OFFICE OF CIVILIAN RADIOACTIVE WASTE MANAGEMENT
CALCULATION COVER SHEET

1. QA QA
Page 1 Of. 17

2. Calculation Title

Type Curve Calculations for Mass Transport in Parallel Fractures Used in Particle-Tracking Scheme in the Saturated Zone

3. Document Identifier (including Revision Number)

B00000000-01717-0210-00089 REV 00

4 Total Attachments	5. Attachment Numbers – Number of pages in each
6	I-21, II-6, III-3, IV-5, V-1, VI-2

	Print Name	Signature	Date
6. Originator	Hubao Zhang	Hubao Zhang	5/3/2000
7. Checker	Stephanie Kuzio	Stephanie Kuzio	5/3/00
8 Lead	Bill Arnold	Bill W. Arnold	5/3/00

9. Remarks

This calculation was originally developed and checked under QAP-3-15Q Rev 0. However, this procedure was superseded by AP-3.12Q Rev 0, ICN 0 on 06/30/99. The final version of this calculation (REV 00) conforms to AP-3.12Q requirements.

Revision History

10 Revision No	11. Description of Revision
REV 00	Initial Issue

Table of Contents:

Item **Page**

1.0 PURPOSE 3
2.0 METHOD 4
2.1 CALCULATION EQUATIONS 4
2.2 NUMERICAL INTEGRATION SCHEME 6
3.0 ASSUMPTIONS 7
4.0 USE OF SOFTWARE ROUTINE 7
4.1 USER GUIDE OF THE SOFTWARE ROUTINE 8
4.2 VERIFICATION OF THE SOFTWARE ROUTINE 8
4.2.1 VERIFICATION OF NUMERICAL INTEGRATION SCHEME 8
4.2.2 VERIFICATION OF TYPE CURVE CALCULATIONS 9
5.0 CALCULATION INPUTS 11
6.0 RESULTS 15
7.0 REFERENCES 16
8.0 ATTACHMENTS 17

1.0 Purpose

The purpose of this calculation is to generate type curves in parallel fractures (Figure 1) to be used in a particle-tracking scheme in the finite element heat and mass transport (FEHM) code developed at the Los Alamos National Laboratory (LANL). The planning for this work is described in CRWMS M&O (1999). The type curves consisting of relative concentration vs. relative time for two dimensionless parameters will be output from this calculation and incorporated in FEHM to account for the mass diffusion in and out of matrix. It is intended in FEHM that relative time will be calculated from the type curves for a given set of dimensionless parameters and relative concentration. This relative time will be used to figure out whether a particle will move to the next location.

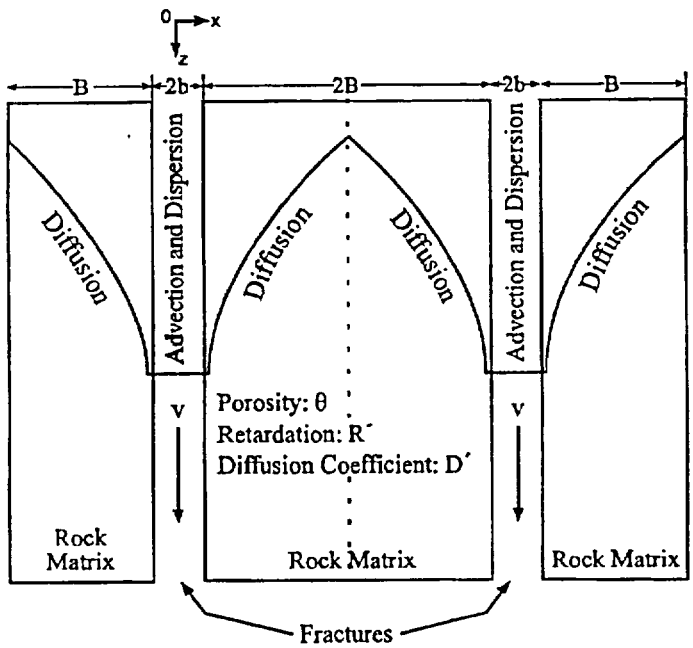

Figure 1. Mass transport in parallel fractures

2.0 Method

The calculation is directly based on an analytical solution given by Sudicky and Frind (1982). To facilitate calculations, the solution is changed to a dimensionless format.

2.1 Calculation Equations

In the fracture systems as shown in Figure 1, z is the spatial coordinate along the fracture (L), t is the time, 2b is the width of the fracture (L), 2B is the fracture spacing (L), v is the linear ground water velocity in the fracture (L/T), θ is the porosity of the matrix, R′ is the retardation factor in the matrix, R is the retardation factor in the fracture, D′ is the effective diffusion coefficient (L^2/T), c is the concentration at z along the fracture (ML^{-3}), c_0 is the source concentration at $z = z_0$ (ML^{-3}). The transient solution for contaminant transport with D = 0 in parallel fractures (Sudicky and Frind 1982) is given by

$$\frac{c}{c_0} = 0, \quad T^0 < 0 \tag{1}$$

$$\frac{c}{c_0} = \frac{1}{\pi}\exp\left(\frac{R\lambda z}{v}\right)\int_0^\infty \frac{\varepsilon}{\lambda^2 + \varepsilon^4/4}\exp(\varepsilon_R{}^0)\left[\exp(-\lambda T^0)\left\{\frac{\varepsilon^2}{2}\sin(\varepsilon_I{}^0) - \lambda\cos(\varepsilon_I{}^0)\right\}\right.$$

$$\left. + \frac{\varepsilon^2}{2}\sin(\Omega_I{}^0) + \lambda\cos(\Omega_I{}^0)\right]d\varepsilon, \quad T^0 > 0 \tag{2}$$

where

$$T^0 = t - \frac{Rz}{v} \tag{3}$$

$$\varepsilon_R{}^0 = -\frac{\omega\varepsilon}{2}\left(\frac{\sinh(\sigma\varepsilon) - \sin(\sigma\varepsilon)}{\cosh(\sigma\varepsilon) + \cos(\sigma\varepsilon)}\right) \tag{4}$$

$$\varepsilon_I{}^0 = \frac{\varepsilon^2 T^0}{2} - \frac{\omega\varepsilon}{2}\left(\frac{\sinh(\sigma\varepsilon) + \sin(\sigma\varepsilon)}{\cosh(\sigma\varepsilon) + \cos(\sigma\varepsilon)}\right) \tag{5}$$

$$\Omega^0 = \frac{\omega\varepsilon}{2}\left(\frac{\sinh(\sigma\varepsilon) + \sin(\sigma\varepsilon)}{\cosh(\sigma\varepsilon) + \cos(\sigma\varepsilon)}\right) \tag{6}$$

with

$$\omega = \frac{\theta (R' D')^{1/2} z}{bv} \tag{7}$$

$$\sigma = (R'/D')^{1/2}(B-b) \tag{8}$$

Define that

$$\tau_0 = \frac{z}{v} \tag{9}$$

$$\varepsilon = \frac{\varepsilon_1}{\tau_0^{1/2}} \tag{10}$$

Equations (1) through (8) can be rewritten as

$$\frac{c}{c_0} = 0, \quad T_1^0 < 0 \tag{1a}$$

$$\frac{c}{c_0} = \frac{1}{\pi}\exp(R\lambda\tau_0)\int_0^\infty \frac{\varepsilon_1}{\lambda^2\tau_0^2 + \varepsilon_1^4/4}\exp(\varepsilon_R{}^0)\left[\exp(-\lambda\tau_0 T_1^0)\left\{\frac{\varepsilon_1^2}{2}\sin(\varepsilon_I{}^0) - \lambda\tau_0\cos(\varepsilon_I{}^0)\right\}\right.$$

$$\left. + \frac{\varepsilon_1^2}{2}\sin(\Omega_I{}^0) + \lambda\tau_0\cos(\Omega_I{}^0)\right]d\varepsilon_1, \quad T_1^0 > 0 \tag{2a}$$

where

$$T_1^0 = \frac{t}{\tau_0} - R \tag{3a}$$

$$\varepsilon_R{}^0 = -\frac{\omega_1\varepsilon_1}{2}\left(\frac{\sinh(\sigma_1\varepsilon_1) - \sin(\sigma_1\varepsilon_1)}{\cosh(\sigma_1\varepsilon_1) + \cos(\sigma_1\varepsilon_1)}\right) \tag{4a}$$

$$\varepsilon_I{}^0 = \frac{\varepsilon_1^2 T_1^0}{2} - \frac{\omega_1\varepsilon_1}{2}\left(\frac{\sinh(\sigma_1\varepsilon_1) + \sin(\sigma_1\varepsilon_1)}{\cosh(\sigma_1\varepsilon_1) + \cos(\sigma_1\varepsilon_1)}\right) \tag{5a}$$

$$\Omega^0 = \frac{\omega_1\varepsilon_1}{2}\left(\frac{\sinh(\sigma_1\varepsilon_1) + \sin(\sigma_1\varepsilon_1)}{\cosh(\sigma_1\varepsilon_1) + \cos(\sigma_1\varepsilon_1)}\right) \tag{6a}$$

with

$$\omega_1 = \frac{\theta(R'D'\tau_0)^{1/2}}{b} \tag{7a}$$

$$\sigma_1 = (\frac{R'}{D'\tau_0})^{1/2}(B-b) \tag{8a}$$

If we further assume that $\lambda = 0$.

$$\frac{c}{c_0} = \frac{1}{\pi}\int_0^\infty \frac{2}{\varepsilon_1}\exp(\varepsilon_R{}^0)\left[\sin(\varepsilon_I{}^0) + \sin(\Omega_I{}^0)\right]d\varepsilon_1 \tag{9a}$$

2.2 Numerical Integration Scheme

Equation (2) or (2a) may be rewritten as a generic form of integration from 0 to infinite.

$$\frac{c}{c_0} = \int_0^\infty f(x)dx \tag{2c}$$

To accurately calculate the integration of the equation (2c), a large number of intervals were divided from 0 to ∞ such that the integration is expressed as

$$\frac{c}{c_0} = \int_0^\infty f(x)dx = \sum_{i=1}^{I}\int_{a_i}^{b_i} f(x)dx \tag{2d}$$

where $a_1=0$, b_1 is a very small number (10^{-7}), a_i (i>1) is the same as b_{i-1}, and b_i (i>1) is the multiplication of b_{i-1} by a factor of greater than 1. The number of integration intervals (I) is increased until a convergent solution is achieved to account for the oscillation of the integrand.

Integration in each interval in equation (2d) may be integrated by Gaussian quadrature or trapezoidal rule. Both methods were implemented in current work. In generating type curves using equation (9a), Gaussian quadrature is used. Trapezoidal rule is implemented to verify the calculations by the Gaussian quadrature technique.

3.0 Assumptions

3.1 Assumptions used in Sudicky and Frind's solution

In deriving the above solution (Section 2), Sudicky and Frind (1982) assumed: (1) The width of each fracture is much smaller than its length; (2) Transverse diffusion and dispersion within each fracture assures complete mixing across its width at all times; (3) The permeability of the intervening porous matrix is low and transport within the matrix will be mainly by molecular diffusion; and (4) Transport along each fracture is much faster than transport within the matrix. The assumptions used by Sudicky and Frind (1982) are valid for the Yucca Mountain saturated zone because (1) The width of each fracture is much smaller than its length in the saturated zone (CRWMS M&O 2000; *ANL-NBS-MD-000011*); (2) Complete mixing within the fracture width is valid because the width of fractures is small; (3) The permeability in the fractured media is usually several order of magnitude smaller than that in the matrix; (4) The transport in fractures is much faster than that in the matrix because the permeability in the fractures is much larger than that in the matrix.

3.2 Assumptions for representing the matrix diffusion using type curves in the particle-tracking scheme

In the FEHM implementation of the particle-tracking scheme, it is assumed that fractures are parallel to flow lines in an aquifer system. The basis for this assumption is that data are available to calculate fracture spacing and fracture porosity assuming that the fractures are parallel to flow (CRWMS M&O 2000, *ANL-NBS-MD-000011*), and the calculated spacing and porosity consist of conservative estimates for transport through the fractured system. This assumption will also be justified in the three-dimensional FEHM models (CRWMS M&O 2000; *ANL-NBS-HS-000030*). This assumption is used in the derivation of the analytical solution in Section 2.

4.0 Use of Software Routine

There is no available software to calculate the type curves. Computer routines were programmed by the originator in order to perform this calculation. The computer routine is called FRACT_P version 1.0 revision 0.

4.1 User Guide of the Software Routine

The solution was implemented in Fortran and may be compiled by any standard Fortran 77 compiler. The source code is listed in Attachment I. A user's guide is summarized in Attachment II. Input and output files are also listed in the attachment II. A copy of the executable subroutine was compiled on a Sun workstation PICARD at Sandia National Laboratories.

4.2 Verification of the Software Routine

4.2.1 Verification of Numerical Integration Scheme

A hypothetical problem was set up to compare calculations by Gaussian quadrature and trapezoidal rule techniques. The input and output files are listed in Attachment III. Figure 2 shows concentration profiles along a fracture simulated by the two integration techniques. The results are identical.

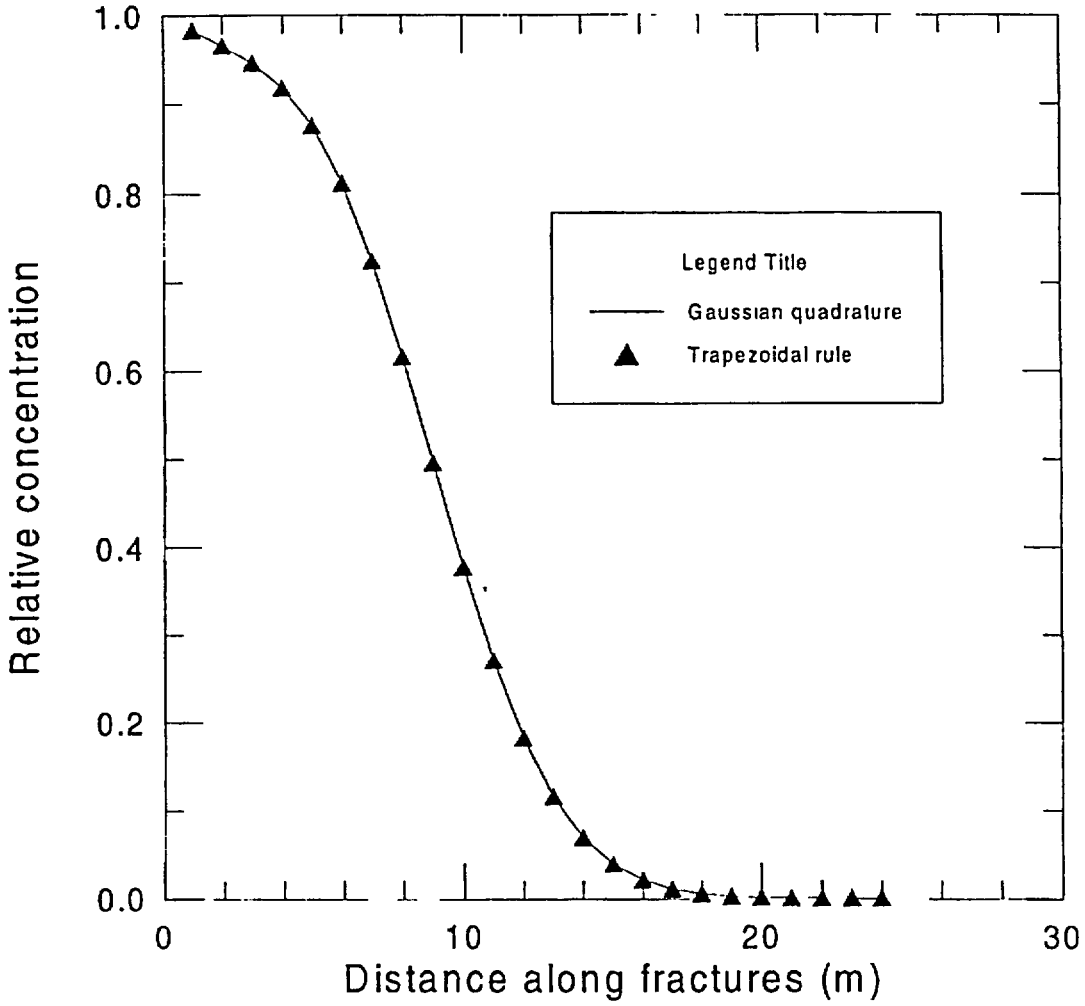

Figure 2. Comparison of Gaussian Quadrature and trapezoidal rule techniques.

4.2.2 Verification of Type Curve Calculations

For infinite matrix domain, an analytical solution was given by Tang et al. (1981). The solution may be used to verify relative concentration calculation in a parallel fracture system with a large value of σ_l. The relative concentration for $\sigma_l = 1000$ and $\omega_l = 0.464159$ with the infinite matrix domain (Tang et al. 1981) using MathCad and Microsoft Excel, respectively, were compared. Comparison indicates that three results are almost identical (Figure 3).

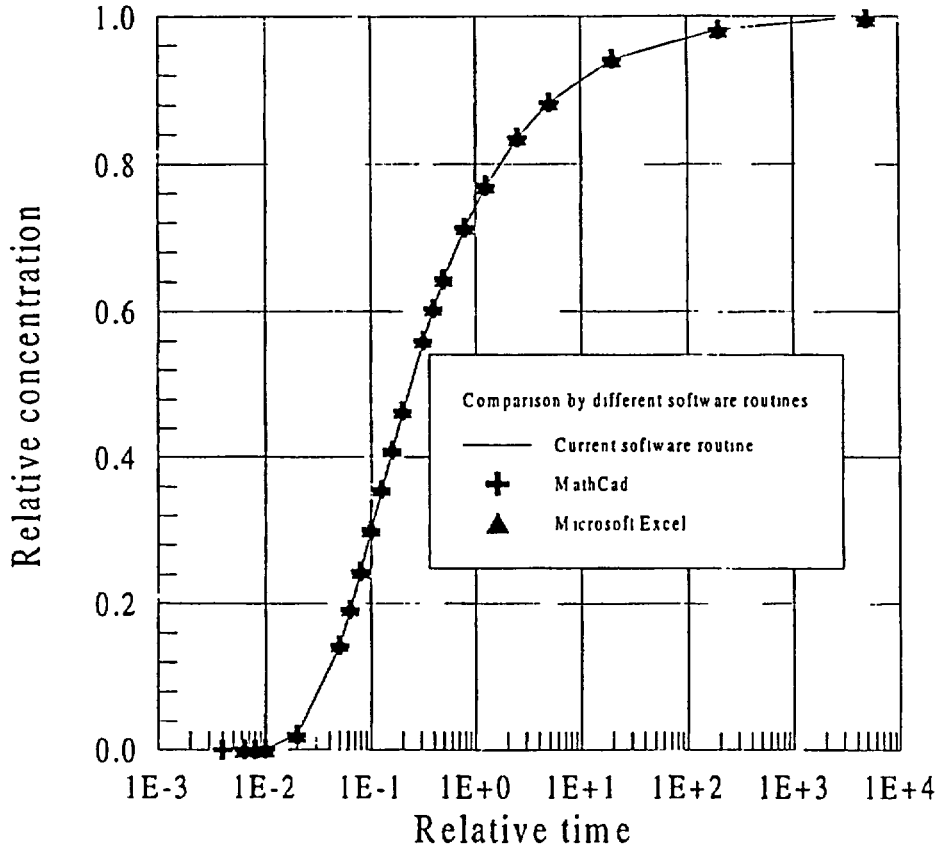

Figure 3. Comparison of the current software routine, MathCad and Microsoft Excel for $\sigma_1 = 1000$ and $\omega_1 = 0.464159$.

5.0 Calculation Inputs

The input file for type curve calculations is listed in Attachment IV. The determination of σ_l and ω_l values was partly based on the range of flow and transport parameters at the Yucca Mountain site. The matrix-diffusion related flow and transport parameters (B, f, D', Q, and Rd) for 100 realizations of three-dimensional saturated zone models for the TSPA-SR were listed in Table 1 (CRWMS M&O 2000; ANL-NBS-MD-000011). The fracture aperture (2b), and average ground water flow velocity (V) were calculated from fracture spacing (2B), fracture porosity (φ_f), and specific discharge (Q) (Table 1). In the three-dimensional FEHM models, the values of z in equation (7) range from 1 to 500 m (CRWMS M&O 2000; *ANL-NBS-HS-000030*).

Based on the flow and transport parameters in the three-dimensional FEHM models for the fractured media of the saturated zone, the minimum and maximum dimensionless parameter σ_l and ω_l were calculated for each realization (Table 1). The maximum and minimum values of the dimensionless parameter σ_l are 3.51×10 and 2.38×10^8 (Table 1), respectively, while the dimensionless parameter ω_l ranges from 4.91×10^{-4} to 9.98×10^4 (Table 1). In calculating the type curves, σ_l ranges from 0.1 to 10^9 while ω_l ranges from 1.0×10^{-4} to 10^6 to include value ranges in the three-dimensional FEHM models. A two-step process is designed to calculate the relative concentration. First, the relative times T_l^0 were automatically searched within the software routine with relative concentrations uniformly distributed close to 34 pre-assigned values. The thirty-four pre-assigned values of relative concentration are 1.0e-8, 1.0e-6, 1.0e-5, 1.0e-4, 1.0e-3, 1.0e-2, 0.1, 0.16, 0.22, 0.28, 0.34, 0 40, 0.46, 0.52, 0.58, 0.64, 0.70, 0.76, 0.80, 0.82, 0.84, 0.86, 0.88, 0.90, 0.92, 0.94, 0.95, 0.96, 0.97, 0.995, 0.999, 0.9999, and 1.0, respectively. Once the values of relative time are determined, a smaller integration interval is used to calculate the relative concentration with much higher accuracy. There are options in the software User's Guide (Page II-1) for initial search of relative times and final calculation of the relative concentrations.

Table 1. Flow and transport parameters used in the 3-D FEHM models and ranges of dimensionless parameters σ_1 and ω_1 determined from these parameters.

Real No	B(m)	ϕ_f	D'(m^2/s)	Q(m/yr)	Rd(RN10)	b(m)	v(m/yr)	τ_0(min)	τ_0(max)	ω_1(min)	ω_1(max)	σ_1(min)	σ_1(max)
1	2 06	6 90E-03	4 36E-11	6 6	8 92E+02	1 46E-02	9 56E+02	3 30E+04	1 65E+07	1.23E-02	1.37E+01	1 53E+02	1.02E+05
2	33 53	3 12E-02	1.70E-11	0 66	1 61E+03	7,12E-02	2 11E+01	1.49E+06	7.47E+08	1 06E-02	1.59E+01	2.97E+02	2.66E+05
3	60 41	9 71E-03	2 00E-11	0.75	1 27E+03	1 44E-03	7.72E+01	4 09E+05	2 04E+08	2 99E-01	3 97E+02	9 45E+02	7.53E+05
4	38.01	1.17E-02	2.40E-13	0 66	7.88E+02	2 89E-02	5 65E+01	5.59E+05	2.79E+08	1 91E-03	1 99E+00	4 63E+03	2.91E+06
5	39 66	3 10E-05	2.39E-11	7 5	4 81E+02	1.19E-05	2.42E+05	1.31E+02	6.53E+04	7 05E-01	5 77E+02	3 17E+04	1.56E+07
6	6 39	1 46E-02	5 65E-13	6 6	1 69E+03	6 06E-02	4 53E+02	6 96E+04	3.48E+07	4 91E-04	7.53E-01	1 43E+03	1.31E+06
7	43 98	1 07E-04	6.49E-11	0 75	1 63E+03	3.33E-05	7.01E+03	4.50E+03	2.25E+06	2.43E+00	3 66E+03	3 64E+03	3 28E+06
8	10 14	5.58E-02	4 79E-13	0 075	1 18E+03	5 41E-02	1.34E+00	2.35E+07	1.17E+10	9.29E-03	1.19E+01	1.34E+02	1.03E+05
9	90 03	8 30E-05	2 94E-12	6 6	1 99E+02	3.33E-05	7 95E+04	3.97E+02	1.99E+05	1.54E-01	8.10E+01	1.18E+05	3.72E+07
10	9 31	8 47E-04	6.11E-13	0 75	1 48E+03	1.51E-03	8.85E+02	3.56E+04	1.78E+07	1 47E-02	2 11E+01	2.82E+03	2.43E+06
11	19 01	1.12E-04	1.11E-11	0 66	9.30E+02	1.19E-05	5 89E+03	5 36E+03	2 68E+06	3 08E+00	3 50E+03	3 48E+03	2.37E+06
12	10 54	5 71E-03	1.99E-13	0 66	1.17E+03	1.26E-02	1.16E+02	2.73E+05	1 37E+08	2 77E-03	3 52E+00	2 02E+03	1.54E+06
13	31 49	2.41E-05	1 85E-11	0 066	7.75E+02	2 42E-05	2 73E+03	1.15E+04	5.77E+06	2 87E+00	2 98E+03	3.05E+03	1.90E+06
14	5 36	1 04E-02	3 05E-12	0 075	1 81E+02	2 15E-02	7.19E+00	4.39E+06	2.20E+09	2 55E-02	1.28E+01	6 53E+01	1.97E+04
15	38 26	9 63E-04	1 61E-11	7 5	3 83E+02	4 20E-03	7 79E+03	4.05E+03	2 03E+06	9 14E-03	6 66E+00	6 69E+03	2.93E+06
16	19 20	8 60E-03	6 78E-11	0 66	1 43E+03	4 94E-03	7 67E+01	4 11E+05	2.06E+08	1 60E-01	2 27E+02	1.63E+02	1.38E+05
17	36 27	6 08E-02	1 09E-12	0 066	6 73E+02	7 04E-02	1 09E+00	2 91E+07	1.45E+10	1 20E-02	1.16E+01	2.88E+02	1 67E+05
18	96 67	2.34E-04	4 24E-12	0 75	8.30E+01	1 30E-04	3 20E+03	9 86E+03	4 93E+06	2 35E-01	7.98E+01	2.12E+04	4.31E+06
19	13 35	1 45E-03	9 79E-13	0 075	8 42E+02	1 16E-03	5 17E+01	6.11E+05	3 05E+08	9 99E-02	1.08E+02	7.72E+02	5 01E+05
20	21 70	2 19E-04	1 88E-12	0 75	3 31E+02	1.09E-04	3.43E+03	9 21E+03	4 60E+06	1 82E-01	1 23E+02	7.37E+03	3.00E+06
21	12 95	4 81E-04	8 53E-13	7 5	9 50E+02	9 66E-04	1.56E+04	2 02E+03	1 01E+06	6 45E-03	7 41E+00	1.39E+04	9.61E+06
22	16 31	1 83E-04	1.50E-11	0 75	1 46E+03	5 66E-04	4 09E+03	7.71E+03	3.85E+06	9.03E-02	1 28E+02	2.14E+03	1.83E+06
23	12.11	8 09E-03	3.99E-11	0 075	4 35E+02	1 46E-03	9 27E+00	3 40E+06	1.70E+09	1 20E+00	9.31E+02	4 64E+01	2.17E+04
24	35 19	2.99E-02	2.80E-13	0 066	4 21E+01	4 67E-03	2.20E+00	1.43E+07	7.16E+09	6 43E-02	1.56E+01	7.86E+02	1.14E+05
25	58.98	3 68E-05	2.48E-13	0 66	2.88E+02	5 37E-05	1 79E+04	1.76E+03	8.80E+05	5 83E-02	3 69E+01	1.26E+05	4.79E+07
26	6 85	3 61E-05	7.21E-12	0 075	9 03E+02	1.06E-04	2 08E+03	1.52E+04	7.59E+06	4 69E-01	5.25E+02	9 27E+02	6 22E+05
27	6.66	2 94E-03	3 48E-11	0 75	2 51E+01	4 36E-03	2.55E+02	1.24E+05	6.19E+07	7.13E-02	1.33E+01	1.44E+02	1 61E+04
28	26 07	2 43E-02	7 57E-12	0 075	1 14E+03	2.53E-02	3 09E+00	1.02E+07	5.11E+09	5.21E-02	6 55E+01	1.32E+02	9.99E+04
29	53 48	3 21E-03	3 30E-12	0 66	1.09E+02	1.22E-02	2 06E+02	1.53E+05	7 66E+07	8 72E-03	3 39E+00	3 36E+03	7 84E+05
30	26.71	1 07E-03	5 64E-11	0 075	1 01E+03	2 05E-03	7 04E+01	4 48E+05	2.24E+08	3.68E-01	4 36E+02	2 38E+02	1.69E+05
31	19 76	4 68E-02	1.42E-11	7 5	3 67E+02	3 96E-02	1 60E+02	1 97E+05	9 85E+07	6.34E-03	4 53E+00	5 27E+02	2.26E+05
32	75 25	7 43E-03	2 20E-12	0 75	1.71E+03	3 11E-03	1 01E+02	3 13E+05	1 56E+08	4 00E-02	6 18E+01	4 05E+03	3.75E+06

Table 1. Flow and transport parameters used in the 3-D FEHM models and ranges of dimensionless parameters σ_1 and ω_1 determined from these parameters (continued).

Real No	B(m)	ϕ_f	D (m^2/s)	Q(m/yr)	Rd(RN10)	b(m)	V(m/yr)	τ_0(min)	τ_0(max)	ω_1(min)	ω_1(max)	σ_1(min)	σ_1(max)
33	34.53	6.13E-04	6.78E-13	0.066	7.16E+02	2.95E-03	1.08E+02	2.93E+05	1.46E+08	2.27E-02	2.26E+01	3.47E+03	2.07E+06
34	16.91	1.31E-04	7.53E-11	6.6	1.42E+02	5.24E-04	5.06E+04	6.24E+02	3.12E+05	6.21E-02	2.76E+01	3.49E+03	9.29E+05
35	10.86	6.60E-02	5.96E-11	0.66	1.72E+03	1.10E-01	1.00E+01	3.16E+06	1.58E+09	1.87E-02	2.89E+01	3.51E+01	3.25E+04
36	102.61	4.85E-02	9.64E-12	0.75	5.35E+02	6.46E-03	1.55E+01	2.04E+06	1.02E+09	1.03E-01	8.87E+01	1.04E+03	5.35E+05
37	78.89	4.62E-03	2.23E-13	0.066	5.29E+02	1.32E-02	1.43E+01	2.21E+06	1.10E+09	7.96E-03	6.83E+00	5.02E+03	2.58E+06
38	29.77	3.01E-05	3.30E-11	0.066	1.50E+02	3.28E-05	2.19E+03	1.44E+04	7.20E+06	3.15E+00	1.44E+03	1.93E+03	5.29E+05
39	5.86	2.16E-02	3.72E-12	0.66	4.20E+02	3.53E-02	3.06E+01	1.03E+06	5.16E+08	8.32E-03	6.35E+00	1.33E+02	6.08E+04
40	50.84	1.28E-03	9.33E-12	0.66	1.06E+03	1.04E-03	5.16E+02	6.11E+04	3.06E+07	1.09E-01	1.32E+02	3.01E+03	2.19E+06
41	20.99	4.15E-03	2.01E-12	0.75	5.77E+02	4.79E-04	1.81E+02	1.75E+05	8.73E+07	1.86E-01	1.66E+02	1.59E+03	8.52E+05
42	28.91	5.72E-04	2.13E-13	0.75	1.04E+03	1.72E-04	1.31E+03	2.41E+04	1.20E+07	6.26E-02	7.54E+01	1.81E+04	1.30E+07
43	16.71	2.56E-05	4.33E-13	0.66	1.12E+03	6.79E-05	2.58E+04	1.22E+03	6.12E+05	5.09E-02	6.34E+01	3.25E+04	2.43E+07
44	5.68	7.06E-05	2.73E-11	0.66	1.07E+03	4.76E-05	9.36E+03	3.37E+03	1.69E+06	9.57E-01	1.17E+03	8.37E+02	6.12E+05
45	8.86	1.75E-02	1.39E-12	6.6	1.24E+03	7.61E-04	3.76E+02	8.39E+04	4.19E+07	6.73E-02	8.82E+01	1.16E+03	9.13E+05
46	47.77	3.45E-04	1.36E-13	0.75	1.37E+03	3.22E-04	2.18E+03	1.45E+04	7.25E+06	2.07E-02	2.86E+01	4.81E+04	3.99E+07
47	14.57	1.53E-05	2.69E-12	0.066	9.83E+02	3.86E-06	4.30E+03	7.33E+03	3.67E+06	5.45E+00	6.36E+03	4.65E+03	3.26E+06
48	3.67	7.79E-04	3.54E-13	7.5	6.99E+01	8.88E-04	9.63E+03	3.28E+03	1.64E+06	5.75E-03	1.79E+00	4.82E+03	9.01E+05
49	18.37	5.69E-05	4.13E-12	6.6	1.58E+03	1.56E-04	1.16E+05	2.72E+02	1.36E+05	3.23E-02	4.79E+01	2.45E+04	2.18E+07
50	8.20	1.42E-03	3.05E-11	0.075	1.57E+03	5.19E-04	5.27E+01	5.99E+05	2.99E+08	1.24E+00	1.82E+03	8.57E+01	7.59E+04
51	21.54	2.06E-02	5.57E-13	0.66	2.68E+02	4.80E-04	3.20E+01	9.87E+05	4.93E+08	2.32E-01	1.41E+02	1.30E+03	4.75E+05
52	49.10	9.71E-05	1.20E-13	0.75	6.62E+02	2.03E-05	7.72E+03	4.09E+03	2.04E+06	1.64E-01	1.57E+02	9.91E+04	5.70E+07
53	9.92	1.30E-05	3.19E-13	0.66	6.32E+02	1.37E-05	5.09E+04	6.20E+02	3.10E+05	1.54E-01	1.44E+02	3.15E+04	1.77E+07
54	3.08	4.53E-05	3.48E-12	0.066	5.02E+02	3.45E-07	1.46E+03	2.17E+04	1.08E+07	1.20E+02	9.98E+04	5.02E+02	2.52E+05
55	7.42	1.20E-03	1.14E-13	0.75	3.18E+02	1.91E-03	6.27E+02	5.03E+04	2.52E+07	5.94E-03	3.95E+00	4.38E+03	1.75E+06
56	20.05	1.14E-05	8.57E-12	0.66	8.26E+02	7.50E-07	5.78E+04	5.46E+02	2.73E+05	1.37E+01	1.47E+04	1.31E+04	8.43E+06
57	2.61	3.95E-03	1.61E-13	0.66	1.01E+03	5.07E-03	1.67E+02	1.89E+05	9.43E+07	5.16E-03	6.10E+00	6.69E+02	4.75E+05
58	28.26	1.78E-05	5.22E-12	6.6	7.26E+02	4.94E-06	3.71E+05	8.51E+01	4.26E+04	6.40E-01	6.43E+02	5.99E+04	3.61E+07
59	4.24	1.91E-03	3.99E-13	6.6	1.42E+03	9.69E-03	3.46E+03	9.13E+03	4.56E+06	9.34E-04	1.31E+00	3.14E+03	2.65E+06
60	23.59	6.32E-05	7.59E-11	0.75	1.51E+03	4.65E-05	1.19E+04	2.66E+03	1.33E+06	1.45E+00	2.10E+03	2.35E+03	2.04E+06
61	8.64	7.00E-02	1.60E-12	0.066	8.10E+02	5.33E-03	9.43E-01	3.35E+07	1.67E+10	2.06E-01	2.18E+02	5.28E+01	3.36E+04
62	124.37	4.22E-04	1.69E-13	0.75	4.78E+00	2.34E-03	1.78E+03	1.78E+04	8.89E+06	3.52E-03	2.87E-01	1.01E+05	4.95E+06
63	15.40	2.05E-05	8.59E-11	7.5	6.97E+02	1.01E-05	3.66E+05	8.62E+01	4.31E+04	1.27E+00	1.25E+03	8.01E+03	4.73E+06
64	4.69	7.78E-02	8.09E-13	7.5	1.38E+03	3.63E-02	9.64E+01	3.27E+05	1.64E+08	2.13E-03	2.94E+00	4.05E+02	3.36E+05
65	139.57	1.69E-04	7.78E-13	7.5	1.61E+03	1.59E-05	4.43E+04	7.12E+02	3.56E+05	2.22E-01	3.32E+02	2.65E+05	2.38E+08
66	11.75	1.54E-04	1.82E-13	0.75	1.76E+03	5.20E-05	4.88E+03	6.46E+03	3.23E+06	9.89E-02	1.55E+02	1.53E+04	1.44E+07
67	11.46	2.26E-05	5.19E-13	7.5	6.42E+02	3.04E-05	3.32E+05	9.50E+01	4.75E+04	3.46E-02	3.27E+01	7.30E+04	4.14E+07

Table 1. Flow and transport parameters used in the 3-D FEHM models and ranges of dimensionless parameters σ_1 and ω_1 determined from these parameters (continued).

Real. No	B(m)	ϕ_f	D (m^2/s)	Q(m/yr)	Rd(RN10)	b(m)	V(m/yr)	τ_0(min)	τ_0(max)	ω_1(min)	ω_1(max)	σ_1(min)	σ_1(max)
68	5.19	8.35E-05	5.90E-12	0.66	1.73E+03	4.43E-05	7.90E+03	9.99E+05	3.99E+06	8.23E+00	1.14E+03	1.07E+03	8.89E+04
69	9.47	2.32E-03	9.20E-13	0.075	1.20E+03	3.05E-03	3.23E+01	2.44E+08	9.77E+08	7.36E-01	8.51E+01	3.16E+02	2.19E+04
70	24.85	1.94E-04	5.58E-12	0.075	3.39E+02	2.95E-04	3.86E+02	2.05E+07	8.18E+07	5.43E+00	3.33E+02	1.16E+03	4.28E+04
71	12.43	2.14E-03	7.89E-12	0.66	1.07E+02	3.04E-03	3.09E+02	2.55E+07	1.02E+08	7.01E-01	2.41E+01	4.38E+02	9.04E+03
72	24.16	3.39E-03	6.43E-12	6.6	1.25E+03	6.26E-03	1.95E+03	4.06E+06	1.62E+07	1.22E-01	1.44E+01	2.37E+03	1.67E+05
73	27.57	2.53E-03	1.18E-11	0.66	1.56E+03	5.88E-03	2.61E+02	3.02E+07	1.21E+08	4.81E-01	6.33E+01	7.30E+02	5.76E+04
74	25.33	6.26E-03	2.18E-11	6.6	1.32E+03	7.68E-03	1.05E+03	7.49E+06	3.00E+07	2.50E-01	3.03E+01	9.92E+02	7.22E+04
75	32.64	1.30E-02	4.74E-12	0.066	1.35E+03	4.33E-04	5.06E+00	1.56E+09	6.23E+09	2.98E+01	3.64E+03	1.90E+02	1.39E+04
76	40.88	1.71E-03	1.31E-13	6.6	2.27E+02	5.97E-03	3.85E+03	2.05E+06	8.20E+06	1.30E-02	6.53E-01	3.95E+04	1.19E+06
77	22.17	1.44E-05	5.01E-11	0.75	1.40E+03	4.80E-05	5.20E+04	1.52E+05	6.07E+05	8.62E+00	1.07E+03	4.02E+03	3.00E+05
78	3.18	3.75E-02	4.48E-11	6.6	4.50E+02	7.58E-03	1.76E+02	4.49E+07	1.79E+08	8.86E-01	6.27E+01	3.54E+01	1.50E+03
79	66.45	3.55E-02	2.96E-13	0.075	1.65E+03	2.30E-02	2.11E+00	3.73E+09	1.49E+10	2.17E-01	2.93E+01	9.99E+02	8.10E+04
80	65.72	4.13E-05	1.32E-12	0.75	4.03E+02	2.47E-05	1.82E+04	4.34E+05	1.74E+06	4.60E+00	3.08E+02	4.33E+04	1.74E+06
81	7.05	3.13E-04	1.15E-12	0.66	8.58E+02	1.37E-04	2.11E+03	3.74E+06	1.50E+07	2.27E+00	2.22E+02	1.70E+03	9.95E+04
82	14.40	2.72E-04	7.35E-13	0.66	4.64E+02	6.45E-05	2.43E+03	3.25E+06	1.30E+07	3.59E+00	2.58E+02	4.66E+03	2.01E+05
83	13.37	1.37E-04	3.33E-13	0.066	1.21E+03	1.22E-04	4.82E+02	1.64E+07	6.54E+07	2.86E+00	3.31E+02	2.87E+03	1.99E+05
84	11.29	5.09E-05	2.51E-12	0.75	1.66E+02	6.40E-05	1.47E+04	5.35E+05	2.14E+06	2.72E+00	1.17E+02	4.87E+03	1.26E+05
85	7.63	6.96E-04	4.81E-12	0.75	1.33E+03	5.34E-04	1.08E+03	7.32E+06	2.93E+07	1.67E+00	2.02E+02	6.43E+02	4.69E+04
86	17.90	1.82E-03	1.76E-12	0.75	8.68E+02	4.71E-03	4.12E+02	1.92E+07	7.66E+07	1.85E-01	1.81E+01	1.54E+03	9.09E+04
87	23.06	4.50E-04	2.57E-11	6.6	1.31E+03	1.34E-03	1.47E+04	5.38E+05	2.15E+06	4.17E-01	5.02E+01	3.10E+03	2.24E+05
88	15.86	1.59E-02	3.56E-11	0.75	1.67E+03	1.37E-02	4.70E+01	1.68E+08	6.71E+08	8.44E-01	1.15E+02	1.02E+02	8.38E+03
89	45.64	2.51E-02	2.24E-12	0.75	6.12E+02	1.51E-04	2.98E+01	2.64E+08	1.06E+09	2.41E+01	1.99E+03	9.38E+02	4.64E+04
90	54.76	6.60E-04	1.04E-13	0.66	9.57E+02	4.65E-04	9.99E+02	7.89E+06	3.16E+07	2.92E-01	3.01E+01	3.03E+04	1.87E+06
91	17.45	1.09E-05	1.57E-12	0.66	6.02E+02	2.60E-05	6.07E+04	1.30E+05	5.20E+05	2.61E+00	2.13E+02	1.93E+04	9.46E+05
92	1.63	3.69E-04	2.21E-11	0.75	7.46E+02	1.33E-03	2.03E+03	3.88E+06	1.55E+07	1.04E+00	9.48E+01	8.79E+01	4.80E+03
93	15.19	1.41E-02	5.26E-11	0.75	5.54E+02	9.58E-02	5.33E+01	1.48E+08	5.92E+08	1.38E-01	1.08E+01	8.56E+01	4.03E+03
94	41.96	4.19E-02	9.68E-11	7.5	1.29E+03	2.67E-02	1.79E+02	4.40E+07	1.76E+08	3.67E-01	4.39E+01	3.21E+02	2.30E+04
95	71.42	9.80E-02	1.27E-12	0.075	1.53E+03	5.83E-01	7.66E-01	1.03E+10	4.12E+10	2.94E-02	3.83E+00	3.10E+02	2.42E+04
96	217.72	2.79E-04	8.79E-11	0.066	2.64E+02	4.76E-04	2.36E+02	3.34E+07	1.33E+08	1.71E+01	9.24E+02	2.01E+03	6.53E+04
97	13.87	8.81E-02	1.07E-11	0.75	2.35E+02	2.26E-02	8.51E+00	9.27E+08	3.71E+09	6.61E-01	3.38E+01	6.95E+01	2.13E+03
98	167.07	5.07E-03	3.88E-13	6.6	1.09E+03	2.57E-04	1.30E+03	6.06E+06	2.42E+07	8.96E-01	9.85E+01	5.45E+04	3.60E+06
99	30.20	6.11E-05	1.48E-13	0.75	1.50E+03	1.14E-04	1.23E+04	6.42E+05	2.57E+06	4.07E-01	5.25E+01	4.90E+04	3.80E+06
100	8.26	1.61E-05	1.28E-11	0.66	1.11E+03	5.15E-05	4.09E+04	1.93E+05	7.71E+05	4.57E+00	5.07E+02	2.63E+03	1.75E+05

6.0 Results

A partial listing of the output files for type curve calculations are listed in Attachment IV. A total number of 61x61 type curves were calculated. These type curves will be used for a particle-tracking scheme in FEHM. Selected type curves for σ_1=0.1 are shown in Figure 4. The calculation contains no information or assumptions that must be confirmed prior to the use of the calculation.

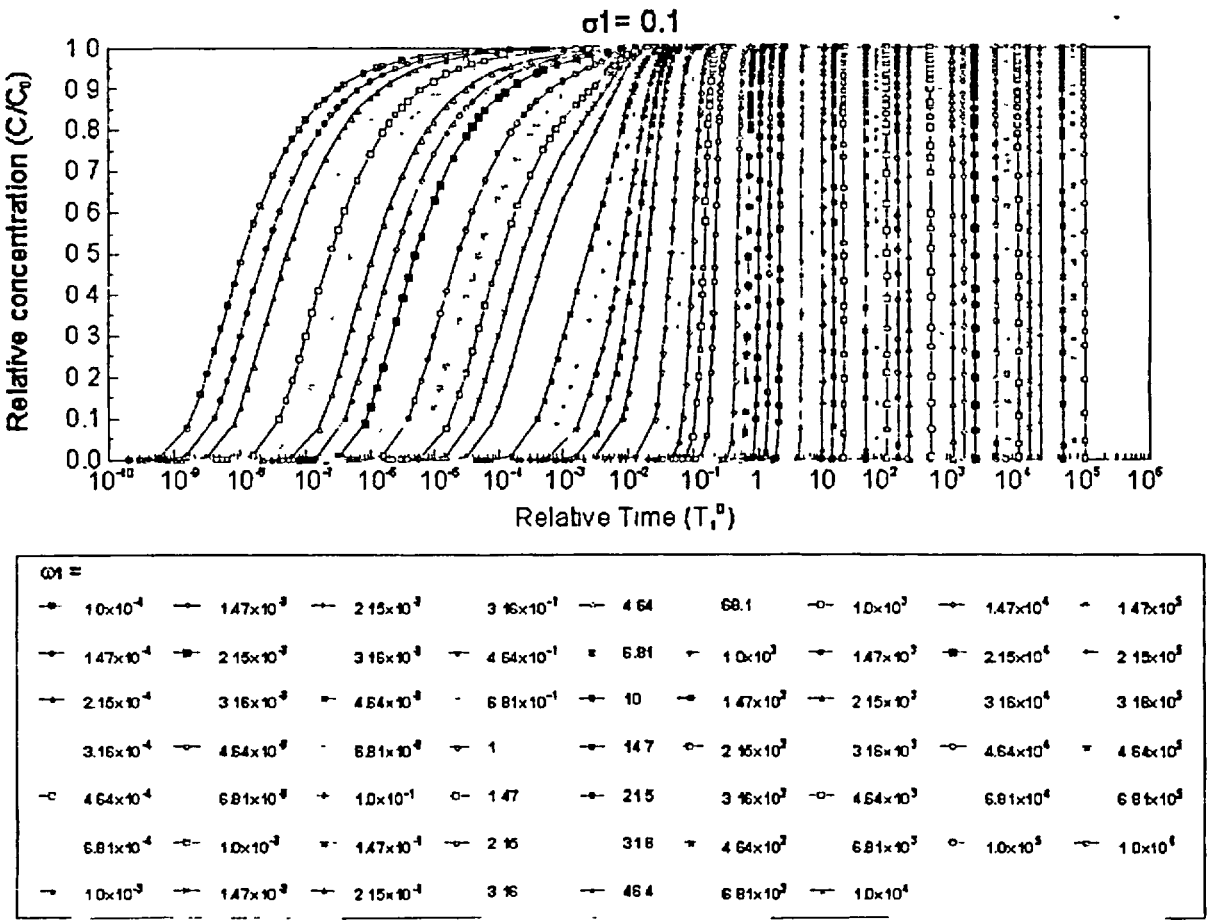

Figure 4. Selected type curves for σ_1=0.1

7.0 References

CRWMS M&O 1999. *Abstraction of Matrix Diffusion for SZ Flow and Transport Analyses, Rev 01. ID: B1080, Activity: SPP3025.* Work Direction and Planning Document. Las Vegas, Nevada. ACC: MOL.19990707.0107.

CRWMS M&O 2000. *Input Transmittal for "Uncertainty Distribution for Stochastic Parameters" (ANL-NBS-MD-000011) and "Input and Results of the Base Case Saturated Zone Flow and Transport Model for TSPA" (ANL-NBS-HS-000030).* Input Transmittal 00204.T. Las Vegas, NV: CRWMS M&O. ACC: MOL.20000404.0198.

Sudicky, E.A., and Frind, E.O. 1982. "Contaminant Transport in Fractured Porous Media: Analytical Solutions for a System of Parallel Fractures." *Water Resources Research,* 18, (6) 1634–1642. Washington, D.C.: American Geophysical Union. TIC: 217475.

Tang, D.H.; Frind, E.O.; and Sudicky, E.A. 1981. "Contaminant Transport in Fractured Porous Media: Analytical Solution for a Single Fracture." *Water Resources Research,* 17, (3), 555-564. Washington, D.C.: American Geophysical Union. TIC: 225358.

8.0 Attachments

ATTACHMENT	TITLE	NUMBER OF PAGES
I	Source Code List of FRACT_P	21
II	User's Guide to FRACT_P	6
III	Examples	3
	III.1. Example 1: Concentration along Fracture using Trapezoidal Rule Integration Scheme	
	III.2. Example 2: Concentration along Fracture using Gaussian Quadrature Integration Scheme	
IV	Input and Output Files for Type Curve Calculations	5
V	Complete File Information	1
VI	DIRS	2

Attachment I. Source Code List of FRACT_P

```
C FRACT_P Version 1.0
c This computer program will calculate concentration along a fracture in
c a parallel fracture system based on Sudicky and Frind (1982) with D=0
      program fract_p
      implicit integer*4 (i-n)
      implicit real*8 (a-h,o-z)
      PARAMETER(NPOINT=1024,ntmx=10000,nsmx=100,nwmx=100,ntmx1=10000)
      dimension x(NPOINT),w(NPOINT),t1(ntmx),s1(nsmx),w1(nwmx),
     $ds1(nsmx),dlam1(nsmx),s5(ntmx),w5(ntmx),
     $c1(ntmx),t4(ntmx),
     $c4(ntmx),it4(ntmx),c5(ntmx*34),t5(ntmx*34)
      COMMON /gauss/x,w,ngauss,xamin,dxam,xamax,errr
      COMMON /ndefi /nttot
      CHARACTER*32  jobid
c input parameters for calculate concentration
        write (*,6001)
6001      format (' Input JOBID for file definitions: ')
        read (*,5001) jobid
5001      format (a32)
        print *, ' '
        lenfil = index(jobid,' ') - 1
        if (lenfil.le.0 .or. lenfil gt.8) then
         print *, ' ***** Invalid JOBID *****'
         print *, '  -> must be 1 to 8 characters with no spaces  '
         stop
        end if
      ngauss=256
      open (10,FILE=jobid(:lenfil)//'.dat',STATUS='old')
      open (11,FILE=jobid(:lenfil)//'.log',STATUS='unknown')
c user's guide
c 1  prepare an input data file with extension of .dat
c 2  type program name
c 3  type input file name without extension at the prompt
c 4  look at results in file with extension .out
c iopt:=0, for type curve calculations; =1, for matching with Sudicky's results.
      READ (10,*)iopt,irestart
      IF(iopt.eq.0) then
      open (12,FILE=jobid(:lenfil)//'.out',STATUS='unknown')
      open (13,FILE=jobid(.lenfil)//'.rst',STATUS='unknown',
     & form='unformatted')
      call typecurve(jobid,irestart,lenfil)
      END if
c for concentration along x direction
      IF(iopt.eq.1.or.iopt.eq.2) then
      IF(iopt.eq.1) then
      WRITE(11,*)'Concentration along x axis using Trapezoidal rule inte
     &gration scheme'
      else
      WRITE(11,*)'Concentration along x axis using Gaussian quadrature i
     &ntegration scheme'
      END if
      read (10,*)ns1,ngauss,xamin,dxam,xamax,errr
```

```
      READ(10,*)vel,bs1,bb1,theta1,tort,dm,half,rf,rm
      READ (10,*)tt1
      READ (10,*)(ds1(i),i=1,ns1)
      WRITE(11,20)ns1,ngauss,xamin,dxam,xamax,errr
20    format(t5,'Number of distances= ',i5/
     $ t5,'Number of Gaussian points = ', i5,/
     $ t5,'Initial integration interval = ', e12.3,/
     $ t5,'Integration interval multiplier = ', e12.3,/
     $ t5,'Maximum integration interval = ', e12.3,/
     $ t5,'Relative integration error = ', e12.3/)
      write(11,30)vel,tt1,bs1,bb1,theta1,tort,dm,half,rf,rm
30    format(t5,'Velocity (m/d) = ',f10.5/
     $ t5,'Simulation time (d) = ', e12.3,/
     $ t5,'Half aperture (m) = ', f10.5,/
     $ t5,'Half fracture spacing (m)= ', f10.5,/
     $ t5,'Matrix porosity = ', f10.5,/
     $ t5,'Tortuosity = ', f10.5,/
     $ t5,'Diffusion coef.(cm2/s) = ', e12 3/
     $ t5,'Half-life (years) = ', e12.3/
     $ t5,'Fracture retardation = ', f10.5/
     $ t5,'Matrix retardation = ', f10.5/)
      Write(11, '(t5,a)')'Distances (m) = '
      write(11,'(10f10.3)')(ds1(i),i=1,ns1)
      write(11,*)
      dm=dm*tort*8.64d0
      dlambda=dlog(2.0d0)/(365.0d0*half)
      do i=1,ns1
        tau0=ds1(i)/vel
        t1(i)=tt1/tau0-rf
        s1(i)=(bb1-bs1)*(rm/tau0/dm)**0.5
        w1(i)=theta1*(tau0*dm*rm)**0.5/bs1
        dlam1(i)=dlambda*tau0
      end do
      X1=-1 0
      X2=1.0
      call gauleg(X1,X2,x,w,ngauss)
         do k=1,ns1
         IF(iopt eq.1) then
          call integra1(t1(k),w1(k),s1(k),c1(k),dlam1(k),rf)
          else
          call integra3(t1(k),w1(k),s1(k),c1(k),dlam1(k),rf)
          END if
          WRITE(11,'(2e20.10)')ds1(k),c1(k)
         end do
      END if
      IF(iopt eq.3) then
      WRITE(11,*)'Breakthrough curves x axis using Gaussian quadrature i
     &ntegration scheme'
      read (10,*)ns1,ngauss,xamin,dxam,xamax,errr
      READ(10,*)tau01,bs1,bb1,theta1,dm01,rf,rm
      tau0=tau01/60.0/60.0/24.0
      dm=dm01*60 0*60.0*24.0
      do i=1,ns1
      READ (10,*)t4(i),c4(i)
      end do
      WRITE(11,32)ns1,ngauss,xamin,dxam,xamax,errr
```

```
32   format(t5,'Number of times= ',i5/
   $ t5,'Number of Gaussian points = ', i5,/
   $ t5,'Initial integration interval = ', e12 3,/
   $ t5,'Integration interval multiplier = ', e12.3,/
   $ t5,'Maximum integration interval = ', e12.3,/
   $ t5,'Relative integration error = ', e12 3/)
    write(11,34)tau0,tau01,bs1,bb1,theta1,dm,dm01,rf,rm
34   format(t5,'Plug flow time (days) = ',e12.3/
   $ t5,'Plug flow time (sec) = ',e12.3/
   $ t5,'Half aperture (m) = ', f10.5,/
   $ t5,'Half fracture spacing (m)= ', f10.5,/
   $ t5,'Matrix porosity = ', f10.5,/
   $ t5,'Diffusion coef.(m2/day) = ', e12 3/
   $ t5,'Diffusion coef.(m2/sec) = ', e12.3/
   $ t5,'Fracture retardation = ', f10 5/
   $ t5,'Matrix retardation = ', f10 5/)
    Write(11, '(t5,a)')'Calculation Time (days) = '
    write(11,'(6(1pe12.3))')(t4(i),i=1,ns1)
    write(11,*)
    X1=-1.0
    X2=1.0
    call gauleg(X1,X2,x,w,ngauss)
    do i=1,ns1
      t1(i)=t4(i)/tau0-rf
    end do
    s1(1)=(bb1-bs1)*(rm/tau0/dm)**0.5
    w1(1)=theta1*(tau0*dm*rm)**0 5/bs1
    WRITE(11,36)s1(1),w1(1)
36   format(t5,'Sigma1 = ',f10 5/
   $ t5,'Omega1 = ', f10.5/)
    WRITE(11,*)
    WRITE(11,*) 'Calculation results.'
    WRITE(11,*)
    WRITE(11,39)'Dimensionless time','Calculation-time',
   $ 'Analy-Solution','FEHM-output'
39   FORMAT(t3,a,t25,a,t45,a,t65,a)
    do k=1,ns1
          if(t1(k).gt.0.0) then
    call integra2(t1(k),w1(1),s1(1),c1(k))
    IF(c1(k).lt.0.0) c1(k)=0.0
          Else
          C1(k)=0.0d0
          End if
    WRITE(11,'(4e20.10)')t1(k),t4(k),c1(k),c4(k)
    end do
    END if
    CLOSE(10)
    CLOSE(11)
    stop
    end
    subroutine typecurve(jobid,irestart,lenfil)
    implicit integer*4 (i-n)
    implicit real*8 (a-h,o-z)
    PARAMETER(NPOINT=1024,ntmx=10000,nsmx=100,nwmx=100,ntmx1=10000)
    dimension x(NPOINT),w(NPOINT),t1(ntmx),s1(nsmx),w1(nwmx),
   $ds1(nsmx),dlam1(nsmx),s5(ntmx),w5(ntmx),
```

```
      $c1(ntmx),t4(ntmx),
      $c4(ntmx),it4(ntmx),c5(ntmx*34),t5(ntmx*34)
       COMMON /gauss/x,w,ngauss,xamin,dxam,xamax,errr
       COMMON /ndefi /nttot
       CHARACTER*32  jobid
       if(irestart.gt.0) then
       read (13)ngauss,xamin,dxam,xamax,errr
       READ (13)tmin,tmax,ntlog
       READ (13)smin,smax,nslog
       READ (13)wmin,wmax,nwlog
       END if
       read (10,*)ngauss,xamin,dxam,xamax,errr
       READ (10,*)tmin,tmax,ntlog
       READ (10,*)smin,smax,nslog
       READ (10,*)wmin,wmax,nwlog
       IF(irestart eq.0) then
       write (13)ngauss,xamin,dxam,xamax,errr
       write (13)tmin,tmax,ntlog
       write (13)smin,smax,nslog
       write (13)wmin,wmax,nwlog
       END if
       write(11,10)ngauss,xamin,dxam,xamax,errr
10     format(t5,'Number of Gaussian points = ', i5,/
      $ t5,'Initial integration interval = ', e12.3,/
      $ t5,'Integration interval multiplier = ', e12.3,/
      $ t5,'Maximum integration interval = ', e12.3,/
      $ t5,'Relative integration error = ', e12.3/)
       Write(11, '(t5,a)')'Simulation times'
       write(11,11)tmin,tmax,ntlog
11     format(t5,'Minimum time = ', e13.5,/
      $ t5,'Maximum time = ', e13 5,/
      $ t5,'Number of points in one-log cycle = ', i5/)
       Write(11, '(t5,a)')'Sigma1 = '
       write(11,12)smin,smax,nslog
12     format(t5,'Minimum sigma1 = ', e13.5,/
      $ t5,'Maximum sigma1 = ', e13.5,/
      $ t5,'Number of points in one-log cycle = ', i5/)
       Write(11, '(t5,a)')'Omega1 = '
       write(11,13)wmin,wmax,nwlog
13     format(t5,'Minimum omega1 = ', e13.5,/
      $ t5,'Maximum omega1 = ', e13 5,/
      $ t5,'Number of points in one-log cycle = ', i5/)
c calculation of t,s,w points
        itmin=dlog10(tmin)-0 5
        itmax=dlog10(tmax)+0 5
        itmin1=itmin
        itmax1=itmax
        itmin2=itmin
        itmax2=itmax
        dtlog=1.0/DBLE(ntlog)
        do i=itmin,itmax-1
        i1=(i-itmin)*ntlog+1
        t1(i1)=10 0d0**(DBLE(i))
        do j=1,ntlog-1
        i1=i1+1
        t1(i1)=10 0d0**(DBLE(i)+dtlog*DBLE(j))
```

```
      end do
      end do
      nt1=ntlog*(itmax-itmin)+1
      t1(i1+1)=10.0d0**(DBLE(itmax))
c sigma1
      ismin=dlog10(smin)-0.5
      ismax=dlog10(smax)+0.5
      dslog=1 0/DBLE(nslog)
      do i=ismin,ismax-1
      i1=(i-ismin)*nslog+1
      s1(i1)=10.0d0**(DBLE(i))
      do j=1,nslog-1
      i1=i1+1
      s1(i1)=10.0d0**(DBLE(i)+dslog*DBLE(j))
      end do
      end do
      ns1=nslog*(ismax-ismin)+1
      s1(i1+1)=10.0d0**(DBLE(ismax))
c omega1
      iwmin=dlog10(wmin)-0.5
      iwmax=dlog10(wmax)+0.5
      dwlog=1 0/DBLE(nwlog)
      do i=iwmin,iwmax-1
      i1=(i-iwmin)*nwlog+1
      w1(i1)=10.0d0**(DBLE(i))
      do j=1,nwlog-1
      i1=i1+1
      w1(i1)=10.0d0**(DBLE(i)+dwlog*DBLE(j))
      end do
      end do
      nw1=nwlog*(iwmax-iwmin)+1
      w1(i1+1)=10.0d0**(DBLE(iwmax))
     Write(11, '(t5,a)')'Simulation times'
     write(11,'(10e12.3)')(t1(i),i=1,nt1)
     write(11,*)
     Write(11, '(t5,a)')'Sigma1 = '
     write(11,'(10e12.3)')(s1(i),i=1,ns1)
     write(11,*)
     Write(11, '(t5,a)')'Omega1 = '
     write(11,'(10e12.3)')(w1(i),i=1,nw1)
     write(12,*)ns1
     write(12,'(10e12.3)')(s1(i),i=1,ns1)
     write(12,*)nw1
     write(12,'(10e12.3)')(w1(i),i=1,nw1)
c concentration values to be calculated
     c1(1)=1.0e-8
     c1(2)=1.0e-6
     c1(3)=1.0e-5
     c1(4)=1.0e-4
     c1(5)=1.0e-3
     c1(6)=1.0e-2
     c1(7)=0 1
     c1(8)=0.16
     c1(9)=0.22
     c1(10)=0.28
     c1(11)=0 34
```

```
      c1(12)=0.40
      c1(13)=0.46
      c1(14)=0.52
      c1(15)=0.58
      c1(16)=0.64
      c1(17)=0 70
      c1(18)=0.76
      c1(19)=0.80
      c1(20)=0.82
      c1(21)=0.84
      c1(22)=0.86
      c1(23)=0.88
       c1(24)=0.90
       c1(25)=0.92
       c1(26)=0.94
       c1(27)=0.95
       c1(28)=0.96
       c1(29)=0.97
       c1(30)=0.98
       c1(31)=0.99
       c1(32)=0.995
       c1(33)=0.999
       c1(34)=0.9999
       C1(35)=1.0
       nttot=34
       nt1=34
      CLOSE(10)
      X1=-1.0
      X2=1.0
      call gauleg(X1,X2,x,w,ngauss)
      write(11,*)
      IF(irestart.eq 0) then
      is2=1
      iw2=1
c      ns1=1
c      nw1=1
c      s1(1)=0.215E+01
c      w1(1)= 0.215E+01
      else
      is2=0
      nt1=34
      nttot=34
       do i=1,ns1
       is2=is2+1
        iw2=0
        do j=1,nw1
            iw2=iw2+1
            read(13,iostat=ijk100)nt1,s1(i),w1(j)
            if (ijk100.lt 0) goto 1999
            if(irestart lt.3) then
            write(11,200)s1(i),w1(j)
            write(12,*)nt1,s1(i),w1(j)
            end if
            do k=1,nt1
            read(13)t4(k),c4(k)
            t5((i-1)*nt1*nw1+(j-1)*nt1+k)=t4(k)
```

```
            c5((i-1)*nt1*nw1+(j-1)*nt1+k)=c4(k)
            if(irestart.lt.3) then
            WRITE(11,'(2e20.10)')t4(k),c4(k)
            IF(c4(k).gt.1.0d0) c4(k)=1.0d0
            if(c4(k) lt.0.0d0) c4(k)=0 0d0
            WRITE(12,'(2e20.10)')t4(k),c4(k)
            end if
            end do
        END do
      END do
1999   continue
    CLOSE(13)
     if(irestart.lt.3) then
    open (13,FILE=jobid(:lenfil)//'.rst',STATUS='unknown',
   & form='unformatted',access='append')
    else
    open (13,FILE=jobid(.lenfil)//' rst',STATUS='unknown',
   & form='unformatted')
    end if
    END if
    IF(irestart.eq.0) then
     ij=0
     do i=1,ns1
      do j=1,nw1
      ij=ij+1
      s5(ij)=s1(i)
      w5(ij)=w1(j)
      end do
     end do
     iws1=1
     iws2=ns1*nw1
    END if
    IF(irestart eq.1) then
     ij=0
     do i=1,ns1
      do j=1,nw1
      ij=ij+1
      s5(ij)=s1(i)
      w5(ij)=w1(j)
      end do
     end do
     iws1=(is2-1)*nw1+iw2
     iws2=ns1*nw1
          if(iws1.gt.iws2) iws1=iws2
    END if
    IF(irestart.eq.2) then
     ij=0
     do i=1,ns1
      do j=1,nw1
      ij=ij+1
      s5(ij)=s1(i)
      w5(ij)=w1(j)
      end do
     end do
     iws1=(is2-1)*nw1+iw2
     iws2=ns1*nw1
```

```
          if(iws1.gt.iws2) iws1=iws2
      END if
      IF(irestart.eq.3) then
       ij=0
       do i=1,ns1
        do j=1,nw1
        ij=ij+1
        s5(ij)=s1(i)
        w5(ij)=w1(j)
        end do
       end do
       iws1=1
       iws2=(is2-1)*nw1+iw2
      END if
        do ij=iws1,iws2
         i=(ij-1)/nw1+1
         j=ij-(i-1)*nw1
         s1(i)=s5(ij)
         w1(j)=w5(ij)
         IF(irestart.gt.1) then
         do k=1,nt1
         t4(k)=t5((i-1)*nt1*nw1+(j-1)*nt1+k)
         c4(k)=c5((i-1)*nt1*nw1+(j-1)*nt1+k)
         end do
         END if
         write(11,200)s1(i),w1(j)
         if (irestart.eq 3.or.irestart eq.2) then
         GOTO 300
         end if
c calculate selected point at 10**i
        IF(irestart.Lt.2) then
         itmin2=-18
         itmax2=18
         itmin=itmin2
         itmax=itmax2
         delt1=1.0
         i3=0
         i4=0
55        k1=0
         do k=itmin,itmax
         k1=k1+1
         t4(k1)=10.0d0**(DBLE(k))
          call integra(t4(k1),w1(j),s1(i),c4(k1))
          IF(c4(k1) gt 0.99999d0) then
          itmax=k
          GOTO 45
          END if
c          WRITE(11,'(2e20.10)')t4(k1),c4(k1)
         end do
c for cases out of range
45        continue
         imin=0
         k1=0
         do k=itmin,itmax
         k1=k1+1
          IF(c4(k1) gt c1(1) and.imin eq 0) then
```

```
        imin=1
        itmin=k
        i3=3
        END if
        IF(c4(k1).ge.c1(nttot)) then
        itmax=k
        i4=4
        END if
       end do
c values of calculated concentration are too small
       IF(c1(1).GE.c4(k1)) then
       itmax1=itmax
       itmax=itmax+(itmax-itmin)
       itmin=itmax1
       GO TO 55
       END if
c values of calculated concentration are too large
       IF(dabs(c4(1)-c1(nttot)).lt.1d-2 or.c4(1).ge.1.0d0)then
       itmin1=itmax
       itmin=itmin-(itmax-itmin)
       itmax=itmin1
       GO TO 55
       END IF
c find itmin
       k2=0
       do k=itmin,itmax-1
       k2=k2+1
       IF(c4(k2).gt.c1(1).and.c4(k2) le c1(2))then
       i3=3
       itmin=k
       END if
c find itmax
      IF(c4(k2).gt.c1(nttot-1) and.c4(k2) le c1(nttot))then
       i4=4
       itmax=k
       END if
      end do
c extend time in the direction of the smaller concentration values
56     IF(itmin.lt itmin2) GOTO 57
      IF(i3.eq.0) then
       itmin=itmin-1
       t4(1)=10.0d0**(DBLE(itmin))
       call integra(t4(1),w1(j),s1(i),c4(1))
       IF(c4(1).gt.c1(1).and.c4(1) le.c1(2))then
      i3=3
      GOTO 57
      else
      i3=0
      GO TO 56
      END if
      end if
c extend time in the direction of the larger concentration values
57     IF(itmax gt.itmax2) GOTO 58
      IF(i4.eq.0) then
       itmax=itmax+1
       t4(k1)=10.0d0**(DBLE(itmax))
```

```
      call integra(t4(k1),w1(j),s1(i),c4(k1))
      IF(c4(k2).gt.c1(nttot-1) and.c4(k2).le.c1(nttot))then
      i4=4
       GOTO 58
      else
      i4=0
      GO TO 57
      END if
      END if
58    continue
c decide range of t to be calculated
      dtlog=1.0/DBLE(ntlog)
      IF(itmin.eq.itmax) then
      itmin=itmin-1
      itmax=itmax+1
      END if
      do ii=itmin,itmax-1
      i1=(ii-itmin)*ntlog+1
      t4(i1)=10.0d0**(DBLE(ii))
      do jj=1,ntlog-1
      i1=i1+1
      t4(i1)=10 0d0**(DBLE(ii)+dtlog*DBLE(jj))
      end do
      end do
      nt1=ntlog*(itmax-itmin)+1
      t4(i1+1)=10.0d0**(DBLE(itmax))
       do k=1,nt1
        call integra(t4(k),w1(j),s1(i),c4(k))
        IF(c4(k).ge.0.99999d0) then
        nt1=k
        GO TO 180
        END if
       end do
180   continue
c end of if(irestart.ne.2) then
      END if
c delete times for larger concentration values
      delt1=1.0d0
      iter=0
190   IF(iter.gt 100.and.nt1.eq nttot) GOTO 300
      iter=iter+1
      ntt2=1
      do k=1,nt1-1
      if (c4(k).lt c1(1).and c4(k+1).gt.c1(1)) then
      ntt2=k
      END if
      end do
      nt1=nt1-ntt2+1
      do k=1,nt1
      t4(k)=t4(k+ntt2-1)
      c4(k)=c4(k+ntt2-1)
      end do
      call timeshort(c1,t4,c4,it4,nt1)
      IF(nt1.eq nttot) GOTO 300
      call sort3(nt1,t4,c4)
c check to see if the time points are less than nttot
```

```
      IF(nt1.lt.nttot) then
      call timeinter(c1,t4,c4,it4,nt1,delt1)
      do k=1,nt1
      IF(it4(k) eq.0) then
      call integra(t4(k),w1(j),s1(i),c4(k))
      IF(c4(k) gt.0.99999d0) then
      nt1=k
      GOTO 190
      END if
      END if
      end do
      END if
      GO TO 190
300     continue
      IF(irestart.eq 3.or.irestart.eq.2) then
       do k=1,nt1
       call integra2(t4(k),w1(j),s1(i),c4(k))
       END do
      END if
      write(12,*)nt1,s1(i),w1(j)
      write(13)nt1,s1(i),w1(j)
      call sort2(nt1,t4,c4)
      do k=1,nt1
        WRITE(11,'(2e20.10)')t4(k),c4(k)
        WRITE(13)t4(k),c4(k)
        IF(c4(k).gt.1.0d0) c4(k)=1.0d0
        if(c4(k).lt.0.0d0) c4(k)=0 0d0
        WRITE(12,'(2e20.10)')t4(k),c4(k)
      end do
      end do
    CLOSE(12)
    CLOSE(13)
200   format(/,5x,'Sigma1=',e12.3,5x,'Omega1=',e12.3/)
    CLOSE(11)
     stop
    end
    subroutine sort3(nt1,t4,c4)
    implicit integer*4 (i-n)
    implicit real*8 (a-h,o-z)
    PARAMETER(NPOINT=1024,ntmx=10000,nsmx=100,nwmx=100,ntmx1=10000)
    COMMON /ndefi /nttot
    DIMENSION c4(ntmx),it4(ntmx),t4(ntmx),t3(ntmx),c1(ntmx),c3(ntmx),
   & it3(ntmx)
    call sort2(nt1,t4,c4)
    do i=1,nt1-1
      IF(c4(i) gt.c4(i+1)) then
      nt1=i
      GOTO 100
      END if
    end do
100  continue
    return
    end

    subroutine timeshort(c1,t4,c4,it4,nt1)
    PARAMETER(NPOINT=1024,ntmx=10000,nsmx=100,nwmx=100,ntmx1=10000)
```

```
      implicit integer*4 (i-n)
      implicit real*8 (a-h,o-z)
      COMMON /ndefi /nttot
      DIMENSION c4(ntmx),it4(ntmx),t4(ntmx),t3(ntmx),c1(ntmx),c3(ntmx),
     & it3(ntmx)
      ntt2=nt1-nttot
      i2=0
      i1=0
       do 10 j=1,nt1
       it4(j)=1
10      continue
      do 20 i=1,nttot
       i2=0
       do 21 j=1,nt1
c figure out how many points needs to be deleted
        IF(c4(j) gt.c1(i) and c4(j).le.c1(i+1)) then
        i2=i2+1
        it3(i2)=j
        END if
21      CONTINUE
       IF(i2.gt.1) then
        do 22 j=1,i2-1
        it4(it3(j))=0
        i1=i1+1
22      continue
       endif
20      continue
c do interpolation
      i3=0
      do 30 i=1,nt1
      IF(it4(i) eq.1) then
      i3=i3+1
      t3(i3)=t4(i)
      c3(i3)=c4(i)
      END if
30      continue
      nt1=i3
      do 40 i=1,nt1
      t4(i)=t3(i)
      c4(i)=c3(i)
40      continue
      return
      end

      subroutine timeinter(c1,t4,c4,it4,nt1,delt1)
      PARAMETER(NPOINT=1024,ntmx=10000,nsmx=100,nwmx=100,ntmx1=10000)
      implicit integer*4 (i-n)
      implicit real*8 (a-h,o-z)
      COMMON /ndefi /nttot
      DIMENSION c4(ntmx),it4(ntmx),t4(ntmx),t3(ntmx),c1(ntmx),c3(ntmx)
      IF(nt1.eq.1) then
       IF(c4(1) gt.0.5) then
       t4(2)=t4(1)
       c4(2)=c4(1)
       t4(1)=t4(1)/10.0d0
       c4(1)=1.0e-6
```

```
      else
      t4(2)=t4(1)*10.0d0
      c4(2)=1.0
      END if
      nt1=2
     END if
     do i=1,nt1
     IF(c4(i).lt.1e-30) c4(i)=1.e-30
     end do
     ntt2=nttot-nt1
     i2=0
     i3=0
     IF(nt1.eq.2.AND.c4(1) lt c1(1) and.c4(2).gt.c1(1)) then
     delt1=delt1*2.0
      it4(nttot)=1
      t4(nttot)=t4(2)
      c4(nttot)=c4(2)
      do i=1,nttot-1
      dtlog=1.0d0/DBLE(nttot)/delt1
      t4(nttot-i)=t4(2)/10**(dtlog*i)
      it4(nttot-i)=0
      END do
     nt1=nttot
     return
     END if
c
     IF(nt1.eq.2.AND c4(2) lt c1(1)) then
      delt1=delt1*2.0
      it4(1)=1
      do i=2,nttot
      dtlog=1.0d0/DBLE(nttot)/delt1
      t4(i)=t4(1)*10**(dtlog*i)
      it4(i)=0
      end do
      nt1=nttot
     return
     END if
c
     do j=1,nttot
      i1=0
      do i=1,nt1
c figure out how many points needs to be interpolated
      IF(c4(i).gt.c1(j) and.c4(i).le c1(j+1)) then
      i1=i1+1
      i2=i
      END if
      end do
      IF(i1.lt.1) then
       IF(i2.eq.0.or.i2.eq nt1) then
       IF(i2.eq.0) then
       i2=1
       caverage=(dlog10(c1(j))+dlog10(c1(j+1)))/2.0d0
       dtlog=(dlog10(t4(i2+1))-dlog10(t4(i2)))
       dtlog=dtlog/(dlog10(c4(i2+1))-dlog10(c4(i2)))
       dtlog=dlog10(t4(i2))+(caverage-dlog10(c4(i2)))*dtlog
       it4(j)=0
```

```
        c3(j)=0.0
        t3(j)=10.0d0**(dtlog)
        END if
        IF(i2.eq nt1) then
        i2=nt1-1
        caverage=(dlog10(c1(j))+dlog10(c1(j+1)))/2.0d0
        dtlog=(dlog10(t4(i2+1))-dlog10(t4(i2)))
        dtlog=dtlog/(dlog10(c4(i2+1))-dlog10(c4(i2)))
        dtlog=dlog10(t4(i2))+(caverage-dlog10(c4(i2)))*dtlog
        it4(j)=0
        c3(j)=0 0
        t3(j)=10.0d0**(dtlog)
        END if
        else
        caverage=(dlog10(c1(j))+dlog10(c1(j+1)))/2.0d0
        dtlog=(dlog10(t4(i2+1))-dlog10(t4(i2)))
        dtlog=dtlog/(dlog10(c4(i2+1))-dlog10(c4(i2)))
        dtlog=dlog10(t4(i2))+(caverage-dlog10(c4(i2)))*dtlog
        it4(j)=0
        c3(j)=0.0
        t3(j)=10.0d0**(dtlog)
        END if
       else
       t3(j)=t4(i2)
       c3(j)=c4(i2)
       it4(j)=1
       END if
      END do
c     it4(nttot)=1
c     t3(nttot)=t4(nt1)
c     c3(nttot)=c4(nt1)
      nt1=nttot
      do i=1,nttot
      t4(i)=t3(i)
      c4(i)=c3(i)
      end do
      return
      end
c integration calculation for type curves without decay
      subroutine integra2(t2,w2,s2,xx)
      implicit integer*4 (i-n)
      implicit real*8 (a-h,o-z)
      PARAMETER(NPOINT=1024)
      dimension x(NPOINT),w(NPOINT)
      COMMON /gauss/x,w,ngauss,xamin,dxam,xamax,errr
      pi=3.14159265d0
      a=0.0d0
      b=xamin
      xx=0 0
      xa1=0.0
      k=0
      k=k+1
      b1=dsqrt(4.0*pi/t2)
      b=b1
10    continue
      xold=xx
```

```
      xa1=xa1+(b-a)
      do 12 i=1,ngauss
      ww1=w(i)*(b-a)/2.d0
      zz1=(x(i)*(b-a)+b+a)/2 d0
      xx=xx+ww1*cinteg(t2,w2,s2,zz1)
12    continue
      a=xa1
c     b1=b1*dxam
c     IF(b1.gt.xamax) b1=xamax
c     b=xa1+b1
      k=k+1
      b=dsqrt(4.0*pi*k*k/t2)
c calculate integration interval
      IF(dabs(xold-xx) ge.dabs(xold)*errr.and.xa1 lt.1.0d30
     & )GO TO 10
      xx=xx*2 0d0/pi
      IF(xx.lt.1.0e-30) xx=1.e-30
      return
      end
c integrand calculation for type curves without decay
      real*8  function cinteg(tt1,ww1,ss1,ee1)
      implicit integer*4 (i-n)
      implicit real*8 (a-h,o-z)
      xx1=er1(ww1,ss1,ee1)
      xx2=ei1(tt1,ww1,ss1,ee1)
      xx3=omeg0(ww1,ss1,ee1)
      cinteg=dexp(xx1)*(dsin(xx2)+dsin(xx3))/ee1
      return
      end
c integration calculation with decay
c Gaussian quadrature technique for integration
      subroutine integra1(t2,w2,s2,xx,dlam2,rf)
      implicit integer*4 (i-n)
      implicit real*8 (a-h,o-z)
      PARAMETER(NPOINT=1024)
      dimension x(NPOINT),w(NPOINT)
      COMMON /gauss/x,w,ngauss,xamin,dxam,xamax,errr
      pi=3 14159265d0
      a=0.0d0
      b=xamin
      b1=xamin
      xx=0.0
      xa1=0.0
10    continue
      xold=xx
      xa1=xa1+(b-a)
      do 12 i=1,ngauss
      ww1=w(i)*(b-a)/2.d0
      zz1=(x(i)*(b-a)+b+a)/2 d0
      xx=xx+ww1*cinteg1(t2,w2,s2,zz1,dlam2)
12    continue
      a=xa1
      b1=b1*dxam
      IF(b1.gt.xamax) b1=xamax
      b=xa1+b1
      IF(dabs(xold-xx) ge.dabs(xold)*errr) GO TO 10
```

```
      xx=xx*1.0d0/pi*dexp(-dlam2*rf)
      return
      end
c integration calculation for type curves without decay
      subroutine integra(t2,w2,s2,xx)
      implicit integer*4 (i-n)
      implicit real*8 (a-h,o-z)
      PARAMETER(NPOINT=1024)
      dimension x(NPOINT),w(NPOINT)
      COMMON /gauss/x,w,ngauss,xamin,dxam,xamax,errr
      pi=3.14159265d0
      a=0.0d0
      b=xamin
      b1=xamin
      xx=0 0
      xa1=0 0
10    continue
      xold=xx
      xa1=xa1+(b-a)
      do 12 i=1,ngauss
      ww1=w(i)*(b-a)/2.d0
      zz1=(x(i)*(b-a)+b+a)/2.d0
      xx=xx+ww1*cinteg(t2,w2,s2,zz1)
12    continue
      a=xa1
      b1=b1*dxam
      IF(b1.gt.xamax) b1=xamax
      b=xa1+b1
      IF(dabs(xold-xx).gt.dabs(xold)*errr and xa1.lt.1.0d40) GO TO 10
      xx=xx*2.0d0/pi
      IF(xx.lt.1.0e-30) xx=1.e-30
      return
      end

c modified from numerical recipe
      SUBROUTINE trapzd(a,b,xx,t2,w2,s2,zz1,dlam2,n)
      implicit integer*4 (i-n)
      implicit real*8 (a-h,o-z)
      if (n eq.1) then
      xx=0.5*(b-a)*(cinteg1(t2,w2,s2,a,dlam2)+cinteg1(t2,w2,s2,b,dlam2))
      else
       it=2**(n-2)
       tnm=it
       del=(b-a)/tnm
       x=a+0.5*del
       sum=0.
       do 11 j=1,it
        sum=sum+cinteg1(t2,w2,s2,x,dlam2)
        x=x+del
11      continue
       xx=0.5*(xx+(b-a)*sum/tnm)
      end if
      return
      END
c modified from numerical recipe
      SUBROUTINE qtrap(a,b,xx,t2,w2,s2,zz1,dlam2)
```

```
      implicit integer*4 (i-n)
      implicit real*8 (a-h,o-z)
      PARAMETER (EPS=1.e-6, JMAX=256)
      olds=-1.e30
      do 11 j=1,JMAX
       call trapzd(a,b,xx,t2,w2,s2,zz1,dlam2,j)
       if (abs(xx-olds).le.EPS*abs(olds)) return
       olds=xx
11    continue
      return
      END
c trapezoidal rule technique for integration
      subroutine integra3(t2,w2,s2,xx,dlam2,rf)
      implicit integer*4 (i-n)
      implicit real*8 (a-h,o-z)
      PARAMETER(NPOINT=1024)
      dimension x(NPOINT),w(NPOINT)
      COMMON /gauss/x,w,ngauss,xamin,dxam,xamax,errr
      pi=3.14159265d0
      a=0 0d0
      b=xamin
      b1=xamin
      xx1=0 0
      xa1=0.0
10    continue
      xold=xx1
      xa1=xa1+(b-a)
      call qtrap(a,b,xx,t2,w2,s2,zz1,dlam2)
      xx1=xx1+xx
      a=xa1
      b1=b1*dxam
      IF(b1.gt.xamax) b1=xamax
      b=xa1+b1
      IF(dabs(xold-xx1) gt.dabs(xold)*errr) GO TO 10
       xx=xx1*1.0d0/pi*dexp(-dlam2*rf)
      return
      end
c integrand calculation with decay
      real*8  function cinteg1(tt1,ww1,ss1,ee1,dlam2)
      implicit integer*4 (i-n)
      implicit real*8 (a-h,o-z)
      xx1=er1(ww1,ss1,ee1)
      xx2=ei1(tt1,ww1,ss1,ee1)
      xx3=omeg0(ww1,ss1,ee1)
      ee2=ee1*ee1*0.5
      cinteg1=dexp(xx1)*(dexp(-dlam2*tt1)*(ee2*dsin(xx2)
     @ -dlam2*dcos(xx2))+ee2*dsin(xx3)+dlam2*dcos(xx3))
     $ /(dlam2*dlam2+ee2*ee2)*ee1
      return
      end
c function calculation for er0
      real*8  function er1(ww1,ss1,ee1)
      implicit integer*4 (i-n)
      implicit real*8 (a-h,o-z)
      er1=-ww1*ee1*0.5d0
      se1=ss1*ee1
```

```
      IF(se1.gt.100.0d0) then
      dexp2=0.0
      else
      dexp2=dexp(-se1*2.0d0)
      END if
      IF(se1.gt.200.0d0) then
      dexp1=0.0
      else
      dexp1=dexp(-se1)
      END if
      er1=er1*(1.0d0-dexp2-2.0*dsin(se1)*dexp1)/(1.0d0+dexp2+
     & 2 0*dcos(se1)*dexp1)
       return
      end
c function calculation for ei0
      real*8   function ei1(tt1,ww1,ss1,ee1)
      implicit integer*4 (i-n)
      implicit real*8 (a-h,o-z)
      er1=-ww1*ee1*0 5d0
      ei1=ee1*ee1*tt1*0.5d0
      se1=ss1*ee1
      IF(se1.gt.100 0d0) then
      dexp2=0 0
      else
      dexp2=dexp(-se1*2.0d0)
      END if
      IF(se1.gt 200.0d0) then
      dexp1=0 0
      else
      dexp1=dexp(-se1)
      END if
      er1=er1*(1 0d0-dexp2+2.0*dsin(se1)*dexp1)/(1.0d0+dexp2+
     & 2 0*dcos(se1)*dexp1)
       ei1=ei1+er1
       return
      end
c function calculation for omega
      real*8 function omeg0(ww1,ss1,ee1)
      implicit integer*4 (i-n)
      implicit real*8 (a-h,o-z)
      omeg0=ww1*ee1*0.5d0
      se1=ss1*ee1
      IF(se1.gt.100 0d0) then
      dexp2=0 0
      else
      dexp2=dexp(-se1*2.0d0)
      END if
      IF(se1 gt.200.0d0) then
      dexp1=0.0
      else
      dexp1=dexp(-se1)
      END if
      omeg0=omeg0*(1.0d0-dexp2+2.0*dsin(se1)*dexp1)/(1.0d0+dexp2+
     & 2 0*dcos(se1)*dexp1)
       return
      end
```

```
c the following subroutine is modified from Numerical Recipes
      SUBROUTINE gauleg(x1,x2,x,w,n)
      implicit integer*4 (i-n)
      implicit real*8 (a-h,o-z)
      dimension x(n),w(n)
      PARAMETER (EPS=3.d-14)
      m=(n+1)/2
      xm=0 5d0*(x2+x1)
      xl=0 5d0*(x2-x1)
      do 12 i=1,m
       z=cos(3.141592654d0*(i-.25d0)/(n+.5d0))
1      continue
        p1=1.d0
        p2=0.d0
        do 11 j=1,n
         p3=p2
         p2=p1
         p1=((2.d0*j-1.d0)*z*p2-(j-1.d0)*p3)/j
11       continue
        pp=n*(z*p1-p2)/(z*z-1.d0)
        z1=z
        z=z1-p1/pp
       if(abs(z-z1).gt EPS)goto 1
       x(i)=xm-xl*z
       x(n+1-i)=xm+xl*z
       w(i)=2.d0*xl/((1.d0-z*z)*pp*pp)
       w(n+1-i)=w(i)
12    continue
      return
      END

      SUBROUTINE sort2(n,arr,brr)
      implicit integer*4 (i-n)
      implicit real*8 (a-h,o-z)
      PARAMETER (M=7,NSTACK=50)
      dimension arr(n),brr(n),istack(NSTACK)
c      INTEGER i,ir,j,jstack,k,l,
c      REAL a,b,temp
       if (n.lt.2) then
       return
       end if
      jstack=0
      l=1
      ir=n
1      if(ir-l.lt.M)then
       do 12 j=l+1,ir
        a=arr(j)
        b=brr(j)
        do 11 i=j-1,1,-1
         if(arr(i) le.a)goto 2
         arr(i+1)=arr(i)
         brr(i+1)=brr(i)
11       continue
        i=0
2        arr(i+1)=a
```

```
        brr(i+1)=b
12      continue
      if(jstack.eq.0)return
      ir=istack(jstack)
      l=istack(jstack-1)
      jstack=jstack-2
    else
      k=(l+ir)/2
      temp=arr(k)
      arr(k)=arr(l+1)
      arr(l+1)=temp
      temp=brr(k)
      brr(k)=brr(l+1)
      brr(l+1)=temp
      if(arr(l+1).gt.arr(ir))then
        temp=arr(l+1)
        arr(l+1)=arr(ir)
        arr(ir)=temp
        temp=brr(l+1)
        brr(l+1)=brr(ir)
        brr(ir)=temp
      endif
      if(arr(l).gt.arr(ir))then
        temp=arr(l)
        arr(l)=arr(ir)
        arr(ir)=temp
        temp=brr(l)
        brr(l)=brr(ir)
        brr(ir)=temp
      endif
      if(arr(l+1).gt.arr(l))then
        temp=arr(l+1)
        arr(l+1)=arr(l)
        arr(l)=temp
        temp=brr(l+1)
        brr(l+1)=brr(l)
        brr(l)=temp
      endif
      i=l+1
      j=ir
      a=arr(l)
      b=brr(l)
3     continue
        i=i+1
      if(arr(i).lt.a)goto 3
4     continue
        j=j-1
      if(arr(j).gt.a)goto 4
      if(j.lt.i)goto 5
      temp=arr(i)
      arr(i)=arr(j)
      arr(j)=temp
      temp=brr(i)
      brr(i)=brr(j)
      brr(j)=temp
      goto 3
```

```
5     arr(l)=arr(j)
      arr(j)=a
      brr(l)=brr(j)
      brr(j)=b
      jstack=jstack+2
      if(jstack.gt.NSTACK)pause 'NSTACK too small in sort2'
      if(ir-i+1.ge.j-l)then
        istack(jstack)=ir
        istack(jstack-1)=i
        ir=j-1
      else
        istack(jstack)=j-1
        istack(jstack-1)=l
        l=i
      endif
    endif
    goto 1
    END
```

Attachment II. User's Guide to FRACT_P

There are one input file required and one or three output files produced by the FRACT_P. The source code first must be compiled as an executable code. When the executable is activated by typing the executable name, the program prompts a user for job identification, which is an eight-character name common for input and output files. The name of an input file is the job id (*) plus an extension *.dat. Main output file name is the job id plus an extension *.log. For type curve calculations, two additional output files are produced, one for restart of the program with an extension *.rst and one for plotting the type curves with an extension *.out.

Instructions for input file

1. READ (10,*)iopt, irestart

iopt: =0, for type curve calculation;

=1, for concentration calculation along a fracture with Gaussian qudrature;

=2, for concentration calculation along a fracture with Trapezoidal rule.

=3, for breakthrough curves at a location along a fracture.

irestart: =0, search relative times for type curves;

=1, continues the search of relative times for type curves from a previous job that was stopped (uses restart file *.rst)

=3, final calculation of type curves using the relative times determined from irestart=0 and/or 1 (the resulting *.rst file should be renamed to such as type2.rst). For irestart > 0, a restart file with the extension *.rst is required. The variable "irestart" is active for iopt =0.

In the following input statements, specify input statements 2 to 5 for iopt=0; statements 6 to 9 for iopt=1 or 2; and statements 10 to 12 for iopt=3.

2. READ (10,*) ngauss,xamin,dxam,xamax,errr

ngauss: the number of Gaussian quadrature points in integration.

xamin: initial integration interval.

dxam: a multiple of the integration interval.

xamax: maximum integration interval.

errr: relative error in the integration.

3. READ (10,*)tmin,tmax,ntlog

tmin: minimum relative time (T_1^0).

tmax: maximum relative time (T_1^0).

ntlog: number of time points in a log cycle.

4. READ (10,*) smin,smax,nslog

smin: minimum σ_1.

smax: maximum σ_1.

nslog: the number of σ_1 in a log cycle.

5. READ (10,*)wmin,wmax,nwlog

wmin: minimum ω_1.

wmax: maximum ω_1

nwlog: number of ω_1 points in a log cycle.

6. READ (10,*)ns1,ngauss,xamin,dxam,xamax,errr

ns1: number of distances.

ngauss: number of Gaussian points for use when iopt=1.

xamin: initial integration interval.

dxam: integration interval multiplier.

xamax: maximum integration interval.

errr: relative integration error.

7. READ (10,*)vel,bs1,bb1,theta1,tort,dm,half,rf,rm

vel: velocity (m/d).

bs1: half aperture (m).

bb1: half fracture spacing (m).

theta1: matrix porosity.

tort: tortuosity.

dm: diffusion coefficient (cm2/s).

half: half-life (years).

rf: fracture retardation.

rm: matrix retardation.

8. READ (10,*)tt1

tt1: simulation time.

9. READ (10,*)(ds1(i),i=1,ns1)

ds1(i): an array containing distances.

10. read (10,*) ns1,ngauss,xamin,dxam,xamax,errr

ns1: number of times.

ngauss: number of Gaussian points for use when iopt=1.

xamin: initial integration interval.

dxam: integration interval multiplier.

xamax: maximum integration interval.

errr: relative integration error.

11. READ(10,*) tau01,bs1,bb1,theta1,dm01,rf,rm

tau01: Plug flow time (days).

bs1: Half aperture (m).

bb1: Half fracture spacing (m).

theta1: matrix porosity.

dm01: effective diffusion coefficient in matrix (m^2/day).

rf: retardation factor in fractures.

rm: retardation factor in matrix.

12. do i=1, ns1

read (10,*) t4(i), c4(i)

end do

t4(i): calculation times for a breakthrough curve (days).

c4(i): relative concentration values generated by other code, such as FEHM for comparison purposes.

Contents of output files for iopt =0

The output files are related to type curve calculations.

1. main output file *.log

   This file contains input echo and concentration along distance for iopt=1 or 2.

2. restart file *.rst

This file is a binary file containing input and calculation results when last simulation is stopped. The *.rst file is used when the calculation of the type curves is continued from last simulation.

3. plot file *.out

The information in this output file is used in FEHM for the stream-line particle-tracking scheme. The following is the content of the output.

ns1: no. of σ_l

s1(i)(i=1, ns1): list of σ_l.

nw1: no. of ω_l.

w1(i)(i=1, nw1): list of ω_l

nt, s1,w1: no. of t1 (fixed no. of 34), σ_l , ω_l. This line and following 34 pairs of time and concentration will be continued for ns1*nw1 times (61x61 curves used in FEHM).

t1(1),c(1)

t1(2),c(2)

t1(3),c(3)

t1(4),c(4)

t1(5),c(5)

... ...

Contents of output files for iopt =1 or 2

Only one output file *.log will be generated. The contents of the output file are self-explanatory.

Contents of output files for iopt =3

Only one output file *.log will be generated. The contents of the output file are self-explanatory.

Attachment III. Examples

III.1. Example 1: Concentration along fracture using Trapezoidal Rule Integration Scheme

Input file: ex1.dat

```
1 0
24 32 1.0e-7 1.01 1.0 1.e-8
0.1 5.e-5 0.05 0.01 0.1 1.6e-5 12.35 1.0 1.0
1000
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
16 17 18 19 20 21 22 23 24
```

Output file: ex1.log

```
Concentration along x axis using Trapezoidal rule integration scheme
    Number of distances=    24
    Number of Gaussian points =    32
    Initial integration interval =    0.100E-06
    Integration interval multiplier =    0.101E+01
    Maximum integration interval =    0.100E+01
    Relative integration error =    0.100E-07

    Velocity (m/d) =    0.10000
    Simulation time (d) =    0.100E+04
    Half aperture (m) =    0.00005
    Half fracture spacing (m)=    0.05000
    Matrix porosity =    0.01000
    Tortuosity =    0.10000
    Diffusion coef.(cm2/s) =    0.160E-04
    Half-life (years) =    0.124E+02
    Fracture retardation =    1.00000
    Matrix retardation =    1.00000

    Distances (m) =
     1.000     2.000     3.000     4.000     5.000     6.000     7.000
8.000     9.000    10.000
    11.000    12.000    13.000    14.000    15.000    16.000    17.000
18.000    19.000    20.000
    21.000    22.000    23.000    24.000

    0.1000000000E+01    0.9833281959E+00
    0.2000000000E+01    0.9662696785E+00
    0.3000000000E+01    0.9464776318E+00
    0.4000000000E+01    0.9188760978E+00
    0.5000000000E+01    0.8763047914E+00
```

```
0.6000000000E+01    0.8122007726E+00
0.7000000000E+01    0.7240732814E+00
0.8000000000E+01    0.6156114648E+00
0.9000000000E+01    0.4960851937E+00
0.1000000000E+02    0.3774117452E+00
0.1100000000E+02    0.2704517361E+00
0.1200000000E+02    0.1823527217E+00
0.1300000000E+02    0.1156502357E+00
0.1400000000E+02    0.6899874056E-01
0.1500000000E+02    0.3873780088E-01
0.1600000000E+02    0.2047380895E-01
0.1700000000E+02    0.1019016710E-01
0.1800000000E+02    0.4777401616E-02
0.1900000000E+02    0.2109978003E-02
0.2000000000E+02    0.8778382967E-03
0.2100000000E+02    0.3439496358E-03
0.2200000000E+02    0.1268605480E-03
0.2300000000E+02    0.4401732716E-04
0.2400000000E+02    0.1435507319E-04
```

III.2. Example 2: Concentration along Fracture using Gaussian Quadrature Integration Scheme

Input file: ex2.dat

```
2 0
24 256 1.0e-7 1.01 1.0 1.e-8
0.1 5.e-5 0.05 0.01 0.1 1.6e-5 12.35 1.0 1.0 ;vel,bs1,bb1,theta1,tort,dm,lamda,rf,rm
1000
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
16 17 18 19 20 21 22 23  24
```

Output file: ex2.log

```
Concentration along x axis using Gaussian quadrature integration scheme
    Number of distances=    24
    Number of Gaussian points =   256
    Initial integration interval =    0.100E-06
    Integration interval multiplier =    0.101E+01
    Maximum integration interval =    0.100E+01
    Relative integration error =    0.100E-07

    Velocity (m/d) =    0.10000
    Simulation time (d) =    0.100E+04
    Half aperture (m) =    0.00005
    Half fracture spacing (m)=    0.05000
    Matrix porosity =    0.01000
    Tortuosity =    0.10000
    Diffusion coef.(cm2/s) =    0.160E-04
    Half-life (years) =    0.124E+02
    Fracture retardation =    1.00000
    Matrix retardation =    1.00000
```

```
    Distances (m) =
     1.000     2.000     3.000     4.000     5.000     6.000     7.000
8.000     9.000    10.000
    11.000    12.000    13.000    14.000    15.000    16.000    17.000
18.000    19.000    20.000
    21.000    22.000    23.000    24.000

    0.1000000000E+01    0.9833280285E+00
    0.2000000000E+01    0.9662695336E+00
    0.3000000000E+01    0.9464774901E+00
    0.4000000000E+01    0.9188759544E+00
    0.5000000000E+01    0.8763046588E+00
    0.6000000000E+01    0.8122006618E+00
    0.7000000000E+01    0.7240731901E+00
    0.8000000000E+01    0.6156113849E+00
    0.9000000000E+01    0.4960851190E+00
    0.1000000000E+02    0.3774116776E+00
    0.1100000000E+02    0.2704516821E+00
    0.1200000000E+02    0.1823526803E+00
    0.1300000000E+02    0.1156502122E+00
    0.1400000000E+02    0.6899873496E-01
    0.1500000000E+02    0.3873780031E-01
    0.1600000000E+02    0.2047381310E-01
    0.1700000000E+02    0.1019017441E-01
    0.1800000000E+02    0.4777412604E-02
    0.1900000000E+02    0.2109982746E-02
    0.2000000000E+02    0.8778420843E-03
    0.2100000000E+02    0.3439508017E-03
    0.2200000000E+02    0.1268614075E-03
    0.2300000000E+02    0.4401622716E-04
    0.2400000000E+02    0.1435375495E-04
```

Attachment IV. Input and Output Files for Type Curve Calculations

A listing of the file type1.dat is not given because type1.dat is the same as type.dat except for "irestart" is equal to 1.

(a) search for relative time by specified concentration values

List of Input file: type.dat

```
0 0
64 1.0e-9 1.01 1.0e10 1.e-40
1.e-17 1.e17 10
1.0e-1 1.0e9 6
1.0e-4 1.0e6 6
```

Partial list of Plot file: type.out

```
61
   0.100E+00   0.147E+00   0.215E+00   0.316E+00   0.464E+00
0.681E+00   0.100E+01   0.147E+01   0.215E+01   0.316E+01
   0.464E+01   0.681E+01   0.100E+02   0.147E+02   0.215E+02
0.316E+02   0.464E+02   0.681E+02   0.100E+03   0.147E+03
   0.215E+03   0.316E+03   0.464E+03   0.681E+03   0.100E+04
0.147E+04   0.215E+04   0.316E+04   0.464E+04   0.681E+04
   0.100E+05   0.147E+05   0.215E+05   0.316E+05   0.464E+05
0.681E+05   0.100E+06   0.147E+06   0.215E+06   0.316E+06
   0.464E+06   0.681E+06   0.100E+07   0.147E+07   0.215E+07
0.316E+07   0.464E+07   0.681E+07   0.100E+08   0.147E+08
   0.215E+08   0.316E+08   0.464E+08   0.681E+08   0.100E+09
0.147E+09   0.215E+09   0.316E+09   0.464E+09   0.681E+09
   0.100E+10
   61
   0.100E-03   0.147E-03   0.215E-03   0.316E-03   0.464E-03   0.681E-
03   0.100E-02   0.147E-02   0.215E-02   0.316E-02
   0.464E-02   0.681E-02   0.100E-01   0.147E-01   0.215E-01   0.316E-
01   0.464E-01   0.681E-01   0.100E+00   0.147E+00
   0.215E+00   0.316E+00   0.464E+00   0.681E+00   0.100E+01
0.147E+01   0.215E+01   0.316E+01   0.464E+01   0.681E+01
   0.100E+02   0.147E+02   0.215E+02   0.316E+02   0.464E+02
0.681E+02   0.100E+03   0.147E+03   0.215E+03   0.316E+03
   0.464E+03   0.681E+03   0.100E+04   0.147E+04   0.215E+04
0.316E+04   0.464E+04   0.681E+04   0.100E+05   0.147E+05
   0.215E+05   0.316E+05   0.464E+05   0.681E+05   0.100E+06
0.147E+06   0.215E+06   0.316E+06   0.464E+06   0.681E+06
   0.100E+07
   34    1.0000000000000D-01    1.0000000000000D-04
    0.2028721937E-09    0.6888548939E-06
    0.2182238575E-09    0.1695675839E-05
    0.2989153252E-09    0.4315903690E-04
    0,3751600911E-09    0.2615219477E-03
    0.5746557241E-09    0.3180596539E-02
    0.1584893192E-08    0.7570458355E-01
```

```
 0.2511886432E-08    0.1582848627E+00
 0.3162277660E-08    0.2085967307E+00
 0.3981071706E-08    0.2624203840E+00
 0.5011872336E-08    0.3178843785E+00
 0.6309573445E-08    0.3733613528E+00
 0.7943282347E-08    0.4275524659E+00
 0.1000000000E-07    0.4795001227E+00
 0.1584893192E-07    0.5743375521E+00
 0.1995262315E-07    0.6166574042E+00
 0.3162277660E-07    0.6908989893E+00
 0.5011872336E-07    0.7521140906E+00
 0.6309573445E-07    0.7783236467E+00
 0.7943282347E-07    0.8018984197E+00
 0.1000000000E-06    0.8230632747E+00
 0.1584893192E-06    0.8590236222E+00
 0.1995262315E-06    0.8742192353E+00
 0.3162277660E-06    0.8999350937E+00
 0.3981071706E-06    0.9107688136E+00
 0.7943282347E-06    0.9367635096E+00
 0.1258925412E-05    0.9497499103E+00
 0.1584893192E-05    0.9552111718E+00
 0.3162277660E-05    0.9682849865E+00
 0.6309573445E-05    0.9775730160E+00
 0.3162277660E-04    0.9896652188E+00
 0.6407730252E-04    0.9904053904E+00
 0.1584893192E-03    0.9961397955E+00
 0.3598460558E-03    0.9927981059E+00
 0.4072089606E-03    0.9992699616E+00
34    1.0000000000000D-01    1.4677992676221D-04
 0.4343931860E-09    0.6365655589E-06
 0.5395411974E-09    0.7885490654E-05
 0.6292292536E-09    0.3509762572E-04
 0.7366396030E-09    0.1312745776E-03
 0.1584893192E-08    0.9132232775E-02
 0.3981071706E-08    0.9998092949E-01
 0.5011872336E-08    0.1426320116E+00
 0.6309573445E-08    0.1913393808E+00
 0.7943282347E-08    0.2442082206E+00
 0.1000000000E-07    0.2993208903E+00
 0.1258925412E-07    0.3549548996E+00
 0.1584893192E-07    0.4096975588E+00
 0.2511886432E-07    0.5125547846E+00
 0.3162277660E-07    0.5594565564E+00
 0.3981071706E-07    0.6029397784E+00
 0.6309573445E-07    0.6794661139E+00
 0.1000000000E-06    0.7427529684E+00
 0.1584893192E-06    0.7943189422E+00
 0.1995262315E-06    0.8162624451E+00
 0.2511886432E-06    0.8359423504E+00
 0.3162277660E-06    0.8535694672E+00
 0.3981071706E-06    0.8693417915E+00
 0.6309573445E-06    0.8960421301E+00
 0.1000000000E-05    0.9173367579E+00
 0.1584893192E-05    0.9342945847E+00
 0.2511886432E-05    0.9477866963E+00
 0.3981071706E-05    0.9585273059E+00
```

```
0.6309573445E-05    0.9670388211E+00
0.1584893192E-04    0.9790364429E+00
0.5121055717E-04    0.9891172636E+00
0.1425195726E-03    0.9917776623E+00
0.5011872336E-03    0.9966647824E+00
0.5685854533E-03    0.9995452049E+00
0.5723924338E-03    0.1000000000E+01
```

(b) Calculate relative concentration values using determined relative time values from type.out

List of Input file: type2.dat

```
0 3
1024 1.0e-9 1.01 1.0e1 1.e-10
1.e-17 1.e17 10
1.0e-1 1.0e9 6
1.0e-4 1.0e6 6
```

Partial list of Plot file: type2.out

```
61
   0.100E+00   0.147E+00   0.215E+00   0.316E+00   0.464E+00
0.681E+00   0.100E+01   0.147E+01   0.215E+01   0.316E+01
   0.464E+01   0.681E+01   0.100E+02   0.147E+02   0.215E+02
0.316E+02   0.464E+02   0.681E+02   0.100E+03   0.147E+03
   0.215E+03   0.316E+03   0.464E+03   0.681E+03   0.100E+04
0.147E+04   0.215E+04   0.316E+04   0.464E+04   0.681E+04
   0.100E+05   0.147E+05   0.215E+05   0.316E+05   0.464E+05
0.681E+05   0.100E+06   0.147E+06   0.215E+06   0.316E+06
   0.464E+06   0.681E+06   0.100E+07   0.147E+07   0.215E+07
0.316E+07   0.464E+07   0.681E+07   0.100E+08   0.147E+08
   0.215E+08   0.316E+08   0.464E+08   0.681E+08   0.100E+09
0.147E+09   0.215E+09   0.316E+09   0.464E+09   0.681E+09
   0.100E+10
   61
   0.100E-03   0.147E-03   0.215E-03   0.316E-03   0.464E-03   0.681E-
03   0.100E-02   0.147E-02   0.215E-02   0.316E-02
   0.464E-02   0.681E-02   0.100E-01   0.147E-01   0.215E-01   0.316E-
01   0.464E-01   0.681E-01   0.100E+00   0.147E+00
   0.215E+00   0.316E+00   0.464E+00   0.681E+00   0.100E+01
0.147E+01   0.215E+01   0.316E+01   0.464E+01   0.681E+01
   0.100E+02   0.147E+02   0.215E+02   0.316E+02   0.464E+02
0.681E+02   0.100E+03   0.147E+03   0.215E+03   0.316E+03
   0.464E+03   0.681E+03   0.100E+04   0.147E+04   0.215E+04
0.316E+04   0.464E+04   0.681E+04   0.100E+05   0.147E+05
   0.215E+05   0.316E+05   0.464E+05   0.681E+05   0.100E+06
0.147E+06   0.215E+06   0.316E+06   0.464E+06   0.681E+06
   0.100E+07
   34    1.0000000000000D-01    1.0000000000000D-04
    0.2028721937E-09    0.6888548937E-06
    0.2182238575E-09    0.1695675840E-05
    0.2989153252E-09    0.4315903695E-04
    0.3751600911E-09    0.2615219478E-03
    0.5746557241E-09    0.3180596540E-02
```

```
 0.1584893192E-08     0.7570458351E-01
 0.2511886432E-08     0.1582848627E+00
 0.3162277660E-08     0.2085967307E+00
 0.3981071706E-08     0.2624203840E+00
 0.5011872336E-08     0.3178843785E+00
 0.6309573445E-08     0.3733613528E+00
 0.7943282347E-08     0.4275524659E+00
 0.1000000000E-07     0.4795001227E+00
 0.1584893192E-07     0.5743375520E+00
 0.1995262315E-07     0.6166574042E+00
 0.3162277660E-07     0.6908989892E+00
 0.5011872336E-07     0.7521140905E+00
 0.6309573445E-07     0.7783236466E+00
 0.7943282347E-07     0.8018984196E+00
 0.1000000000E-06     0.8230632746E+00
 0.1584893192E-06     0.8590236221E+00
 0.1995262315E-06     0.8742192351E+00
 0.3162277660E-06     0.8999350923E+00
 0.3981071706E-06     0.9107687991E+00
 0.7943282347E-06     0.9367632375E+00
 0.1258925412E-05     0.9497498192E+00
 0.1584893192E-05     0.9552083849E+00
 0.3162277660E-05     0.9682816509E+00
 0.6309573445E-05     0.9775421782E+00
 0.3162277660E-04     0.9899673386E+00
 0.6407730252E-04     0.9929514745E+00
 0.1584893192E-03     0.9955195651E+00
 0.3598460558E-03     0.9970755674E+00
 0.4072089606E-03     0.9972589945E+00
34    1.0000000000000D-01    1.4677992676221D-04
 0.4343931860E-09     0.6365655579E-06
 0.5395411974E-09     0.7885490650E-05
 0.6292292536E-09     0.3509762572E-04
 0.7366396030E-09     0.1312745776E-03
 0.1584893192E-08     0.9132232765E-02
 0.3981071706E-08     0.9998092951E-01
 0.5011872336E-08     0.1426320115E+00
 0.6309573445E-08     0.1913393808E+00
 0.7943282347E-08     0.2442082206E+00
 0.1000000000E-07     0.2993208903E+00
 0.1258925412E-07     0.3549548997E+00
 0.1584893192E-07     0.4096975588E+00
 0.2511886432E-07     0.5125547847E+00
 0.3162277660E-07     0.5594565564E+00
 0.3981071706E-07     0.6029397783E+00
 0.6309573445E-07     0.6794661138E+00
 0.1000000000E-06     0.7427529683E+00
 0.1584893192E-06     0.7943189420E+00
 0.1995262315E-06     0.8162624450E+00
 0.2511886432E-06     0.8359423503E+00
 0.3162277660E-06     0.8535694671E+00
 0.3981071706E-06     0.8693417914E+00
 0.6309573445E-06     0.8960421316E+00
 0.1000000000E-05     0.9173367352E+00
 0.1584893192E-05     0.9342947641E+00
 0.2511886432E-05     0.9477866720E+00
```

0.3981071706E-05	0.9585145440E+00
0.6309573445E-05	0.9670414486E+00
0.1584893192E-04	0.9792010000E+00
0.5121055717E-04	0.9884282257E+00
0.1425195726E-03	0.9930627385E+00
0.5011872336E-03	0.9963185137E+00
0.5685854533E-03	0.9965552597E+00
0.5723924338E-03	0.9965672779E+00

Attachment V. Complete file information

Table V-1 lists the software routine source code, MS DOS executable, and input/output files for this calculation. Intermediate results are not listed. These files have been submitted to the Technical Data Management System and Records Processing Center under DTN: SN0004T0501600.006.

Table V-1. File information in this calculation

Name	Date	Time	Size in bytes	Function
Fract_p.for	10/25/99	3:34p	32,309	Source code for the software routine
Fract_p exe	11/09/99	9:28a	123,392	Executable for the software routine
Ex1.dat	4/15/99	10:38a	192	Input file for concentration calculation along a fracture with Trapezoidal rule.
ex1 log	8/18/99	2:24p	1,992	Output file for concentration calculation along a fracture with Trapezoidal rule.
ex2 dat	4/15/99	10.38a	193	Input file for concentration calculation along a fracture with Gaussian quadrature.
ex2.log	8/18/99	2:25p	1,995	Output file for concentration calculation along a fracture with Gaussian quadrature.
ex3.dat	8/17/99	3:04p	667	Input file for breakthrough curves at a location along a fracture.
ex3.log	8/17/99	3.05p	4,948	Output file for breakthrough curves at a location along a fracture.
type dat	8/17/99	11:16a	86	Input file for type curve calculation from the start.
type1.dat	8/17/99	11 15a	86	Input file for type curve calculation continued from previous calculation
type2.dat	8/17/99	11:16a	87	Input file for type curve calculation after the relative times are determined.
type2.zip	10/06/99	10.01a	1,021,494	Zipped output file for type curves used in FEHM includes type2.out

Attachment VI. DIRS

OFFICE OF CIVILIAN RADIOACTIVE WASTE MANAGEMENT
DOCUMENT INPUT REFERENCE SHEET

1. Document Identifier No./Rev.:	Change:	Title:
B00000000-01717-0210-00089 Rev. 00	N/A	TYPE CURVE CALCULATIONS FOR MASS TRANSPORT IN PARALLEL FRACTURES USED IN PARTICLE-TRACKING SCHEME IN THE SATURATED ZONE

Input Document			4. Input Status	5. Section Used in	6. Input Description	7. TBV/ TBD Priority	8. TBVDue To		
2a.	2. Technical Product Input Source Title and Identifier(s) with Version	3. Section					Unqual.	From Uncontrolled Source	Un-Confirmed
1	CRWMS M&O 1999. *Abstraction of Matrix Diffusion for SZ Flow and Transport Analyses, Rev. 01. ID: B1080, Activity: SPP3025.* Work Direction and Planning Document. Las Vegas, Nevada. CRWMS M&O. ACC: MOL.19990707 0107.	Entire	N/A - Reference Only	1	Work direction and planning document.	N/A	N/A	N/A	N/A
2	CRWMS M&O 2000. *Input Transmittal for "Uncertainty Distribution for Stochastic Parameters" (ANL-NBS-MD-000011) and "Input and Results of the Base Case Saturated Zone Flow and Transport Model for TSPA" (ANL-NBS-HS-000030).* Input Transmittal 00204.T. Las Vegas, Nevada: CRWMS M&O. ACC: MOL.20000404.0198.	ANL-NBS-MD-000011	N/A – Technical Product Output	3.1,3.2,5.0	Draft AMR for uncertainty distributions for stochastic parameters	N/A	N/A	N/A	N/A
		ANL-NBS-HS-000030	N/A - Technical Product Output	3.2,5.0	Draft AMR on input and results of base-case saturated-zone flow and transport model for TSPA	N/A	N/A	N/A	N/A
3	Sudicky, E.A. and Frind, E.O. 1982 "Contaminant Transport in Fractured Porous Media: Analytical Solutions for a System of Parallel Fractures." *Water Resources Research, 18,* (6), 1634-1642.	Entire	N/A - Reference Only	2.0, 2.1, 3.1	Analytical solution for transport in parallel fractures.	N/A	N/A	N/A	N/A

OFFICE OF CIVILIAN RADIOACTIVE WASTE MANAGEMENT
DOCUMENT INPUT REFERENCE SHEET

1. Document Identifier No./Rev.:	Change	Title:
B00000000-01717-0210-00089 Rev. 00	N/A	TYPE CURVE CALCULATIONS FOR MASS TRANSPORT IN PARALLEL FRACTURES USED IN PARTICLE-TRACKING SCHEME IN THE SATURATED ZONE

Input Document			4. Input Status	5. Section Used in	6. Input Description	7. TBV/TBD Priority	8. TBVDue To		
2a.	2. Technical Product Input Source Title and Identifier(s) with Version	3. Section					Unqual	From Uncontrolled Source	Un-Confirmed
	Washington, D C.. American Geophysical Union. TIC: 217475.								
4	Tang, D.H.; Frind, E.O.; and Sudicky, E.A. 1981. "Contaminant Transport in Fractured Porous Media: Analytical Solution for a Single Fracture." *Water Resources Research, 17*, (3), 555-564. Washington, D C.: American Geophysical Union. TIC: 225358.	Entire	N/A - Reference Only	4.2.2	Analytical solution for transport in a single fracture.	N/A	N/A	N/A	N/A