AM 129, Fall 2023
Final Project Type C – Numerical PDE:
Linear Advection-Diffusion Equation

1. Overview

We are interested in solving the linear advection-diffusion partial differential equation (PDE)

$$u_t + au_x = \kappa u_{xx}, \tag{1}$$

where $a \in \mathbb{R}$ is a constant advection velocity and $\kappa \geq 0$ is a constant diffusion coefficient. Note that if $\kappa < 0$ equation (1) becomes a *backward heat equation* which is an ill-posed problem that we won't consider.

2. Initial and boundary conditions

We impose an initial condition at $t = 0$,

$$u(x, 0) = u_0(x), \tag{2}$$

and (Dirichlet) boundary conditions at $x_a \leq x \leq x_b$ for all time

$$u(x_a, t) = g_a(t) \text{ and } u(x_b, t) = g_b(t), \text{ for } t > 0. \tag{3}$$

3. Discretization in space and time

To handle this problem numerically we need to discretize both the spatial and temporal domains. Consider N equally spaced grid points in space, and M equally spaced points in time

$$x_i = x_a + (i - \frac{1}{2})\Delta x, \;\; i = 1, ..., N, \tag{4}$$

$$t^n = n\Delta t, \;\; n = 0, ...M, \tag{5}$$

where the uniform grid spacing Δx is given by $\Delta x = (x_b - x_a)/N$ as a function of N. On the other hand, the temporal spacing Δt should be chosen carefully according to the CFL condition for numerical stability (see Section 5.3.).
Cell interface grid points are written using the 'half-integer' indices:

$$x_{i+\frac{1}{2}} = x_i + \frac{\Delta x}{2}. \tag{6}$$

Our goal in the following is to approximate the solution of equation (1) on these points. As useful shorthand we'll use the notation u_i^n to denote the approximate solution at x_i and time t^n. That is, $u(x_i, t^n) \approx u_i^n$.

4. Imposing Boundary Conditions via Guard-cell (or ghost-cell)

We can introduce so-called ‘guard-cells’ or ‘ghost-cells’ (simply GCs) on each end of the domain

$$x_0 = x_a - \Delta x/2 \tag{7}$$
$$x_{N+1} = x_b + \Delta x/2. \tag{8}$$

These two extra GC points will help us impose boundary conditions. The finite difference methods discussed in the next section will advance the solution in time over the interior cells, $x_i,\ 1 \le i \le N$. However, this evolution will depend on the value of the solution on these ghost cells

$$u_0^n = g_a(t^n), \ \ u_{N+1}^n = g_b(t^n), \tag{9}$$

where $g_a(t)$ and $g_b(t)$ are the prescribed boundary conditions in (3)

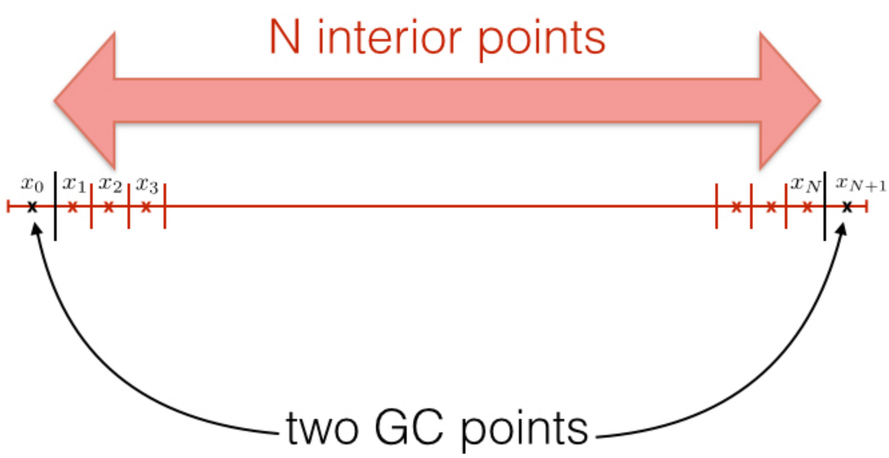

Figure 1. Discrete grid configuration including guard cells.

5. Discrete schemes for the advection and diffusion equations

5.1. Finite difference scheme for 1D advection

First consider a simple advection equation with constant speed $a > 0$,

$$u_t + au_x = 0, \text{ with } u(x,0) = u_0(x), \tag{10}$$

which arises from (1) by taking $\kappa = 0$.

Recall that our approximate solution at each (x_i, t^n) point is,

$$u_i^n = u(x_i, t^n). \tag{11}$$

The forward difference approximation scheme for first-order spatial and temporal derivatives writes, respectively, as,

$$u_x(x,t) = \frac{u(x+\Delta x, t) - u(x,t)}{\Delta x} + O(\Delta x), \tag{12}$$

$$u_t(x,t) = \frac{u(x, t+\Delta t) - u(x,t)}{\Delta t} + O(\Delta t). \tag{13}$$

Dropping the truncation error terms $O(\Delta x)$ and $O(\Delta t)$ yields a simple first-order difference scheme that approximates the advection PDE. As a result, we arrive at a first-order accurate (both spatially and temporally) discrete difference equation from an analytic differential equation,

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} + a\frac{u_{i+1}^n - u_i^n}{\Delta x} = 0, \tag{14}$$

which gives a temporal update scheme of u_i^{n+1} in terms of the known data at $t = t^n$,

$$u_i^{n+1} = u_i^n - a\frac{\Delta t}{\Delta x}\Big(u_{i+1}^n - u_i^n\Big). \tag{15}$$

Notice that the future solution u_i^{n+1} at each grid point depends on the past solution at that grid point, u_i^n, as well as the past solution at the grid point u_{i-1}^n. For $i = 1$ you can see that the evolution will require setting the boundary condition u_0^n at the left GC point.
On the other hand, if we use a backward difference scheme for u_x

$$u_x(x,t) = \frac{u(x,t) - u(x-\Delta x, t)}{\Delta x} + O(\Delta x), \tag{16}$$

we arrive at another first-order difference equation

$$u_i^{n+1} = u_i^n - a\frac{\Delta t}{\Delta x}\Big(u_i^n - u_{i-1}^n\Big). \tag{17}$$

Another approximation is available using the centered differencing scheme,

$$u_i^{n+1} = u_i^n - a\frac{\Delta t}{2\Delta x}\Big(u_{i+1}^n - u_{i-1}^n\Big). \tag{18}$$

The choice of Δt should be small enough, satisfying:

$$|a|\Delta t \le \Delta x. \tag{19}$$

This is called the Courant–Friedrichs–Lewy (CFL) condition (or simply the Courant condition). The CFL condition describes a necessary (but *not* sufficient) condition for convergence when solving time-dependent PDEs numerically (e.g., by finite difference, finite volume, or Galerkin methods).

5.2. Finite difference scheme for 1D diffusion

Consider now the heat equation (or diffusion equation)

$$u_t = \kappa u_{xx}, \tag{20}$$

where $\kappa > 0$. This arises from the original equation (1) by taking $a = 0$. We will discretize this equation in a very similar manner to the last one. The key difference is that equation (20) now contains a second-order derivative in space. To handle this, we'll adopt the standard second-order central difference difference scheme for u_{xx},

$$u_{xx}(x,t) = \frac{u(x+\Delta x,t) - 2u(x,t) + u(x-\Delta x,t)}{\Delta x^2} + O(\Delta x^2), \tag{21}$$

which gives a discrete form of our explicit finite difference scheme for the heat equation,

$$u_i^{n+1} = u_i^n + \kappa\frac{\Delta t}{\Delta x^2}\Big(u_{i+1}^n - 2u_i^n + u_{i-1}^n\Big). \tag{22}$$

Similar to the 1D advection case, we choose Δt satisfying

$$\kappa \Delta t \le \frac{\Delta x^2}{2}. \tag{23}$$

Notice that equation (21) can be obtained by applying the forward and backward difference schemes consecutively,

$$\begin{aligned} u_{xx}(x,t) &= \frac{\partial}{\partial x}\Big(\frac{\partial u}{\partial x}\Big) && (24)\\ &\approx \frac{\partial}{\partial x}\frac{u(x+\Delta x,t)-u(x,t)}{\Delta x} && (25)\\ &\approx \frac{u(x+\Delta x,t) - 2u(x,t) + u(x-\Delta x,t)}{\Delta x^2}, && (26) \end{aligned}$$

though finding the error term is harder from this perspective. *Aside:* You can also find this by using centered differences at half-integer grid points, then a centered difference on those to arrive at the same formula.

5.3. The CFL condition

As mentioned, the CFL condition provides an upper bound for Δt that is necessary (but not sufficient) for the numerical scheme to converge to the true solution as Δx and Δt are taken to zero. Let C to be the CFL number defined as

$$C = \max_p |a_p| \frac{\Delta t}{\Delta x}, \tag{27}$$

for the advection problem, and

$$C = \max_p \kappa_p \frac{2\Delta t}{\Delta x^2}, \tag{28}$$

for the diffusion problem. Here p is the number of all available wave speeds a_p or diffusion coefficients κ_p, respectively. Note that $p = 1$ for a *scalar* equation, which is the current case. The CFL condition is thus satisfied by ensuring $0 < C \leq 1$.

It is important to note that the CFL condition is only a *necessary* condition for stability (and hence convergence). It is not always *sufficient* to guarantee stability, and a numerical method satisfying the CFL condition can still fail to converge.

Note that the above CFL conditions in equations (19) and (23) for choosing Δt_{advect} and Δt_{diff}, respectively, need to be combined together for a linear advection-diffusion equation:

$$\Delta t = C \min\Big(\frac{\Delta x}{|a|}, \frac{\Delta x^2}{2\kappa}\Big), \tag{29}$$

where again $0 < C \leq 1$.

6. A short List of Finite Difference Methods for Linear Problems

There are a couple of finite difference (FD) methods for solving the advection part of the PDE, $u_t + au_x = 0$. We assume $a > 0$ for Beam-Warming and Fromm's methods. One can easily get appropriate forms for these two methods for $a < 0$.

- Upwind for $a > 0$ (FTBS – Forward Time Backward Space)

$$u_i^{n+1} = u_i^n - \frac{a\Delta t}{\Delta x}\Big(u_i^n - u_{i-1}^n\Big) \tag{30}$$

- Downwind for $a > 0$ (FTFS – Forward Time Forward Space)

$$u_i^{n+1} = u_i^n - \frac{a\Delta t}{\Delta x}\Big(u_{i+1}^n - u_i^n\Big) \tag{31}$$

- Centered for any a (FTCS – Forward Time Centered Space)

$$u_i^{n+1} = u_i^n - \frac{a\Delta t}{2\Delta x}\Big(u_{i+1}^n - u_{i-1}^n\Big) \tag{32}$$

- Leapfrog for any a

$$u_i^{n+1} = u_i^{n-1} - \frac{a\Delta t}{2\Delta x}\Big(u_{i+1}^n - u_{i-1}^n\Big) \tag{33}$$

- Lax-Friedrichs (LF) for any a

$$u_i^{n+1} = \frac{1}{2}\Big(u_{i+1}^n + u_{i-1}^n\Big) - \frac{a\Delta t}{2\Delta x}\Big(u_{i+1}^n - u_{i-1}^n\Big) \tag{34}$$

- Lax-Wendroff (LW) for any a

$$u_i^{n+1} = u_i^n - \frac{a\Delta t}{2\Delta x}\Big(u_{i+1}^n - u_{i-1}^n\Big) + \frac{1}{2}\Big(\frac{a\Delta t}{\Delta x}\Big)^2\Big(u_{i+1}^n - 2u_i^n + u_{i-1}^n\Big) \quad (35)$$

- Beam-Warming (BW) for $a > 0$

$$u_i^{n+1} = u_i^n - \frac{a\Delta t}{2\Delta x}\Big(3u_i^n - 4u_{i-1}^n + u_{i-2}^n\Big) + \frac{1}{2}\Big(\frac{a\Delta t}{\Delta x}\Big)^2\Big(u_i^n - 2u_{i-1}^n + u_{i-2}^n\Big) \quad (36)$$

- Fromm's method for $a > 0$

$$\begin{aligned} u_i^{n+1} = \quad & u_i^n - \frac{a\Delta t}{\Delta x}\Big(u_i^n - u_{i-1}^n\Big) - \frac{1}{4}\frac{a\Delta t}{\Delta x}\Big(1 - \frac{a\Delta t}{\Delta x}\Big)\Big(u_{i+1}^n - u_i^n\Big) \\ & + \frac{1}{4}\frac{a\Delta t}{\Delta x}\Big(1 - \frac{a\Delta t}{\Delta x}\Big)\Big(u_{i-1}^n - u_{i-2}^n\Big) \end{aligned} \quad (37)$$

Note: On the contrary, there is not so much to do when spatially discretizing the diffusion part of PDE. This is because the physical process described by parabolic PDEs is diffusive and smooth. As such, it does not require the same numerical attention present in resolving the advective process, or more generally from tracing the wave information in hyperbolic PDEs.

Alternatively, you might notice that the CFL condition on the diffusive term is much more restrictive. There are more complicated discretizations in time that deal with this fact, but they are beyond the scope of this outline.

7. Examples of advection: continuous and discontinuous

In Fig. 2 we display five different numerical solutions to the linear advection equation with two different initial conditions. In all cases periodic boundary conditions have been used. The panels on the left column show a smooth $\sin(2\pi x)$ wave initialized on $x \in [0, 1]$. The sine wave is solved numerically with – from top to bottom – (1) Upwind method, (2) Lax-Friedrichs, (3) Lax-Wendroff, (4) Beam-Warming, and (5) Fromm's method. In the right column, the same methods are applied, in the same order, to solve the equation with a discontinuous initial condition

$$u_0(x) = \begin{cases} 1 \text{ for } x < 0.5 \\ -1 \text{ for } x > 0.5. \end{cases} \quad (38)$$

All numerical methods solve the sine wave problem until the wave completes one full cycle on the periodic domain. Spatially, 64 grid cells are used ($N = 64$).

The same number of grid cells is used for the discontinuous case where the solutions have been integrated on a domain with inflow and outflow boundary conditions at the left and right sides of the domain respectively. The solution is advanced until the location of the discontinuity reaches $x = 0.8$.

There are two first-order accurate methods (upwind and Lax-Friedrichs) and three second-order accurate methods (Lax-Wendroff, Beam-Warming, and Fromm's

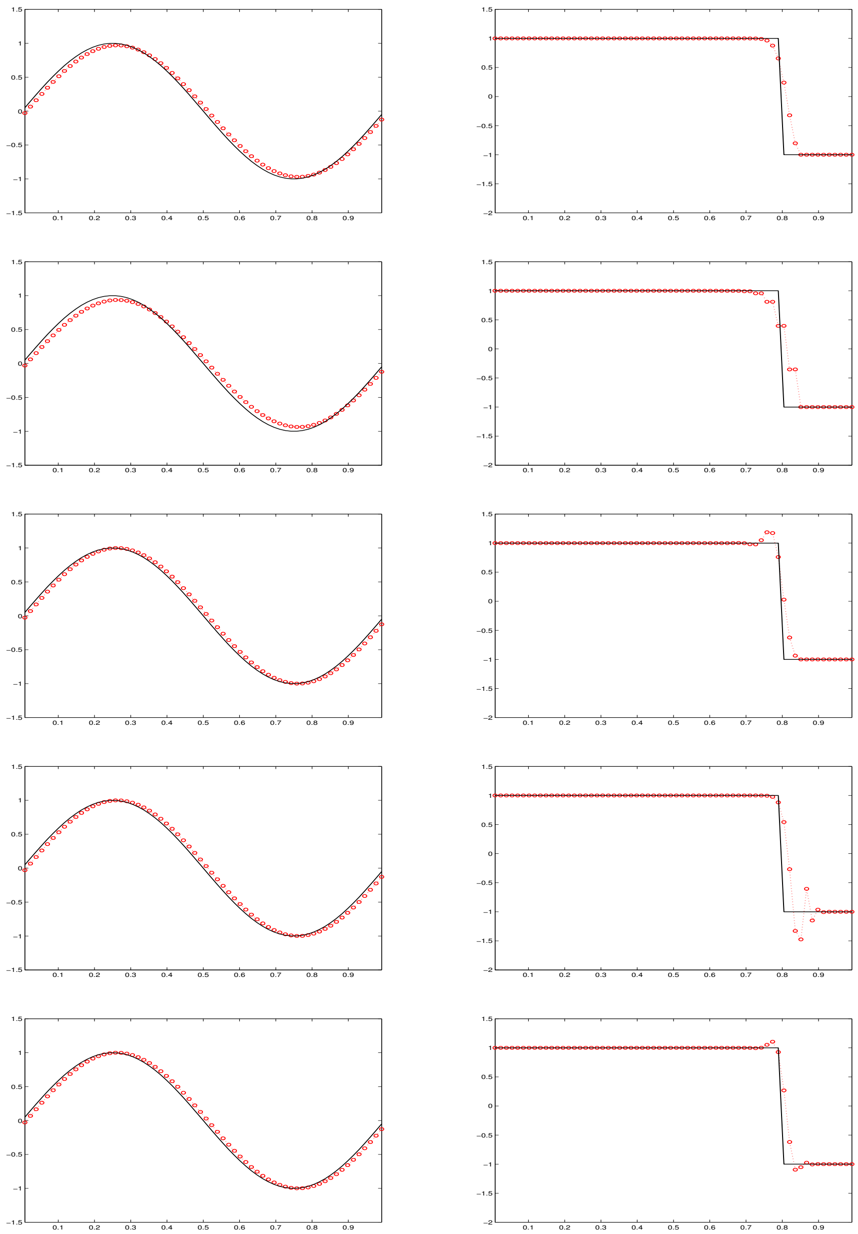

Figure 2. Numerical (red circles) and exact (solid black lines) solutions to the scalar advection equation $u_t + au_x = 0, a > 0$ with two different initial conditions: a sine wave in the left column, and a discontinuous profile in the right column. Solutions from five different schemes are shown from top to bottom: (1) Upwind, (2) Lax-Friedrichs, (3) Lax-Wendroff, (4) Beam-Warming, (5) Fromm's method.

method). We note that all methods behave similarly on the smooth flow. On the contrary, there are two distinctive solution characteristics – dissipation and oscillations – on the discontinuous flow: the first-order methods give very smeared solutions, while the second-order methods give oscillations. Understanding these types of behaviors is a key topic in computational fluid dynamics.