CHAPTER

2

Basics of Numerical Solutions by Finite Difference

Some models give rise to relatively simple analytic solutions for a wide range of initial and boundary conditions. But this is generally not true for more complex partial differential equations of increased dimension. As the dimensions and complexity of the coefficients and boundary conditions increase, finding analytic solutions becomes prohibitively difficult, and, in fact, some nonlinear PDEs have no known analytic solutions. To circumvent this problem, numerical solution schemes have been developed that involve finding discrete solutions at specific points in time and space. Of these schemes, the simplest are of the finite difference type, and we restrict our discussion to them. In essence, the approach is to convert the differential equations of a well-posed problem into a set of linear algebraic equations written in terms of the dependent variable(s). Because matrix algebra plays such a large role in solving such sets, we first briefly review it here. For a more complete introduction to discretization and finite difference methods, the reader is referred to Fletcher (1991) and Hoffman and Chiang (2000).

First Some Matrix Algebra

A matrix is a table or array of numbers or algebraic variables arranged in rows and columns such as this table for matrix **A**:

$$\mathbf{A} = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{bmatrix}. \tag{2.1}$$

Matrices are usually denoted in bold. The elements or entries in the matrix are usually denoted by indices reflecting the row and column of the entry, with the row number first. A matrix such as equation 2.1 having three rows and three columns is said to be a 3 by 3, or 3×3, matrix. If the number of rows and columns is equal, then the matrix is said to be square. Let the number of rows be m and the number of columns be n. Then if $m = 3$ and $n = 1$, such as

$$\mathbf{u} = \begin{pmatrix} a \\ b \\ c \end{pmatrix}, \tag{2.2}$$

matrix **u** is said to be a column matrix or column vector.

Matrices can be added or subtracted only if their number of rows and number of columns is identical, in which case corresponding entries are added or subtracted. Two matrices **A** and **B** can be multiplied only if the number of columns of **A** is equal to the number of rows of **B**. For example, if matrix **A** has size m by n, then it may premultiply a matrix **B** with size n by q, in which case the product matrix $\mathbf{C} = \mathbf{AB}$ will be size m by q. In component form, this multiplication takes the form

$$C_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj}. \tag{2.3}$$

A scalar multiplies a matrix by multiplying each entry. Likewise, differentiation of a matrix is done on each element.

A diagonal matrix is a square matrix in which $a_{ij} = 0$ if i does not equal j. For example, this is a special diagonal matrix called the identity matrix **I**:

$$\mathbf{I} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{2.4}$$

The transpose of a matrix (call it matrix **B**), denoted by $\mathbf{B}^{\mathrm{T}}$, is obtained by reflecting the entries about the diagonal from upper left to lower right.

The inverse of a square matrix **A** is defined such that

$$\mathbf{A}^{-1}\mathbf{A} \equiv \mathbf{A}\mathbf{A}^{-1} = \mathbf{I} \tag{2.5}$$

and is denoted $\mathbf{A}^{-1}$. A matrix is invertible if and only if its determinant is nonzero. The determinant is a special number that can be computed for any square matrix **A** and is denoted det(**A**) or |**A**|. Computing the determinant depends upon the dimensions of the matrix. For example, the determinant of the 3×3 matrix

$$\mathbf{A} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \tag{2.6}$$

is given by: $\det(\mathbf{A}) = aei - afh + bfg - bdi + cdh - ceg$.

Solution of Linear Systems of Algebraic Equations

Consider the system of m algebraic equations in n unknowns:

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2 \\ \vdots \qquad\qquad & \qquad \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n &= b_m. \end{aligned} \tag{2.7}$$

Using the rules noted above, one can write this system of equations in the compact notation of matrix algebra as

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix} \tag{2.8}$$

or

$$\mathbf{A}x = \mathbf{b}. \tag{2.9}$$

Usually, we want to solve for the column vector x of unknowns. Because it will always be the case that $m = n$ in these systems, we can use the definition from equation 2.5 to obtain

$$\mathbf{A}^{-1}\mathbf{A}x = \mathbf{A}^{-1}\mathbf{b}$$

or

$$x = \mathbf{A}^{-1}\mathbf{b}. \tag{2.10}$$

General Finite Difference Approach

To introduce the basics of the finite difference technique, consider the generic partial differential equation representing one-dimensional diffusion:

$$\frac{\partial T}{\partial t} - D\frac{\partial^2 T}{\partial x^2} = 0. \tag{2.11}$$

At this point, it is not necessary to know how the equation is derived, nor even what property T represents, only that $T(x,t)$ is a continuous function. To solve for $T(x,t)$ over specific intervals of time and distance and for specific initial and boundary conditions using the finite difference method, the general approach is to rewrite equation 2.11 as an algebraic equation and solve that equation at discrete points in space and time. Figure 2.1 shows the steps. We begin by discretizing the x–t plane.

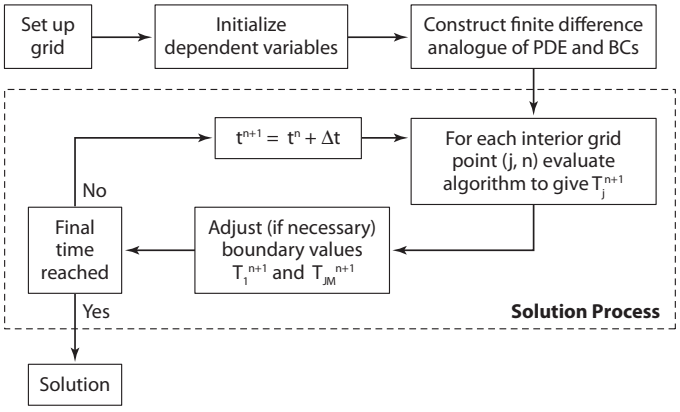

Figure 2.1. Steps in obtaining a finite difference solution to a PDE. [Modified from Fletcher, C.A.J. (1991). *Computational Techniques for Fluid Dynamics*. Berlin, Springer-Verlag.]

Discretization

Consider the domain in x–t space in figure 2.2 as the region in which we seek a solution to equation 2.11. In reality the solution is a continuous surface, but in numerical solutions the space–time plane is discretized into a set of points. The points are not of necessity at equal intervals, but for simplicity here we take the space step as a constant Δx and the time step as a constant Δt. Thus points in x and t lie at $x = j\Delta x$ and $n\Delta t$ where $j = 1, 2, 3, \ldots$ JM (maximum value of j) and $n = 1, 2, 3, \ldots$ NM (maximum value of n). We seek the values of the solution only at these discrete points. Of course if Δx and Δt are very small, then the coverage of solutions approximates the continuous solution surface. For an in-depth discussion of discretization for structured grids, see Hoffmann and Chiang (2000) or Anderson (1995).

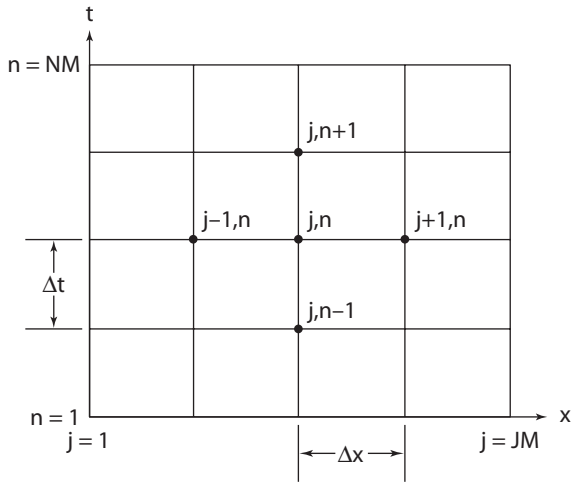

Figure 2.2. Hypothetical domain in space and time within which solutions of the one-dimensional diffusion equation are sought.

Obtaining Difference Operators by Taylor Series

To obtain an algebraic equation representing equation 2.11, we use the Taylor series. Consider a function $u(x,t)$. Equation 2.12 estimates the value of a function u at a point Δx ahead of the point x where the function is known, and equation 2.13 estimates the function at a point one space step behind:

$$u(x+\Delta x) = u(x) + \Delta x \frac{\partial u}{\partial x} + \frac{\Delta x^2}{2}\frac{\partial^2 u}{\partial x^2} + \frac{\Delta x^3}{6}\frac{\partial^3 u}{\partial x^3} + O(\Delta x^4) \tag{2.12}$$

$$u(x-\Delta x) = u(x) - \Delta x \frac{\partial u}{\partial x} + \frac{\Delta x^2}{2}\frac{\partial^2 u}{\partial x^2} - \frac{\Delta x^3}{6}\frac{\partial^3 u}{\partial x^3} + O(\Delta x^4). \tag{2.13}$$

The term $O(\Delta x)^4$ means that there exists a positive constant K, depending upon u, such that the difference between u at the $x + \Delta x$ node and the first three terms of the expansion, all evaluated at the xth node, is numerically less than $K(\Delta x)^3$ for all sufficiently small Δx.

Finite Difference Operators

Solving equation 2.12 for $\partial u/\partial x$ and dropping all higher-order terms yields the forward difference operator

$$\frac{\partial u}{\partial x} = \frac{u(x+\Delta x) - u(x)}{\Delta x} + O(\Delta x). \tag{2.14}$$

Similarly, equation 2.13 yields the backwards difference operator

$$\frac{\partial u}{\partial x} = \frac{u(x) - u(x-\Delta x)}{\Delta x} + O(\Delta x). \tag{2.15}$$

And subtracting equation 2.13 from equation 2.12 yields the centered difference operator

$$\frac{\partial u}{\partial x} = \frac{u(x+\Delta x) - u(x-\Delta x)}{2\Delta x} + O(\Delta x^2). \tag{2.16}$$

Note that the forwards and backwards approximations are first-order accurate, whereas the centered is second-order accurate. This means that for the same Δx, it should be more accurate. Forwards and backwards difference schemes of higher-order accuracy are possible, too, but they use values of u at two and three Δx away, making them more computationally expensive and difficult to use near boundaries of the computational domain. Likewise, if equation 2.12 and equation 2.13 are added, the resulting equation can be solved for $\partial^2 u/\partial x^2$ such that

$$\frac{\partial^2 u}{\partial x^2} = \frac{u(x+\Delta x) - 2u(x) + u(x-\Delta x)}{\Delta x^2} + O(\Delta x^2). \tag{2.17}$$

This approximation uses three nodes and is centered in space. There are many more possibilities. For an extensive listing of finite different approximations to various differentials, see Hoffmann and Chiang (2000) and Fletcher (1991).

Explicit Schemes

The next step in formulating a finite difference approximation to the one-dimensional (1-D) diffusion equation is to substitute the above definitions of the derivatives into equation 2.11. Before we do so, it is convenient to change the notation such that $u(x + \Delta x)$ is represented by u_{j+1} and $u(t + \Delta t)$ is represented by u^{n+1} where $t = n\Delta t$ and $x = j\Delta x$ and n and j are the integer series 0, 1, 2, 3, ... Letting $u = T$, substituting equation 2.14 and equation 2.17 into equation 2.11, and solving for the unknown values of T at the new time step yields the forward-in-time, centered-in-space (FTCS) finite difference scheme

$$T_j^{n+1} = sT_{j-1}^n + (1 - 2s)T_j^n + sT_{j+1}^n, \tag{2.18}$$

where

$$s = D\frac{\Delta t}{\Delta x^2}$$

is called the *diffusion number.*

Inspection of equation 2.18 indicates that values of T known at time n are used to approximate the second derivative of T with respect to x (i.e., its curvature). This quantity added to the value of T_j^n (represented by the coefficient 1 in the second term on the right-hand side) provides an estimate of how T changes from its initial value over one time step. The grid in figure 2.2 can be swept from $j = 2$ to $j = \mathrm{JM} - 1$ for each successive time step. Values of the function at $j = 1$ and $j = \mathrm{JM}$ are not computed because they are known from the boundary conditions (presuming Dirichlet boundary conditions). Notice that the scheme is set up to estimate the value of the function at $f(j, n + 1)$ by using the value of the function at $f(j, n)$, $f(j - 1, n)$, and $f(j + 1, n)$ (i.e., by using values that are all known at the time of the computation). For that reason the scheme is called *explicit*.

One can imagine, however, that the structure of the solution surface as plotted in x,t space may look like a topographic map with domes and hollows. Explicit schemes estimate the curvature of the solution in space *not* at the same time as we want the solution but at an earlier time when the geometry of the solution surface may be different. Wouldn't it be better (more internally consistent) to estimate the curvature at the same point in the space–time plane as the temporal derivative? Usually the answer is yes. We say usually, because the most obvious approach to effect this for equation 2.11 is to use a centered difference operator for the time derivative. Unfortunately, this centered in time and centered in space (CTCS) scheme, also called the Richardson scheme, is unconditionally unstable and of no practical use.

Alternatively, one could estimate the curvature at the point in the space–time plane at the new time step where the new value of T is being computed. This leads to *implicit* schemes as the following example demonstrates.

Implicit Schemes

We rewrite equation 2.18 to estimate the curvature at the $n + 1$ time step. Gathering all unknowns on the left-hand side (LHS):

$$-sT_{j-1}^{n+1} + (1+2s)\,T_j^{n+1} - sT_{j+1}^{n+1} = T_j^n. \tag{2.19}$$

This is called the Laasonen fully implicit scheme. But now there are three unknowns and only one equation. The path out of this dilemma is to notice that we can write equation 2.19 for each node in space (at the $n + 1$ time step), thereby generating just enough equations for the number of unknowns. Figure 2.3 shows a simple example of two unknown points surrounded by known values provided by the boundary and initial conditions: Writing equation 2.19 at $n = 2$ and at $j = 2$ and then at $j = 3$ yields

$$\begin{aligned} -sa + (1+2s)\,T_2^2 - sT_3^2 &= c \\ -sT_2^2 + (1+2s)\,T_3^2 - sf &= d, \end{aligned} \tag{2.20}$$

where the exponents on T indicate time step 2 (not a squaring operation), which assembled in matrix notation becomes

$$\begin{pmatrix} (1+2s) & -s \\ -s & (1+2s) \end{pmatrix} \begin{pmatrix} T_2^2 \\ T_3^2 \end{pmatrix} = \begin{pmatrix} c+sa \\ d+sf \end{pmatrix}. \tag{2.21}$$

Thus the resulting equations constitute a linear system that can be solved by matrix methods.

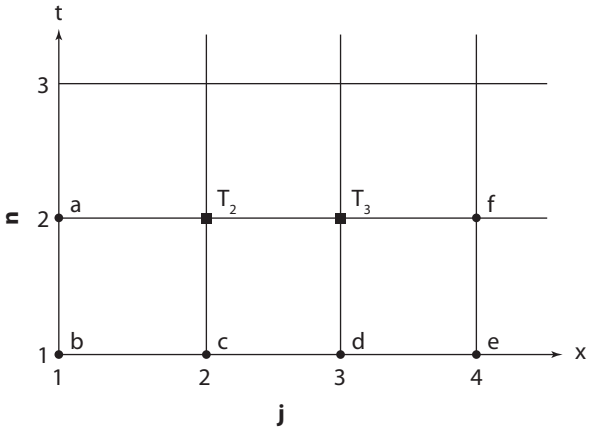

Figure 2.3. Example problem grid for two nodes solved by the fully implicit method.

Another very popular implicit scheme is the Crank–Nicolson scheme. It uses the logic that a forward-in-time approximation of the time derivative is actually estimating the slope of the function with respect to time at a point halfway between n and $n + 1$. Consequently, we should center our estimate of the curvature on that point, too, and that means calculating the curvature at time n *and* at time $n + 1$ and averaging the two. The resulting computation equation is

$$\begin{aligned} -0.5sT_{j-1}^{n+1} + (1+s)T_j^{n+1} - 0.5sT_{j+1}^{n+1} \\ = 0.5sT_{j-1}^{n} + (1-s)T_j^{n} + 0.5sT_{j+1}^{n}. \end{aligned} \tag{2.22}$$

Some schemes weight the estimate of curvature a little toward the $n + 1$ time step by using proportions other than 0.5. In any case, equation 2.22 written for each node in the j direction will contain an unknown at $j - 1$, j, and $j + 1$ of the form

$$a_j T_{j-1} + b_j T_j + c_i T_{j+1} = d_j. \tag{2.23}$$

When are all assembled in matrix form, the structure is

$$\begin{bmatrix} b_1 & c_1 & & & 0 \\ a_2 & b_2 & c_2 & & \\ & a_3 & b_3 & \cdot & \\ & & \cdot & \cdot & c_{n-1} \\ 0 & & & & b_n \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \\ \cdot \\ \cdot \\ T_n \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ \cdot \\ \cdot \\ d_n \end{bmatrix}. \tag{2.24}$$

Notice that the first or coefficient matrix is tridiagonal; that is, it contains terms only along the center and adjacent two diagonals. Tridiagonal systems of equations like this can be solved efficiently using a simplified form of Gaussian elimination known as Thomas' algorithm. An insightful discussion on this very efficient form of Gaussian elimination and solvers in C and Fortran can be obtained at http://www.nr.com/.

A handy method of summarizing finite difference schemes is to present the basic computational module graphically. Figure 2.4 shows the templates for the four schemes discussed above.

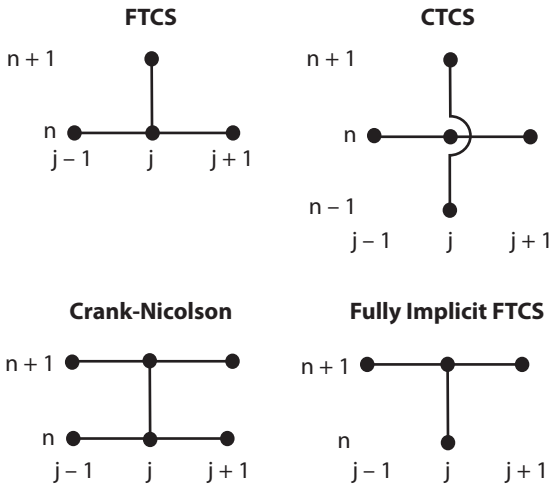

Figure 2.4. Nodes used in approximations to differentials of a one-dimensional diffusion equation. FT, forward in time; CS, centered in space; CT, centered in time.

How Good Is My Finite Difference Scheme?

A choice of finite difference schemes begs the question of which scheme is better. Better can be defined in a number of ways, but we define it as a scheme that is *accurate, efficient,* and *easy.* A scheme is accurate (also called convergent) if its solution approaches the analytic solution as the discretization steps are reduced in size. A scheme is efficient if it minimizes computation time. Easy refers to our ability to comprehend and code the scheme. Judging a scheme's efficiency and ease of use is straightforward, but how do we guarantee its accuracy? The answer is that we must guarantee its *consistency* and *stability.*

A scheme's system of algebraic equations is consistent if, as $\Delta x, \Delta t \to 0$, the system becomes equivalent to the differential equations (DEs) at each grid point. To determine consistency, expand the finite difference equation about x

and t by Taylor series to recover the ODE or PDE. In addition, there will be a remainder of higher-order terms. If the remainder tends to zero as $\Delta x, \Delta t \rightarrow 0$, then the finite difference equation is consistent.

A scheme is stable if spontaneous perturbations (such as round-off error) in the solution of the algebraic equations decay as the computations proceed. A variety of methods exist to determine a scheme's stability such as a von Neumann stability analysis, but these are beyond the scope of this book. See Fletcher (1991) or Hoffmann and Chiang (2000) for excellent summaries.

Finally, a solution of the algebraic equations approximating a DE is convergent if the approximate solution approaches the exact solution as grid size tends to zero. To guarantee convergence, we make use of the Lax equivalence theorem. The Lax equivalence theorem states: *"Given a properly posed linear initial value problem, if a finite difference approximation is consistent and stable, it is convergent."*

Figure 2.5 summarizes these concepts.

So which schemes in figure 2.4 are better? Generally, it can be said that higher-order approximations are more accurate unless they are (1) unstable or (2) the exact solution contains discontinuities or steep gradients. For example, the CTCS (Richardson) scheme would seem to be better than the FTCS explicit scheme because it is of

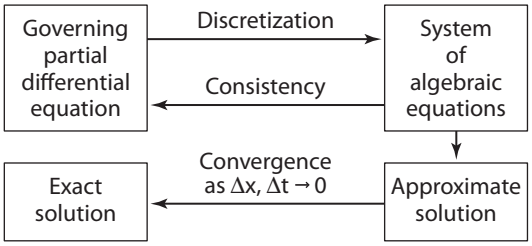

Figure 2.5. Relationship between consistency, stability, and convergence. [Modified from Fletcher, C.A.J. (1991). *Computational Techniques for Fluid Dynamics*. Berlin, Springer-Verlag.]

second order in both time and space discretization (equation 2.16 and equation 2.17), whereas the FTCS scheme is only first-order accurate in time. But the CTCS scheme is unconditionally unstable. The FTCS scheme is stable under certain conditions. As indicated by a von Neumann stability analysis, the explicit FTCS scheme approximating equation 2.11 is stable if

$$s = \frac{D\Delta t}{\Delta x^2} \leq 0.5. \tag{2.25}$$

The fully implicit FTCS and Crank–Nicolson schemes are second-order accurate in both time and space and unconditionally stable.

Stability Is Not Accuracy

As an example of how stability depends upon s, consider solutions to the 1-D diffusion equation describing viscous flow of a Newtonian fluid adjacent to a solid wall. At $t > 0$ the wall at $x = 0$ begins to move instantaneously at a velocity V_0. If the resulting flow is nonturbulent as it is dragged along, then the equation describing the fluid velocity parallel to the wall at various distances y away from the wall, $V(y)$, is described by

$$\frac{\partial V}{\partial t} - \nu \frac{\partial^2 V}{\partial y^2} = 0, \tag{2.26}$$

where ν is the kinematic viscosity of the fluid. This equation is derived in chapter 4. Note the similar form to equation 2.11. There is an analytic solution to this problem given by

$$V = V_0 \left\{ \sum_{n=0}^{\infty} erfc[2n\eta_1 + \eta] - \sum_{n=0}^{\infty} erfc[2(n+1)\eta_1 - \eta] \right\}, \tag{2.27}$$

where

$$\eta_1 = \frac{h}{(2\sqrt{\nu t})}$$

$$\eta = \frac{y}{(2\sqrt{\nu t})}.$$

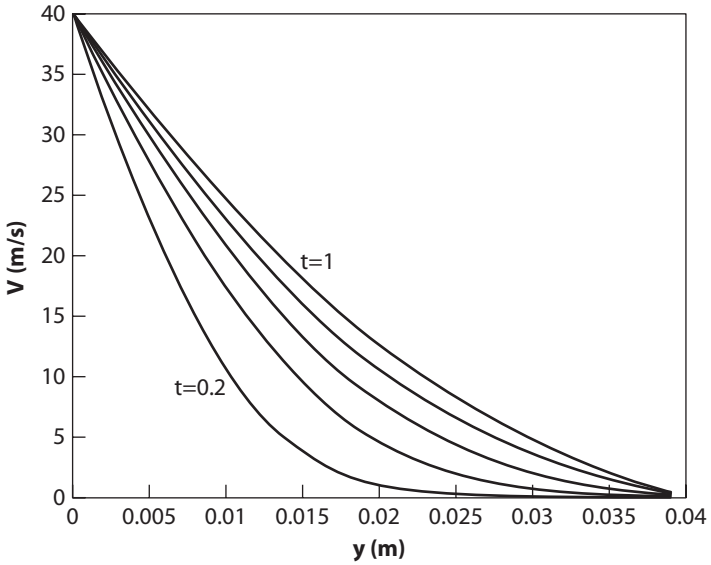

Figure 2.6. Solid lines are analytic solutions to equation 2.26 under the ICs and BCs specified in the text. Dashed lines are solutions from the FTCS scheme with $s = 0.2$. The dashed and solid lines are indistinguishable.

h is the thickness of the fluid, and *erfc* is the complementary error function.

For the purposes of comparing finite difference solutions with the analytic solution, consider the particular problem of an oil of kinematic viscosity equal to 2×10^{-4} $m^2\ s^{-1}$ sitting in a 40-mm-thick space bounded by a fixed wall at $y = 0.04$ m and a wall at $y = 0$ that at $t > 0$ begins to move at $V_0 = 40$ m s^{-1}. The analytic solutions are given in figure 2.6 at five equally spaced times from 0.2 to 1 second, showing the development of the velocity profile toward a steady state. Also shown are the numerical solutions to equation 2.26 obtained by the explicitly FTCS scheme with $s = 0.2$. The numerical solution is indistinguishable from the analytic solution.

However, inaccuracies appear as s is increased. The difference between the analytic and numerical solutions

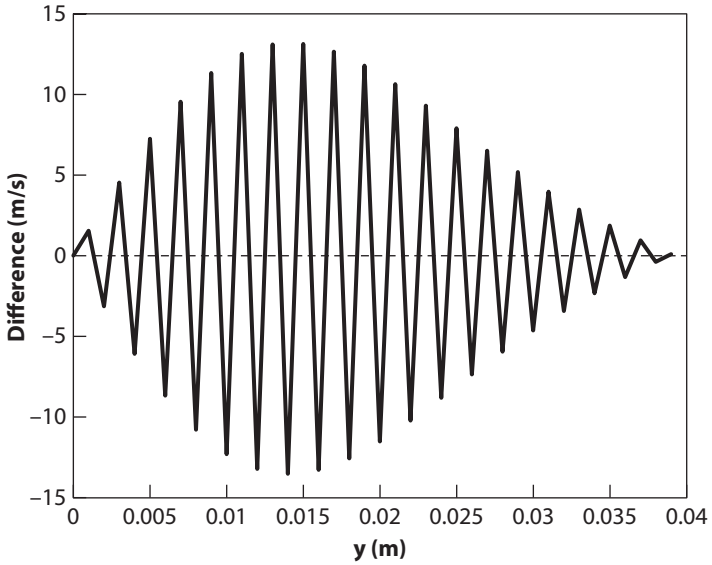

Figure 2.7. Difference between the analytic and FTCS solutions to equation 2.26 for the velocity profile in a viscous fluid. ICs and BCs defined in text. Dashed line computed at $t = 0.5$ second with a diffusion number, $s = 0.5$; solid line computed at same time with $s = 0.508$, illustrating that s must be less than or equal to 0.5 for stability of the FTCS scheme.

to equation 2.26 at $t = 0.5$ and $s = 0.5$ remain small (fig. 2.7). Remember that a von Neumann stability analysis shows that s must be less than or equal to 0.5 for stability of the FTCS scheme. With a slight increase in the diffusion number to 0.508, the scheme becomes unstable and therefore inaccurate.

Summary

A numerical solution to a differential equation or equation set is obtained by converting the equations into an algebraic equation or equation set. This is accomplished by approximating values of the derivatives using Taylor series.

Then the resulting algebraic equations are solved for the dependent variables either explicitly or implicitly at discrete points in the space–time plane. Subsequent chapters will first (and foremost) focus on the process of translating natural phenomena into sets of differential equations but then also explore how numerical solutions to these equations can be obtained. We begin with problems that through translation lead to ordinary differential equations.

Modeling Exercises

1. **Matrix Algebra**
   Consider the set of equations

   $$3x + 2y + 5z = 0$$
   $$7x + 6y + 4z = -2$$
   $$x + 3y + 2z = -6.$$

   Write the set in matrix form $\mathbf{Ax} = \mathbf{b}$. Is the $\mathbf{A}$ matrix invertible? Use your favorite math software (MATLAB, Mathematica, etc.) to solve the equation set for x, y, and z.
2. **The First Numerical Model**
   Write a simple code to calculate the time evolution of the viscous velocity profile that arises from the numerical solution to equation 2.26. Use the FTCS scheme given by equation 2.18, the initial and boundary conditions given in the text, and a diffusion number of 0.2. You will have an outer loop (for loop or do loop) that progresses through time (the n index) and an inner loop that sweeps the grid from left to right (the j index). Your solutions should be identical to those given in figure 2.6. Then reproduce the analysis of figure 2.7.
3. **Practice with Implicit Schemes**
   Now apply the fully implicit scheme (equation 2.22) to the viscous velocity profile problem discussed in the text. Assess its accuracy for various diffusion numbers.