CHAPTER 9

Finite Difference Methods for Transport Equations

Numerical Methods in the Hydrological Sciences
Special Publication Series 57

doi:10.1029/057SP14

9. Finite Difference Methods for Transport Equations

9.1. Background

In Chapter 8, we considered problems related to time-dependent diffusion of quantities such as heat, solutes, and head. In those problems, diffusion was the only process responsible for the flux or transport of the quantity of interest. However, a solute in a moving fluid not only will be transported by diffusion, but will also be advected with the fluid. In this case both transport processes – advection and diffusion – must be included in the equation for the time-dependent change in concentration,

$$\frac{\partial c}{\partial t} + u\frac{\partial c}{\partial x} = D\frac{\partial^2 c}{\partial x^2} \tag{9.1}$$

where u is the advection velocity and D is the diffusion or dispersion coefficient. (Vertical and transverse variations in flow velocity increase rates of mixing compared to molecular or turbulent diffusion. The increased rate of mixing is commonly parameterized by a dispersion coefficient. See, e.g, Fischer et al., 1979 or Hemond and Fechner-Levy, 2000.) Equation (9.1) is commonly referred to as the advection-diffusion or advection-dispersion equation (or simply the transport equation). Here we'll refer to (9.1) as the advection-dispersion equation.

9.2. Numerical dispersion

When considering numerical solutions to the advection- dispersion equation, it is helpful to non-dimensionalize equation (9.1) using the dimensionless parameters $\hat{t} = tu/L$, and $\hat{x} = x/L$,

$$\frac{\partial c}{\partial \hat{t}} + \frac{\partial c}{\partial \hat{x}} = \frac{1}{\mathbf{Pe}}\frac{\partial^2 c}{\partial \hat{x}^2} \tag{9.2}$$

where L is a length scale. The dimensionless coefficient on the right-hand-side of the equation is the Peclet number, $\mathbf{Pe} = uL/D$. The Peclet number represents the ratio of the time scale for advection to the time scale for diffusion or dispersion. When the Peclet number is large, the term on the right can be neglected, i.e., dispersion is relatively unimportant. When the Peclet number is small, transport is dominantly by dispersion.

To illustrate one of the difficulties associated with solving equations having the form of the advection-dispersion equation numerically, we first consider the equation for large Peclet numbers. In this case the dispersion term is negligible compared to the advection term and the equation we want to solve is:

$$\frac{\partial c}{\partial \hat{t}} + \frac{\partial c}{\partial \hat{x}} = \frac{\partial c}{\partial t} + u\frac{\partial c}{\partial x} = 0.$$

Let c be the concentration of some tracer in a stream flowing at velocity u. The problem is to describe the downstream transport of the tracer when it is injected at location $x = 0$ such that the concentration in the stream at the injection point is held at 1 unit. That is, initially c is equal to zero everywhere, then $c = 1$ at $x = 0$ for all subsequent time. We can see

intuitively what the solution to the problem is: the concentration "front" (c = 1 on the upstream side and c = 0 on the downstream side) propagates downstream at speed u.

The advection equation provides an easy demonstration that apparently sensible finite-difference schemes do not always work. Take the straightforward explicit representation of the (dimensional version of the) advection equation:

$$c_i^{j+1} = c_i^j - \left(\frac{u\Delta t}{2\Delta x}\right)\left(c_{i+1}^j - c_{i-1}^j\right) \tag{9.3}$$

This benign-looking method is actually *unconditionally* unstable! That is, there are no values of the time and space steps, no matter how small, that will make this method stable. You can do the simple "worst-case" stability analysis (Chapter 8.3) to show this.

The Lax scheme often is used to solve equations in which the advective term is dominant. To use this scheme, we replace the c_i^j on the right-hand side of the finite-difference equation above with $\left(c_{i+1}^j + c_{i-1}^j\right)/2$.

$$c_i^{j+1} = \left(\frac{c_{i+1}^j + c_{i-1}^j}{2}\right) - \left(\frac{u\Delta t}{2\Delta x}\right)\left(c_{i+1}^j - c_{i-1}^j\right), \text{ or,}$$

$$c_i^{j+1} = \left(\frac{1}{2} + \frac{u\Delta t}{2\Delta x}\right)c_{i-1}^j + \left(\frac{1}{2} - \frac{u\Delta t}{2\Delta x}\right)c_{i+1}^j$$

Now if you use our simple stability analysis procedure, you will find that the Lax scheme is stable if the following stability criterion (referred to as the Courant condition) is met:

$$\frac{|u|\Delta t}{\Delta x} \le 1.$$

Let's look at the solution to our problem using the Lax scheme.

```
% Lax solution for the advection problem
c=zeros(1,21);       % initial conditions
c(1)=1;           % boundary condition
u=1; dx=2; dt=2;        % set velocity and grid steps
% note: solution for dt=dx/u=2 should preserve sharp front
for i=1:20
   c(2:20)=(0.5+u*dt/(2*dx))*c(1:19)+(0.5-u*dt/(2*dx))*c(3:21);
   cc(i,1:21)=c;
end
% repeat for dt=1.5
c=zeros(1,21);
c(1)=1;
dt=1.5;
for i=1:20
      c(2:20)=(0.5+u*dt/(2*dx))*c(1:19)+(0.5-u*dt/(2*dx))*c(3:21);
   cc2(i,1:21)=c;
end
% plot the profile at t=24 (12 steps for dt=2; 16 for dt=1.5)
z=0:dx:40;
```

```
plot(z,cc(12,:),'b',z,cc2(16,:),'r')
xlabel('Distance from injection point')
ylabel('Reduced concentration')
legend('dt=2','dt=1.5')
```

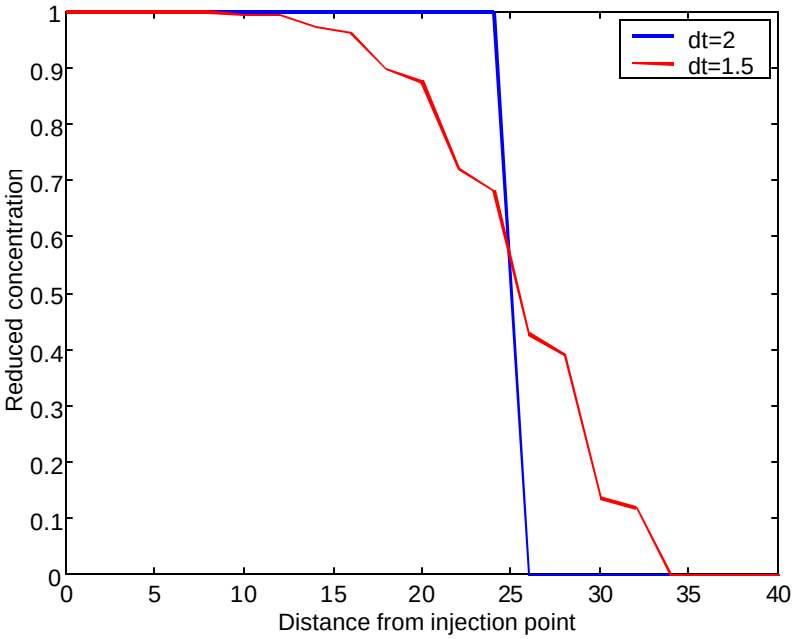

Figure 9.1. Lax solution for a simple advection problem for two time steps

Curiously, the solution for $\Delta t = 2$ is better than the solution for $\Delta t = 1.5$. What is going on? It turns out that the change we made to ensure stability was not totally benign. Let's consider our modification of equation (9.3) more carefully. The forward-in-time, central-in-space approximation that is unconditionally unstable is:

$$\frac{c_i^{j+1}}{\Delta t} = \frac{c_i^j}{\Delta t} - \left(\frac{u}{2\Delta x}\right)\left(c_{i+1}^j - c_{i-1}^j\right)$$

To achieve stability, we replaced the first term on the right-hand side with an average of the concentrations at the surrounding nodes. It turns out that this is equivalent to adding the term $\frac{1}{2}\frac{(\Delta x)^2}{\Delta t}\left(\frac{c_{i+1}^j - 2c_i^j + c_{i-1}^j}{(\Delta x)^2}\right)$ to the right-hand side of the equation. But note that the term that we had to add is a numerical approximation to a dispersion term equal to $\frac{1}{2}\frac{(\Delta x)^2}{\Delta t}\frac{\partial^2 c}{\partial x^2}$. That is, we introduced a *numerical* dispersion term.

Of course, the dispersion term in a differential equation is not exactly represented by the numerical approximation – recall that we truncated a Taylor series to get the approximation to the second derivative [Chapter 3.2]. The actual differential equation that we are approximating in using the Lax scheme can be written

$$\frac{\partial c}{\partial t} + u\frac{\partial c}{\partial x} = \frac{1}{2}u\Delta x\left(\frac{1}{\mathbf{Cr}} - \mathbf{Cr}\right)\frac{\partial^2 c}{\partial x^2} + \cdots$$

where $\mathbf{Cr}$ is the Courant number ($\mathbf{Cr} = u\Delta t / \Delta x$) and the ellipses (…) represent higher order terms in the equation. Now it is clear why the solution for Δt = 2 is good – the Courant number is exactly unity so the dispersion term is knocked out of the equation above. The solution for Δt = 2 "moves" the front exactly one grid space in a time interval and all of the truncation errors magically cancel. In general, however, the only way we have to control numerical dispersion errors in solving transport problems is to use small time and space grid sizes. (There are some special ways to use finite-differences to control numerical dispersion – for example, the "method of characteristics" – but we will not consider these here.)

9.3. The advection-dispersion equation

With this bit of background on numerical dispersion, we return to the full advection-disperion equation.

$$\frac{\partial c}{\partial \hat{t}} + \frac{\partial c}{\partial \hat{x}} = \frac{1}{\mathbf{Pe}}\frac{\partial^2 c}{\partial \hat{x}^2}$$

A Crank-Nicolson approximation for this equation is:

$$\frac{c_i^{j+1} - c_i^j}{\Delta t} + \frac{1}{2}\left[\frac{c_{i+1}^{j+1} - c_{i-1}^{j+1}}{2\Delta x} + \frac{c_{i+1}^{j} - c_{i-1}^{j}}{2\Delta x}\right]$$
$$= \left(\frac{1}{\mathbf{Pe}}\right)\left(\frac{1}{2}\right)\left(\frac{c_{i+1}^{j+1} - 2c_i^{j+1} + c_{i-1}^{j+1}}{\Delta x^2} + \frac{c_{i+1}^{j} - 2c_i^{j} + c_{i-1}^{j}}{\Delta x^2}\right).$$

(Note that the Lax method is unstable when the dispersion term is included in the equation.) Let's look at a *MATLAB* implementation of this solution for a couple of grid spacings and compare the solution to an analytical solution for a semi-infinite domain. (The comparison will be valid as long as the concentration at the downstream end of the domain that we are simulating remains very close to the zero background concentration.)

```
% adv_disp.m
% Crank-Nicolson solution to advection-dispersion equation
% Solution for a Peclet number of 30, a distance of 4 dimensionless
% units and a total time of 1.5.
Pe=30; length=4; endtime=1.5;
%
% First do the solution for dx=0.2 (21 nodes) and dt=0.25.
nnodes=21; nsoln=nnodes-2; dx=length/(nnodes-1); dt=0.25;
ntimes=endtime/dt;
x1=0:dx:length;
cold=zeros(nnodes,1); cold(1)=1;  %set initial and boundary conditions
Mterm=1/dt+1/(Pe*dx^2);
Uterm=1/(4*dx)-1/(2*Pe*dx^2);
Lterm=-1/(4*dx)-1/(2*Pe*dx^2);
M_diag=sparse(1:nsoln,1:nsoln,Mterm,nsoln,nsoln);
U_diag=sparse(1:nsoln-1,2:nsoln,Uterm,nsoln,nsoln);
L_diag=sparse(2:nsoln,1:nsoln-1,Lterm,nsoln,nsoln);
```

```
A=M_diag+U_diag+L_diag;   %construct the left-hand-side matrix
Mterm=1/dt-1/(Pe*dx^2);
Uterm=-1/(4*dx)+1/(2*Pe*dx^2);
Lterm=1/(4*dx)+1/(2*Pe*dx^2);
M_diag=sparse(1:nsoln,1:nsoln,Mterm,nsoln,nsoln);
U_diag=sparse(1:nsoln-1,2:nsoln,Uterm,nsoln,nsoln);
L_diag=sparse(2:nsoln,1:nsoln-1,Lterm,nsoln,nsoln);
rhsmatrix=M_diag+U_diag+L_diag;   %construct the right-hand-side matrix
for j=1:ntimes
   rhs=rhsmatrix*cold(2:nnodes-1);   %calculate rhs vector
      %adjust rhs vector for boundary condition at x=0
   rhs(1)=rhs(1)+2*(1/(4*dx)+1/(2*Pe*dx^2))*cold(1);
   cnew=A\rhs;   %solve the equation
   cnew=[cold(1);cnew;cold(nnodes)];
   cold=cnew;
end
c1=cnew;
%
% now do the solution for 81 nodes and for dt=0.1
%
nnodes=81; nsoln=nnodes-2; dx=length/(nnodes-1); dt=0.1;
ntimes=endtime/dt;
x2=0:dx:length;
cold=zeros(nnodes,1); cold(1)=1;
Mterm=1/dt+1/(Pe*dx^2);
Uterm=1/(4*dx)-1/(2*Pe*dx^2);
Lterm=-1/(4*dx)-1/(2*Pe*dx^2);
M_diag=sparse(1:nsoln,1:nsoln,Mterm,nsoln,nsoln);
U_diag=sparse(1:nsoln-1,2:nsoln,Uterm,nsoln,nsoln);
L_diag=sparse(2:nsoln,1:nsoln-1,Lterm,nsoln,nsoln);
A=M_diag+U_diag+L_diag;
Mterm=1/dt-1/(Pe*dx^2);
Uterm=-1/(4*dx)+1/(2*Pe*dx^2);
Lterm=1/(4*dx)+1/(2*Pe*dx^2);
M_diag=sparse(1:nsoln,1:nsoln,Mterm,nsoln,nsoln);
U_diag=sparse(1:nsoln-1,2:nsoln,Uterm,nsoln,nsoln);
L_diag=sparse(2:nsoln,1:nsoln-1,Lterm,nsoln,nsoln);
rhsmatrix=M_diag+U_diag+L_diag;
for j=1:ntimes
   rhs=rhsmatrix*cold(2:nnodes-1);
   rhs(1)=rhs(1)+2*(1/(4*dx)+1/(2*Pe*dx^2))*cold(1);
   cnew=A\rhs;
   cnew=[cold(1);cnew;cold(nnodes)];
   cold=cnew;
end
c2=cnew;
%
% The analytical solution.
xanalyt=0:0.02:4;
arg1=(xanalyt-endtime)./(2*sqrt(endtime/Pe));
arg2=(xanalyt+endtime)./(2*sqrt(endtime/Pe));
o=ones(size(xanalyt));
canalyt=0.5*(o-erf(arg1)+exp(1/Pe)*(o-erf(arg2)));
plot(x1,c1,'bo-',x2,c2,'r+-',xanalyt,canalyt,'g')
xlabel('Dimensionless distance')
ylabel('Reduced concentration')
legend('coarse spacing','finer spacing','analytical solution')
```

The results of the computation (Figure 9.2) indicate that the effects of numerical dispersion are easily noticeable for the coarse spacing but appear to be greatly diminished for the finer spacing. One way to check for numerical dispersion is to halve the grid spacing and see if the answer remains the same. Note also the "overshoot" near the front for the coarse spacing. This, like the more subtle effects of numerical dispersion, is a feature that also is seen quite often in numerical solutions for problems in which advection is important.

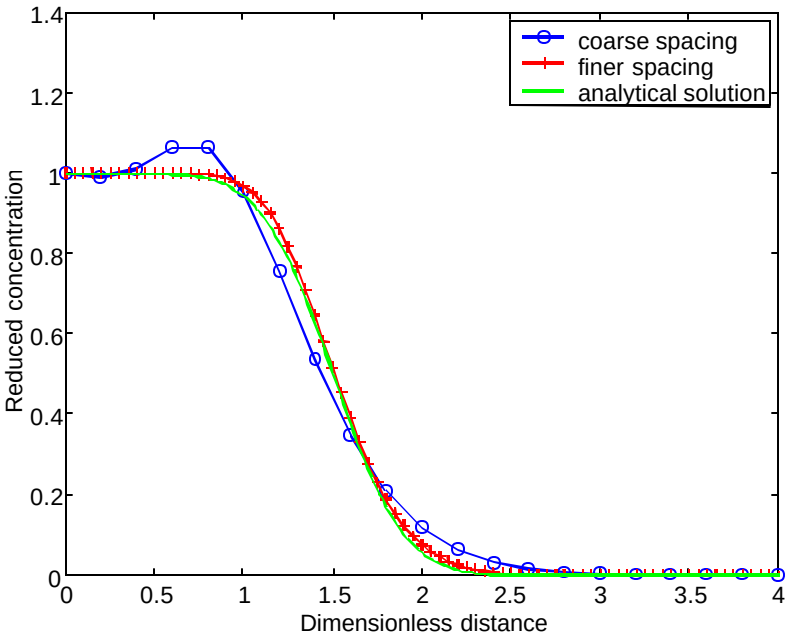

Figure 9.2. Results for numerical solution of the advection-dispersion equation.

9.4. Transport of reactive solutes

Often, we are concerned with solutes that interact with the mineral grains – of the aquifer if we are dealing with groundwater transport, of the streambed material if we are dealing with streamflow. To illustrate some of the embellishments of the methods that we have considered so far when applied to such transport problems, we will look at the transport of silica colloid through sand (Saiers et al., 1994). The interaction between the colloidal particles and the sand grains can be described using first-order kinetics. The transport equations are:

$$\frac{\partial c}{\partial t} + u\frac{\partial c}{\partial x} = D\frac{\partial^2 c}{\partial x^2} - k_f\, c + k_b\, s$$

$$\frac{\partial s}{\partial t} = k_f c - k_b\, s$$

where s is the amount of adsorbed constituent (equal to the bulk density of the aquifer material divided by the porosity times the adsorbed concentration expressed in mass of constituent per mass of mineral grains), and k_f and k_b are "forward" and "backward" rate coefficients, respectively, for the transfer of colloidal mass between the aqueous (c) and the adsorbed (s) phases. A fully implicit finite-difference approximation for these equations is:

$$\frac{c_i^{j+1} - c_i^j}{\Delta t} + u\frac{c_{i+1}^{j+1} - c_{i-1}^{j+1}}{2\Delta x} = D\frac{c_{i+1}^{j+1} - 2c_i^{j+1} + c_{i-1}^{j+1}}{\Delta x^2} - k_f c_i^{j+1} + k_b s_i^{j+1}$$

$$\frac{s_i^{j+1} - s_i^j}{\Delta t} = k_f c_i^{j+1} - k_b s_i^{j+1}$$

The way to picture the formulation of (and hence the solution to) this problem in matrix-vector notation is to note that we have twice as many unknowns now – the values of the c_i^{j+1} for the "regular" advection-dispersion problem and the s_i^{j+1} as well. To gain a solution, we again number the *c*'s sequentially, c_1 through c_n. Now, rather than doing the same for the *s*'s, we let the first *s* be c_{n+1}, the second *s* be c_{n+2}, and so on so that the n^{th} *s* is c_{2n}. Our vector of unknowns is then

$$c = [c_1, c_2, ..., c_n, c_{n+1}, c_{n+2}, ..., c_{2n}]'$$

where the second *n* elements are the unknown values of the sorbed concentration.

How does the coefficient matrix look for our approach? Consider the matrix to be blocked into $n \times n$ submatrices. The upper part of the matrix is for the equations for the aqueous concentrations. The diagonal elements in the upper left block (M in matrix below) are the coefficients on c_i^{j+1}: $1/\Delta t + 2D/\Delta x^2 + k_f$. The upper diagonal elements (U) are the coefficients on c_{i+1}^{j+1}: $u/2\Delta x - D/\Delta x^2$. Likewise the lower diagonal (L) contains the coefficients on the c_{i-1}^{j+1}: $-u/2\Delta x - D/\Delta x^2$. The matrix block in the upper right (r) contains the coefficients on the *s*'s in the equations for the aqueous concentrations. The diagonal elements (for the *block*) are the coefficients on s_i^{j+1}, $-k_b$.

M U L M U L M U L M U L M U L M U L M U L M U L M U L M	r r r r r r r r r r
d d d d d d d d d d	m m m m m m m m m m

The lower half of the matrix is for the equations for the adsorbed concentrations. The diagonal elements in the right, lower block (m) are $1/\Delta t+k_b$, and the coefficients in the lower left block (d) are $-k_f$. The right-hand-side vector is composed of the appropriate "known" quantities involving the j time step.

The solution in which we are interested is the breakthrough curve at the end of a column of sand. Initial conditions are zero concentration everywhere. The top boundary condition is a reduced concentration of 1 during pulse injection and a reduced concentration of zero thereafter. The bottom boundary condition is subtler. Flow occurs out of the bottom and the concentration changes with time. Thus, the bottom boundary condition is neither a fixed concentration nor a specified flux. The appropriate condition is that the flux occurs only by advection, i.e. dispersion=0. This implies that $\partial^2 c / \partial x^2 = 0$ (since $D \neq 0$) or, in terms of finite differences, $c_{n+1} - 2c_n + c_{n-1} = 0$. Solving for the concentration at the fictitious "n+1" node gives $c_{n+1} = 2c_n - c_{n-1}$.

```
% kinsorp.m  Implicit solution to advection-dispersion equation with
% kinetic sorption
% Solution for u=15.38cm/h, D=2.43cm^2/h, L=14.5cm, kf=0.13h^-1,
% kb=1.08h^-1, and a pulse input of 2.28 h (2.42 pore volumes).
v=15.38; length=14.5; D=2.43; kf=0.13; kb=1.08; pulse=2.28; endtime=5.0;
%
% do the solution for 49 nodes and for dt=0.02h.
%
nnodes=49; nsoln=nnodes; dx=length/(nnodes-1); dt=0.02;
ntimes=endtime/dt; npulse=pulse/dt;
ntotal=2*nsoln;
time=0:dt:endtime-dt;
time=time/0.94;  %correction specific to this data set
cbottom=zeros(size(time));
c=zeros(nnodes,1);c(1)=1;
s=zeros(nnodes,1);
cold=[c;s];
%
% Set the coefficient matrix
Mterm=1/dt+2*D/(dx^2)+kf;
Uterm=v/(2*dx)-D/(dx^2);
Lterm=-v/(2*dx)-D/(dx^2);
M_diagUL=sparse(1:nsoln,1:nsoln,Mterm,ntotal,ntotal);
U_diagUL=sparse(1:nsoln-1,2:nsoln,Uterm,ntotal,ntotal);
L_diagUL=sparse(2:nsoln,1:nsoln-1,Lterm,ntotal,ntotal);
M_diagUR=sparse(1:nsoln,nsoln+1:ntotal,-kb,ntotal,ntotal);
M_diagLR=sparse(nsoln+1:ntotal,nsoln+1:ntotal,1/dt+kb,ntotal,ntotal);
M_diagLL=sparse(nsoln+1:ntotal,1:nsoln,-kf,ntotal,ntotal);
A=M_diagUL+U_diagUL+L_diagUL+M_diagUR+M_diagLR+M_diagLL;
% Bottom boundary
A(nnodes,nnodes)=1/dt+v/dx+kf;
A(nnodes,nnodes-1)=-v/dx;
% Set the right hand side
rhs=cold./dt;
% Do the solution
for j=1:ntimes
   % top boundary
      if j<=npulse
        rhs(1)=rhs(1)+v/(2*dx)+D/(dx^2);
```

```
        end
    cnew=A\rhs;
    cold=cnew;
    rhs=cold./dt;
    cbottom(j)=cnew(nsoln);
end
load silica.dat;
plot(time,cbottom,'b',silica(:,1),silica(:,2),'+r')
xlabel('Dimensionless time, (pore volumes)')
ylabel('Reduced concentration')
legend('model','data')
```

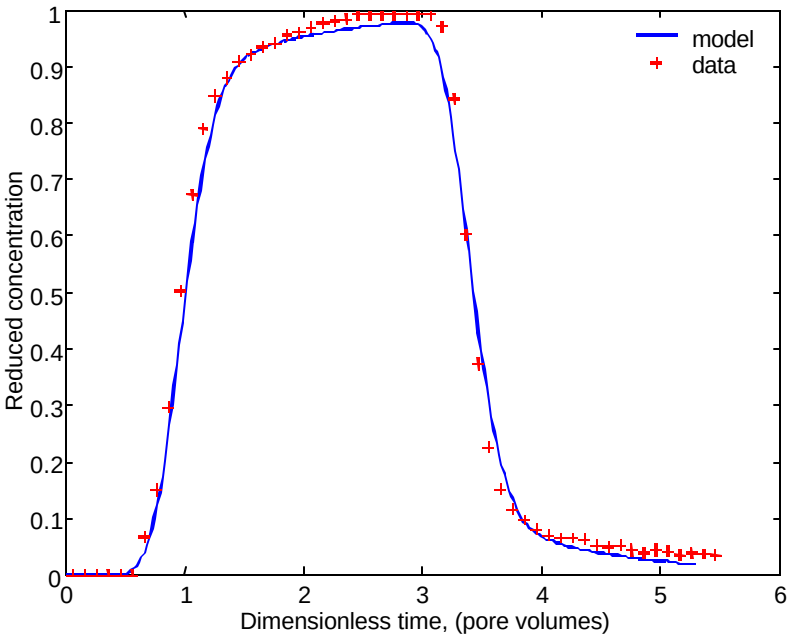

Figure 9.3. Breakthrough curve for example.

9.5. Problems

1. The transport of solute in a stream flowing over alluvium can be approached by considering the water velocity in the alluvium to be negligible with respect to that in the stream and by considering the exchange of solutes between the stream and the groundwater in the sediments to be governed by a simple first-order expression. (If you need to read up on the notion of the "transient-storage" model, you can start with the paper by Bencala and Walters, 1983.) The equations to solve (for a somewhat more simplified representation of the channel and the flow processes than that given by Bencala and Walters) are

$$\frac{\partial c}{\partial t} + u\frac{\partial c}{\partial x} = D\frac{\partial^2 c}{\partial x^2} + \alpha\,(s - c)$$

$$\frac{ds}{dt} = -\alpha\,\frac{A}{A_s}(s - c)$$

where c represents concentration in the stream, s represents concentration in the substream sediments (the "hyporheic zone"), α is a first-order exchange coefficient, A is the stream cross-sectional area , and A_s is the cross-sectional area of the storage zone.

Write a code to solve the transient-storage-zone model. Apply the code to the stream at Shaver Hollow, Virginia using the data below (from Castro and Hornberger, 1991). [To get into the ballpark with the parameter values that you will need, note that Bencala and Walters obtained values of α in the 10^{-5} per second range, D on the order of 0.1 $m^2\ s^{-1}$, and A/A_s in the range of 0.3 – 1 for Uvas Creek, a small gravel-bed stream; other data indicate that A/A_s is larger for larger streams and rivers (Bencala and Walters, 1983).]

Data below are for bromide concentrations at a station 131 m downstream of the injection point. At the injection point, concentrations in the stream were maintained approximately constant at 47.8 mg L^{-1} from 1015 on 2 Aug 1988 through 0700 on 6 Aug 1988. Concentrations at the injection site were zero thereafter.

Day of August 1988	*Time*	*Bromide Concentration* (mg L^{-1})
2	1406	1.67
2	1506	4.96
2	1606	7.60
2	1704	10.38
2	1806	12.61
2	1906	14.37
2	2058	16.82
2	2214	18.92
2	2359	21.29
3	0206	23.04
3	0413	24.92
3	0613	25.92
3	0802	25.92
3	1204	28.04
3	1609	28.04
3	2016	31.55
4	0800	34.13
4	1155	31.55

4	1602	30.34
4	1955	32.81
5	0818	36.92
5	1202	34.13
5	1558	34.13
5	2004	35.50
6	0640	38.41
6	0730	36.92
6	0829	38.41
6	0956	37.31
6	1156	34.51
6	1409	27.30
6	1616	15.81
6	1815	9.89
7	0823	3.39
7	1147	2.66
7	1628	2.18
8	0825	1.64
9	0712	0.99
12	0825	0.52
20	0920	0.14

9.6. References

Bencala, K.E. and R.A. Walters, Simulation of solute transport in a mountain pool and-riffle stream: A transient storage model, *Water Resour. Res., 19*: 718-724, 1983.

Castro, N.M. and G.M. Hornberger, Surface-subsurface water interactions in an alluviated mountain stream channel, *Water Resour. Res., 27*: 1613-1621, 1991.

Fischer, H.B., E.J. List, R.C.Y. Koh, J. Imberger, and N.H. Books, *Mixing in Inland and Coastal Waters*, 483 pp., Academic Press, New York, 1979.

Hemond, H.F. and E.J. Fechner-Levy, *Chemical Fate and Transport in the Environment*, 433 pp., Academic Press, San Diego, 2000.

Hornberger, G.M., Mills, A.L., and J.S. Herman, Bacterial transport in porous media: evaluation of a model using laboratory observations, *Water Resour. Res., 28*: 915-938, 1992.

Saiers, J.E., Hornberger, G.M., and C. Harvey, Colloidal silica transport through structured, heterogeneous porous media, *J. Hydrology 163*: 271-288, 1994.