-3.799305912780D-02+P*( 5.764919041267D-02+
     2       P*(-1.160990640255D-01+P*( 2.915913992307D-01+
     3       P*(-8.776669695100D-01+P*( 3.079453030173D+00+
     4       P*(-1.234157333235D+01+P*( 5.562278536591D+01+
     5       P*(-2.784650807776D+02+P*( 1.533169432013D+03+
     6       P*(-9.207206599726D+03))))))))))))
      AI=A/P1RT2
      RETURN
C
C     ----- integral representation for Ai(ZA) (for 3.0 =< ZA =< 5.0) -----
   30 ZK = 2.0D0*(ZA**1.5D0)/3.0D0
      TMP = 70.D0/ZK
      IF(TMP.LE.1.0D0) THEN
      XL=20.0D0/ZK
      ELSE
      XL = DACOSH(TMP)
      ENDIF
C
      BA2 = XL/2.D0
      SUM = 0.0D+0
      SUM1 = 0.0D+0
C
      DO 35 I=1,NG
      Y =  BA2*(XG(I) + 1.0D0)
      Y1= -BA2*(XG(I) - 1.0D0)
      SUM = SUM + WG(I)*FN(Y)
      SUM1 = SUM1 + WG(I)*FN(Y1)
   35 CONTINUE
C
      SUM = BA2*(SUM+SUM1)
      AI = DEXP(ZK)*(ZA**C.75D0)*SUM/PISQR3
```

```
      RETURN
C
900   FORMAT (' *** WARNING *** SUBROUTINE AI(Z) WILL NOT EVALUATE A
     # NON-POSITIVE ARGUMENT OF AI(Z).')
      END
C
C*********************************************************************
C
C     SUBROUTINE LINV  --  FINDS THE STEHFEST WEIGHTING COEFFICIENTS
C
C*********************************************************************
      SUBROUTINE LINV(IA,IB,N,V,G,H)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION G(IA),V(IA),H(IB)
C
      G(1)=1.DO
      NH=N/2
      DO 10 I=2,N
      G(I)=G(I-1)*DBLE(I)
   10 CONTINUE
      H(1)=2.DO/G(NH-1)
C
      DO 20 I=2,NH
      FI=DBLE(I)
      IF(I.EQ.NH) GOTO 15
      H(I)=(FI**NH)*G(2*I)/(G(NH-I)*G(I)*G(I-1))
      GOTO 20
   15 H(I)=(FI**NH)*G(2*I)/(G(I)*G(I-1))
   20 CONTINUE
      ISN=2*(NH-(NH/2)*2)-1
C
      DO 25 I=1,N
      V(I)=0.DO
      K1=(I+1)/2
      K2=I
      IF(K2.GT.NH)K2=NH
      DO 40 K=K1,K2
      IF(2*K-I.EQ.0) GOTO 30
      IF(I.EQ.K) GOTO 35
      V(I)=V(I)+H(K)/(G(I-K)*G(2*K-I))
      GOTO 40
   30 V(I)=V(I)+H(K)/G(I-K)
      GOTO 40
   35 V(I)=V(I)+H(K)/G(2*K-I)
   40 CONTINUE
      V(I)=ISN*V(I)
      ISN=-ISN
   25 CONTINUE
      RETURN
      END
C
C*********************************************************************
C
C     SUBROUTINE INTRAC -- ALLOWS INTERACTIVE INPUT
C
C     (This subroutine and the calling statement in the main program
C     can be removed if interactive input is not required)
C*********************************************************************
C
      SUBROUTINE INTRAC(TITLE,IBC,N,R1,R2,D,B,N2,DM2,LAM,Q,NR,RO
     # ,R,DR,NZ,DZ,NT,T)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION N2,LAM
      DIMENSION T(10)
      CHARACTER TITLE(3)*70
      COMMON IN,IT,IO,IL
C
  800 FORMAT(1X,A50,$)
  805 FORMAT(1X,A50)
      CALL VTPOSI
      DO 10 I=1,3
      WRITE(IT,805)' GIVE A LINE OF TITLE                              '
   10 READ(IT,'(A)') TITLE(I)
      CALL VTPOSI
      WRITE(IT,805)' GIVE 0: for DECAYING BOUNDARY CONDITION           '
      WRITE(IT,800)' or   1: for CONSTANT CONCENTRATION BOUNDARY  ==> '
      READ(IT,*) IBC
      CALL VTPOSI
      WRITE(IT,800)' GIVE THE NUMBER OF WEIGHTING FACTORS [N]     ==> '
      READ(IT,*) N
      CALL VTPOSI
      WRITE(IT,800)' GIVE RETARDATION FACTOR [FRACTURE: R1]       ==> '
      READ(IT,*) R1
      CALL VTPOSI
      WRITE(IT,800)' GIVE RETARDATION FACTOR [POROUS MATRIX: R2]  ==> '
      READ(IT,*) R2
      CALL VTPOSI
      WRITE(IT,800)' GIVE DISPERSIVITY IN THE FRACTURE [d]        ==> '
      READ(IT,*) D
      CALL VTPOSI
      WRITE(IT,800)' GIVE HALF FRACTURE APERATURE DIMENSION [b]   ==> '
      READ(IT,*) B
      CALL VTPOSI
      WRITE(IT,805)' GIVE POROSITY OF POROUS MATRIX [n ]               '
      WRITE(IT,800)'                                 2            ==> '
      READ(IT,*) N2
      CALL VTPOSI
      WRITE(IT,805)' GIVE DIFFUSION COEFFICIENT IN MATRIX [Dm ]        '
      WRITE(IT,800)'                                       2      ==> '
      READ(IT,*) DM2
      CALL VTPOSI
      WRITE(IT,800)' GIVE RADIOACTIVE DECAY CONSTANT [lambda]     ==> '
      READ(IT,*) LAM
      CALL VTPOSI
      WRITE(IT,800)' GIVE THE INJECTION RATE [Q]                  ==> '
      READ(IT,*) Q
      CALL VTPOSI
      WRITE(IT,805)' GIVE THE NUMBER OF RADII [NT], WELL RADIUS [RO], '
      WRITE(IT,805)' START RADIUS [R] AND DISTANCE BETWEEN RADII [DR] '
      WRITE(IT,800)'                                              ==> '
      READ(IT,*) NR,RO,R,DR
      CALL VTPOSI
      WRITE(IT,805)' GIVE THE NUMBER OF TIMES THE CONCENTRATION        '
      WRITE(IT,800)' PROFILE IS TO BE CALCULATED [NT]             ==> '
      READ(IT,*) NT
      CALL VTPOSI
      DO 15 I=1,NT
      CALL VTPOSI
      WRITE(IT,810) I
   15 READ(IT,*) T(I)
  810 FORMAT(1X,' GIVE THE ',I2,'th TIME [T(I)]',20X,'==> ',$)
      CALL VTPOSI
      WRITE(IT,805)' GIVE THE NUMBER OF VERTICAL POSITIONS WHERE A     '
      WRITE(IT,805)' POROUS MATRIX CONCENTRATION IS TO BE CALCULATED   '
      WRITE(IT,800)'                                                ==> '
      READ(IT,*) NZ
      IF(NZ.EQ.0) RETURN
      WRITE(IT,800)' GIVE THE SPACING BETWEEN VERTICAL POSITIONS  ==> '
      READ(IT,*) DZ
      RETURN
      END
C
C
C*********************************************************************
C
C     SUBROUTINES VT*** -- VIDEO DRIVERS FOR VT-100 AND VT-52
C
C*********************************************************************
      SUBROUTINE VT100
      CHARACTER*1 ESC
      DATA ESC /#1B/
      WRITE(5,900) ESC
  900 FORMAT ('+',1A1,'<')
      RETURN
      END
C
      SUBROUTINE VT52
      CHARACTER*1 ESC
      DATA ESC /#1B/
      WRITE(5,900) ESC
  900 FORMAT ('+',1A1,'[?2l')
      RETURN
      END
C
      SUBROUTINE VTPOSI
      CHARACTER*1 ESC
      CHARACTER CMD1*5,CMD2*3
      DATA ESC /#1B/,ILINE/8/,ICOL/1/
      DATA CMD1 /'[1;1f'/,
     # CMD2 /'[2J]'/
C
      WRITE(5,900) ESC,CMD1,ESC,CMD2
      WRITE(5,905)  ESC,ILINE,ICOL
      RETURN
C
      ENTRY VTPOS1
      WRITE(5,900) ESC,CMD1,ESC,CMD2
  900 FORMAT ('+',A,A,A,A,A,$)
  905 FORMAT ('$',A1,'[',I2.2,';',I3.3,'f')
      RETURN
      END
C
      SUBROUTINE VTWAIT
      CHARACTER*1 ESC
      CHARACTER CMD1*5,CMD2*3
      COMMON IN,IT,IO,IL
      DATA ESC /#1B/,ILINE/24/,ICOL/1/
      DATA CMD1 /'[1;1f'/,
     # CMD2 /'[2J]'/
C
C     ----- if output device is the printer -----
      IF(IO.EQ.IT) GOTO 10
      WRITE(IO,900)
      RETURN
C
C     ----- if output device is the terminal -----
   10 WRITE(5,905)  ESC,ILINE,ICOL
      WRITE(IO,910)
      READ(IT,915) TMP
      WRITE(5,920) ESC,CMD1,ESC,CMD2
      RETURN
C
  900 FORMAT('1')
  905 FORMAT('$',A1,'[',I2.2,';',I3.3,'f')
  910 FORMAT('+ Type return to continue >>> ',$)
  915 FORMAT(G1.0)
  920 FORMAT('+',A,A,A,A,$)
      END
```

Appendix 2. Examples of Program Input and Output

```
     Example input data set
1
16 1.0 1.0 0.1 5.0E-5 .01 1.0E-3 .01 3.65
10 0.1 1.0 1.0
1  .01
5 .002
```

```
*************************************************************************
*                                                                       *
*                                                                       *
*       Example input data set                                          *
*                                                                       *
*                                                                       *
*************************************************************************

PROBLEM SPECIFICATIONS
======================

Vertical concentrations will be calculated
A constant boundary condition exists at the well
16 Stehfest weighting factors will be used to invert the Laplace transform
```

INPUT PARAMETERS
==========

```
Dispersivity of the fracture [L]..... 1.000000E-01
Retardation coefficient for fracture walls [0]..... 1.000000E+00
Half width of fracture aperature [L]..... 5.000000E-05
Diffusion coefficient of porous matrix [L*L/T]..... 1.000000E-03
Retardation coefficient for porous matrix [0]..... 1.000000E+00
Porosity of the porous matrix [0]..... 1.000000E-02
Radioactive decay constant [1/T]..... 1.000000E-02
Constant injection rate [L*L*L/T]..... 3.650000E+00
```

CALCULATED PARAMETERS
==========

```
Advection parameter (A)..... 5.809155E+03
Dimensionless radius of the well (RHOO)..... 1.000000E+00
Dimensionless distance between radii (DRHO)..... 1.000000E+01
Ratio of diffusive loss to injection (ALPHA)..... 8.298001E-03
Dimensionless radioactive decay constant (ALPHA1)..... 1.721421E-08
Dimensionless vertical spacing (DXI)..... 4.820438E+01
```

STEHFEST WEIGHTING FACTORS
==========

I	V(I)	II	V(II)
1	-3.9682540E-04	9	-1.0525395E+09
2	2.1337302E+00	10	2.2590133E+09
3	-5.5101667E+02	11	-3.3997020E+09
4	3.3500161E+04	12	3.5824505E+09
5	-8.1266511E+05	13	-2.5914941E+09
6	1.0076184E+07	14	1.2270498E+09
7	-7.3241383E+07	15	-3.4273456E+08
8	3.3905963E+08	16	4.2841819E+07

CONCENTRATION DISTRIBUTION
==========

Time = 1.00000E-02 Tau = 5.80916E+03

I	R	Z	C/Co
1	1.000	.0000	.9961
2	1.000	.0020	.6494
3	1.000	.0040	.3659
4	1.000	.0060	.1758
5	1.000	.0080	.0713
6	1.000	.0100	.0243
7	2.000	.0000	.9857
8	2.000	.0020	.6353
9	2.000	.0040	.3521
10	2.000	.0060	.1655
11	2.000	.0080	.0654
12	2.000	.0100	.0215
13	3.000	.0000	.9682
14	3.000	.0020	.6118
15	3.000	.0040	.3296
16	3.000	.0060	.1493
17	3.000	.0080	.0563
18	3.000	.0100	.0175
19	4.000	.0000	.9427
20	4.000	.0020	.5777
21	4.000	.0040	.2981
22	4.000	.0060	.1274
23	4.000	.0080	.0447
24	4.000	.0100	.0128
25	5.000	.0000	.9066
26	5.000	.0020	.5311
27	5.000	.0040	.2568
28	5.000	.0060	.1008
29	5.000	.0080	.0320
30	5.000	.0100	.0081
31	6.000	.0000	.8560
32	6.000	.0020	.4696
33	6.000	.0040	.2055
34	6.000	.0060	.0713
35	6.000	.0080	.0196
36	6.000	.0100	.0042
37	7.000	.0000	.7904
38	7.000	.0020	.3870
39	7.000	.0040	.1456
40	7.000	.0060	.0424
41	7.000	.0080	.0095
42	7.000	.0100	.0016
43	8.000	.0000	.6861
44	8.000	.0020	.2774
45	8.000	.0040	.0842
46	8.000	.0060	.0191
47	8.000	.0080	.0031
48	8.000	.0100	.0003
49	9.000	.0000	.5012
50	9.000	.0020	.1548
51	9.000	.0040	.0347
52	9.000	.0060	.0052
53	9.000	.0080	.0004
54	9.000	.0100	.0000
55	10.000	.0000	.2643
56	10.000	.0020	.0561
57	10.000	.0040	.0072
58	10.000	.0060	.0001
59	10.000	.0080	.0000
60	10.000	.0100	.0000

References

Abramowitz, M. and I. A. Stegun. 1970. Handbook of Mathematical Functions. Dover, New York. 1046 pp.

Barker, J. A. 1982. Laplace transform solutions for solute transport in fissured aquifers. Adv. Water Resour. v. 5, pp. 98-104.

Chen, C. S. 1986. Solutions for radionuclide transport from an injection well into a single fracture in a porous formation. Water Resour. Res. v. 22, no. 4, pp. 508-518.

Hodgkinson, D. P. and D. A. Lever. 1983. Interpretation of a field experiment on the transport of sorbed and nonabsorbed tracers through a fracture in crystalline rock. Radioact. Waste Manage., Nucl. Fuel Cycle. v. 4, no. 2, pp. 129-158.

Hsieh, P. 1986. A new formula for the analytical solution of the radial dispersion problem. Water Resour. Res. v. 22, no. 11, pp. 1597-1605.

Neretnieks, I. 1980. Diffusion in the rock matrix: an important factor in radionuclide retardation? J. Geophys. Res. v. 85, no. B8, pp. 4379-4397.

Neretnieks, I. and A. Rasmuson. 1984. An approach to modeling radio-nuclide migration in a medium with strongly varying velocity and block sizes along the flow path. Water Resour. Res. v. 20, no. 12, pp. 1823-1836.

Rasmuson, A. 1984. Migration of radionuclides in fissured rock: analytical solutions for the case of constant source strength. Water Resour. Res. v. 20, no. 10, pp. 1435-1442.

Rasmuson, A. and I. Neretnieks. 1981. Migration of radionuclides in fissured rock: the influence of micropore diffusion and longitudinal dispersion. J. Geophys. Res. v. 86, no. B5, pp. 3749-3758.

Stehfest, H. 1970a. Numerical inversion of Laplace transforms. Commun. ACM. v. 13, no. 1, pp. 47-49.

Stehfest, H. 1970b. Remark on algorithm 368, numerical inversion of Laplace transforms. Commun. ACM. v. 13, no. 10, p. 624.

Sudicky, E. A. and E. O. Frind. 1982. Contaminant transport in fractured porous media: analytical solutions for a system of parallel fractures. Water Resour. Res. v. 18, no. 6, pp. 1634-1642.

Tang, D. H., E. O. Frind, and E. A. Sudicky. 1981. Contaminant transport in fractured porous media: analytical soltuion for a single fracture. Water Resour. Res. v. 18, no. 3, pp. 555-564.

* * * * *

Chia-Shyun Chen is an Associate Professor of Hydrology in the Geoscience Department of New Mexico Institute of Mining and Technology. He did his M.S. and Ph.D. in the Agricultural Engineering Department of Colorado State University and Texas A & M University, respectively. Prior to joining academics, he worked in a consulting firm for three years working on numerical simulation of transport problems under field conditions. His research interest is mathematical modeling of transport phenomena in subsurface hydrology.

Scott R. Yates received his B.S. in Geology from the University of Wisconsin in 1980, M.S. in Hydrology from New Mexico Institute of Mining and Technology in 1982, and Ph.D. in Soil and Water Science from the University of Arizona in 1985. He is currently a Soil Scientist for the USDA/ARS at the U.S. Salinity Laboratory in Riverside, California. His research interests include soil physics, modeling the fate and transport of contaminants in variably saturated media, and geostatistics.