Finite Difference Techniques

Consider a time-dependent partial differential equation in one state, $x \in [\underline{x}, \bar{x}]$

$$\frac{\partial V}{\partial t} + \alpha \frac{\partial V}{\partial x} + \beta \frac{\partial^2 V}{\partial x^2} = S, \tag{2.1}$$

where $V = V(x,t)$, and α, β, and S are functions of x, V, and derivatives of V. To write a stable and accurate finite-difference approximation to this equation we will consider each term independently. In the following discussion we will consider a uniform spacing in x and t, of size Δx and Δt, respectively.[1] We adopt the shorthand notation that $V_i^n = V(i\Delta x, n\Delta t)$, where i and n are integers.

Let us set $\beta = 0$ and $S = 0$ initially. This new equation is known as the *convection problem* in the computational fluid dynamics literature and has been studied extensively for a constant convection speed, α. This equation is known as the convection problem because the initial function $V(x, 0)$ is shifted, or convected at a speed α to the right (increasing x) if $\alpha > 0$ and to the left (decreasing x) if $\alpha < 0$. That is, the exact solution to this equation is $V(x,t) = V(x - \alpha t, 0)$, where α is constant. A wide variety of algorithms have been developed for solving this problem.[2] Here we consider the upwind method first proposed by Steger and Warming (1979). This method uses a one-sided finite-difference approximation to the spatial derivative, $\partial V/\partial x$. Backward and forward first-order one-sided difference approximations to this derivative are respectively

$$\begin{aligned} \left.\frac{\partial V(x,t)}{\partial x}\right|_{x=i\Delta x,\, t=n\Delta t} &= \frac{V_i^n - V_{i-1}^n}{\Delta x} + \mathcal{O}(\Delta x^2) \\ \left.\frac{\partial V(x,t)}{\partial x}\right|_{x=i\Delta x,\, t=n\Delta t} &= \frac{V_{i+1}^n - V_i^n}{\Delta x} + \mathcal{O}(\Delta x^2) \end{aligned} \tag{2.2}$$

The backward difference approximation is used for $\alpha > 0$ and the forward difference for $\alpha < 0$. The reasoning for this switch in difference scheme depending on the sign of α is based on the direction of the shift of the solution. Observe that the solution at x is only a function of the solution to the left of x if $\alpha > 0$. Thus when $\alpha > 0$, we use a backward difference that uses the solution from the left only. A finite-difference approximation to the derivative that switches depending on the sign of α is then

$$\alpha \left.\frac{\partial V(x,t)}{\partial x}\right|_{x=i\Delta x,\, t=n\Delta t} = \frac{1}{2}(\alpha - |\alpha|)\frac{V_{i+1}^n - V_i^n}{\Delta x} + \frac{1}{2}(\alpha + |\alpha|)\frac{V_i^n - V_{i-1}^n}{\Delta x} + \mathcal{O}(\Delta x^2). \tag{2.3}$$

[1] We are using a uniform mesh spacing for convenience. A non-equispaced mesh is a simple extension.

[2] For example, see Anderson, Tannehill, and Pletcher (1984).

This differencing scheme has been studied for the constant α case, but it also has been used for the case where α is variable, as in the problems of interest here. In this case, the local value of α is used in the expression above and the same reasoning applies for the shift in the differencing scheme depending on the sign of $\alpha(x,t)$. Then we have

$$\alpha(x,t)\frac{\partial V(x,t)}{\partial x}\bigg|_{x=i\Delta x,\,t=n\Delta t} = \frac{1}{2}(\alpha_i^n - |\alpha_i^n|)\frac{V_{i+1}^n - V_i^n}{\Delta x} + \frac{1}{2}(\alpha_i^n + |\alpha_i^n|)\frac{V_i^n - V_{i-1}^n}{\Delta x} + \mathcal{O}(\Delta x^2). \tag{2.4}$$

A higher order approximation to the derivative may be used to increase the accuracy of the method so long as it is one-sided.

The next question is how to advance the solution in time. There are essentially two types of time-advancement schemes that may be used. The first is the class of explicit methods where the solution at time level $n+1$ is obtained from a direct extrapolation in time from the solution at time level n. Secondly, the implicit methods use the solution at time level n and at the unknown time level $n+1$ to advance the solution to that time. Generally, the explicit methods are easier to program and take less computational effort per time step than the implicit methods. However, the maximum time step is strictly limited, whereas the implicit methods are stable for much larger time steps. Therefore, for problems where time accuracy is important, an explicit method will likely be used, while for cases where a steady-state solution is required, an implicit method is usually a good choice and requires less total computer time to obtain the result.

A first-order explicit time advancement scheme for the differential equation of interest is

$$\frac{V_i^{n+1} - V_i^n}{\Delta t} + \alpha_i^n \frac{DV_i^n}{\Delta x} + \mathcal{O}(\Delta x^2) + \mathcal{O}(\Delta t^2) = 0 \tag{2.5}$$

where $\alpha_i^n DV_i^n/\Delta x + \mathcal{O}(\Delta x^2)$ is our shorthand notation for the right hand side of (2.4). With this explicit method, we can solve for V_i^{n+1} directly from the solution at time level n. This method is stable for the constant coefficient case for time steps such that

$$CFL \equiv \alpha\frac{\Delta t}{\Delta x} \leq 1. \tag{2.6}$$

This is the Courant or CFL (Courant-Friedrichs-Lewy) condition, which implies that for large convection speeds or for small Δx the time step is severely limited[3]. For the non-constant α case, the method will usually be stable for a time step chosen such that

$$\alpha_i\frac{\Delta t}{\Delta x} \leq 1, \quad \text{for all } i. \tag{2.7}$$

[3] For the definition of stability and the derivation of the stability constraint, see Anderson, Tannehill, and Pletcher (1984).

A simple first-order implicit time advancement scheme for the same differential equation is

$$\frac{V_i^{n+1} - V_i^n}{\Delta t} + \alpha_i^{n+1} \frac{DV_i^{n+1}}{\Delta x} = \mathcal{O}(\Delta x^2) + \mathcal{O}(\Delta t^2). \tag{2.8}$$

In this case we cannot solve directly for V_i^{n+1} because the operator has terms involving V_{i+1}^{n+1} and V_{i-1}^{n+1}. However, equation (2.8) can be solved by rewriting it in terms of the change in each variable

$$\begin{aligned} \delta V_i^n &= V_i^{n+1} - V_i^n \\ \delta \alpha_i^n &= \alpha_i^{n+1} - \alpha_i^n \end{aligned} \tag{2.9}$$

Then equation (2.8) becomes

$$\begin{aligned} \delta V_i^n \left(\frac{\Delta x}{\Delta t} + |\alpha_i^n| \right) + \frac{1}{2} \delta V_{i+1}^n \left(\alpha_i^n - |\alpha_i^n| \right) + \frac{1}{2} \delta V_{i-1}^n \left(\alpha_i^n + |\alpha_i^n| \right) = \\ - \frac{1}{2} \left\{ \alpha_i^n (V_{i+1}^n - V_{i-1}^n) - |\alpha_i^n| (V_{i+1}^n - 2V_i^n + V_{i-1}^n) \right\} \\ + \frac{1}{2} \left\{ |\delta \alpha_i^n| (V_{i+1}^n - V_{i-1}^n) - \delta \alpha_i^n (V_{i+1}^n - 2V_i^n + V_{i-1}^n) \right\} \end{aligned} \tag{2.10}$$

where we have neglected terms of second order in δ and we have assumed that $|\alpha + \delta\alpha| = |\alpha| + |\delta\alpha|$ because $\delta\alpha$ is small enough that it does not change the sign of α. The change in α can be related to the change in V through the expression

$$\delta \alpha_i^n \simeq \left. \frac{\partial \alpha}{\partial V} \right|_{x = i\Delta x,\, t = n\Delta t} \delta V_i^n \tag{2.11}$$

If we substitute this expression into (2.10), we obtain an equation involving δV_{i+1}^n, δV_i^n, and δV_{i-1}^n. The solution of this equation involves the inversion of a tri-diagonal matrix of dimension equal to the number of mesh points in x.

For the constant α case, the method is theoretically stable for all Δt, and for the variable case α, there is generally a practical time step limitation which is typically an order of magnitude greater than the explicit method. Therefore, the implicit method requires more computational effort with the benefit of allowing a larger time step than the explicit method. As asserted above, implicit methods are usually faster than explicit methods when a steady-state solution is required. However the explicit method is generally preferred if a time-dependent result is required because time accuracy is lost when large time steps are taken. Examples of the convergence rates and computer time required for sample programs will be given in the following section.

Now consider (2.1) with $\alpha = 0$ and $S = 0$. In this case we are solving the diffusion problem that has been studied extensively for the application to heat transfer and viscous fluid dynamics problems. This differential equation is easier to solve numerically than the convection problem and the differencing of the second derivative, $\partial^2 V/\partial x^2$ is done as follows

$$\beta \frac{\partial^2 V}{\partial x^2}\bigg|_{x=i\Delta t,\, t=n\Delta t} = \beta_i^n \frac{V_{i+1}^n - 2V_i^n + V_{i-1}^n}{\Delta x^2} + \mathcal{O}(\Delta x^3) \tag{2.12}$$

for an explicit first-order time advancement scheme. For an implicit method, the finite-difference approximation is

$$\beta \frac{\partial^2 V}{\partial x^2}\bigg|_{x=i\Delta t,\, t=n\Delta t} = \beta_i^{n+1} \frac{V_{i+1}^{n+1} - 2V_i^{n+1} + V_{i-1}^{n+1}}{\Delta x^2} + \mathcal{O}(\Delta x^3) \tag{2.13}$$

Unlike in the convection case, there is no need to use a different differencing scheme for each sign of β. The explicit algorithm will be stable for all time steps such that

$$Fo \equiv \beta \frac{\Delta t}{\Delta x^2} \leq \frac{1}{2} \tag{2.14}$$

for the case where β is a constant. The implicit algorithm is unconditionally stable for this case. The implicit method involves solving a tridiagonal matrix as in equation (2.10), whereas the explicit method can be advanced directly. As a result many of the same comments made for the convection problem apply for this differential equation.

The final portion of the differential equation (2.1), is the the source term S

$$\frac{\partial V}{\partial t} = S. \tag{2.15}$$

For the case where S is a constant, the solution is trivial and an explicit or an implicit time advancement scheme will work well. However, for the case where S is proportional to V, say $S = \rho V$, the finite difference algorithm that should be used depends on the sign of ρ. This can be seen by considering the two first-order methods that we have using above. For the explicit case, the solution at time level $n+1$ is

$$V_i^{n+1} = \big(1 + \rho\Delta t\big)V_i^n, \tag{2.16}$$

and for the implicit method it is

$$V_i^{n+1} = \frac{V_i^n}{1 - \rho\Delta t}. \tag{2.17}$$

If we are interested in achieving a steady-state solution in a minimum of time steps, we would like to take as large a time step as possible. For the two previous aspects of the problem, this required us to use an implicit method. However, in this case, the sign of ρ determines the type of time advancement scheme that should be used. We can see this by plotting the results of equations (2.16) and (2.17) for several time steps. A constant $\rho\Delta t = 2$ and an initial V of 1 are used. The result of this simple test is plotted in Figure 0a. We see that for this case where $\rho > 0$, the explicit method lags behind the exact solution, but moves in the right direction. However, the implicit method oscillates between plus and minus one. For the case where $\rho\Delta t = -2$ similar results are obtained except in reverse as seen in Figure 0b. We see that the implici method moves with the exact solution, whereas the explicit method oscillates. It should be noted that neither of these solutions is time accurate as Δt would be chosen so that V_i^{n+1} would only change by some small portion of V_i^n if a time-accurate solution was required. From this simple demonstration, we see that using an explicit method for cases where there is exponential growth ($\rho > 0$) is recommended, while for problems where $\rho < 0$, an implicit method should be used. Using an explicit time-advancement scheme for the source term in equation (2.1) does not preclude using an implicit method for the other terms. The finite-difference algorithm would just evaluate S at the n time level while the other terms would be evaluated at time level $n + 1$.