Roache, P.J. 1976, *Computational Fluid Dynamics* (Albuquerque: Hermosa). [7]
Woodward, P., and Colella, P. 1984, *Journal of Computational Physics*, vol. 54, pp. 115–173. [8]
Rizzi, A., and Engquist, B. 1987, *Journal of Computational Physics*, vol. 72, pp. 1–69. [9]

19.2 Diffusive Initial Value Problems

Recall the model parabolic equation, the diffusion equation in one space dimension,

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x}\left(D\frac{\partial u}{\partial x}\right) \tag{19.2.1}$$

where D is the diffusion coefficient. Actually, this equation is a flux-conservative equation of the form considered in the previous section, with

$$F = -D\frac{\partial u}{\partial x} \tag{19.2.2}$$

the flux in the x-direction. We will assume $D \geq 0$, otherwise equation (19.2.1) has physically unstable solutions: A small disturbance evolves to become more and more concentrated instead of dispersing. (Don't make the mistake of trying to find a stable differencing scheme for a problem whose underlying PDEs are themselves unstable!)

Even though (19.2.1) is of the form already considered, it is useful to consider it as a model in its own right. The particular form of flux (19.2.2), and its direct generalizations, occur quite frequently in practice. Moreover, we have already seen that numerical viscosity and artificial viscosity can introduce diffusive pieces like the right-hand side of (19.2.1) in many other situations.

Consider first the case when D is a constant. Then the equation

$$\frac{\partial u}{\partial t} = D\frac{\partial^2 u}{\partial x^2} \tag{19.2.3}$$

can be differenced in the obvious way:

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = D\left[\frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{(\Delta x)^2}\right] \tag{19.2.4}$$

This is the FTCS scheme again, except that it is a second derivative that has been differenced on the right-hand side. But this makes a world of difference! The FTCS scheme was unstable for the hyperbolic equation; however, a quick calculation shows that the amplification factor for equation (19.2.4) is

$$\xi = 1 - \frac{4D\Delta t}{(\Delta x)^2}\sin^2\left(\frac{k\Delta x}{2}\right) \tag{19.2.5}$$

The requirement $|\xi| \leq 1$ leads to the stability criterion

$$\frac{2D\Delta t}{(\Delta x)^2} \leq 1 \tag{19.2.6}$$

The physical interpretation of the restriction (19.2.6) is that the maximum allowed timestep is, up to a numerical factor, the diffusion time across a cell of width Δx.

More generally, the diffusion time τ across a spatial scale of size λ is of order

$$\tau \sim \frac{\lambda^2}{D} \tag{19.2.7}$$

Usually we are interested in modeling accurately the evolution of features with spatial scales $\lambda \gg \Delta x$. If we are limited to timesteps satisfying (19.2.6), we will need to evolve through of order $\lambda^2/(\Delta x)^2$ steps before things start to happen on the scale of interest. This number of steps is usually prohibitive. We must therefore find a stable way of taking timesteps comparable to, or perhaps — for accuracy — somewhat smaller than, the time scale of (19.2.7).

This goal poses an immediate "philosophical" question. Obviously the large timesteps that we propose to take are going to be woefully inaccurate for the small scales that we have decided not to be interested in. We want those scales to do something stable, "innocuous," and perhaps not too physically unreasonable. We want to build this innocuous behavior into our differencing scheme. What should it be?

There are two different answers, each of which has its pros and cons. The first answer is to seek a differencing scheme that drives small-scale features to their *equilibrium* forms, e.g., satisfying equation (19.2.3) with the left-hand side set to zero. This answer generally makes the best physical sense; but, as we will see, it leads to a differencing scheme ("fully implicit") that is only *first-order* accurate in time for the scales that we are interested in. The second answer is to let small-scale features *maintain* their initial amplitudes, so that the evolution of the larger-scale features of interest takes place superposed with a kind of "frozen in" (though fluctuating) background of small-scale stuff. This answer gives a differencing scheme ("Crank-Nicholson") that is *second-order* accurate in time. Toward the end of an evolution calculation, however, one might want to switch over to some steps of the other kind, to drive the small-scale stuff into equilibrium. Let us now see where these distinct differencing schemes come from:

Consider the following differencing of (19.2.3),

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = D\left[\frac{u_{j+1}^{n+1} - 2u_j^{n+1} + u_{j-1}^{n+1}}{(\Delta x)^2}\right] \tag{19.2.8}$$

This is exactly like the FTCS scheme (19.2.4), except that the spatial derivatives on the right-hand side are evaluated at timestep $n+1$. Schemes with this character are called *fully implicit* or *backward time*, by contrast with FTCS (which is called *fully explicit*). To solve equation (19.2.8) one has to solve a set of simultaneous linear equations at each timestep for the u_j^{n+1}. Fortunately, this is a simple problem because the system is tridiagonal: Just group the terms in equation (19.2.8) appropriately:

$$-\alpha u_{j-1}^{n+1} + (1+2\alpha)u_j^{n+1} - \alpha u_{j+1}^{n+1} = u_j^n, \qquad j = 1, 2 \ldots J-1 \tag{19.2.9}$$

where

$$\alpha \equiv \frac{D\Delta t}{(\Delta x)^2} \tag{19.2.10}$$

Supplemented by Dirichlet or Neumann boundary conditions at $j = 0$ and $j = J$, equation (19.2.9) is clearly a tridiagonal system, which can easily be solved at each timestep by the method of §2.4.

What is the behavior of (19.2.8) for very large timesteps? The answer is seen most clearly in (19.2.9), in the limit $\alpha \to \infty$ ($\Delta t \to \infty$). Dividing by α, we see that the difference equations are just the finite-difference form of the equilibrium equation

$$\frac{\partial^2 u}{\partial x^2} = 0 \tag{19.2.11}$$

What about stability? The amplification factor for equation (19.2.8) is

$$\xi = \frac{1}{1 + 4\alpha \sin^2\left(\dfrac{k\Delta x}{2}\right)} \tag{19.2.12}$$

Clearly $|\xi| < 1$ for any stepsize Δt. The scheme is unconditionally stable. The details of the small-scale evolution from the initial conditions are obviously inaccurate for large Δt. But, as advertised, the correct equilibrium solution is obtained. This is the characteristic feature of implicit methods.

Here, on the other hand, is how one gets to the second of our above philosophical answers, combining the stability of an implicit method with the accuracy of a method that is second-order in both space and time. Simply form the average of the explicit and implicit FTCS schemes:

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = \frac{D}{2}\left[\frac{(u_{j+1}^{n+1} - 2u_j^{n+1} + u_{j-1}^{n+1}) + (u_{j+1}^n - 2u_j^n + u_{j-1}^n)}{(\Delta x)^2}\right] \tag{19.2.13}$$

Here both the left- and right-hand sides are centered at timestep $n + \frac{1}{2}$, so the method is second-order accurate in time as claimed. The amplification factor is

$$\xi = \frac{1 - 2\alpha \sin^2\left(\dfrac{k\Delta x}{2}\right)}{1 + 2\alpha \sin^2\left(\dfrac{k\Delta x}{2}\right)} \tag{19.2.14}$$

so the method is stable for any size Δt. This scheme is called the *Crank-Nicholson* scheme, and is our recommended method for any simple diffusion problem (perhaps supplemented by a few fully implicit steps at the end). (See Figure 19.2.1.)

Now turn to some generalizations of the simple diffusion equation (19.2.3). Suppose first that the diffusion coefficient D is not constant, say $D = D(x)$. We can adopt either of two strategies. First, we can make an analytic change of variable

$$y = \int \frac{dx}{D(x)} \tag{19.2.15}$$

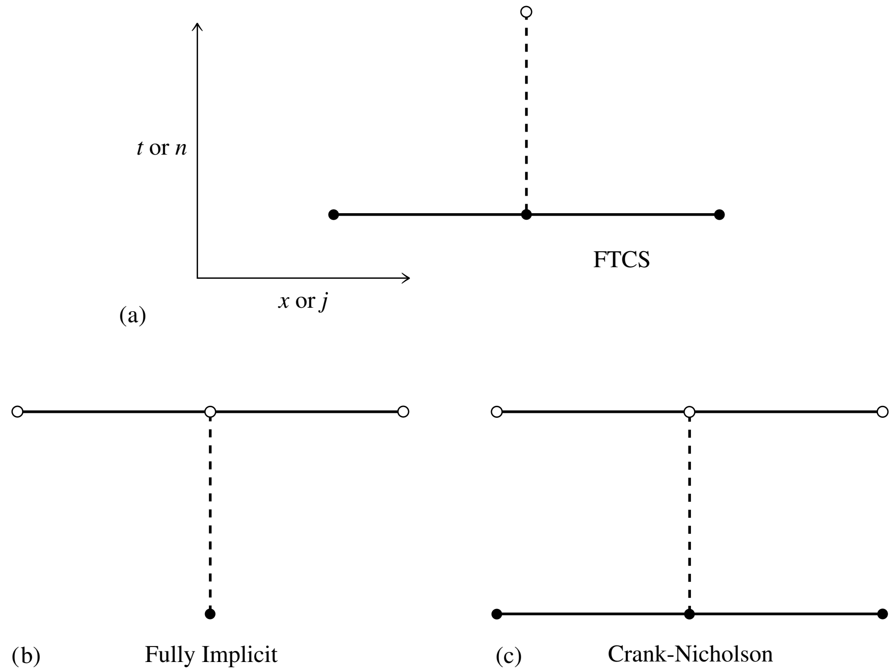

Figure 19.2.1. Three differencing schemes for diffusive problems (shown as in Figure 19.1.2). (a) Forward Time Center Space is first-order accurate, but stable only for sufficiently small timesteps. (b) Fully Implicit is stable for arbitrarily large timesteps, but is still only first-order accurate. (c) Crank-Nicholson is second-order accurate, and is usually stable for large timesteps.

Then

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x} D(x) \frac{\partial u}{\partial x} \tag{19.2.16}$$

becomes

$$\frac{\partial u}{\partial t} = \frac{1}{D(y)} \frac{\partial^2 u}{\partial y^2} \tag{19.2.17}$$

and we evaluate D at the appropriate y_j. Heuristically, the stability criterion (19.2.6) in an explicit scheme becomes

$$\Delta t \leq \min_j \left[\frac{(\Delta y)^2}{2 D_j^{-1}} \right] \tag{19.2.18}$$

Note that constant spacing Δy in y does not imply constant spacing in x.

An alternative method that does not require analytically tractable forms for D is simply to difference equation (19.2.16) as it stands, centering everything appropriately. Thus the FTCS method becomes

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = \frac{D_{j+1/2}(u_{j+1}^n - u_j^n) - D_{j-1/2}(u_j^n - u_{j-1}^n)}{(\Delta x)^2} \tag{19.2.19}$$

where

$$D_{j+1/2} \equiv D(x_{j+1/2}) \tag{19.2.20}$$

and the heuristic stability criterion is

$$\Delta t \leq \min_j \left[\frac{(\Delta x)^2}{2D_{j+1/2}} \right] \tag{19.2.21}$$

The Crank-Nicholson method can be generalized similarly.

The second complication one can consider is a nonlinear diffusion problem, for example where $D = D(u)$. Explicit schemes can be generalized in the obvious way. For example, in equation (19.2.19) write

$$D_{j+1/2} = \frac{1}{2}\left[D(u_{j+1}^n) + D(u_j^n)\right] \tag{19.2.22}$$

Implicit schemes are not as easy. The replacement (19.2.22) with $n \to n+1$ leaves us with a nasty set of coupled nonlinear equations to solve at each timestep. Often there is an easier way: If the form of $D(u)$ allows us to integrate

$$dz = D(u)du \tag{19.2.23}$$

analytically for $z(u)$, then the right-hand side of (19.2.1) becomes $\partial^2 z/\partial x^2$, which we difference implicitly as

$$\frac{z_{j+1}^{n+1} - 2z_j^{n+1} + z_{j-1}^{n+1}}{(\Delta x)^2} \tag{19.2.24}$$

Now linearize each term on the right-hand side of equation (19.2.24), for example

$$\begin{aligned} z_j^{n+1} \equiv z(u_j^{n+1}) &= z(u_j^n) + (u_j^{n+1} - u_j^n) \left.\frac{\partial z}{\partial u}\right|_{j,n} \\ &= z(u_j^n) + (u_j^{n+1} - u_j^n) D(u_j^n) \end{aligned} \tag{19.2.25}$$

This reduces the problem to tridiagonal form again and in practice usually retains the stability advantages of fully implicit differencing.

Schrödinger Equation

Sometimes the physical problem being solved imposes constraints on the differencing scheme that we have not yet taken into account. For example, consider the time-dependent Schrödinger equation of quantum mechanics. This is basically a parabolic equation for the evolution of a complex quantity ψ. For the scattering of a wavepacket by a one-dimensional potential $V(x)$, the equation has the form

$$i\frac{\partial \psi}{\partial t} = -\frac{\partial^2 \psi}{\partial x^2} + V(x)\psi \tag{19.2.26}$$

(Here we have chosen units so that Planck's constant $\hbar = 1$ and the particle mass $m = 1/2$.) One is given the initial wavepacket, $\psi(x, t = 0)$, together with boundary conditions that $\psi \to 0$ at $x \to \pm\infty$. Suppose we content ourselves with first-order accuracy in time, but want to use an implicit scheme, for stability. A slight generalization of (19.2.8) leads to

$$i\left[\frac{\psi_j^{n+1}-\psi_j^n}{\Delta t}\right] = -\left[\frac{\psi_{j+1}^{n+1}-2\psi_j^{n+1}+\psi_{j-1}^{n+1}}{(\Delta x)^2}\right] + V_j\psi_j^{n+1} \tag{19.2.27}$$

for which

$$\xi = \frac{1}{1+i\left[\dfrac{4\Delta t}{(\Delta x)^2}\sin^2\left(\dfrac{k\Delta x}{2}\right)+V_j\Delta t\right]} \tag{19.2.28}$$

This is unconditionally stable, but unfortunately is not *unitary*. The underlying physical problem requires that the total probability of finding the particle somewhere remains unity. This is represented formally by the modulus-square norm of ψ remaining unity:

$$\int_{-\infty}^{\infty} |\psi|^2 dx = 1 \tag{19.2.29}$$

The initial wave function $\psi(x,0)$ is normalized to satisfy (19.2.29). The Schrödinger equation (19.2.26) then guarantees that this condition is satisfied at all later times.

Let us write equation (19.2.26) in the form

$$i\frac{\partial\psi}{\partial t} = H\psi \tag{19.2.30}$$

where the operator H is

$$H = -\frac{\partial^2}{\partial x^2} + V(x) \tag{19.2.31}$$

The formal solution of equation (19.2.30) is

$$\psi(x,t) = e^{-iHt}\psi(x,0) \tag{19.2.32}$$

where the exponential of the operator is defined by its power series expansion.

The unstable explicit FTCS scheme approximates (19.2.32) as

$$\psi_j^{n+1} = (1 - iH\Delta t)\psi_j^n \tag{19.2.33}$$

where H is represented by a centered finite-difference approximation in x. The stable implicit scheme (19.2.27) is, by contrast,

$$\psi_j^{n+1} = (1 + iH\Delta t)^{-1}\psi_j^n \tag{19.2.34}$$

These are both first-order accurate in time, as can be seen by expanding equation (19.2.32). However, neither operator in (19.2.33) or (19.2.34) is unitary.

The correct way to difference Schrödinger's equation [1,2] is to use *Cayley's form* for the finite-difference representation of e^{-iHt}, which is second-order accurate *and* unitary:

$$e^{-iHt} \simeq \frac{1 - \frac{1}{2}iH\Delta t}{1 + \frac{1}{2}iH\Delta t} \tag{19.2.35}$$

In other words,

$$\left(1 + \tfrac{1}{2}iH\Delta t\right)\psi_j^{n+1} = \left(1 - \tfrac{1}{2}iH\Delta t\right)\psi_j^n \tag{19.2.36}$$

On replacing H by its finite-difference approximation in x, we have a complex tridiagonal system to solve. The method is stable, unitary, and second-order accurate in space and time. In fact, it is simply the Crank-Nicholson method once again!

CITED REFERENCES AND FURTHER READING:

Ames, W.F. 1977, *Numerical Methods for Partial Differential Equations*, 2nd ed. (New York: Academic Press), Chapter 2.

Goldberg, A., Schey, H.M., and Schwartz, J.L. 1967, *American Journal of Physics*, vol. 35, pp. 177–186. [1]

Galbraith, I., Ching, Y.S., and Abraham, E. 1984, *American Journal of Physics*, vol. 52, pp. 60–68. [2]

19.3 Initial Value Problems in Multidimensions

The methods described in §19.1 and §19.2 for problems in $1 + 1$ dimension (one space and one time dimension) can easily be generalized to $N + 1$ dimensions. However, the computing power necessary to solve the resulting equations is enormous. If you have solved a one-dimensional problem with 100 spatial grid points, solving the two-dimensional version with 100×100 mesh points requires *at least* 100 times as much computing. You generally have to be content with very modest spatial resolution in multidimensional problems.

Indulge us in offering a bit of advice about the development and testing of multidimensional PDE codes: You should always first run your programs on *very small* grids, e.g., 8×8, even though the resulting accuracy is so poor as to be useless. When your program is all debugged and demonstrably stable, *then* you can increase the grid size to a reasonable one and start looking at the results. We have actually heard someone protest, "my program would be unstable for a crude grid, but I am sure the instability will go away on a larger grid." That is nonsense of a most pernicious sort, evidencing total confusion between accuracy and stability. In fact, new instabilities sometimes do show up on *larger* grids; but old instabilities never (in our experience) just go away.

Forced to live with modest grid sizes, some people recommend going to higher-order methods in an attempt to improve accuracy. This is very dangerous. Unless the solution you are looking for is known to be smooth, and the high-order method you