CHAPTER 10

The Finite Element Method: An Introduction

Numerical Methods in the Hydrological Sciences
Special Publication Series 57

doi:10.1029/057SP15

10. The Finite Element Method: An Introduction

10.1. Background

Finite-difference methods for solving differential equations are attractive for a number of reasons. First, they are conceptually straightforward – they “make sense”. Second, the equations are relatively easy to derive, at least for a regular mesh. Third, programming to obtain a solution is not too difficult.

Finite-difference methods have some serious drawbacks, however. First, it often is desirable to have an irregular mesh (unequally spaced nodes). The finite difference equations are not easy to write in general terms under this condition. Second, it is quite cumbersome to handle irregular boundaries using finite differences. And finally, anisotropy is not easy to incorporate into finite-difference methods.

A popular numerical method that overcomes the drawbacks of finite-difference methods (admittedly at the expense of some of the attractive features of the method) is the finite-element method. In the finite-element method, the solution to the differential equation is approximated as a continuous function of the independent variables, as opposed to being approximated at only a discrete number of points, the nodal points of a mesh. The presentation below starts with the method of collocation to illustrate the idea of approximation with a continuous function and then goes on to the finite-element method itself.

10.2. Collocation

The basic idea for the method of collocation is to consider some functional approximation to the solution to a differential equation and then find coefficients in the approximate function to make it "close" to the actual solution. For example, we can consider the approximating function to be a polynomial of degree n with the $(n+1)$ coefficients selected to match the boundary conditions and to fit the solution as well as possible. An illustrative example should help clarify the idea.

Consider the problem of groundwater flow between two drains (Figure 10.1). With the Dupuit assumptions[1], the equation describing the problem is (e.g., see Fetter 2001):

$$
\begin{aligned}
&\frac{d}{dx}\left[Kh\frac{dh}{dx}\right] = -w \\
&h(0) = h_1 \qquad\qquad (10.1)\\
&h(L) = h_2
\end{aligned}
$$

where K is hydraulic conductivity and w is recharge rate. To make the problem even more concrete, let h_1=5 m, h_2=10 m, L=100 m, K=10 cm d^{-1}, and w=0.15 cm d^{-1}.

[1] The Dupuit assumptions are used to simplify the problem of flow in an unconfined aquifer. Basically, the assumptions are that the vertical component of flow in a horizontal aquifer is negligible – that the head gradient is the slope of the water table and that flow is horizontal.

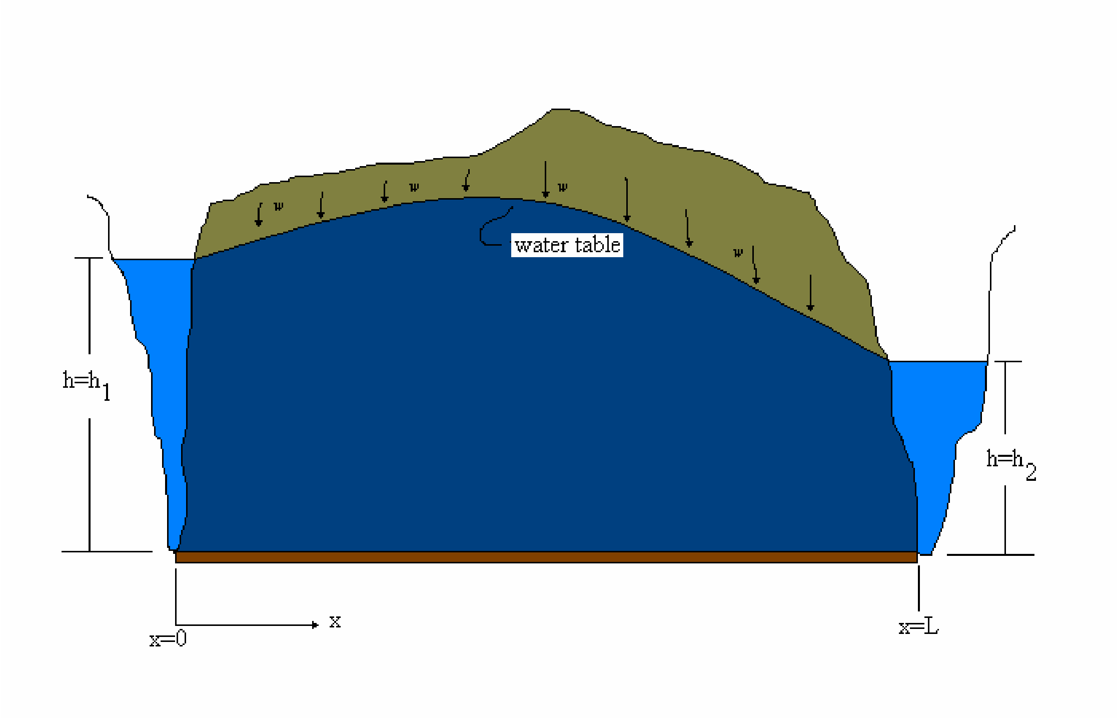

Figure 10.1. Schematic of groundwater flow between two drains.

Choose a second-degree polynomial to represent the true solution. The polynomial can be made to satisfy the boundary conditions by judicious choice of two of the coefficients.

$$\hat{h} = 5 + 0.05x + Cx(x - 100) \qquad (10.2)$$

where C is a constant at our disposal to fit the approximate solution to the true solution. First, express the left side of (10.1) in terms of the trial function (10.2).

$$\hat{h} = 5 + 0.05x + Cx(x - 100) = 5 + (0.05 - 100C)x + Cx^2$$

$$\frac{d\hat{h}}{dx} = (0.05 - 100C) + 2Cx$$

$$K\hat{h}\frac{d\hat{h}}{dx} = K\left[5 + (0.05 - 100C)x + Cx^2\right]\left[(0.05 - 100C) + 2Cx\right]$$

$$= K\left\{5(0.05 - 100C) + \left[(0.05 - 100C)^2 + 10C\right]x + 3C(0.05 - 100C)x^2 + 2C^2x^3\right\}$$

$$\frac{d}{dx}\left[K\hat{h}\frac{d\hat{h}}{dx}\right] = K\left\{\left[(0.05 - 100C)^2 + 10C\right] + 6C(0.05 - 100C)x + 6C^2x^2\right\}$$

For the exact solution to the differential equation, the derivative that we just calculated is equal to -w for all values of x. Because $\hat{h}$ is approximate, there will be a *residual* between the derivative and w for at least some values of x:

$$residual = r = -w - K\left\{\left[(0.05-100C)^2 + 10C\right] + 6C(0.05-100C)x + 6C^2x^2\right\}$$

The objective of the method of collocation is to make the residual as close to zero as possible. With one free coefficient at our disposal, we can select one value of x at which to make the residual equal zero exactly. A logical choice might be at $x=50$, halfway between the drains. This choice leads to the following solution.

$$\begin{aligned}
&-w - K\left\{\left[(0.05-100C)^2 + 10C\right] + 6C(0.05-100C)50 + 6C^2(50)^2\right\} = 0\\
&-w/K - \left\{\left[(0.05-100C)^2 + 10C\right] + 6C(0.05-100C)50 + 6C^2(50)^2\right\} = 0\\
&-0.0175 - 15C + 5000C^2 = 0\\
&C = \text{-}8.9792\text{x}10^{-4}.
\end{aligned}$$

[Note that there are two solutions for C (the equation is quadratic) but only the one that makes sense for the problem at hand is presented. Also note that reducing the residual at x=50 to zero is *not* the same as reducing the difference between approximate and actual values of h at x=50 to zero.]

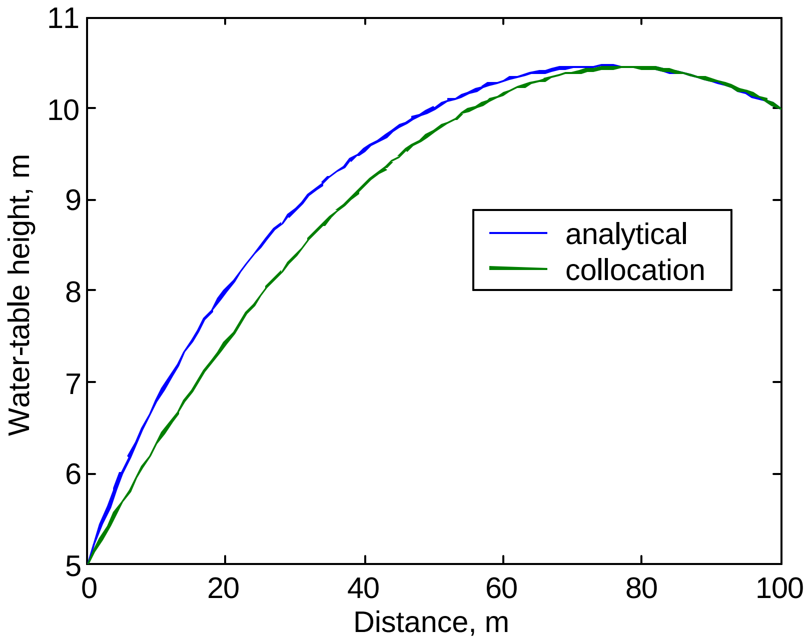

Figure 10.2. Approximate solution to the groundwater-drain problem determined using collocation compared with the analytical solution.

The approximate solution is "similar" to the analytical solution[2] (Figure 10.2), but quite possibly not as good as we might want. In the method of collocation, a better fit can be obtained by using a higher order polynomial as the trial function. For example, if we used a cubic equation, we would

[2] The *MATLAB* symbolic toolbox can be used to derive the analytical solution. Letting u be dh/dx, the problem can be represented as two differential equations, h*du/dx+u^2=-w/K and dh/dx=u. The solution is obtained by the statement:

```
[h,u]=dsolve('h*Du+u^2=-.015','Dh=u','h(0)=5','h(100)=10','x').
```

have had two free coefficients to fit and could have chosen two points in the domain to match the differential equation exactly.

The approximate solution with the second-degree polynomial would have been much better for the example problem had the drains been only 10 m apart and the head difference between them only 0.5 m. Given the stated problem, however, increased accuracy requires a more complex approximating polynomial and, consequently, a solution to more complicated equations. The finite-element method overcomes the problems of collocation by breaking the domain into a number of elements and using simple polynomials to approximate the solution over each "small" element.

10.3. Weighted residual method

In the method of collocation applied to the example problem, the residual between the approximate solution and the right-hand side of the differential equation was minimized (set to zero) at one point in the domain. A general method for determining a good approximation is to reduce the integral of the residuals over the domain to zero. In fact, in the general method of *weighted residuals*, the residuals are multiplied by some function $W(x)$ (the weighting function) and the integral of the product is set to zero. For the example problem, we would require:

$$\int_0^{100} W(x) r(x) dx = 0 \tag{10.3}$$

The method of collocation as we applied it is a special case of the weighted residual method with the weighting function set to the Dirac delta function, a function that "selects" the collocation points ($x=50$ in the example) from the integral and sets the residual at these points to zero.

10.4. The finite element approach: Galerkin weighted residual method

Consider again the example problem of groundwater flow between drains. This time, we will simplify the problem by considering the linearized version – i.e., by replacing the term Kh with a constant transmissivity, T, which we will take to be 80 m^{2}d^{-1}.

$$\frac{d}{dx}\left[T\frac{dh}{dx}\right] = T\frac{d^2h}{dx^2} = -w$$

or,

$$\begin{aligned} &T\frac{d^2h}{dx^2} = -w = -0.15cm/day \\ &h(0) = 5m \\ &h(100) = 10m \end{aligned} \tag{10.4}$$

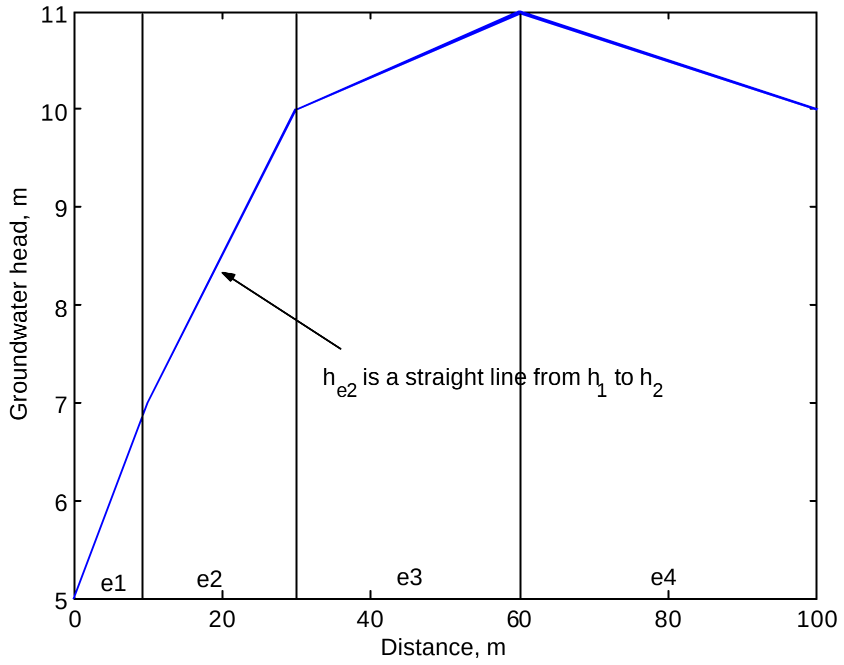

Figure 10.3. The domain is broken into four elements, e1 through e4. The groundwater head is approximated by a straight line over each element, with the endpoints denoted by the heads at the nodes, h_i.

The first step in the finite-element method is to divide the domain into *elements*, in this case four line segments (Figure 10.3). The elements are bounded by *nodes*, the endpoints of the line segments. The next step is to approximate the solution over each element. The simplest approximating polynomials are straight lines, which can be expressed as:

$$
\begin{aligned}
h_{e1} &= h_1\left(\frac{x_2 - x}{x_2 - x_1}\right) + h_2\left(\frac{x - x_1}{x_2 - x_1}\right) \\
h_{e2} &= h_2\left(\frac{x_3 - x}{x_3 - x_2}\right) + h_4\left(\frac{x - x_2}{x_3 - x_2}\right) \\
h_{e3} &= h_3\left(\frac{x_4 - x}{x_4 - x_3}\right) + h_4\left(\frac{x - x_3}{x_4 - x_3}\right) \\
h_{e4} &= h_4\left(\frac{x_5 - x}{x_5 - x_4}\right) + h_5\left(\frac{x - x_4}{x_5 - x_4}\right)
\end{aligned}
\qquad (10.5)
$$

It is easy to check that equations (10.5) are straight lines connecting the heads at the two endpoints of an element.

The straight-line approximation for an element can be written in terms of general “basis functions”, $x(x)$, defined for each element. The i^{th} basis function has a value of one at node i and drops linearly to zero at the adjacent nodes (Figure 10.4). The basis functions are written explicitly as functions of x as follows.

$$x_i = \begin{cases} \dfrac{x_{i+1} - x}{x_{i+1} - x_i}, & \text{for } x_i \le x \le x_{i+1} \\[2ex] \dfrac{x - x_{i-1}}{x_i - x_{i-1}}, & \text{for } x_{i-1} \le x \le x_i \end{cases} \tag{10.6}$$

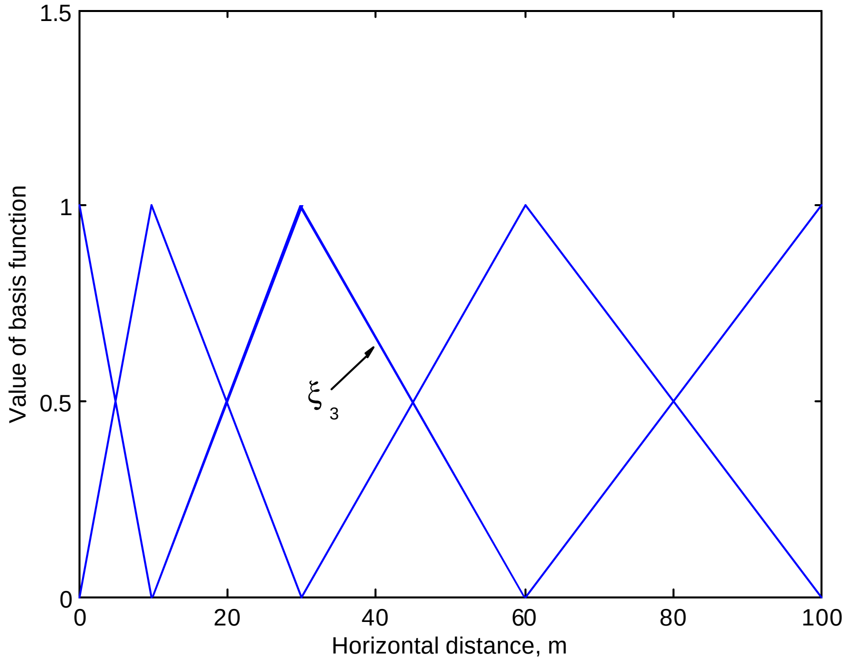

Figure 10.4. The basis functions (“chapeau functions”) for the example problem. There is a basis function for each node. The function for node three is shown by the bold line.

The approximations for the groundwater heads in terms of the basis functions are:

$$\hat{h}_{ei} = h_i x_i + h_{i+1} x_{i+1} \tag{10.7}$$

where i represents the i$^{\text{th}}$ element. (Compare equation 10.7 with equation 10.5 given the definition of the x_i in equation 10.6.) The expression for h over the entire domain is then:

$$\hat{h} = \sum_k x_k h_k \tag{10.8}$$

The next step is to substitute the approximate function for h into the governing equation and form the residuals as we did in section 10.2.

$$residual = r = T\frac{d^2\hat{h}_e}{dx^2} + w$$

This residual is weighted and integrated over the domain of h as suggested in equation (10.3). The Galerkin method uses the basis functions themselves as the weighting functions. Although this choice may seem arbitrary, it turns out that it can be shown that use of the basis functions as weights is "best" in a sense. The specific form for equation 10.3 for the example problem is then:

$$\int_R x\left(T\frac{d^2\hat{h}_e}{dx^2} + w \right)dx = 0 \qquad (10.9)$$

where R is the domain over which the integration is carried out. The second derivative is eliminated from equation (10.9) by integrating by parts[3]. Let u be x and dv be $T\frac{d^2\hat{h}_e}{dx^2}dx$ (giving $v = T\frac{dh_e}{dx}$). Equation (10.9) can be written:

$$xT\frac{d\hat{h}_e}{dx}\bigg|_0^{100} - \int_0^{100}\frac{dx}{dx}T\frac{d\hat{h}_e}{dx}dx + \int_0^{100} xwdx = 0 \qquad (10.10)$$

We started our solution to the problem by dividing the domain into a number of elements. Instead of integrating across the entire domain R – from $x=0$ to $x=100$ – it makes more sense to integrate over the elements and then sum them to get the total integral. That is, we can rewrite equation (10.10) as:

$$\sum_e \left\{ x_e T_e \frac{d\hat{h}_e}{dx}\bigg|_{x_i}^{x_{i+1}} - T_e \int_{x_i}^{x_{i+1}} \frac{dx_e}{dx}\frac{d\hat{h}_e}{dx}dx \right\} + \int_{x_i}^{x_{i+1}} xwdx = 0 \qquad (10.11)$$

where the e subscripts refer to elements. Note that in this particular example T_e is constant, but even if it were variable, it could be approximated as constant over an element and still be brought outside the integral for the element. In such a case, the coefficients derived would vary from element to element as T_e varied.

Next we evaluate equation (10.11) for a single element. First, note that the first term on the left-hand side of (10.11) is simply the total water flow into the element at x_i ($Q_i = T\frac{dh}{dx}\Big|_{x=x_i}$) minus the flow out of the element at x_{i+1} (Q_{i+1}). Now consider the second integral in equation (10.11). For the element, there are two portions of the basis functions to consider (Figure 10.5). For x_i, the second integral on the left-hand side of (10.11) reads:

$$T_e\int_{x_i}^{x_{i+1}} \frac{dx_i}{dx}\frac{d}{dx}\left(h_i x_i + h_{i+1}x_{i+1}\right)dx = T_e\int_{x_i}^{x_{i+1}} \left(\frac{dx_i}{dx}\frac{dx_i}{dx}h_i + \frac{dx_i}{dx}\frac{dx_{i+1}}{dx}h_{i+1}\right)dx \qquad (10.12)$$

and for x_{i+1} it is:

[3] $\int udv = uv - \int vdu.$

$$T_e \int_{x_i}^{x_{i+1}} \frac{dx_{i+1}}{dx}\frac{d}{dx}\left(h_i x_i + h_{i+1}x_{i+1}\right)dx = T_e \int_{x_i}^{x_{i+1}} \left(\frac{dx_{i+1}}{dx}\frac{dx_i}{dx}h_i + \frac{dx_{i+1}}{dx}\frac{dx_{i+1}}{dx}h_{i+1}\right)dx \quad (10.13)$$

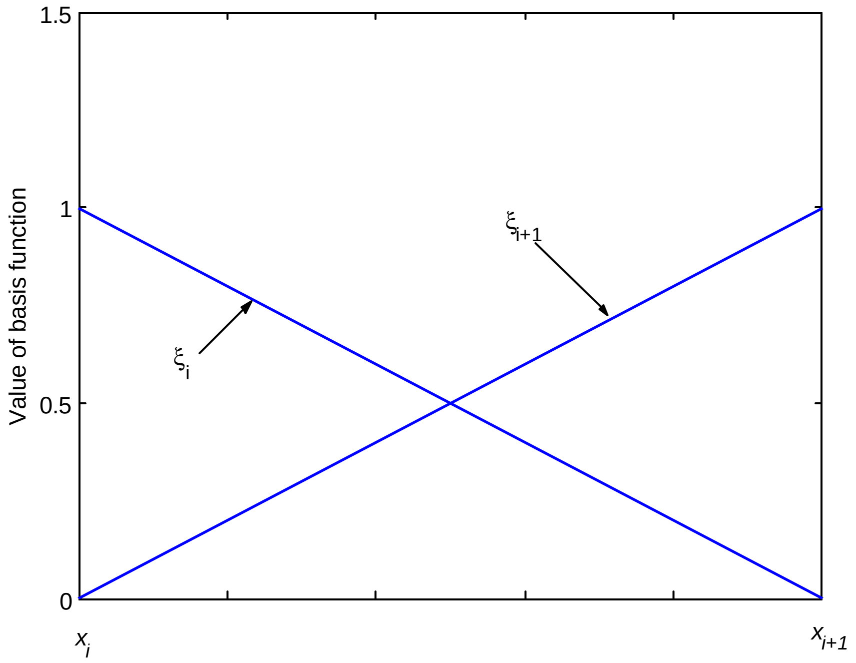

Figure 10.5. The two segments of the basis function for a general element (cf. Figure 10.4).

The integral can be evaluated easily once we find the derivatives of the basis functions with respect to x. From equation (10.6) it is clear that:

$$\frac{dx_i}{dx} = \frac{1}{(x_{i+1} - x_i)}, \quad \frac{dx_{i+1}}{dx} = \frac{-1}{(x_{i+1} - x_i)}$$

Thus, the integrands in equations (10.12) and (10.13) are not functions of x and the necessary evaluation is simply the integral of dx.

The last integral on the left-hand side of equation (10.11) is easily evaluated. Consider the integral for x_i and note that we can use equation (10.6) to make a substitution to convert the integral with respect to x to an integral with respect to x_i. In making the substitution, the limits of integration change to 0 to 1 (when $x=x_i$, $x_i = 0$) and dx is replaced by $\left(x_{i+1} - x_i\right)dx_i$.

$$w\int_{x_i}^{x_{i+1}} \frac{x - x_i}{x_{i+1} - x_i}\,dx = w\int_0^1 x_i \left(x_{i+1} - x_i\right)dx_i$$

$$= w\left(x_{i+1} - x_i\right)\frac{x_i^2}{2}\Bigg|_0^1$$

$$= \frac{w\left(x_{i+1} - x_i\right)}{2}$$

The integration for x_{i+1} is similar.

Carrying out all of the integrations, the equations corresponding to (10.11) [treating (10.12) and (10.13) separately] become:

$$\begin{aligned} &Q_i - T_e\left\{\frac{h_i}{\left(x_{i+1}-x_i\right)^2} - \frac{h_{i+1}}{\left(x_{i+1}-x_i\right)^2}\right\}\left(x_{i+1}-x_i\right) + w\left(x_{i+1}-x_i\right)/2 = 0 \\ &-Q_{i+1} - T_e\left\{\frac{-h_i}{\left(x_{i+1}-x_i\right)^2} + \frac{h_{i+1}}{\left(x_{i+1}-x_i\right)^2}\right\}\left(x_{i+1}-x_i\right) + w\left(x_{i+1}-x_i\right)/2 = 0 \end{aligned} \qquad (10.14)$$

Now note that when the sum is done for all elements, the outflow from one element is equal to the inflow to the next. Thus, the "*Q*'s" for all internal nodes cancel and these terms can be dropped. For the example problem, we do not have flow boundary conditions, so the *Q*'s can be dropped there as well. The equations for a single element can be written in matrix-vector form.

$$\begin{pmatrix} 1/\Delta_e & -1/\Delta_e \\ -1/\Delta_e & 1/\Delta_e \end{pmatrix}\begin{pmatrix} h_i \\ h_{i+1} \end{pmatrix} = \begin{pmatrix} w\Delta_e/2T_e \\ w\Delta_e/2T_e \end{pmatrix} \qquad (10.15)$$

where Δ_e is the element size, in this case $\left(x_{i+1}-x_i\right)$.

The finite-element procedure is completed by assembling all of the element matrices – the leftmost matrix in equation (10.15) – into a "global" matrix and solving the resulting equations for the nodal values of head. The global matrix for our 4-element example will be 3 x 3 because the heads at each end of the domain are fixed. (For the first element, the first row and first column of the element matrix disappear because the head is known. The known value of head at node 1 is transposed to the right-hand side of the equation for the first element. Likewise for the last element, the second row and column disappear.) The contribution from the first element to the global matrix is:

$$\begin{pmatrix} 1/\left(x_2-x_1\right) & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

The contribution from the second element is:

$$\begin{pmatrix} 1/\left(x_3-x_2\right) & -1/\left(x_3-x_2\right) & 0 \\ -1/\left(x_3-x_2\right) & 1/\left(x_3-x_2\right) & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

$$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 1/(x_4-x_3) & -1/(x_4-x_3) \\ 0 & -1/(x_4-x_3) & 1/(x_4-x_3) \end{pmatrix}$$

and

$$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1/(x_5-x_4) \end{pmatrix}$$

The global matrix is simply the sum of the element matrices. In the same fashion, the right-hand vector is assembled by adding contributions from each element. Also, the known values of head at the endpoints of the domain are transfered to the right-hand vector. The resulting equations follow.

$$\begin{pmatrix} 1/(x_2-x_1)+1/(x_3-x_2) & -1/(x_3-x_2) & .0 \\ -1/(x_3-x_2) & 1/(x_3-x_2)+1/(x_4-x_3) & -1/(x_4-x_3) \\ 0. & -1/(x_4-x_3) & 1/(x_4-x_3)+1/(x_5-x_4) \end{pmatrix}\begin{pmatrix} h_2 \\ h_3 \\ h_4 \end{pmatrix}=$$

$$\begin{pmatrix} w(x_2-x_1)/2T_1+w(x_3-x_2)/2T_2+5/(x_2-x_1) \\ w(x_3-x_2)/2T_2+w(x_4-x_3)/2T_3 \\ w(x_4-x_3)/2T_3+w(x_5-x_4)/2T_4+10/(x_5-x_4) \end{pmatrix}$$

(The boundary conditions have been placed in the first and last equations as appropriate. Alternatively, we could have kept 5 equations and simply made the first and fifth simple statements of the boundary conditions – e.g., h_1=5.)

To obtain the numerical solution, assume that transmissivity is constant at 0.8 m^{2}day^{-1}. The finite-element equations are then:

$$\begin{pmatrix} \frac{1}{10}+\frac{1}{20} & -\frac{1}{20} & 0 \\ -\frac{1}{20} & \frac{1}{20}+\frac{1}{30} & -\frac{1}{30} \\ 0 & -\frac{1}{30} & \frac{1}{30}+\frac{1}{40} \end{pmatrix} \begin{pmatrix} h_2 \\ h_3 \\ h_4 \end{pmatrix} = \begin{pmatrix} 0.0281+0.5 \\ .0469 \\ .0656+0.25 \end{pmatrix}$$

The numerical solution is an excellent approximation to the exact solution[4] (Figure 10.6), even with the fairly coarse mesh.

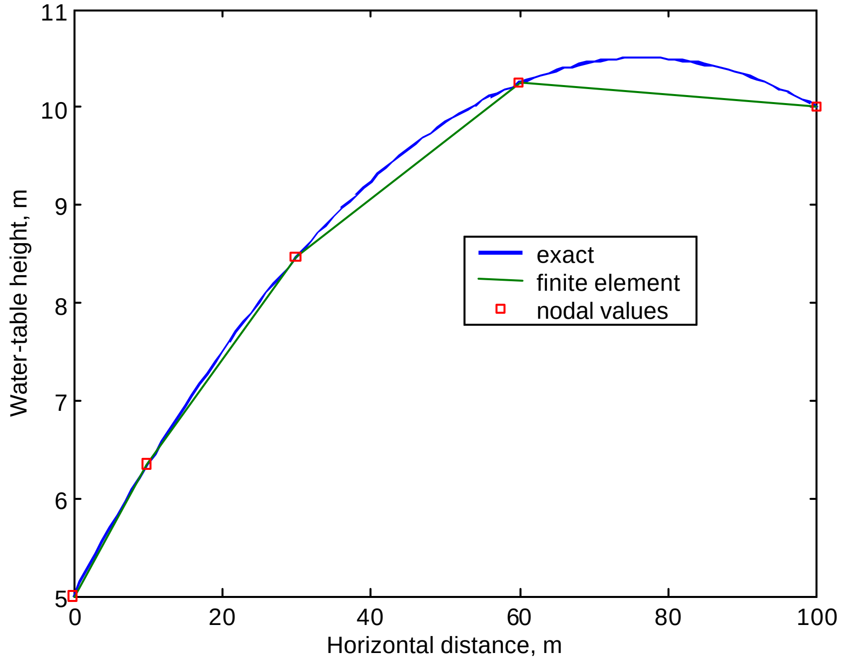

Figure 10.6. Finite element solution to the example problem.

10.5. Steady diffusion into sediment

As an example that demonstrates how variable properties, as well as unequal element size, can be accommodated in the finite-element method, consider the diffusion of oxygen into the sediment at the bottom of a lake or estuary. The equation used is a combination of Fick's law of diffusion and conservation of mass – much the same as the groundwater equation is a combination of Darcy's law and conservation of mass (see Boudreau, 1997). For steady-state conditions, the equation to solve is:

$$\frac{d}{dz}\left(fD\frac{dc}{dz}\right)+R=0. \tag{10.16}$$

[4] The *MATLAB* statement to obtain the exact solution is:

```
h=dsolve('D2h=-0.0015/0.8','h(0)=5','h(100)=10','x').
```

where f is the sediment porosity, *D* is the diffusion coefficient of oxygen in the pore water (corrected for tortuosity), and *R* is the rate of oxygen consumption by reactions in the sediment. Extension of the method to conditions other than those assumed here is straightforward [Box 10.1].

Consider a 0.005-m thick layer of sediment with $z=0$ at depth and $z=0.005$ m at the sediment surface. The flux of oxygen into the sediment from the overlying water is 0.0125 n mol cm^{-2} s^{-1}. The oxygen concentration at the 0.005m (*z*=0) depth is zero. The conditions are given mathematically by:

$$c(0) = 0$$

$$J = fD\left.\frac{dc}{dz}\right|_{z=0.005} = 0.0125 \text{ n mol cm}^{-2}\text{s}^{-1}$$

The diffusion coefficient, *D* is the product of the free-water diffusion coefficient, D_{fw} (=11.6 ×10^{-6} cm^2 s^{-1} for this problem) and the square of the sediment porosity. That is, $D = f^2 D_{fw}$. The porosity, f, varies with depth:

Depth interval (cm) below surface	*z* at bottom of interval (cm)	Porosity	Rate of oxygen consumption n mol cm^{-3} s^{-1}
0-0.05	0.45	0.9	0.02
0.05-0.1	0.4	0.85	0.07
0.1-0.2	0.3	0.8	0.04
0.2-0.3	0.2	0.75	0.02
0.3-0.5	0	0.7	0.01

We proceed as above to define the basis functions, do the integrations, and so forth, assuming that *R* and f vary from element to element but are constant within any given element. The finite-element equations are very similar to those for the groundwater problem (the equations are nearly identical.). There are some differences. First, the top boundary (the *n*th nodal value) condition is a *flux* condition. Look at equation (10.14). The “natural” boundary condition for the finite-element method *is* a flux condition. Thus, all that needs to be done is to place the flux on the right-hand side of the equation. Also, the porosity and diffusion coefficient vary and so are kept as part of the coefficient matrix. To simplify the notation somewhat, define l as $fD = f\left(f^2 D_{fw}\right) = f^3 D_{fw}$. The finite-element equations (i.e., the global matrix-vector equation) are:

$$\begin{bmatrix} 1 & & & & & \\ -l_1/\Delta_1 & l_1/\Delta_1 + l_2/\Delta_2 & -l_2/\Delta_2 & & & \\ & & \ddots & & & \\ & & & -l_{n-2}/\Delta_{n-2} & l_{n-2}/\Delta_{n-2} + l_{n-1}/\Delta_{n-1} & -l_{n-1}/\Delta_{n-1} \\ & & & & -l_{n-1}/\Delta_{n-1} & l_{n-1}/\Delta_{n-1} \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \bullet \\ \bullet \\ \bullet \\ c_{n-1} \\ c_n \end{bmatrix} = \begin{bmatrix} cbot \\ R_1\Delta_1/2 + R_2\Delta_2/2 \\ \\ \\ \\ R_{n-2}\Delta_{n-2}/2 + R_{n-1}\Delta_{n-1}/2 \\ R_{n-1}\Delta_{n-1}/2 - 0.0125 \end{bmatrix}$$

where the subscripted Δ's indicate the element lengths.

The solution in *MATLAB* is now easy to construct. The m-file below does the calculation and plots the results.

```
% diagen.m
% This is a finite element solution to calculate the steady state
% concentration distribution in a heterogeneous (layered) sediment.
% Set free-water diffusion coefficient
D_fw=11.6e-6;     %cm^2 s^-1
% set nodal values of z
z=[0 0.05 0.1 0.15 0.2 0.25 0.3 0.325 0.35 0.375 0.4 0.4125 0.425 0.4375
0.45 0.4625 0.475 0.4875 0.5];
dz=diff(z);
% n=number of nodes, ne=number of elements
n=length(z);
ne=length(dz);
% phi=array of element porosities; R is array of consumption rates.
i1=find(z<0.2);i2=find(z>=0.2&z<0.3);i3=find(z>=0.3&z<0.4);
i4=find(z>=0.4&z<0.45);i5=find(z>=0.45&z<0.5);
phi=zeros(size(z));
phi(i1)=0.7;phi(i2)=0.75;phi(i3)=0.8;phi(i4)=0.85;phi(i5)=0.9;
lam_e=phi.^3*D_fw;
R=zeros(size(z));
R(i1)=-0.01;R(i2)=-0.02;R(i3)=-0.04;R(i4)=-0.07;R(i5)=-0.02;
% flux=flux at top in n mol cm^-2 s^-1, cbot=constant bottom boundary
% concentration in n mol cm^-3.
flux=0.0125;
cbot=0;
% Set the coefficient matrix and the right-hand vector.
% preallocate matrix
G=zeros(n,n);
rhs=zeros(n,1);
for e=1:ne
    node1=e;node2=e+1;            %nodes for the element
    delta=z(node2)-z(node1);      %size of element
    elmat=zeros(2,2);             %initialize element matrix
    elmat(1,1)=lam_e(e)/delta;
    elmat(1,2)=-lam_e(e)/delta;
    elmat(2,1)=-lam_e(e)/delta;
    elmat(2,2)=lam_e(e)/delta;
    % assemble the global matrix by adding the element matrix
    G(node1,node1)=G(node1,node1)+elmat(1,1);
    G(node2,node2)=G(node2,node2)+elmat(2,2);
    G(node1,node2)=G(node1,node2)+elmat(1,2);
    G(node2,node1)=G(node2,node1)+elmat(2,1);
    % assemble rhs
    rhs(node1)=rhs(node1)+R(node1)*delta/2;
    rhs(node2)=rhs(node2)+R(node1)*delta/2;
end
% Change the first equation for constant concentration
% and the last to include the flux
G(1,1:n)=0; G(1,1)=1;
rhs(1,1)=cbot;
rhs(n)=rhs(n)+flux;
% The finite element equations are in the form G*c=rhs, where "G" is
% the global coefficient matrix, "c" is the vector of unknowns,
% and "rhs" is the vector of known quantities. The MATLAB "\" function
% solves the system of equations.
c=G\rhs;
% Plot the concentrations.
plot(c,z,'r+',c,z,'-')
xlabel('Oxygen concentration (n mol cm^-^3)');
ylabel('z (cm)');
```

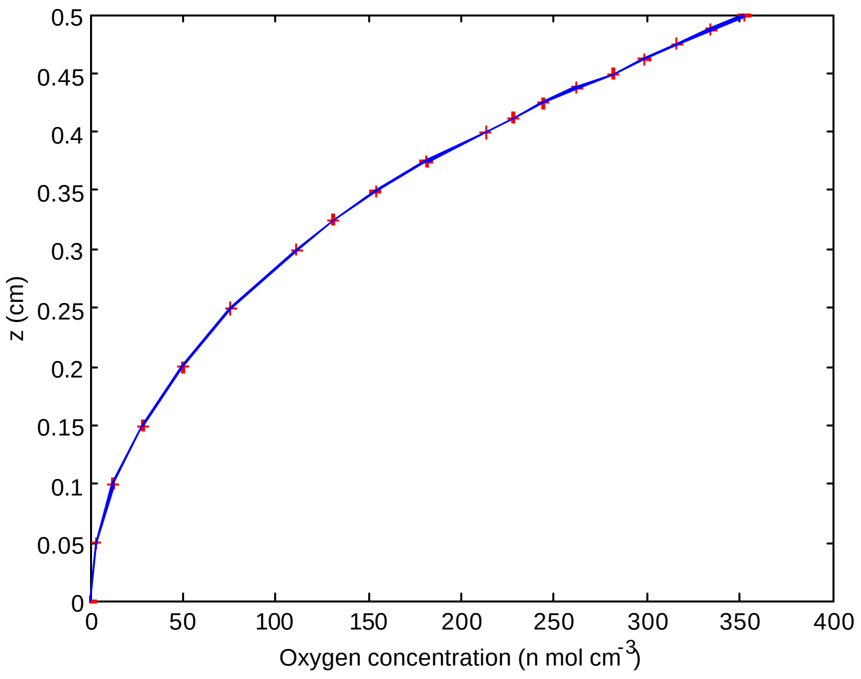

Figure 10.7. Oxygen profile calculated using the finite-element code. Note that the steady-state profile is curved because porosity (and thus diffusion coefficient) and rate of consumption vary with depth.

10.6. Problems

1. Write the finite difference equations for equation (10.1) and compare with the matrix derived using finite elements. What are the similarities and differences?

2. Berg et al. (1998) report measurements of "microprofiles" of NO_3 in freshwater sediment. In the case of nitrate, there can be "consumption" (reducing processes) or "production" (oxidizing processes). The porosity of the sediment is 0.93 and the free-water diffusion coefficient for NO_3 is 1.58×10^{-5} cm^2 s^{-1}. Given estimates of NO_3 consumption in layers of the sediment (Table 1), use a finite-element code to compute and plot the nitrate profile in the sediment. Plot your results against the measurements (Table 2) of the concentration profile. Use the surface flux boundary condition to "calibrate" the model, i.e., adjust the flux to get a good match between the computed and measured values. You might try a flux value of 0.0005 n mol cm^{-2} s^{-1} as a starting point. (Note that **negative** consumption rates represent production.)

Depth interval (cm)	NO_3 consumption rate (n mol cm^{-3} s^{-1})
0-0.16	-0.01
0.16-0.23	-0.08
0.23-0.28	0.11
0.28-0.32	0.01
0.32-0.40	0.00

Measured concentrations.

Depth	NO_3 (n mol cm^{-3})
0	69
0.05	63
0.1	57
0.15	47
0.2	30
0.25	13
0.3	2
0.35	0
0.4	0

3. The equation for early diagenesis in sediments can be modified to take account of *irrigation*, the pumping activity of tube-dwelling animals. In this case equation (10.16) is expanded to:

$$\frac{d}{dz}\left(fD\frac{dc}{dz}\right)+j\,a\left(c_0-c\right)+R=0,$$

where a is an irrigation coefficient and c_0 is the concentration at the sediment-water interface. Write a finite-element code to solve this problem and apply it to the conditions of problem 1. Examine results for a = 1×10^{-5} s^{-1} and 5×10^{-5} s^{-1}.

4. Write a finite-element code to solve the problem of heat flow in the earth, including advection of thermal energy:

$$\frac{d}{dz}\left(l \frac{dT}{dz} \right) - r_f c_f q \frac{dT}{dz} = 0.$$

where q is the specific discharge (units of, e.g., m s^{-1}), and r_f, and c_f, are the fluid density and heat capacity, respectively. [A code for the case of heat flow without advection, `geotherm.m`, is available as a template (see the m-file index).] Use a constant l = 2.5 W m^{-1} °C and a constant r_f = 1000 kg m^{-3} and c_f = 4200 J kg^{-1} °C. Solve the problem subject to the boundary conditions: T_{top}=20 °C and T_{bottom}=145 °C. Plot results for the following values of *q:*

q = 0.0 cm yr^{-1} (static case)
q = +0.1 cm yr^{-1} (up)
q=+*0.5* cm yr^{-1} (up)
q=+*l.0* cm yr^{-1} (up)
q = -*0.*1 cm yr^{-1} (down)
q = -0.5 cm yr^{-1} (down)
q=-*l.0* cm yr^{-1} (down)

Comment on the implications of the results for interpretations made in Problem 1 in Chapter 8.

10.7. References

Berg, P., Risgaard-Petersen, N., and S. Rysgaard, Interpretation of measured concentration profiles in sediment pore water. *Limnol. Oceanogr. 43*: 1500-1510, 1998.

Boudreau, BP. *Diagenetic Models and their Implementation*, 414 pp., Springer-Verlag, Berlin. 1997.

Fetter, C.W. Jr., *Applied Hydrogeology*, 598 pp., Prentice-Hall, Upper Saddle River, NJ, 2001.

Domenico, P.A. and F.W. Schwartz, *Physical and Chemical Hydrogeology*, 2nd edition, 506 pp., Wiley, New York, 1998.

Box 10.1. The finite element approach for transient conditions with advection

As an example of how the finite element method is extended to problems beyond the solutions for the steady state heat equation, consider the transport of a solute through a porous medium under transient conditions including advection (movement of the solute with the average velocity of the water) and dispersion (the "mixing" of solute due to differing velocities along different flow paths). Recall that we solved a form of the advection-dispersion equation using finite differences in Chapter 9. Here, we also let the solute decay according to a first-order rate law. More details about processes and the equation representing them can be found in hydrogeology texts (e.g., Fetter, 2001; Domenico and Schwartz, 1998). The equation is

$$\frac{\partial c}{\partial t} - D\frac{\partial^2 c}{\partial x^2} + u\frac{\partial c}{\partial x} + l\,c = 0$$

where c is solute concentration, t is time, x is distance, D is the dispersion coefficient, u is average velocity, and l is the decay coefficient.

The weighted residual integral for a typical element is

$$\int_{x_i}^{x_{i+1}} x \left(\frac{\partial c}{\partial t} - D\frac{\partial^2 c}{\partial x^2} + u\frac{\partial c}{\partial x} + l\,c \right) dx$$

The dispersion term is similar to the one for the steady state heat equation discussed in Chapter 10. The element matrix for the dispersion term is (cf. equation (10.15)

$$\begin{pmatrix} D/\Delta_e & -D/\Delta_e \\ -D/\Delta_e & D/\Delta_e \end{pmatrix}$$

The integrations for the other terms are straightforward. First take the advection term for x_i. Note that the element concentration can be written

$$c = c_i x_i + c_{i+1} x_{i+1} = c_i x_i + c_{i+1}(1 - x_i)$$

We can then evaluate the integral (for x_i) as follows, using once again a substitution of x_i for x in the integral and noting that $\frac{\partial c}{\partial x}dx = \frac{\partial c}{\partial x_i}dx_i$.

$$\begin{aligned} \int_{x_i}^{x_{i+1}} x_i u \frac{\partial c}{\partial x} dx &= u\int_0^1 x_i \frac{\partial c}{\partial x_i} dx_i \\ &= u\int_0^1 x_i \left(c_i - c_{i+1} \right) dx_i \\ &= u(c_i - c_{i+1}) \left. \frac{x_i^{\,2}}{2} \right|_0^1 \\ &= \frac{u}{2}c_i - \frac{u}{2}c_{i+1} \end{aligned}$$

The integral for x_{i+1} is evaluated in a similar way. The element matrix for the advection term is thus

$$\begin{pmatrix} \frac{u}{2} & -\frac{u}{2} \\ \frac{u}{2} & -\frac{u}{2} \end{pmatrix}$$

The integral for the term that accounts for the decay of the solute (for x_i) is evaluated as

$$\int_{x_i}^{x_{i+1}} x_i\, l\, c_e dx = \int_o^1 l x_i \left[c_i x_i + c_{i+1}(1-x_i)\right]\Delta_e dx_i$$

$$= l\Delta_e \left(\frac{x_i^3}{3}c_i + \frac{x_i^2}{2}c_{i+1} - \frac{x_i^3}{3}c_{i+1}\right)\Bigg]_0^1$$

$$= l\Delta_e \left(\frac{c_i}{3} + \frac{c_{i+1}}{6}\right)$$

The evaluation for x_{i+1} is similar. Thus the element matrix for the decay term is:

$$\begin{pmatrix} \frac{l\Delta_e}{3} & \frac{l\Delta_e}{6} \\ \frac{l\Delta_e}{6} & \frac{l\Delta_e}{3} \end{pmatrix}$$

The time derivative also must be integrated. The order of differentiation and integration can be interchanged, so the weighted residual to be evaluated is:

$$\frac{\partial}{\partial t}\int_{x_i}^{x_{i+1}} x\, c_e dx$$

Note that the integral is essentially the same as the term evaluated for the decay term. Thus the element matrix equation for the time derivative term is

$$\begin{pmatrix} \frac{\Delta_e}{3} & \frac{\Delta_e}{6} \\ \frac{\Delta_e}{6} & \frac{\Delta_e}{3} \end{pmatrix} \begin{pmatrix} \frac{\partial c_i}{\partial t} \\ \frac{\partial c_{i+1}}{\partial t} \end{pmatrix}$$

The time derivative term is replaced with an implicit finite difference approximation. The part of the approximation containing the known concentration at time t^j is placed on the right-hand side of the equation. The part of the approximation containing the unknown concentrations at time t^{j+1} is added to the left-hand side of the equation. The element matrices for the left and right sides are the same because of the sign change when the knowns are moved to the right-hand side:

$$\frac{\Delta_e}{3\Delta t}$$

The global matrix is assembled and the equations are solved one time step at a time as for the finite difference method. The code below shows how the solution can be implemented in *MATLAB*.

```
% ad_w_decay.m
% This is a finite element solution to calculate transport of
% a dissolved substance undergoing decay.
% dc/dt+u(dc/dx)=D(d2c/dx2)-(lambda)c

% Assume a 100-meter reach with constant concentration boundary
% conditions
% set nodal values of z and delta_t
z=0:1:100;
dz=diff(z);
delta_t=0.2;       % second
% n=number of nodes, e=number of elements
n_nodes=length(z);
n_elements=length(dz);
% set parameter values
u=0.2;       %m/s
D=0.1;       %m^2/s
lambda=0.01;  % per second
% conc at x=0 is one and at x=100 is zero
c0=1;c100=0;
% Set the coefficient matrix.
% preallocate matrix for left and rh sides
% The time derivative is done using a finite difference with
% an implicit approximation.
G=zeros(n_nodes,n_nodes);
R=zeros(n_nodes,n_nodes);
for e=1:n_elements
    node1=e;node2=e+1;          %nodes for the element
    delta=z(node2)-z(node1);    %size of element
    elmat=zeros(2,2);           %initialize element matrix
    rmat=zeros(2,2);            %initialize rhs matrix
    elmat(1,1)=D/delta-u/2+lambda*delta/3+delta/(3*delta_t);
    elmat(1,2)=-D/delta+u/2+lambda*delta/6+delta/(6*delta_t);
    elmat(2,1)=-D/delta-u/2+lambda*delta/6+delta/(6*delta_t);
    elmat(2,2)=D/delta+u/2+lambda*delta/3+delta/(3*delta_t);
    rmat(1,1)=delta/(3*delta_t);
    rmat(1,2)=delta/(6*delta_t);
    rmat(2,1)=delta/(6*delta_t);
    rmat(2,2)=delta/(3*delta_t);
    % assemble the global matrix by adding the element matrix
         G(node1,node1)=G(node1,node1)+elmat(1,1);
         G(node2,node2)=G(node2,node2)+elmat(2,2);
         G(node1,node2)=G(node1,node2)+elmat(1,2);
         G(node2,node1)=G(node2,node1)+elmat(2,1);
    % assemble the matrix for the rhs
         R(node1,node1)=R(node1,node1)+rmat(1,1);
         R(node2,node2)=R(node2,node2)+rmat(2,2);
         R(node1,node2)=R(node1,node2)+rmat(1,2);
         R(node2,node1)=R(node2,node1)+rmat(2,1);
end
```

```
% set the first and last equation for a fixed concentration boundary
% conditions
G(1,:)=0;G(1,1)=1; G(n_nodes,:)=0; G(n_nodes,n_nodes)=1;
R(1,:)=0;R(1,1)=1; R(n_nodes,:)=0; R(n_nodes,n_nodes)=1;
%
% Set the vector of "knowns" using specified concentration values
rhs=zeros(n_nodes,1);
rhs(1)=c0;
rhs(n_nodes)=c100;
endtime=300;       % seconds
% time loop
dz=dz';
time=0;j=1;k=1;
c_old=zeros(n_nodes,1);c_old(1)=rhs(1);c_old(n_nodes)=rhs(n_nodes);
while time<endtime
    rhs=R*c_old;
    c_new=G\rhs;
    if mod(j,300)==0
        conc(:,k)=c_new;
        k=k+1;
    end
    c_old=c_new;
    j=j+1;
    time=time+delta_t;
end
% Plot the profiles.
plot(z,conc)
ylabel('concentration'); xlabel('z (m)');
```

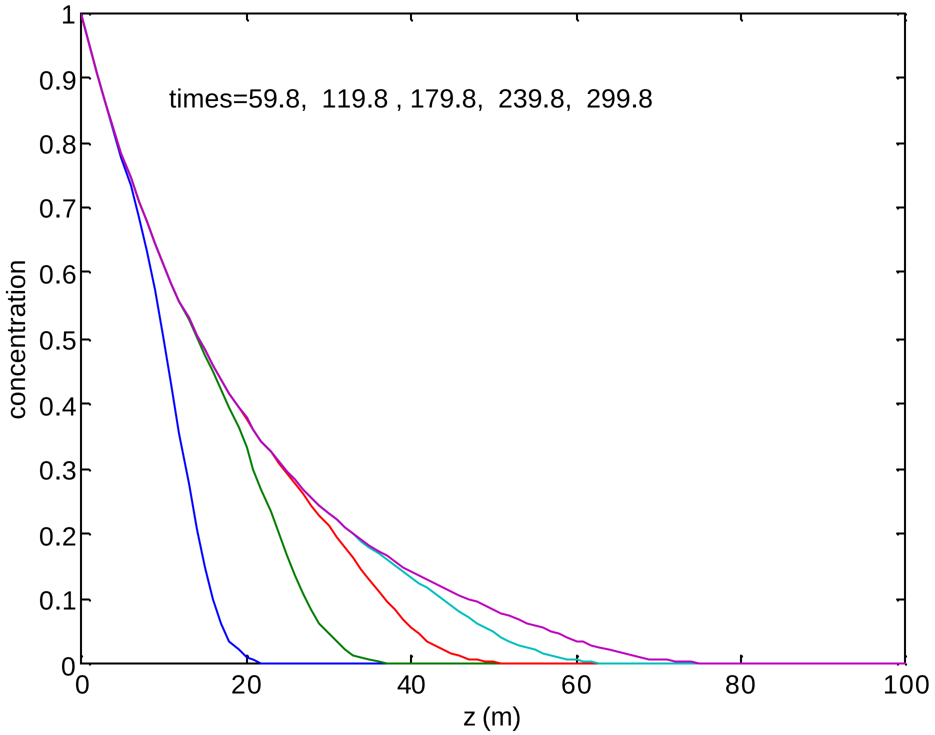

Figure B10.1.1 Finite element solution to advection-dispersion equation.