REPORT ON

ANALYSIS OF TWO-WELL TRACER TESTS WITH A PULSE INPUT

PREPARED BY

LYNN W. GELHAR

FOR

ROCKWELL INTERNATIONAL CORPORATION
ROCKWELL HANFORD OPERATIONS
ENERGY SYSTEMS GROUP
BASALT WASTE ISOLATION PROJECT

P.O. BOX 800 RICHLAND, WA 99352

APRIL 1982

Summary

Dispersion of a conservative solute which is introduced as a pulse in the recharge well of a two-well flow system is analyzed using the general theory for longitudinal dispersion in nonuniform flow along streamlines. Results for the concentration variation at the pumping well are developed using numerical integration and are presented in the form of dimensionless type-curves which can be used to design and analyze tracer tests.

Application of the results is illustrated by analyzing the preliminary tracer test run at boreholes DC7/8 on the Hanford site by Science Applications, Inc., a subcontractor to Rockwell Hanford Operations, in December 1979.

1. GENERAL THEORY

The objective of this analysis is to describe the tracer concentration which evolves in the pumping well of a two-well (pumping-recharge) flow system (see Figure 1) when an instantaneous pulse (slug) of conservative tracer is introduced in the recharge well. The streamline pattern for steady flow in a homogeneous confined aquifer is used in conjunction with the general theoretical results of Gelhar and Collins (1971) for longitudinal dispersion in nonuniform flows. With a pulse input their general result for the concentration (Equation 25) is

$$c(s,t) = \frac{m}{u(s)\sqrt{4\pi\alpha\omega}} \exp[-\eta^2/4\alpha\omega] \qquad (1)$$

s = distance along streamline

t = time

α = longitudinal dispersivity

$\eta = \tau(s) - t$

$\tau(s) = \int_{s_0}^{s} ds/u(s)$, travel time to s

$\omega(t) = \int_{s_0}^{\bar{s}} ds/[u(s)]^2$

$\bar{s}(t)$ = mean location of the pulse at time t

u(s) = seepage velocity

m = mass of tracer per net area of aquifer injected at $s = s_0$ at time t = 0

Equation (1) is applied along each streamline identified by the

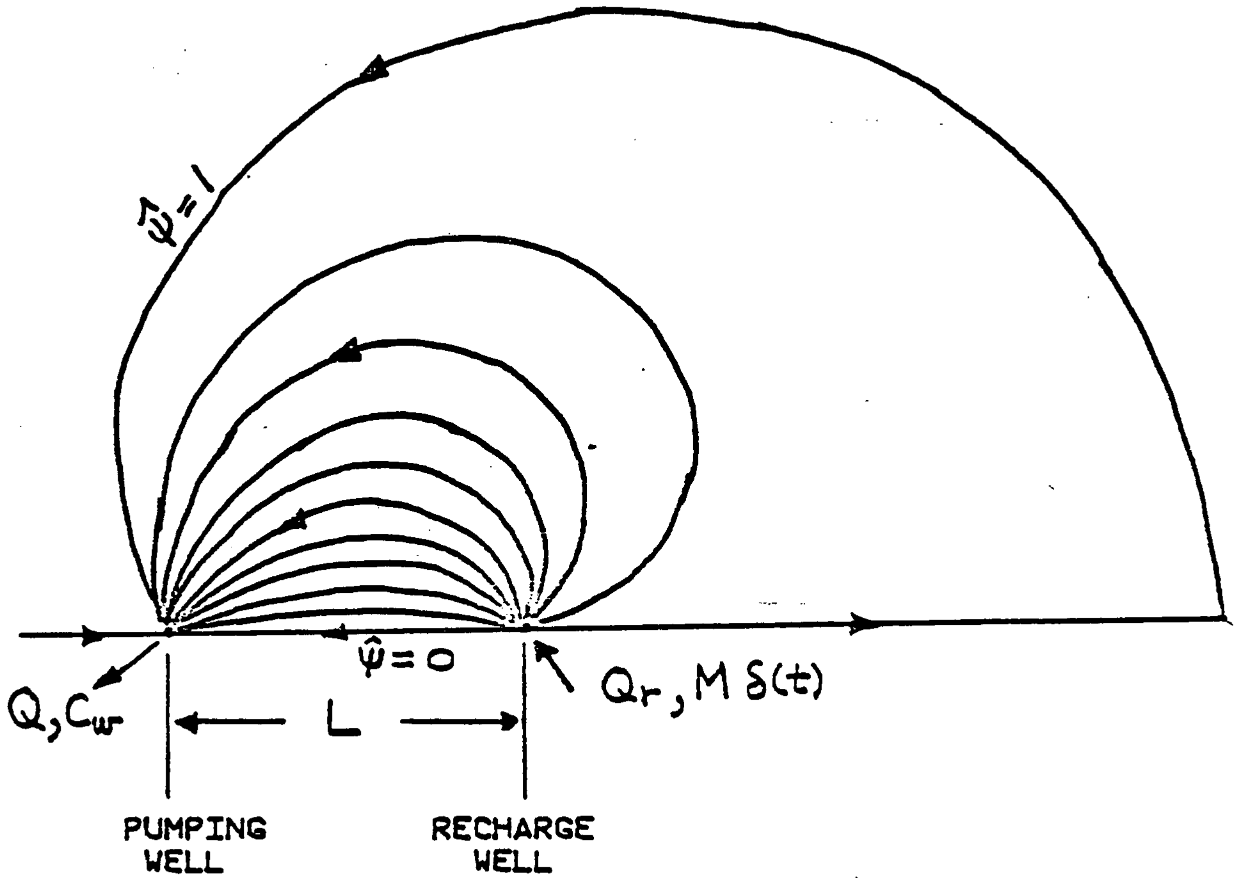


FIGURE 1. STREAMLINE PATTERN FOR TWO-WELL FLOW SYSTEM WITH $Q/Q_r = 2/3$.

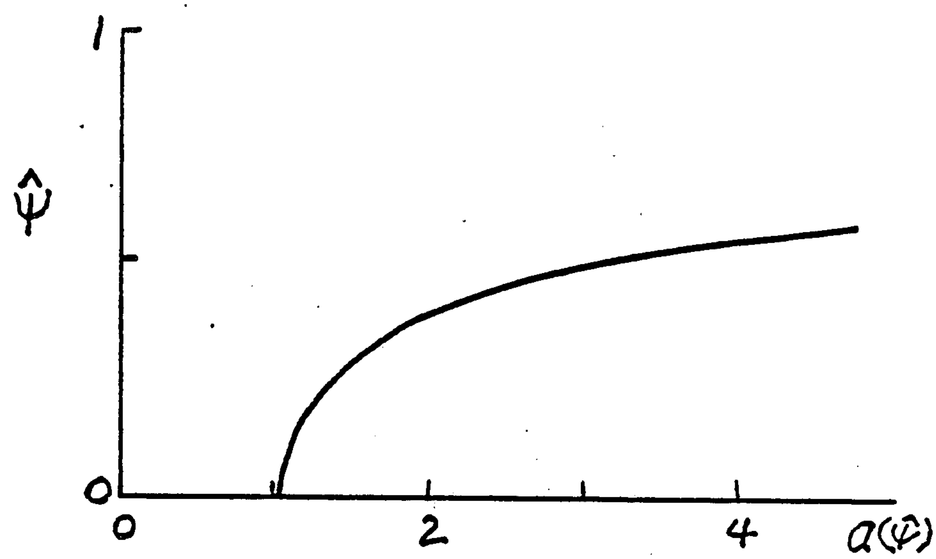


FIGURE 2. DIMENSIONLESS TRAVEL TIME FUNCTION.

value of the stream function ψ; therefore the velocity u depends on ψ and as a result, $\tau(s,\psi)$ and $\omega(t,\psi)$. These flow integrals can be evaluated either analytically or graphically as described in the next section. The coefficient $m/u(s_o)$ in (1) is evaluated by noting that at the recharge well

$$u(s_o) = Q_r/(2\pi r_w n H)$$

Q_r = recharge rate

r_w = well radius

n = effective porosity

H = aquifer thickness

$m = M/2\pi r_w nH$

M = mass of tracer injected

$$m/u(s_o) = M/Q_r$$

The concentration in the pumping well is found by calculating the flow-weighted concentration as the following integral:

$$C_w = \frac{2H}{Q}\int_{\psi=0}^{Q/2H}(M/\beta Q)(4\pi\alpha\omega)^{-1/2}\exp[-(\tau-t)^2/4\alpha\omega]\,d\psi \qquad (2)$$

where $\beta = Q_r/Q$. In this integral the flow integrals τ and ω depend on the stream function, ψ.

2. EVALUATION OF FLOW INTEGRALS AND WELL CONCENTRATION

For the case of equal recharge and discharge rates ($\beta = 1$), the integrals for τ and ω can be evaluated analytically from the velocity variation along a given streamline. The details of this analysis are given in Appendix A; the results for τ and ω are given by (A3) and (A4) expressed in dimensionless form as

$$a(\hat{\psi}) = \frac{Q\tau_w}{nHL^2} \quad , \qquad b(\xi,\hat{\psi}) = \left(\frac{Q}{nH}\right)^2 \frac{\omega}{L^3} \tag{3}$$

where τ_w is the travel time to the pumping well. The well concentration is evaluated using these dimensionless forms and the variable of integration $\hat{\psi} = \psi/(\beta Q/2H)$. Then, for the upper half plane in Figure 1, the flow from the recharge well is represented by the range $0 \leq \hat{\psi} \leq 1$. Furthermore, the concentration will always be zero for the streamlines with $\hat{\psi} > 1$ because lateral dispersion is neglected. Then, using (3) in (2),

$$C_w = \frac{M}{Q}\int_{\hat{\psi}=0}^{1} \frac{Q \exp[-(a-T)^2/4\epsilon b]}{nHL^2\ (4\pi\epsilon b)^{1/2}}\, d\hat{\psi}$$

$$\hat{C} = \frac{nHL^2}{M} C_w = \int_{\hat{\psi}=0}^{1} \frac{\exp[-(a-T)^2/4\epsilon b]}{(4\pi\epsilon b)^{1/2}}\, d\hat{\psi} \tag{4}$$

$$T = \frac{Qt}{nHL^2} \quad , \qquad \epsilon = \frac{\alpha}{L}$$

The integral in (4) was evaluated numerically using a and b from Appendix A; a listing of the computer program is included in Appendix C.

For the case of unequal flow τ and ω were determined graphically from a flow net constructed for the case β = 2/3 as described in Appendix B.

The limiting result for no dispersion in (4) is found by noting that, as $\epsilon \rightarrow 0$,

$$\frac{\exp[-(a-T)^2/4\epsilon b]}{(4\pi\epsilon b)^{1/2}} \rightarrow \delta(a-T)$$

where $\delta(x)$ is the Dirac delta function. Changing the variable of integration to $a(\hat{\Psi})$, (4) becomes

$$\hat{c} = \int_{a(0)}^{\infty} \delta(a-T)\,\frac{d\hat{\Psi}}{da}\,da = \begin{cases} \left.\dfrac{d\hat{\Psi}}{da}\right|_{a=T}, & T > a(0) \\ 0, & T < a(0) \end{cases} \qquad (5)$$

The function $a(\hat{\Psi})$ is simply the dimensionless travel time to the pumping well along a given streamline $\hat{\Psi}$; the general form of this function is shown in Figure 2. The non-zero portion of (5) can be evaluated by taking

$$\frac{d\hat{\Psi}}{da} = \left(\frac{da}{d\hat{\Psi}}\right)^{-1}$$

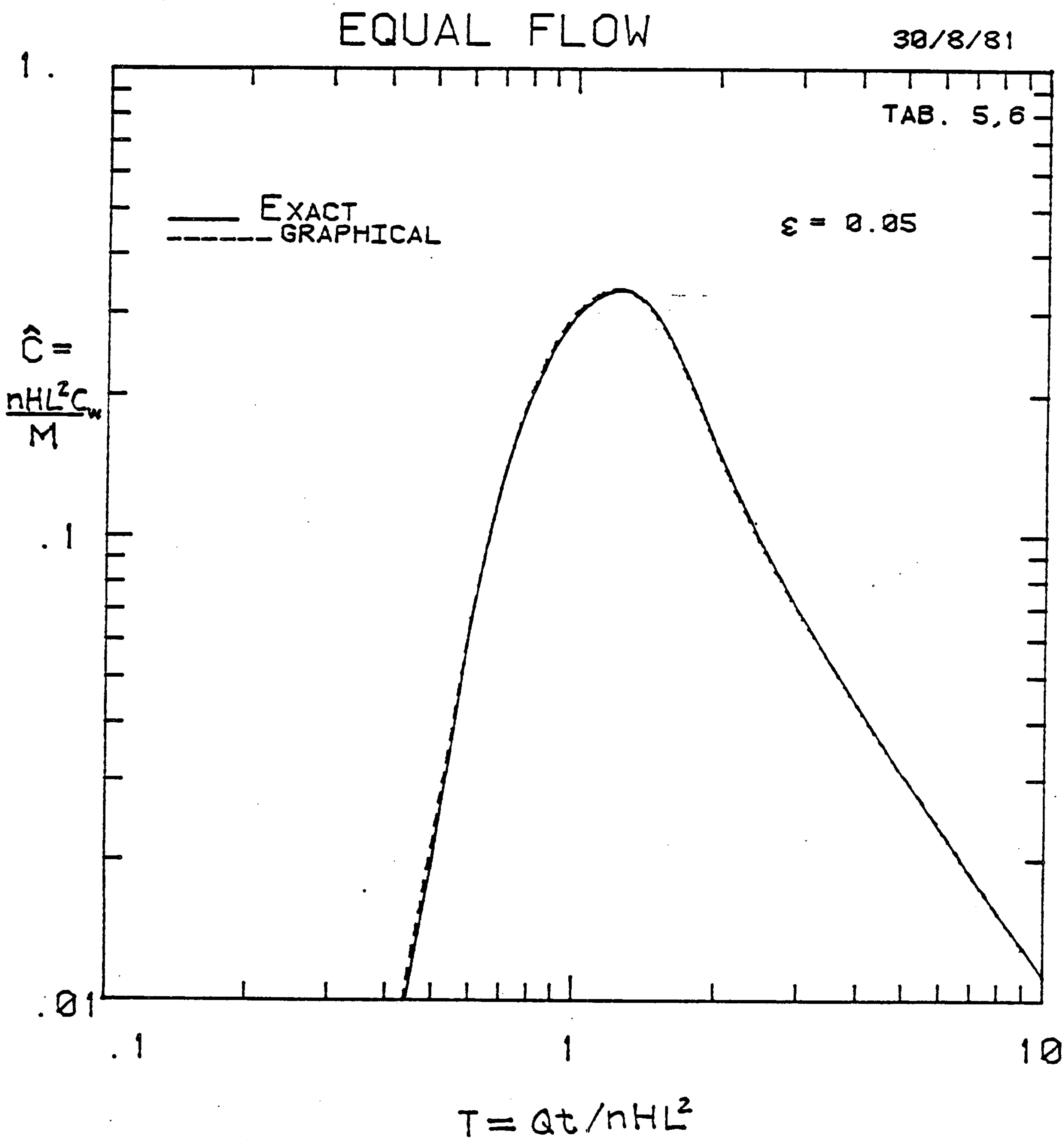
and $\hat{c}(T)$ is then determined by taking $T = a(\hat{\Psi})$ and $da/d\hat{\Psi}|_{\hat{\Psi}_1}$ where $\hat{\Psi}_1$ is an assigned value of $\hat{\Psi}$.

3. RESULTS

Numerical evaluation of the concentration in the pumping well as given by (4) was carried out using the computer program listed in Appendix C. Several runs were made to test the effects of model parameters and approximations. The results are listed in tabular form in Appendix D.

Figure 3 shows a comparison of the graphically based

FIGURE 3. COMPARISON OF EXACT AND GRAPHICAL FLOWNET CALCULATIONS.



results using m = 1 (Appendix B) and the exact result using the equations from Appendix A. The excellent agreement demonstrates the adequacy of the graphical approximation with m = 1 in (B2); that value was used in all subsequent calculations. Figure 4 illustrates the effect of using the approximate form of (B2); some difference is observed at lower concentrations for the rising limb of the curve with the large value of $\epsilon = 0.2$. For smaller ϵ the differences are generally smaller, as shown in Figure 3. The effects the increment $\Delta\hat{\Psi}$ used to approximate the integral in (4) are illustrated in Figure 5. When $\epsilon = 0.01$, the larger $\Delta\hat{\Psi} = 0.05$ produces oscillations in the tail of the curve, but these are eliminated when $\Delta\hat{\Psi} = 0.01$; for $\epsilon = 0.2$ the results are non-oscillatory when $\Delta\hat{\Psi} = 0.05$. Generally the oscillations are eliminated if $\Delta\hat{\Psi} \leq \epsilon$.

The overall results are summarized in Figures 6 and 7; the unequal flow case (Fig. 7 with $\beta = 2/3$) corresponds to the SAI tracer test of December 1979. These results show that the dispersion parameter $\epsilon = \alpha/L$ affects the rising part of the curve and the peak but not the tail. This point is further demonstrated by the behavior the non-dispersive solution using (5) and (A5) as shown in Figure 6. This shows that all of the results approach the non-dispersive analytical result for large time, and further demonstrates the adequacy of the numerical procedure. Generally, the breakthrough curves are characterized by a steep rising limb and an elongated tail as shown in the linear plots of Figure 8. The log-log plots (Figure 6 and 7) are

FIGURE 4. COMPARISON OF EXACT CALCULATION WITH APPROXIMATION USING EQ. B2.

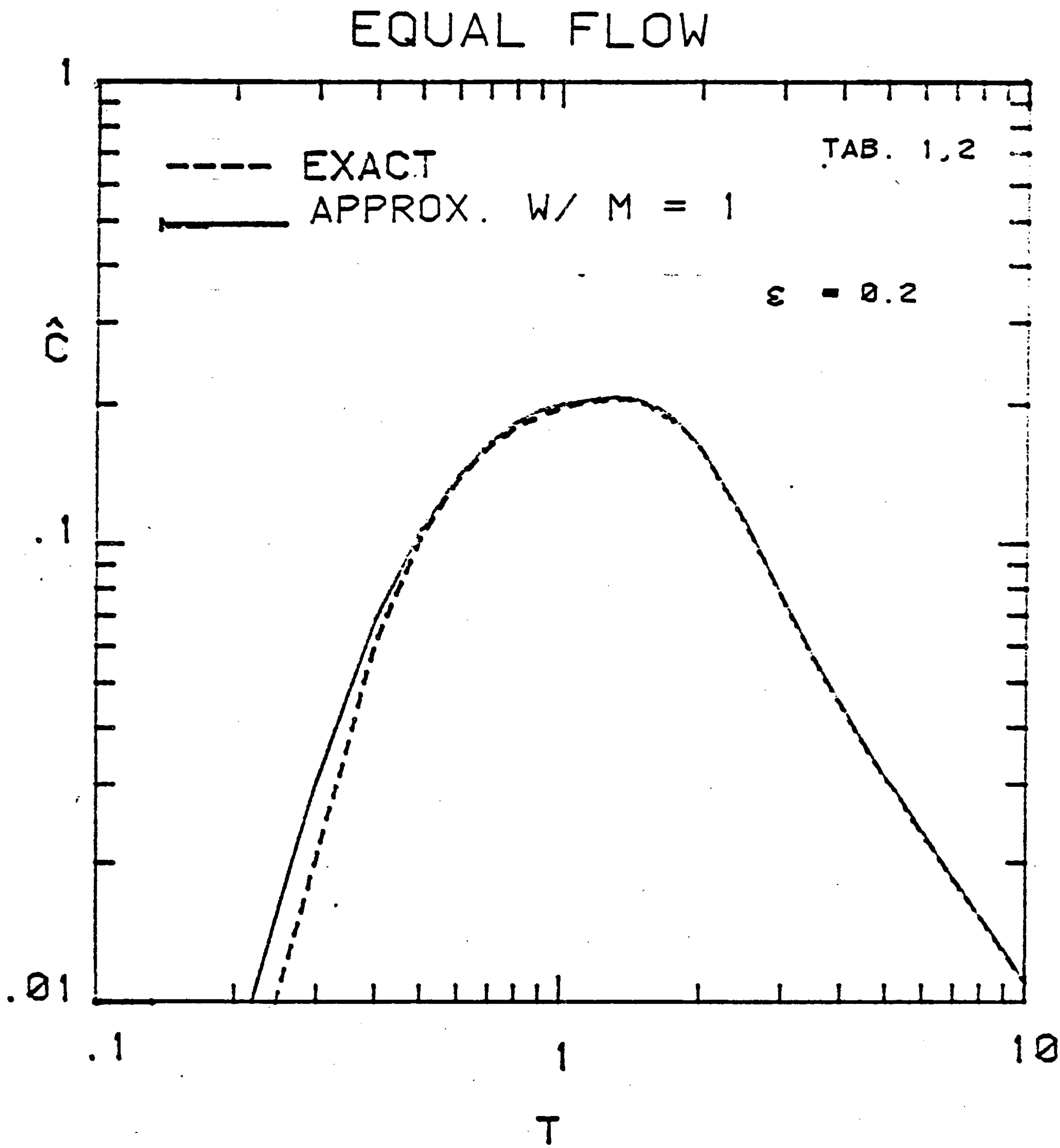


FIGURE 5. EFFECT OF INTEGRATION INCREMENT.

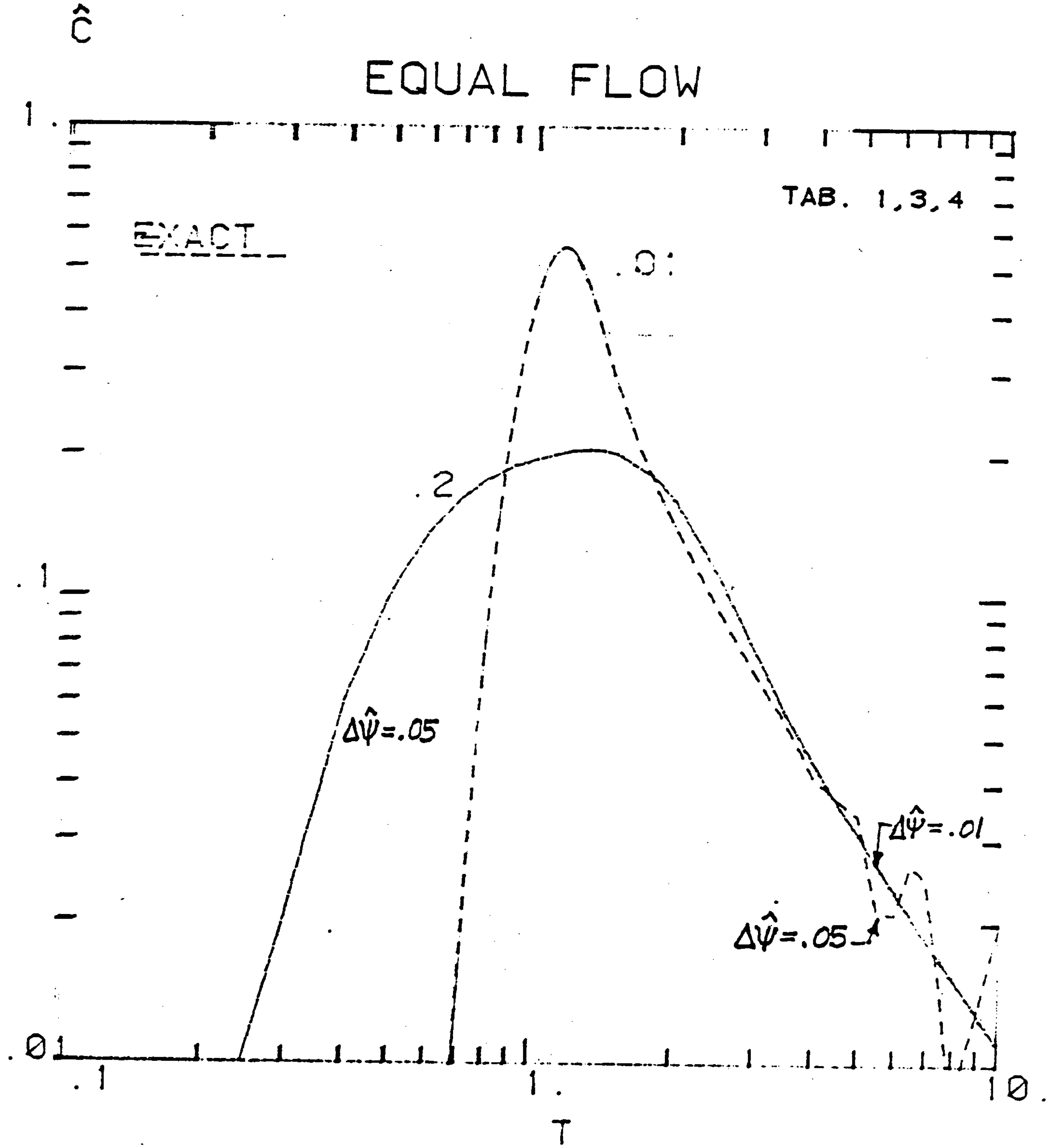
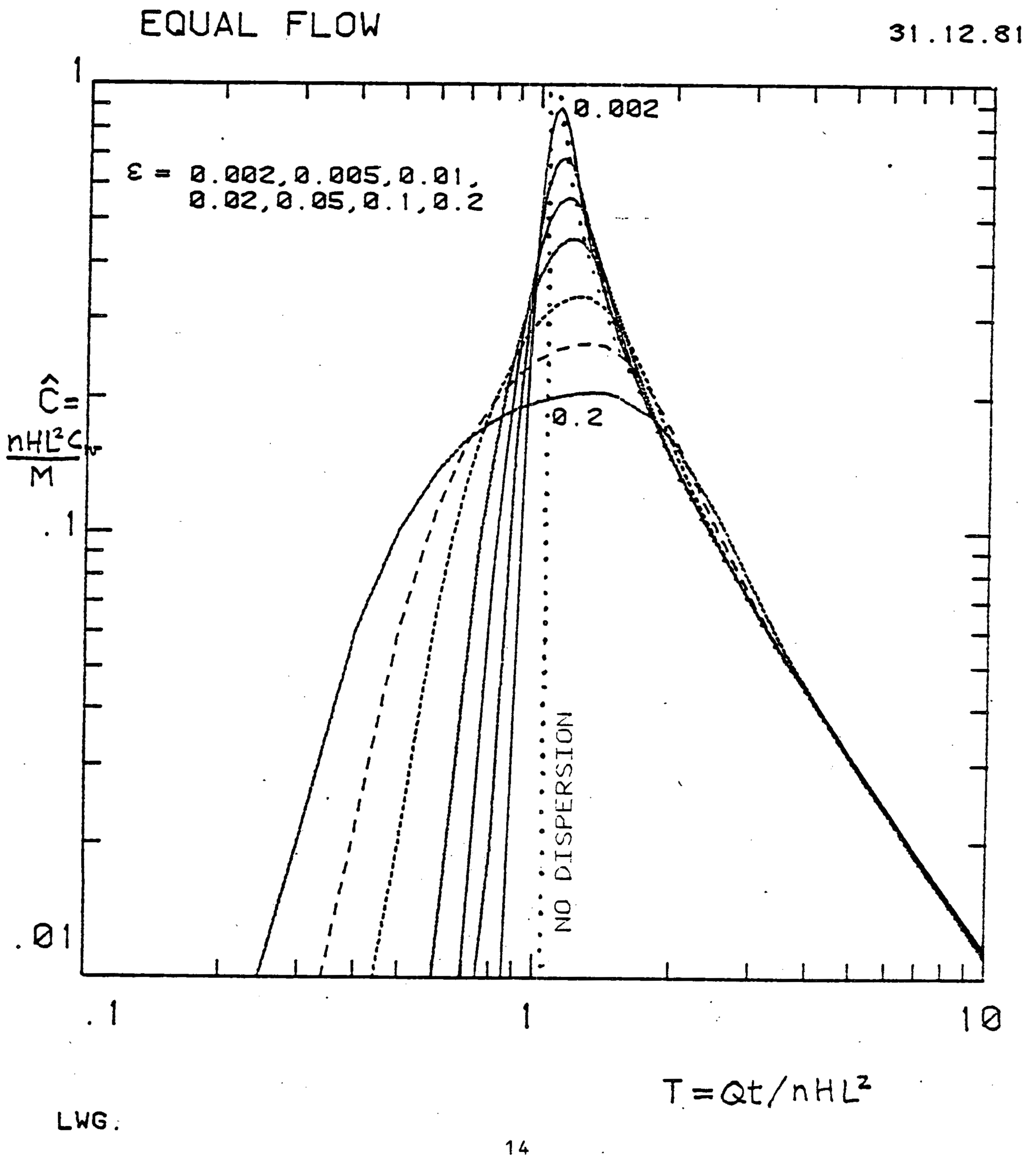


FIGURE 6. TYPE-CURVES FOR TWO-WELL PULSE INPUT TEST WITH EQUAL FLOW.



LWG.

FIGURE 7. TYPE-CURVE FOR TWO-WELL PULSE INPUT WHEN THE RECHARGE RATE IS TWO-THIRDS OF THE DISCHARGE RATE, Q.

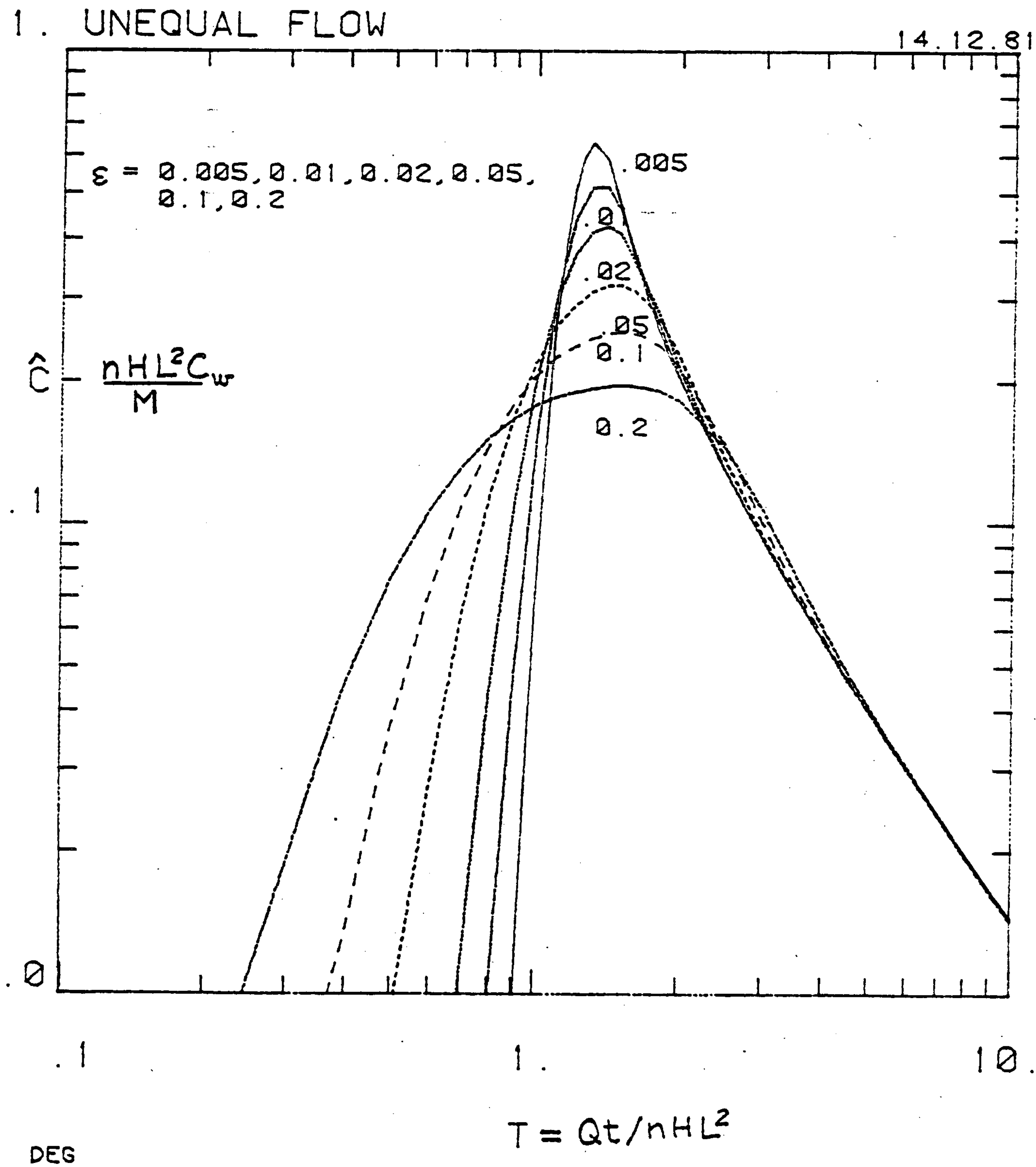
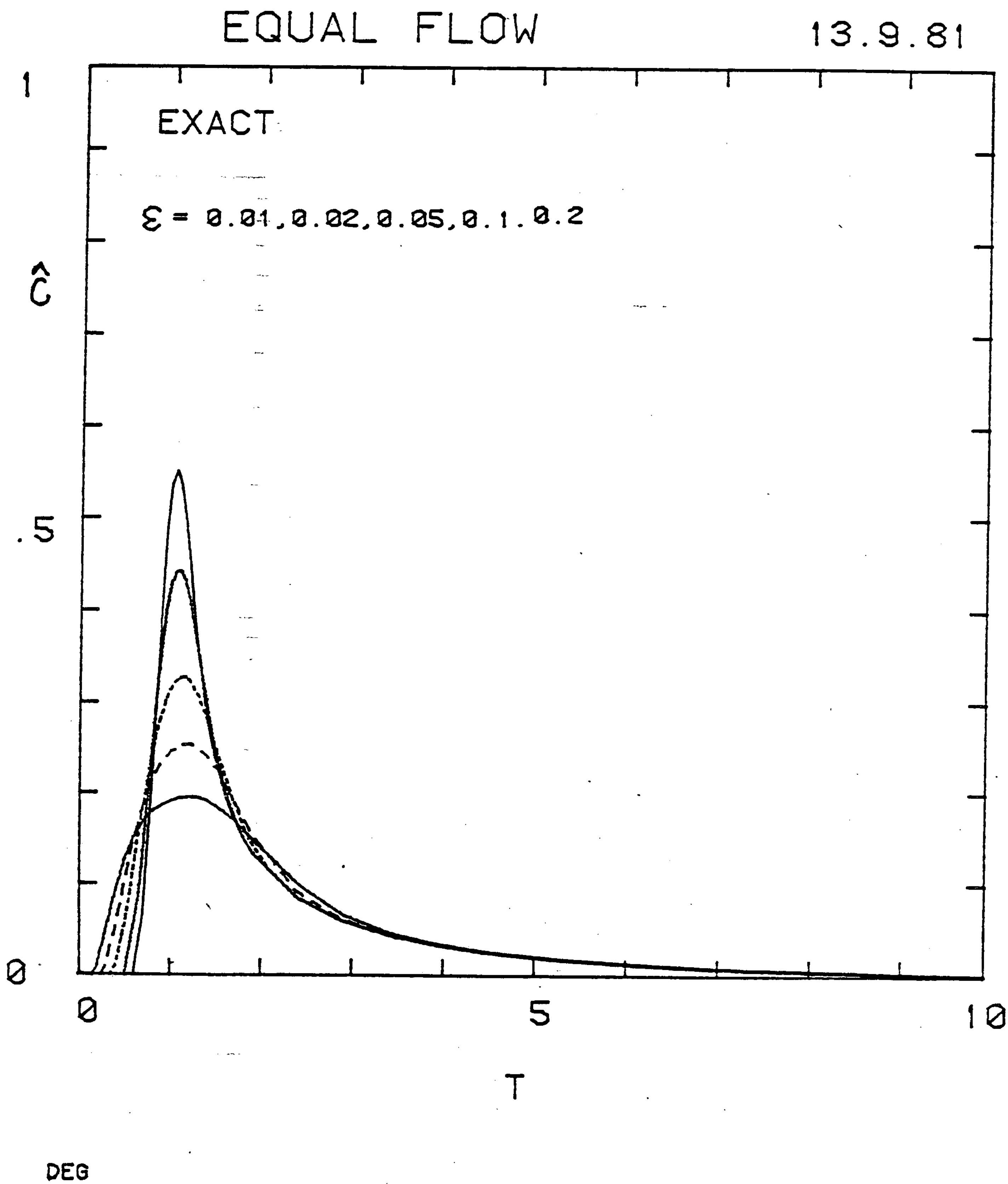


FIGURE 8. PULSE INPUT TYPE-CURVES PLOTTED WITH LINEAR SCALES.



convenient for estimating the dispersivity and effective porosity from tracer test data as illustrated in the following section.

4. APPLICATION METHODOLOGY

The procedure for interpretation of tracer test data using the results of the above analysis is illustrated by analyzing the test conducted by Science Applications, Inc. (SAI) at DC7/8 in December, 1979. Since some of the conditions of that test were not fully defined, the results of this interpretation are considered to be preliminary; this example is present primarily to illustrate the procedures which can be used to analyze such tests.

The SAI test was reanalyzed using the type curves in Figure 7. The flow rates for the test were estimated using the average rates implied for the period 15:13 - 23:00 in Table G4 (p. G-38, SAI draft report); Q_r = 2.31 gpm (injection rate) and Q = 3.42 gpm (pumping rate) or $Q_r/Q = \beta \cong 2/3$. Based on these rates and estimates of the volume in the borehole flow conduits and connecting plumbing the following travel times to and from the test horizon were estimated:

time down in injection well = 153 min.

time up in pumping well = 258 min.

These times were subtracted from the observed times to give the actual elapsed time since the tracer entered the formation. Also the elapsed time was corrected to correspond to a constant pumping rate of 3.42 gpm based of the actual metered volume in

Table G4. After these time corrections were made, the actual data points in Figure 21 (SAI draft report) were plotted with the estimated background of 20 counts subtracted; the corrected data are shown in Figure 9, a log-log plot of the same scale (K&E 46 7323 2x3 cycles) as the dimensionless type-curves in Figure 7 for $\beta = 2/3$. Overlaying the data on the type-curve, we find a reasonable fit for ϵ in the range 0.02 to 0.05; using $\epsilon = 0.035 = \alpha/L$ and L = 56 ft., the indicated longitudinal dispersivity is

$$\alpha = 0.035(56) = 1.96 \text{ ft.}$$

Matching the time scales in Figure 9,

$$T = 1 = Qt/nHL^2, \ t(\text{hours}) = 1.18 \text{ hrs}$$

yields the effective thickness

$$nH = Qt/L^2 = 0.0105 \text{ ft.}$$

The data in Figure 9 seem to show some systematic departures from the theoretical type-curve. This could be a reflection of experimental ambiguities such as the following:

1) The background concentration is n t clearly determined and small changes in this level could drastically alter the low concentration parts of the curve.

2) Unobserved flow rate variations during the period that the tracer was passing the sensor would distort the shape the curve.

3) Uncertainty about the volume in the connecting conduits could introduce errors in the travel time correction and alter the shape of the curves.

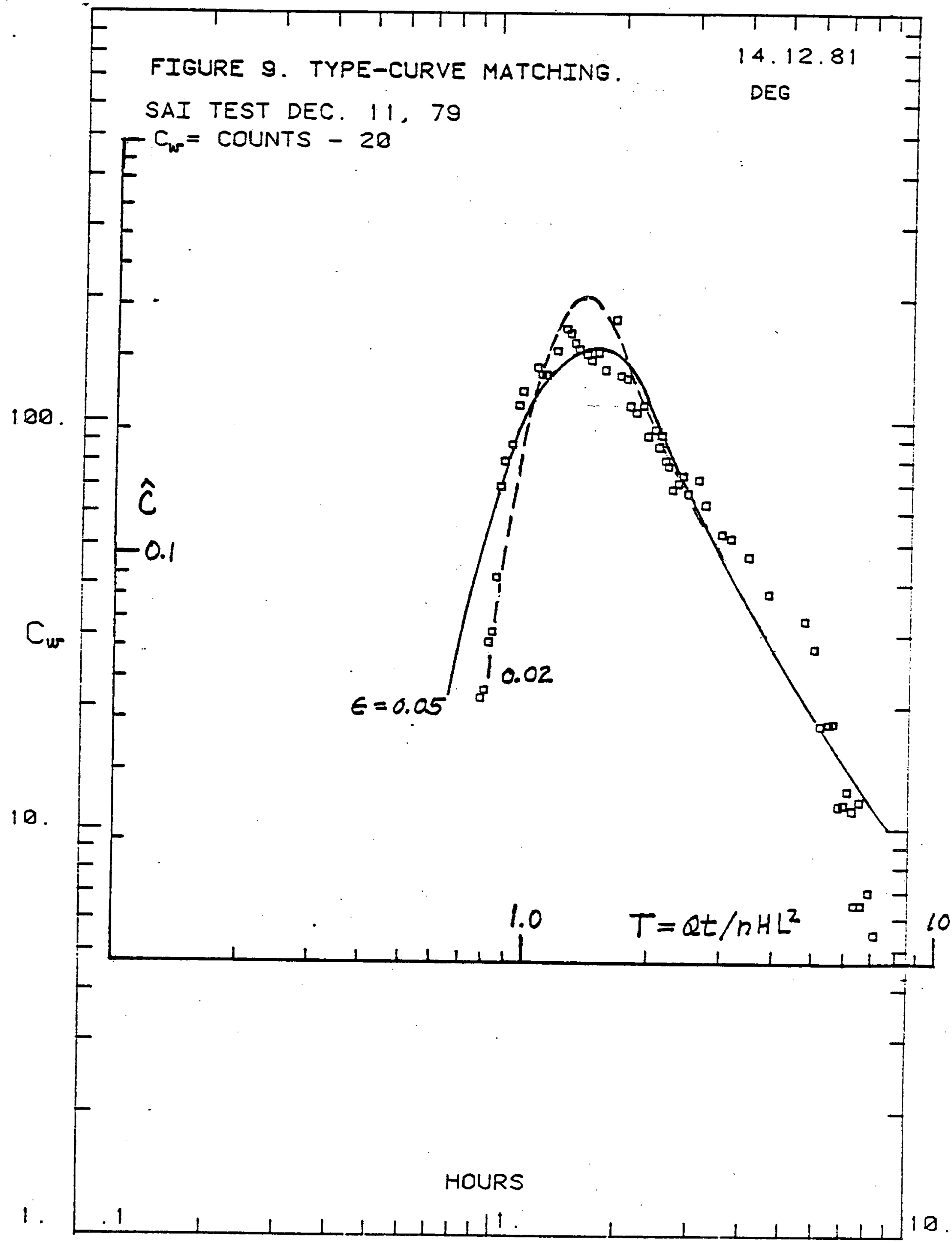


FIGURE 9. TYPE-CURVE MATCHING.
SAI TEST DEC. 11, 79
Cw= COUNTS - 20
14.12.81
DEG
100.
10.
1.
Ĉ
0.1
Cw
0.02
ε=0.05
1.0
T=Qt/nHL²
10
HOURS
.1
1.
10.

The differences in Figure 9 could also indicate that the tested zone does not behave as a homogeneous, constant thickness aquifer. If other sources of errors were eliminated, departures on the tail of the curve would be diagnostic of that possibility because that part of the type-curves is determined solely by convection; i.e., the travel time distribution.

Calculations for the SAI test were also made using the approximate method developed in Appendix H of the SAI draft report. From Figure 9 the peak time t = I.5 hrs and the time to rise from one-half of the peak is t = 0.45 hrs. Then using equation (15) of Appendix H with F = 0.202, G = 0.0488 (β = 2/3)

$$\frac{\alpha}{L} = \frac{1}{4\ln 2}\,\frac{F^2}{G}\,\frac{(\Delta t)^2}{t_p^2} = 0.0271$$

α = 1.52 ft.

and from equation (14) with Q = 3.42 gpm,

$$nH = \frac{Q t_p}{2\pi L^2 F} = 0.0103 \text{ ft.}$$

These results show good agreement with those from the type-curve approach and indicate that the approximate method in the SAI report is reliable. Of course, the type curve method has the advantage that it uses the complete breakthrough curve.

The type-curves of Figures 6 and 7 can also be used to design tracer tests; using estimates of $\epsilon = \alpha/L$, the actual concentration level that will result from a given mass of tracer M can be determined in terms of the effective thickness nH and the well spacing L; i.e., $c_w = M\hat{c}/nHL^2$.

5. COMMENTS AND RECOMMENDATIONS

These results demonstrate the feasibility of the two-well tracer test with a pulse input as a method of determining effective porosity and dispersivity. This type of test has the advantage that the shape of the breakthrough curve is very sensitive to the dispersivity. This is in contrast to the more frequently used step input (Webster et al, 1970; Grove and Beetem, 1971; Robson, 1974; Mercer and Gonzales, 1981) in which dispersion affects the shape of the curve only in the initial low concentration portion of the curve.

The type-curves developed here provide a simple method of designing and analyzing two-well pulse input tracer tests.

The method of analysis used here presumes that α/L is relatively small; results in Gelhar and Collins (1971) indicate that the method should be reasonably accurate for $\alpha/L < 0.1$. If the method is to be used for larger values of α/L, some comparative testing with numerical solutions is suggested. However, it should be recognized that, under those conditions, (large α/L) other factors such as displacement dependent dispersivity and non-Fickian effect (Gelhar et al, 1979) may complicate the interpretation. Also, transverse dispersion is neglected in this analysis; this assumption is reasonable for small α/L because then the dispersion effect occurs primarily along the more direct streamlines for which the fronts will be nearly perpendicular to the streamlines. For larger α/L the dispersion effect along a wider range of streamlines becomes important; numerical testing would also be required in this case. When α/L is large, a finite difference or finite element solution should be routine because a relatively coarse grid could be used.

The type-curves for the pulse input can also be used to treat other inputs by convolution. In particular, this would apply to recirculating tests in which the pulse is routed through the aquifer several times. This aspect is important in the Hanford tests because analysis of the secondary peaks would provide a check on the borehole travel time. Some preliminary work has been done on numerical convolution of the pulse input results. It is recommended that this be developed for analysis of the Hanford data.

6. REFERENCES

Gelhar, L. W., and M. A. Collins, General analysis of longitudinal dispersion in nonuniform flow, Water Resources Res., 7(6), 1511-1521, 1971.

Gelhar, L. W., A. L. Gutjahr, and R. L. Naff, Stochastic analysis of macrodispersion in a stratified aquifer, Water Resources Res., 15(6), 1387-1397, 1979.

Grove, D. B., and W. A. Beetem, Porosity and dispersion constant calculations for a fractured carbonate aquifer using the two-well tracer method, Water Resources Res., 7(1), 128-134, 1971.

Mercer, J. W., and D. Gonzalez, Geohydrology of the proposed Waste Isolation Pilot Plant in southeastern New Mexico, New Mexico Geological Society Special Publication 10, 123-131, 1981.

Robson, S. G., Feasibility of water-quality modeling illustrated by application at Barstow, California, Water Resour. Invest. Rep. 46-73, U. S. Geol. Survey, Menlo Park, Cal., 66 pp., 1974.

Webster, D. S., J. F. Proctor, and I. W. Marine, Two-well tracer test in fractured crystalline rock, U. S. Geol. Survey, Water Supply Paper 1544-I, 22 pp., 1970.

APPENDIX A
τ and ω for the Equal Flow Case

When $Q = Q_r$ it is easily shown that the streamlines are circular arcs. In that case the flow is conveniently described in the polar coordinate system shown in Figure A-1. The piezometric head h for steady flow in this system is

$$h = \frac{Q}{4\pi T_r} \ln (r_1/r_2)^2$$

T_r = transmissivity

$r_1^2 = (x + \ell)^2 + y^2, \; x = -R \sin\gamma$

$r_2^2 = (x - \ell)^2 + y^2, \; y = R \cos\gamma - B$

and using the Darcy equation, seepage velocity along the streamline is

$$U_\gamma = -\frac{T_r}{nHR}\frac{\partial h}{\partial \gamma} = -\frac{Q}{4\pi nHR}\left(\frac{1}{r_1^2}\frac{\partial r_1^2}{\partial \gamma} - \frac{1}{r_2^2}\frac{\partial r_2^2}{\partial \gamma}\right)$$

After extensive algebraic manipulation, this reduces to

$$U_\gamma = \frac{2QR \sin^3\phi}{\pi nHL^2(\cos\gamma - \cos\phi)} \tag{A1}$$

Using this velocity in the travel time integral

$$\tau = \int_{s_0}^{s}\frac{ds}{u(s)} = \int_{\gamma=-\phi}^{\gamma}\frac{R\,d\gamma}{U_\gamma}$$

$$= \frac{nHL^2}{Q}\frac{\pi}{2\sin^3\phi}\left(\sin\gamma + \sin\phi - (\gamma+\phi)\cos\phi\right) \tag{A2}$$

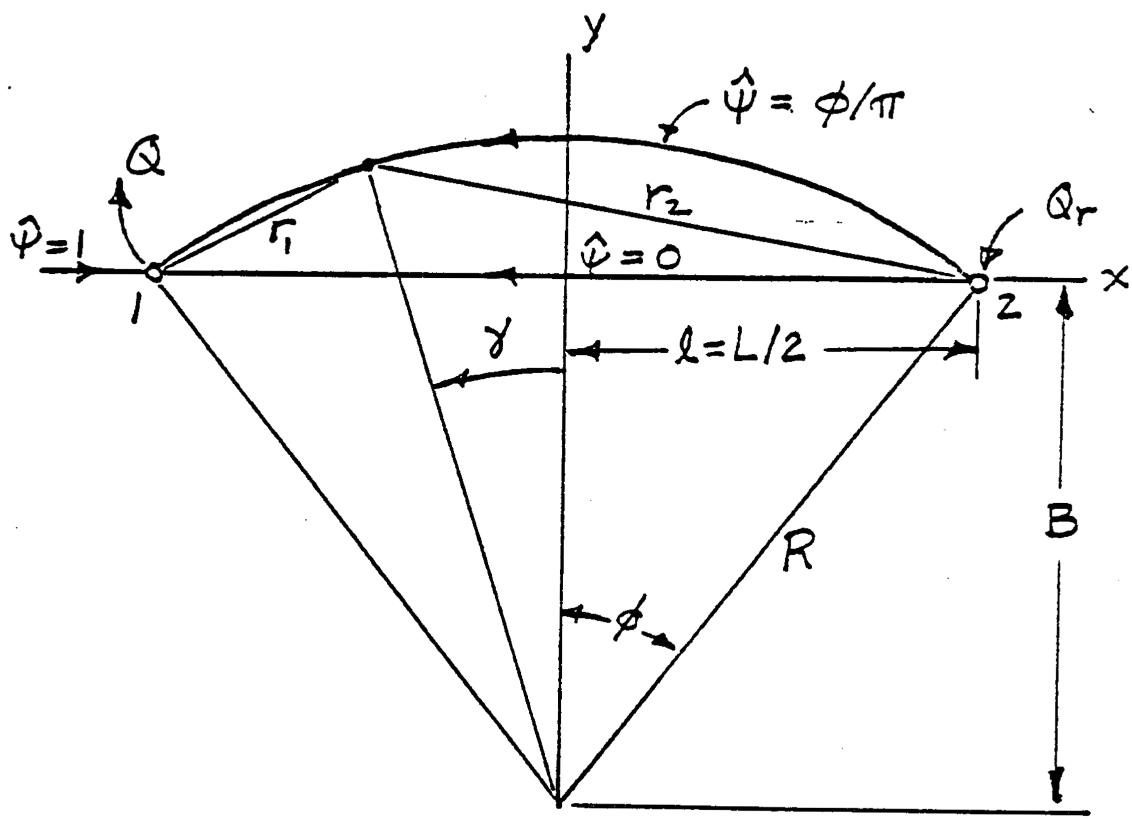


FIGURE A-1. POLAR COORDINATE SYSTEM FOR THE EQUAL FLOW CASE.

When $\gamma = \phi$, (A2) gives the travel time to the pumping well or

$$a(\hat{\Psi}) = \frac{Q\tau_w}{nHL^2} = \frac{\pi}{\sin^3\phi}(\sin\phi - \phi\cos\phi) \qquad \text{(A3)}$$

Equation (A3) gives the dimensionless travel time between the wells as a function of $\hat{\Psi}$; this result is used for a in (4). Similarly, the ω flow integral is evaluated from (A1) as

$$\omega = \int_{s_0}^{\bar{s}} \frac{ds}{[u(s)]^2} = \int_{\gamma=\phi}^{\bar{\gamma}} \frac{R\,d\gamma}{v_\gamma^2} = \frac{1}{L}\left(\frac{nHL^2}{Q}\right)^2 b(\bar{\gamma},\phi)$$

$$b(\bar{\gamma},\phi) = \frac{\pi^2}{2\sin^5\phi}\left[\frac{\bar{\gamma}+\phi}{2} + \frac{\sin\bar{\gamma}\cos\bar{\gamma}}{2} + \frac{\sin\phi\cos\phi}{2} - 2\cos\phi(\sin\bar{\gamma}+\sin\phi) + (\bar{\gamma}+\phi)\cos^2\phi\right] \qquad \text{(A4)}$$

Here $\bar{\gamma}$ indicates the position of the pulse along the streamline corresponding to $\phi = \pi\hat{\Psi}$. At a given time the position $\bar{\gamma}$ is found by solving (A2) for $\gamma = \bar{\gamma}$ when $\tau = t$; this is done iteratively using the program listed in Appendix C. When $\bar{\gamma} > \phi$ the pulse for a given streamline has moved into the pumping well; in this case ω (or b in (4)) was calculated from (A4) using $\bar{\gamma} = \phi$.

If dispersion is neglected, the concentration is found from (5) using (A3) to find $d\hat{\Psi}/da|_{a=T}$:

$$\frac{da}{d\hat{\Psi}} = -3\pi^2\csc^2\pi\hat{\Psi}\cot\pi\hat{\Psi} + 2\pi^3\hat{\Psi}\csc^2\pi\hat{\Psi}\cot^2\pi\hat{\Psi} + \pi^3\hat{\Psi}\csc^4\pi\hat{\Psi}$$

(A5)

The concentration $\hat{c}$ is then found by assigning values of $\hat{\Psi}$ between 0 and 1, calculating T = a from (A3) and $d\hat{\Psi}/da$ from (A5).

APPENDIX B
Graphical Evaluation of τ and ω

The flow integrals τ and ω can be determined from a graphical construction of the streamline pattern. This approach is necessary in the unequal flow case where the analytical description of the flow field is very complicated. The streamline pattern for $Q_r/Q = \beta = 2/3$ (see Figure 1) was constructed by standard superposition of the ray streamlines of the appropriate source and sink strength. If each streamtube has a flow Q_o and width w(s) as a function of the distance along the centerline of the streamtube, s, the velocity is

$$u(s) = Q/(n\ H\ w(s))$$

and then the flow integrals are expressed as

$$\tau = \int \frac{ds}{u} = \frac{nH}{Q_o} \int w\,ds$$

$$\omega = \int \frac{ds}{u^2} = \left(\frac{nH}{Q_o}\right)^2 \int w^2 ds$$

These integrals were approximated by measuring the width of the streamtubes at intervals Δs along each of the streamtubes and summing the appropriate quantities ($w\Delta s$ or $w^2\Delta s$). The integrals were evaluated for each streamtube for intervals in $\hat{\Psi}$ of 0.1 at the pumping and well normalized as in (3). These data for $a(\hat{\Psi})$ and $b(\hat{\Psi})$ at the pumping well were then fit to polynomials of the form

$$\ln a = \sum_{n=0}^{4} a_n \hat{\Psi}^{2n} \qquad \text{(B1)}$$

$$\ln b = \sum_{n=0}^{4} b_n \hat{\Psi}^{2n} \tag{B1}$$

These expressions were then used in the integrations of (4) to find the concentration in the pumping well. Note that (B1) gives only the value of b at the pumping well, b_w. In general, b will increase with time as the pulse approaches the well along a given streamline; this behavior was represented in the form

$$b/b_w = \begin{cases} (T/a)^m, & T \leq a \\ 1, & T > a \end{cases} \tag{B2}$$

where m is a positive exponent to be specified. a and b from (B1) and (B2) were then used in (4) to find $\hat{c}$.

In order to evaluate the above graphical procedure, the flow net evaluation was done first for the equal flow case $\beta = 1$ and the results were compared with the exact analytical approach in Appendix A.

APPENDIX C

LISTING OF FORTRAN PROGRAM FOR NUMERICAL INTEGRATION OF EQUATION 4.

WELL.FOR.156

```
c
c
c

c      This program calculates the concentration (Cw) of
c      tracer appearing in a well as a function of the time
c      elapsed since the tracer was pumped down another well.

c      The user is asked to supply several initializing parameters,
c      and the values of T (time) desired.  First, the all-purpose parameter
c      epsilon (which accounts for dispersion effects) is input.
c      Then, you are asked to input the last value
c      of psi and the increment of psi to use.  (Psi
c      has something to do with what direction the tracer is
c      coming from.)  Using a smaller delta psi can be more
c      accurate, and definitely takes longer.  (We are approximating
c      an integral here, so using smaller steps tends to be more
c      accurate.  Good results are obtained in a reasonable amount
c      of time with dpsi between 0.05 and 0.2 (use smaller dpsi
c      for smaller values of epsilon.))

c      The output of the program, a table of values of time (T)
c      and concentration (Cw), can be sent either to a file
c      (for further processing, plotting etc.) or directly to
c      the line printer.

c      For each value of T, a set of values for the parameter Y
c      must be found.  You can choose from two methods:  'Exact'
c      (equal flow only!) which analytically determines y (a
c      function of psi and T); and 'Nice', in which Y is set to
c      (T/a)**m where m is a user-supplied constant. (usually 1.0)

c      Next, the program asks how to get the magic sets of values
c      A and B, used in the concentration integral.  For the equal
c      flow case, these can be found directly as a simple analytical
c      function of psi.  In unequal flow problems, however, these
c      values cannot be determined exactly, so you must supply instead
c      five coefficients for a polynomial approximation to the functions.

c      Now that initialization is completed, the program will read in
c      values of T from the terminal and calculate and display Cw.
c      When a negative T is input, the program will write out the results
c      (on the printer or to a file called Table.dat) and stop.  To look
c      at the A and B values, or to see T and Cw to more significant
c      figures, you can look at the files A.dat, B.dat, Cw.dat, T.dat,
c      respectively.
```

```
c          Initialize useful variables and open output files

c          FILES USED:
c            A.dat:      A values
c            B.dat:      B values
c            Cw.dat:     calculated Cws
c            T.dat:      T values input by user
c            W.dat:      temporary storage of Ws

        real last,m
        double precision pi
        dimension a(0:4),b(0:4),iflag(2)
        open(unit=36,file='t.dat',access='seqout')  !   store results here
        open(unit=37,file='cw.dat',access='seqout') !   for table

        pi = 3.14159265358979
        innum = -1        ! counts number of Ts supplied
```

```
c            Read in user-supplied parameters

             write(5,1)
1            format(t5,'Input epsilon:  ',$)
             read(5,2) eps
2            format(f6.4)

c              find out what range of psi values to use

             write(5,3)
3            format(t5,'Input last psi, and delta psi:  ',$)
             read(5,4) last,dpsi
4            format(3(f6.4))

c              step evenly, starting at stepsize/2

             first = dpsi/2

             number = (last-first+dpsi)/dpsi ! how many values of psi


c              Output, in table form, can go either to a file or to the
c              line printer.  Here, we find out which is desired, and open
c              the appropriate device as unit 38.

40           write(5,13)
13           format(t5,'Where do you want the output to go?',/,t10,
          1  '(0=file,1=printer):  ',$)
             read(5,6)ians
             if((ians.lt.0).or.(ians.gt.1)) goto 40 ! ignore invalid response
             if(ians.eq.0)    open(unit=38,file='table.dat',access='seqout')
             if(ians.eq.1)    open(unit=38,device='lpt',access='seqout')

c              Y can be calculated in two ways:  a nice, simple method
c              (using a user-specified fudge factor); or a more complex
c              exact method (which works only for the equal flow case).
c              Find out which method is required and set a flag.  If the
c              nice process is to be used, read in the fudge factor now.

80           write(5,14)
14           format(t5,'Exact or nice values for y?',/,t10,
          1  '(0=exact,1=nice):  ',$)
             read(5,6)iyflag ! set iyflag:  0 exact, 1 nice
             if((iyflag.lt.0).or.(iyflag.gt.1)) goto 80 ! bad response
             if (iyflag.eq.0)goto 50 ! exact, no fudge factor
             write(5,15)
15           format(t5,'Input fudge factor M:  ',$)
             read(5,2)m

c              Find which values of a and b to use and set flag

50           write(5,5)
5            format(t5,'Exact or curve fit values for a?',/,t10,
          1  '(0=exact,1=fit):  ',$)
             read(5,6)ians
6            format(i)
             if((ians.lt.0).or.(ians.gt.1)) goto 50 !ignore invalid response
             if(ians.eq.0) call Aexact(number,first,dpsi,pi,iflag(1))!call the
             if(ians.eq.1) call Afit(number,first,dpsi,iflag(1),a)      !procedure
```

```
60      write(5,7)
7       format(t5,'Exact or curve fit values for b?',/,t10,
     1  '(0=exact,1=fit):  ',$)
        read(5,6) ians
        if((ians.lt.0).or.(ians.gt.1)) goto 60 !ignore invalid response
        if(ians.eq.0) call Bexact(number,iflag(2),first,dpsi,pi)
        if(ians.eq.1) call Bfit(number,first,dpsi,iflag(2),b)


        write(5,8)
8       format(t5,'To stop, input a negative value for t')

c         MAIN LOOP:
c         Keep calculating until a negative t is input,
c         storing values of t and cw in data files

70      write(5,9)
9       format(t5,'t = ',$)
        read(5,10) t
10      format(f21.10)
        innum = innum + 1 ! increment counter
        write(36,10)t   ! write t to file

c         as soon as a negative t is input, call a routine
c         to print the results and terminate

        if (t.lt.0) call output(eps,dpsi,first,last,innum,iflag,a,b,
     1  m,iyflag)

c         get y values for this t

        if (iyflag.eq.1) Call Ynice(number,m,t)
        if (iyflag.eq.0) Call Yexact(first,dpsi,number,t,pi)

c         Here we finally call the routine to get the
c         number we are after.

        result = Cw(t,dpsi,number,pi,eps)
        write(37,10)result ! write Cw to file
        write(5,11) result ! and terminal
11      format(t10,'Cw = ',f9.5)
        goto 70 ! loop forever until negative t is input
        end ! end of main program
```

```
      Subroutine Aexact(number,first,dpsi,pi,ifl)
c       This routine calculates values for the parameter a
c       using an exact equation.  The As, one for each value
c       of psi used, are written to file a.dat for use later
c       in the program.  The user specifies whether this routine
c       or the approximate version 'Afit' is to be used.  Note
c       that the exact method works ONLY for equal flow problems.

c       initialize variables and open output file

      double precision pi
      open(unit=33,file='a.dat',access='seqout') ! open data file
      ifl = 0 ! set flag indicating exact method
      psi = first ! initialize psi

c       loop through all values of psi, calculating an a for each

      do 1 i = 1,number ! how many values we need
      a = pi * (1/sin(pi*psi)**2)*(1-pi*psi*(cos(pi*psi)/sin(pi*psi)))
      write(33,2)a ! write it to the file
2     format(f21.10)
      psi = psi + dpsi !increment psi
1     continue
      close(unit=33) ! close file
      return
      end
```

```
      Subroutine Bexact(number,ifl,first,dpsi,pi)

c        Bexact gets values for b with an exact equation.
c        Bs are written to the file β.dat for later use.
c        Flag 'ifl' is set so the output routine knows that
c        exact b values were used.  Remember, the exact
c        method can be used only for equal flow.

      double precision phi,b,pi
      ifl = 0 ! set flag:  exact values
      open (unit=34,file='b.dat',access='seqout') ! open b.dat

c        loop:  get a b; write it out

      do 1 i = 1,number
      phi = pi*(first+(i-1)*dpsi) ! phi = psi*pi
      b = pi**2*(phi-3*cos(phi)*sin(phi)+2*phi*(cos(phi))**2)
     1 /(2*(sin(phi))**5)
      write (34,2)b ! output b to file
2     format(f21.10)
1     continue
      close(unit=34) ! close output file b.dat
      return
      end
```

```
      Subroutine Afit(number,first,dpsi,ifl,a)

c       This routine opens the file a.dat and calls
c       the routine 'Fit', which does a polynomial curve
c       fit using user-supplied coefficients.  The five
c       coefficients are passed back up to the main program
c       in the array 'a', and flag ifl is set to indicate
c       the use of approximate values.

      dimension a(0:4) ! coefficient array
      open (unit=32,file='a.dat',access='seqout') ! open output file
      ifl = 1  ! flag approximate a values
      call Fit(number,first,dpsi,a) ! call routine to generate As
      close(unit=32) !clean up
      return
      end


      Subroutine Bfit(number,first,dpsi,ifl,b)

c       This routine is identical to Afit, but the file b.dat
c       is opened instead of a.dat.  Coefficients are returned in
c       the array 'b'.

      dimension b(0:4) ! array to hold 5 coefficients
      open(unit=32,file='b.dat',access='seqout') ! output to b.dat
      ifl = 1  ! flag approximate b values
      call Fit(number,first,dpsi,b) ! get b values
      close(unit=32)
      return
      end
```

```
      Subroutine Fit(number,first,dpsi,c)
c
c       Fit reads in 5 coefficients and uses them to approximate
c       either a or b, depending on whether it was called by Afit
c       or Bfit (in the first case, data goes to a.dat, in the
c       second, b.dat).  The variable c is equivalent to either
c       a or b, whichever is appropriate.
c
      dimension c(0:4) ! array of curve fit coefficients
      do 10 i=0,4 ! get them one at a time
      write(5,1)
1     format(t5,'Input a coefficient')
      read(5,2) c(i) ! read one in
2     format(f8.4)
10    write(5,2) c(i) !write it to the terminal to confirm
c
c       calculate c and write it to file
c       unit 32 has been opened by our caller
c       as the correct output file (a.dat or b.dat)
c
      psi = first ! go through all psis
      do 3 i = 1,number
      temp = 0
      do 4 n = 0,4
c
c       the coefficients are for a polynomial fit in
c       psi squared
c
4     temp = temp + c(n)*psi**(2*n)
c
c       the natural log of the data was used to determine
c       coefficients, so we must take the exponential of
c       the result
c
      result = exp(temp)
      write(32,5)result ! write to a.dat or b.dat
5     format(f21.10)
      psi = psi + dpsi
3     continue
      return
      end
```

```
      Subroutine Ynice(number,m,t)
c       Here we calculate y values with sleazy-but-nice formula.
c       (using only a and a user-supplied fudge factor).  Actually,
c       we write out not Y itself but a close relative W (=b*y).
c       The w values are put in (guess what?) w.dat.

c       define variables and open useful data files

      real m
      double precision y,b
      open(unit=33,file='a.dat',access='seqin')
      open(unit=34,file='b.dat',access='seqin')
      open(unit=35,file='w.dat',access='seqout')
      ifl = 1 ! set flag: approximate y values
      Do 1 j = 1,number ! 1 y for each psi
      read(33,2)a ! read in a
      read(34,2)b ! and b
2     format(f21.10)
      y = (t/a)**m ! m is fudge factor
      if((t/a).gt.1) y = 1
      w = b*y ! get w
      write(35,2)w ! write w to w.dat
1     continue

c       tidily close all files...

      close(unit=33)
      close(unit=34)
      close(unit=35)
      return
      end
```

```
      Subroutine Yexact(first,dpsi,number,t,pi)
c       This routine gets exact values for y (and w)
c       using a rather messy equation (equal flow only).

c       define variables and open files

      double precision phi,temp1,temp2,w,gamma
      open(unit=33,file='a.dat',access='seqin') ! a values
      open(unit=34,file='b.dat',access='seqin') ! b values
      open(unit=35,file='w.dat',access='seqout') ! put Ws here

      itl = 0 ! flag exact y values
      Do 1 i = 1,number ! loop through all psis
      read(33,2)a ! get a
      read(34,2)b ! and b
2     format(f21.10)
      phi = pi*(first+(i-1)*dpsi) ! phi = pi*psi
      if((t/a).gt.1.0)goto 999 ! special case, w = b

c       invoke function to find gamma, used in
c       calculating y (w)

      gamma = Gappr(phi,a,t)

      temp1 = ((gamma+phi)/2+(sin(gamma)*cos(gamma)/2)+sin(phi)*
     1 cos(phi)/2 - 2*cos(phi)*(sin(gamma)+sin(phi))+(gamma+phi)*
     2 cos(phi)**2)
      temp2 = pi**2/(2*sin(phi)**5)
      w = temp1*temp2
      goto 1000
999   w = b
1000  if (w.le.0.001)w = 0.001 ! ugly things happen if w = 0
      write(35,2)w ! write to w.dat
1     continue ! get next w
      return
      end
```

```
      Function Gappr(phi,a,t)
c       returns an approximation to gamma
      double precision gamma
      oldf = t/a
      oldx = 0
      x = 2*oldf - 1
10    if(abs(oldx-x).lt.0.0001) goto 20 ! close enough yet?
      g = phi*x
      f = (sin(phi)+sin(g)-(phi+g)*cos(phi))
     1 /(2*(sin(phi)-phi*cos(phi)))
      oldx = x
      x = x + (oldf - f)/2 ! get new x
      goto 10 ! check again
20    Gappr = phi*x
      return
      end
```

```
      Function Cw(t,dpsi,number,pi,eps)
c       Approximate the integral for the concentration Cw
      double precision temp,c,pi,W,aaa
      c = 0
      open(unit=33,file='a.dat',access='seqin')
      open(unit=34,file='b.dat',access='seqin')
      open(unit=35,file='w.dat',access='seqin')
      do 1 i = 1,number
      read(33,2) a
      read(34,2) b
      read(35,2) w
2     format(f21.10)
      temp = exp(-((a-t)**2. / (4.*eps*w)))
      c = c + dpsi*temp/sqrt(4.*pi*eps*w)
1     continue
      cw = c
      return
      end
```

```
      Subroutine Output(eps,dpsi,first,last,innum,iflag,a,b,m,
     1  iyflag)

c        Output prints the results in a table, either
c        in a file (table.dat) or on the line printer

      dimension iflag(2) ! flags for a & b:  exact or approx
      dimension a(0:4) ! a coefficients
      dimension b(0:4) ! b coefficients

c        open files with results

      close(unit=36)
      close(unit=37)
      open(unit=36,file='t.dat',access='seqin')
      open(unit=37,file='cw.dat',access='seqin')

c        unit 38 has already been opened as either
c        table.dat or Lpt:
c        Display the parameters input by user:

      write(38,1)eps,dpsi,first,last
1     format(///,10x,'epsilon = ',f8.4,5x,'dpsi = ',f8.4,5x,
     1  'first psi = ',f8.4,5x,'last psi = ',f8.4,/,t10)

c        if exact a & b values, say so

      if(iflag(1).eq.0) write(38,5)
5     format(t10,'Exact a values')
      if(iflag(2).eq.0) Write(38,6)
6     format(t10,'Exact b values')

c        if curve fit was used, print coefficients

      if(iflag(1).eq.1) write(38,7)(a(i),i=0,4)
7     format(t10,'A coefficients are:  ',5(f10.4,2x))
      if(iflag(2).eq.1) write(38,8)(b(i),i=0,4)
8     format(t10,'B coefficients are:  ',5(f10.4,2x))

c        tell how y was calculated

      if(iyflag.eq.0)write(38,10
10    format(t10,'Exact y values',//)
      if(iyflag.eq.1)write(38,11)m
11    format(t10,'Approximate y values, m is:  ',f8.4,//)

c        print header for table

      write(38,9)
9     format(t31,'T',12x,'Cw',/)

c        read in t and Cw from t.dat and Cw.dat;
c        write them out together

      do 2 i = 1,innum ! innum = # of Ts supplied
      read(36,4)t ! get t
4     format(f21.10)
      read(37,4)cw ! get Cw
      write(38,3)t,cw ! make table
3     format(25x,f8.4,5x,f8.4)
```

```
2       continue
c         execution stops here
        stop 'Hydrology is all wet.'
        end
```

APPENDIX D

TABLES OF TYPE CURVE DATA

epsilon = .2000   dpsi = .0500   first psi = .0250   last psi = .9750
beta = 1.0000
Exact a values
Exact b values
Exact y values

T	$\hat{C}$
0.1000	0.0000
0.2000	0.0013
0.3000	0.0205
0.4000	0.0608
0.5000	0.1031
0.6000	0.1373
0.7000	0.1619
0.8000	0.1786
0.9000	0.1895
1.0000	0.1963
1.1000	0.2012
1.2000	0.2045
1.3000	0.2055
1.4000	0.2041
1.5000	0.2005
1.8000	0.1787
2.0000	0.1590
2.5000	0.1099
3.0000	0.0760
3.5000	0.0562
4.0000	0.0444
4.5000	0.0364
5.0000	0.0308
5.5000	0.0266
6.0000	0.0232
6.5000	0.0206
7.0000	0.0184
7.5000	0.0167
8.0000	0.0152
9.0000	0.0128
10.0000	0.0110

Tab. 1

epsilon = .2000 dpsi = .0500 first psi = .0250 last psi = .9750
beta = 1.0000
Exact a values
Exact b values
Approximate y values, m is: 1.0000

T	$\hat{C}$
0.1000	0.0000
0.2000	0.0056
0.3000	0.0308
0.4000	0.0689
0.5000	0.1076
0.6000	0.1404
0.7000	0.1656
0.8000	0.1833
0.9000	0.1947
1.0000	0.2011
1.1000	0.2035
1.2000	0.2029
1.3000	0.2001
1.4000	0.1956
1.5000	0.1901
1.8000	0.1698
2.0000	0.1555
2.5000	0.1225
3.0000	0.0961
3.5000	0.0762
4.0000	0.0611
4.5000	0.0503
5.0000	0.0420
5.5000	0.0357
6.0000	0.0307
6.5000	0.0269
7.0000	0.0237
7.5000	0.0212
8.0000	0.0191
9.0000	0.0159
10.0000	0.0135

Tab. 2

epsilon = .0100 dpsi = .0500 first psi = .0250 last psi = .9750
beta = 1.0000
Exact a values
Exact b values
Exact y values

T	$\hat{C}$
0.1000	0.0000
0.2000	0.0000
0.3000	0.0000
0.4000	0.0000
0.5000	0.0000
0.6000	0.0004
0.6500	0.0027
0.7000	0.0113
0.7500	0.0337
0.8000	0.0778
0.8500	0.1471
0.9000	0.2369
0.9500	0.3354
1.0000	0.4277
1.0500	0.5010
1.1000	0.5470
1.1500	0.5608
1.2000	0.5447
1.3000	0.4577
1.4000	0.3588
1.4500	0.3183
1.5000	0.2850
1.6000	0.2356
1.7000	0.2022
1.8000	0.1771
1.9000	0.1577
2.0000	0.1423
2.5000	0.0941
3.0000	0.0687
3.5000	0.0544
4.0000	0.0415
4.5000	0.0373
5.0000	0.0343
5.5000	0.0215
6.0000	0.0209
6.5000	0.0270
7.0000	0.0249
7.5000	0.0150
8.0000	0.0077
9.0000	0.0130
10.0000	0.0198

Tab. 3

epsilon = .0100 dpsi = .0100 first psi = .0050 last psi = .9750
beta = 1.0000
Exact a values
Exact b values
Exact y values

T	$\hat{C}$
0.1000	0.0000
0.2000	0.0000
0.3000	0.0000
0.4000	0.0000
0.5000	0.0000
0.6000	0.0004
0.7000	0.0113
0.8000	0.0778
0.8500	0.1471
0.9000	0.2369
0.9500	0.3354
1.0000	0.4277
1.0500	0.5010
1.1000	0.5470
1.1500	0.5608
1.2000	0.5446
1.2500	0.5068
1.3000	0.4576
1.4000	0.3589
1.5000	0.2849
1.6000	0.2358
1.7000	0.2021
1.8000	0.1772
1.9000	0.1578
2.0000	0.1421
2.5000	0.0937
3.0000	0.0687
3.5000	0.0536
4.0000	0.0435
4.5000	0.0363
5.0000	0.0310
5.5000	0.0269
6.0000	0.0236
6.5000	0.0210
7.0000	0.0189
7.5000	0.0171
8.0000	0.0156
9.0000	0.0132
10.0000	0.0114

Tab. 4

epsilon = .0500 dpsi = .0200 first psi = .0100 last psi = .9700
beta = 1.0000
Exact a values
Exact b values
Exact y values

T	$\hat{C}$
0.1000	0.0000
0.2000	0.0000
0.3000	0.0000
0.4000	0.0024
0.5000	0.0200
0.6000	0.0647
0.6500	0.0955
0.7000	0.1292
0.7500	0.1635
0.8000	0.1966
0.9000	0.2543
0.9500	0.2775
1.0000	0.2966
1.0500	0.3120
1.1000	0.3241
1.1500	0.3323
1.2000	0.3363
1.2500	0.3363
1.3000	0.3325
1.4000	0.3152
1.5000	0.2889
1.6000	0.2583
1.7000	0.2275
1.8000	0.1993
1.9000	0.1748
2.0000	0.1544
2.2000	0.1242
2.5000	0.0958
3.0000	0.0690
3.5000	0.0533
4.0000	0.0431
4.5000	0.0359
5.0000	0.0306
5.5000	0.0265
6.0000	0.0233
6.5000	0.0207
7.0000	0.0186
7.5000	0.0168
8.0000	0.0153
9.0000	0.0130
10.0000	0.0112

Tab. 5

epsilon = 0.0500 dpsi = 0.0200 first psi = 0.0100 last psi = 0.9700
beta = 1.0000
A coefficients are: a_0 = 0.0480 3.7756 3.3016 -4.9376 7.0424
B coefficients are: b_0 = 0.2730 6.7200 3.9700 -7.3120 13.3020
Approximate y values, m is: 1.0000

T	$\hat{C}$
0.1000	0.0000
0.2000	0.0000
0.3000	0.0001
0.4000	0.0035
0.5000	0.0216
0.6000	0.0649
0.6500	0.0951
0.7000	0.1290
0.7500	0.1643
0.8000	0.1989
0.8500	0.2312
0.9000	0.2597
0.9500	0.2836
1.0000	0.3025
1.0500	0.3163
1.1000	0.3272
1.1500	0.3346
1.2000	0.3380
1.2500	0.3374
1.3000	0.3331
1.4000	0.3149
1.5000	0.2880
1.6000	0.2569
1.7000	0.2259
1.8000	0.1975
1.9000	0.1731
2.0000	0.1528
2.2000	0.1229
2.5000	0.0950
3.0000	0.0687
3.5000	0.0534
4.0000	0.0433
4.5000	0.0361
5.0000	0.0308
5.5000	0.0268
6.0000	0.0235
6.5000	0.0209
7.0000	0.0187
7.5000	0.0170
8.0000	0.0154
9.0000	0.0130
10.0000	0.0112

Tab. 6

epsilon = 0.0200 dpsi = 0.0200 first psi = 0.0100 last psi = 0.9700

A coefficients are: a_0 = 0.2010 3.3090 -0.9750 3.2250 -1.3020
B coefficients are: b_0 = 0.6570 4.9770 -1.1240 4.6500 -1.4520
Approximate y values, @ is: 1.0000

unequal flow

T	$\hat{C}$
0.1000	0.0000
0.2000	0.0000
0.3000	0.0000
0.4000	0.0000
0.5000	0.0000
0.6000	0.0014
0.7000	0.0110
0.8000	0.0438
0.9000	0.1105
1.0000	0.2033
1.1000	0.2983
1.2000	0.3722
1.3000	0.4160
1.4000	0.4278
1.5000	0.4092
1.8000	0.2796
2.0000	0.2112
2.5000	0.1294
3.0000	0.0927
3.5000	0.0711
4.0000	0.0570
4.5000	0.0471
5.0000	0.0399
5.5000	0.0345
6.0000	0.0302
6.5000	0.0269
7.0000	0.0240
8.0000	0.0196
9.0000	0.0167
10.0000	0.0144

Tab. 7

epsilon = 0.0500 dpsi = 0.0200 first psi = 0.0100 last psi = 0.9700

A coefficients are: a_0 = 0.2010 3.3090 -0.9750 3.2250 -1.3020
B coefficients are: b_0 = 0.6570 4.9770 -1.1240 4.6500 -1.4520
Approximate y values, m is: 1.0000

unequal flow

T	$\hat{C}$
0.1000	0.0000
0.2000	0.0000
0.3000	0.0000
0.4000	0.0009
0.5000	0.0077
0.6000	0.0279
0.7000	0.0652
0.8000	0.1157
0.9000	0.1705
1.0000	0.2214
1.1000	0.2627
1.2000	0.2915
1.3000	0.3102
1.4000	0.3201
1.5000	0.3196
1.6000	0.3101
1.7000	0.2934
1.8000	0.2719
2.0000	0.2247
2.5000	0.1364
3.0000	0.0945
3.5000	0.0717
4.0000	0.0573
4.5000	0.0473
5.0000	0.0399
5.5000	0.0345
6.0000	0.0302
6.5000	0.0267
7.0000	0.0239
7.5000	0.0216
8.0000	0.0196
9.0000	0.0165
10.0000	0.0143

Tab. 8

epsilon = 0.1000 dpsi = 0.0200 first psi = 0.0100 last psi = 0.9700

A coefficients are: a_0 = 0.2010 3.3090 -0.9750 3.2250 -1.3020
B coefficients are: b_0 = 0.6570 4.9770 -1.1240 4.6500 -1.4520
Approximate y values, m is: 1.0000

unequal flow

T	θ
0.1000	0.0000
0.2000	0.0000
0.3000	0.0021
0.4000	0.0134
0.5000	0.0378
0.6000	0.0726
0.7000	0.1115
0.8000	0.1489
0.9000	0.1814
1.0000	0.2072
1.1000	0.2262
1.2000	0.2386
1.3000	0.2472
1.4000	0.2529
1.5000	0.2546
1.6000	0.2521
1.7000	0.2463
1.8000	0.2373
2.0000	0.2132
2.5000	0.1458
3.0000	0.0998
3.5000	0.0738
4.0000	0.0584
4.5000	0.0478
5.0000	0.0403
5.5000	0.0347
6.0000	0.0303
6.5000	0.0267
7.0000	0.0239
7.5000	0.0216
8.0000	0.0196
9.0000	0.0165
10.0000	0.0143

Tab. 9

epsilon = 0.0200 dpsi = 0.0200 first psi = 0.0100 last psi = 0.9700
beta = 1.0000
Exact a values
Exact b values
Exact y values

T	$\hat{C}$
0.1000	0.0000
0.2000	0.0000
0.3000	0.0000
0.4000	0.0000
0.5000	0.0006
0.6000	0.0102
0.7000	0.0543
0.8000	0.1475
0.9000	0.2667
1.0000	0.3721
1.1000	0.4378
1.2000	0.4523
1.3000	0.4202
1.4000	0.3628
1.5000	0.3021
1.8000	0.1827
2.0000	0.1441
2.5000	0.0939
3.0000	0.0686
3.5000	0.0534
4.0000	0.0432
4.5000	0.0361
5.0000	0.0308
5.5000	0.0267
6.0000	0.0235
6.5000	0.0209
7.0000	0.0188
7.5000	0.0170
8.0000	0.0155
9.0000	0.0131
10.0000	0.0113

Tab. 10

epsilon = 0.1000 dpsi = 0.0200 first psi = 0.0100 last psi = 0.9700
beta = 1.0000
Exact a values
Exact b values
Exact y values

T	$\hat{C}$
0.1000	0.0000
0.2000	0.0000
0.3000	0.0024
0.4000	0.0213
0.5000	0.0623
0.6000	0.1125
0.7000	0.1593
0.8000	0.1971
0.9000	0.2247
1.0000	0.2436
1.1000	0.2564
1.2000	0.2636
1.3000	0.2644
1.4000	0.2592
1.5000	0.2487
1.6000	0.2344
1.7000	0.2176
1.8000	0.1996
1.9000	0.1815
2.0000	0.1642
2.5000	0.1011
3.0000	0.0707
3.5000	0.0539
4.0000	0.0433
4.5000	0.0359
5.0000	0.0305
5.5000	0.0264
6.0000	0.0232
6.5000	0.0206
7.0000	0.0185
7.5000	0.0167
8.0000	0.0152
8.5000	0.0139
9.0000	0.0128
10.0000	0.0111

Tab. 11

epsilon = 0.0200 dpsi = 0.0100 first psi = 0.0050 last psi = 0.9700
beta = 1.0000
Exact a values
Exact b values
Exact y values

T	$\hat{C}$
0.1000	0.0000
0.2000	0.0000
0.3000	0.0000
0.4000	0.0000
0.5000	0.0006
0.6000	0.0102
0.7500	0.0954
0.8000	0.1475
0.8500	0.2064
0.9000	0.2667
0.9500	0.3233
1.0000	0.3721
1.0500	0.4107
1.1000	0.4378
1.1500	0.4518
1.2000	0.4523
1.2500	0.4410
1.3000	0.4202
1.4000	0.3628
1.5000	0.3021
1.6000	0.2507
1.7000	0.2116
1.8000	0.1827
1.9000	0.1610
2.0000	0.1441
2.5000	0.0939
3.0000	0.0686
3.5000	0.0534
4.0000	0.0433
4.5000	0.0361
5.0000	0.0308
5.5000	0.0267
6.0000	0.0235
6.5000	0.0209
7.0000	0.0188
7.5000	0.0170
8.0000	0.0155
9.0000	0.0131
10.0000	0.0113

Tab. 12

epsilon = 0.0100 dpsi = 0.0200 first psi = 0.0100 last psi = 0.9700

A coefficients are: a_0 = 0.2010 3.3090 -0.9750 3.2250 -1.3020
B coefficients are: b_0 = 0.6570 4.9770 -1.1240 4.6500 -1.4520
Approximate y values, m is: 1.0000

unequal flow

T	$\hat{C}$
0.1000	0.0000
0.2000	0.0000
0.3000	0.0000
0.4000	0.0000
0.5000	0.0000
0.6000	0.0000
0.7000	0.0004
0.8000	0.0078
0.9000	0.0469
1.0000	0.1492
1.1000	0.3018
1.2000	0.4412
1.3000	0.5171
1.4000	0.5156
1.5000	0.4566
1.8000	0.2640
2.0000	0.2021
2.5000	0.1283
3.0000	0.0924
3.5000	0.0711
4.0000	0.0570
4.5000	0.0473
5.0000	0.0400
5.5000	0.0345
6.0000	0.0303
6.5000	0.0269
7.0000	0.0240
7.5000	0.0218
8.0000	0.0198
9.0000	0.0167
10.0000	0.0144

Tab. 13

epsilon = 0.2000 dpsi = 0.0200 first psi = 0.0100 last psi = 0.9700

A coefficients are: a_0 = 0.2010 3.3900 -0.9750 3.2250 -1.3020
B coefficients are: b_0 = 0.6570 4.9770 -1.1240 4.6500 -1.4520
Approximate y values, m is: 1.0000

unequal flow

T	$\hat{c}$
0.1000	0.0000
0.2000	0.0029
0.3000	0.0189
0.4000	0.0468
0.5000	0.0784
0.6000	0.1082
0.7000	0.1332
0.8000	0.1533
0.9000	0.1683
1.0000	0.1791
1.1000	0.1861
1.2000	0.1904
1.3000	0.1934
1.4000	0.1957
1.5000	0.1968
1.8000	0.1914
2.0000	0.1814
2.5000	0.1443
3.0000	0.1068
3.5000	0.0790
4.0000	0.0612
4.5000	0.0493
5.0000	0.0413
5.5000	0.0352
6.0000	0.0308
6.5000	0.0271
7.0000	0.0241
7.5000	0.0218
8.0000	0.0198
9.0000	0.0167
10.0000	0.0143

Tab. 14

epsilon = 0.0050 dpsi = 0.0050 first psi = 0.0025 last psi = 0.9700

A coefficients are: a_0 = 0.2010 3.3090 -0.9750 3.2250 -1.3020
B coefficients are: b_0 = 0.6570 4.9770 -1.1240 4.6500 -1.4520
Approximate y values, m is: 1.0000

unequal flow

T	$\hat{C}$
0.1000	0.0000
0.2000	0.0000
0.3000	0.0000
0.4000	0.0000
0.5000	0.0000
0.6000	0.0000
0.7000	0.0000
0.7500	0.0000
0.8000	0.0002
0.8500	0.0016
0.9000	0.0078
0.9500	0.0271
1.0000	0.0722
1.0500	0.1531
1.1000	0.2677
1.1500	0.3979
1.2000	0.5159
1.2500	0.5994
1.3000	0.6381
1.4000	0.5667
1.5000	0.4630
1.6000	0.3617
1.7000	0.2970
1.8000	0.2543
1.9000	0.2233
2.0000	0.1991
2.2000	0.1635
2.5000	0.1280
3.0000	0.0924
3.5000	0.0712
4.0000	0.0572
4.5000	0.0474
5.0000	0.0402
5.5000	0.0347
6.0000	0.0304
6.5000	0.0270
7.0000	0.0242
8.0000	0.0199
9.0000	0.0168
10.0000	0.0145

Tab. 18

epsilon = 0.0050 dpsi = 0.0100 first psi = 0.0050 last psi = 0.9750

Exact a values
Exact b values
Exact y values

T	$\hat{C}$
0.1000	0.0000
0.2000	0.0000
0.3000	0.0000
0.4000	0.0000
0.5000	0.0000
0.6000	0.0000
0.7000	0.0005
0.8000	0.0197
0.9000	0.1635
1.0000	0.4722
1.0250	0.5461
1.0500	0.6077
1.0750	0.6532
1.1000	0.6797
1.1250	0.6866
1.1500	0.6758
1.1750	0.6505
1.2000	0.6153
1.2250	0.5743
1.2500	0.5316
1.3000	0.4514
1.4000	0.3377
1.5000	0.2722
1.6000	0.2297
1.7000	0.1990
1.8000	0.1755
2.0000	0.1415
2.2000	0.1181
2.5000	0.0939
3.0000	0.0689
3.5000	0.0538
4.0000	0.0436
4.5000	0.0364
5.0000	0.0311
5.5000	0.0270
6.0000	0.0237
6.5000	0.0211
7.0000	0.0190
8.0000	0.0157
9.0000	0.0132
10.0000	0.0114

Tab. 19

epsilon = 0.0020 dpsi = 0.0100 first psi = 0.0050 last psi = 0.9750

Exact a values
Exact b values
Exact y values

T	$\hat{C}$
0.1000	0.0000
0.2000	0.0000
0.3000	0.0000
0.4000	0.0000
0.5000	0.0000
0.6000	0.0000
0.7000	0.0000
0.8000	0.0003
0.8500	0.0054
0.9000	0.0452
0.9500	0.1944
0.9750	0.3265
1.0000	0.4854
1.0250	0.6464
1.0500	0.7804
1.0750	0.8642
1.1000	0.8871
1.1250	0.8556
1.1500	0.7883
1.1750	0.7065
1.2000	0.6265
1.2250	0.5572
1.2500	0.5007
1.3000	0.4194
1.2500	0.5007
1.4000	0.3244
1.5000	0.2671
1.6000	0.2274
1.8000	0.1750
2.0000	0.1415
2.2000	0.1182
2.5000	0.0941
3.0000	0.0692
3.5000	0.0540
4.0000	0.0438
4.5000	0.0366
5.0000	0.0312
5.5000	0.0271
6.0000	0.0238
6.5000	0.0212
7.0000	0.0190
7.5000	0.0172
8.0000	0.0157
9.0000	0.0133
10.0000	0.0115

Tab. 20