Numerical Methods in ESP

Numerical Methods II

Graziano Giuliani

International Centre for Theoretical Physics

Second Semester 2023-24

/afs/ictp.it/public/g/ggiulian/WORLD/num2_lesson2.pdf

The linear advection equation

Consider the partial linear differential equation:

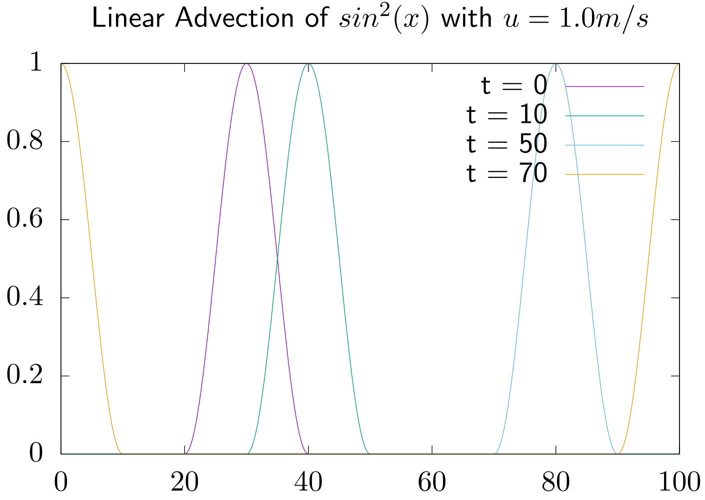
$$\frac{\partial \phi}{\partial t} + u\frac{\partial \phi}{\partial x} = 0 \tag{1}$$

where u is a constant velocity in the domain $0 \leq x \leq 1$ with periodic boundary conditions $\phi(0,t) = \phi(1,t)$ and initial condition given as $\phi(x,0) = F(x)$, with F being a given function.
The analytical solution of the problem is:

$$\phi(x,t) = F(x - ut) \tag{2}$$

We call this equation the Linear Advection Equation as it describes the shape preserving movement of the initial profile $F(x)$ advected with speed u over time t.

Shape shifting with time



Advection discretization: FTFS

- approximate the spatial and temporal derivative with forward difference formula. i.e:

$$\frac{\phi_j^{n+1} - \phi_j^n}{\Delta t} + u\frac{\phi_{j+1}^n - \phi_j^n}{\Delta x} = 0 \tag{3}$$

- rearranging for the unknown at next timestep gives the formula for the FORWARD IN TIME FORWARD IN SPACE (FTFS) scheme:

$$\begin{gathered} \phi_j^{n+1} = (1+c)\phi_j^n - c\phi_{j+1}^n \\ c = u\frac{\Delta t}{\Delta x} \end{gathered} \tag{4}$$

Advection discretization: FTBS

- approximate the spatial derivative with backward difference formula and temporal derivative with forward difference formula. i.e:

$$\frac{\phi_j^{n+1} - \phi_j^n}{\Delta t} + u\frac{\phi_j^n - \phi_{j-1}^n}{\Delta x} = 0 \tag{5}$$

- rearranging for the unknown at next timestep gives the formula for the FORWARD IN TIME BACKWARD IN SPACE (FTBS) scheme:

$$\begin{gathered} \phi_j^{n+1} = (1-c)\phi_j^n + c\phi_{j-1}^n \\ c = u\frac{\Delta t}{\Delta x} \end{gathered} \tag{6}$$

Upstream and Downstream

- We identify a scheme as upstream (downstream) according to the relation between the advection discretization formula and the direction in which the initial shape is moving with the notation:
  - UPSTREAM - the advection discretization formula computes the new value in a point by using the old value of the function in a point upstream of that point wrt the advection velocity.
  - DOWNSTREAM - the advection discretization formula computes the new value in a point by using the old value of the function in a point downstream of that point wrt the advection velocity.

Given the above definitions, we have these possible cases:

(1) $u > 0$

- FTBS is upstream (uses ϕ_{j-1}^{n} to compute ϕ_j^{n+1})
- FTFS is downstream (uses ϕ_{j+1}^{n} to compute ϕ_j^{n+1})

(2) $u < 0$

- FTBS is downstream (uses ϕ_{j-1}^{n} to compute ϕ_j^{n+1})
- FTFS is upstream (uses ϕ_{j+1}^{n} to compute ϕ_j^{n+1})

Von Neumann stability analysis

- On a periodic domain any function of L^1 class can be decomposed into Fourier components. If the function is the solution of a linear differential equation with constant coefficients then its behavior can be determined by looking at the behavior of each Fourier component.
- Similarly, the stability of a linear numerical method can be found by considering a single Fourier component and seeing whether it grows in time.
- Replacing $\phi_j^n \approx \phi(x_j, t^n)$ with $A^n e^{ikj\Delta x}$ in our numerical method, we first obtain an equation for the amplification factor A.
- the von Neumann method requires usually the amplification factor to be bounded such that $|A| \leq 1$
- the von Neumann stability analysis is a necessary and sufficient condition for the stability of a linear finite difference equation with constant coefficient. For nonlinear equations however, it is a necessary but not a sufficient condition.

Stability analysis for FTFS scheme 1

- Substitute $\phi_j^n = A^n e^{ikj\Delta x}$ into equation [3] to obtain:

$$\frac{A^{n+1}e^{ikj\Delta x} - A^n e^{ikj\Delta x}}{\Delta t} = -u\left(\frac{A^n e^{ik(j+1)\Delta x} - A^n e^{ikj\Delta x}}{\Delta x}\right) \tag{7}$$

- If we set $c = u\Delta t/\Delta x$, we can write [7] as:

$$A * A^n e^{ikj\Delta x} = A^n e^{ikj\Delta x} - cA^n e^{ikj\Delta x}\left(e^{ik\Delta x} - 1\right) \tag{8}$$

$$A = 1 - c(e^{ik\Delta x} - 1) \tag{9}$$

- The numerical solution will be stable if $|A| \leq 1$. Given that for a complex number $|A|^2 = A * \overline{A}$ we can write:

$$|A|^2 = (1 - c(e^{ik\Delta x} - 1)) * (1 - c(e^{-ik\Delta x} - 1)) \tag{10}$$

Stability analysis for FTFS scheme 2

- Remembering now some trigonometry:

$$\begin{aligned} |A|^2 &= 1 + 2c + 2c^2 - c(e^{ik\Delta x} + e^{-ik\Delta x}) - c^2(e^{ik\Delta x} + e^{-ik\Delta x}) \\ &= 1 + 2c(1 - cos(k\Delta x)) + 2c^2(1 - cos(k\Delta x)) \\ &= 1 + 2c(1 + c)(1 - cos(k\Delta x)) \leq 1 \end{aligned} \tag{11}$$

- Since the term $(1 - cos(k\Delta x)) \geq 0$ for all wavenumbers k, the above inequality reduces to

$$c(1 + c) \leq 0 \tag{12}$$

- IF $u > 0$ the inequality is never met, since $c > 0$. Thus:
  - *The FTFS scheme is unconditionally unstable for the solution of a linear advection problem with a positive velocity*. We can generalize this affirming that a downstream scheme is always unconditionally unstable.

Stability analysis for FTBS scheme 1

Let us know examine the upstream scheme in the case of positive velocity $u > 0$: the FTBS scheme

- Substitute $\phi_j^n = A^n e^{ikj\Delta x}$ into equation [5] to obtain:

$$\frac{A^{n+1}e^{ikj\Delta x} - A^n e^{ikj\Delta x}}{\Delta t} = -u\left(\frac{A^n e^{ikj\Delta x} - A^n e^{ik(j-1)\Delta x}}{\Delta x}\right) \quad (13)$$

- If we set $c = u\Delta t/\Delta x$, we can write [13] as:

$$A * A^n e^{ikj\Delta x} = A^n e^{ikj\Delta x} - cA^n e^{ikj\Delta x}\left(1 - e^{ik\Delta x}\right) \quad (14)$$

$$A = 1 - c(1 - e^{ik\Delta x}) \quad (15)$$

- The numerical solution will be stable if $|A| \leq 1$. Given that for a complex number $|A|^2 = A * \overline{A}$, we can write:

$$|A|^2 = (1 - c(1 - e^{ik\Delta x})) * (1 - c(1 - e^{-ik\Delta x})) \quad (16)$$

Stability analysis for FTBS scheme 2

- Remembering now some trigonometry:

$$\begin{aligned}|A|^2 &= 1 - 2c + 2c^2 + c(e^{ik\Delta x} + e^{-ik\Delta x}) - c^2(e^{ik\Delta x} + e^{-ik\Delta x}) \\ &= 1 - 2c + 2c\ cos(k\Delta x) + 2c^2 - 2c^2 cos(k\Delta x) \\ &= 1 - 2c(1-c)(1 - cos(k\Delta x)) \le 1 \end{aligned} \tag{17}$$

- Since the term $(1 - cos(k\Delta x)) \ge 0$ for all wavenumbers k, the above inequality reduces to

$$c(1-c) \ge 0 \tag{18}$$

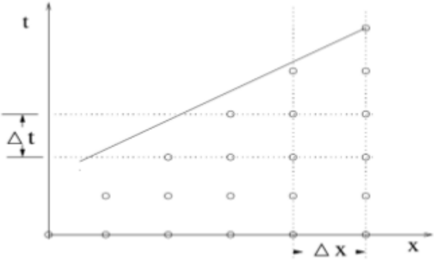
- IF $u > 0$ the inequality is met if $0 \le c \le 1$. Thus:
  - *The FTBS scheme is conditionally stable for the solution of a linear advection problem with a positive velocity*. We can generalize this affirming that an upstream scheme is conditionally stable.

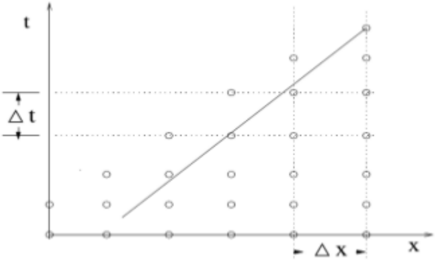
The Courant-Friedrichs-Lewy condition

- The CFL criterion states that a necessary condition for stability is that the domain of dependence of the numerical solution should include the domain of dependence of the original partial differential equation
- CFL condition requires that at each point x_j the fastest that the numerical solution can propagate is one grid point per time step. Time step Δt cannot exceed the limit $\Delta x/u$.
- The CFL condition has a clear geometric interpretation: The numerical domain of dependence must be larger than the physical domain of dependence, and not the other way around. If this wasn't the case, it would be impossible for the numerical solution to converge to the exact solution, since as the grid is refined there will always be relevant physical information that would remain outside the numerical domain of dependence. And, as we have seen, Lax theorem implies that if there is no convergence then the system is unstable.

The Courant-Friedrichs-Lewy graph

- CFL stability condition. For $u\Delta t \leq \Delta x$, the numerical domain of dependence is containing the physical domain of dependence, and the system is stable. For $u\Delta t > \Delta x$ we have the opposite situation, and the system is unstable.

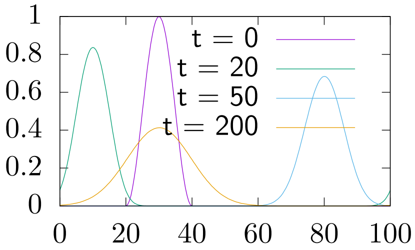




Notes on upstream scheme

- In the general case when u might change sign we must use FTBS when $u > 0$ and FTFS when $u < 0$, so that we always use information from the upstream side of the point whose new value we are trying to calculate.
  - FTBS scheme is stable if $u > 0$
  - FTFS scheme is stable if $u < 0$
- Excessive numerical diffusion is the dominant truncation error in the first-order upwind scheme.

Advection of $sin^2(x)$ with $u = -1.0m/s$ FTFS



Exercise on upstream scheme

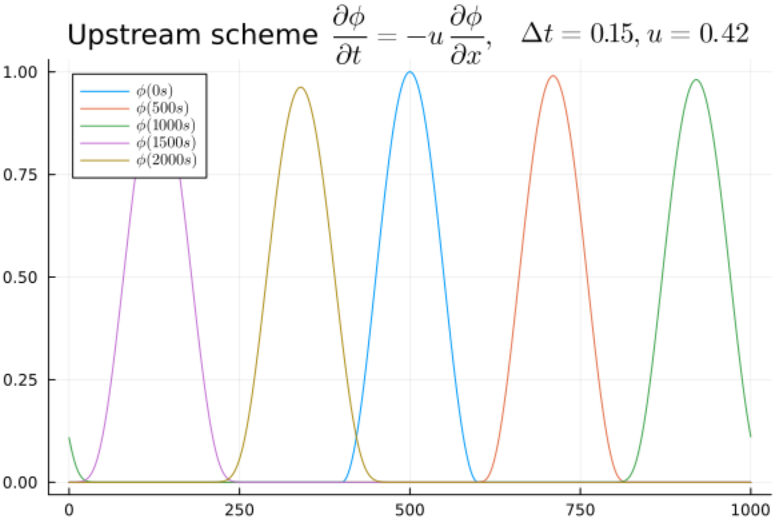
- Write a Fortran program to integrate the linear advection equation in [1] using an upwind scheme in the domain $0 \leq x \leq 1000m$ with advection velocity $u = 0.42m/s$. Let $\Delta x = 0.25m$ and assume periodic boundary conditions. Assume the initial shape to be:

$$\phi(x,0) = \begin{cases} 0.0 & for \quad x < 400 \\ sin(\pi * (x-400)/200))^2 & for \quad 400 \leq x < 600 \\ 0.0 & for \quad x > 600 \end{cases} \qquad (19)$$

- Integrate forward and show solutions from $t = 0s$ up to $t = 2000s$ every $500s$ and explain the characteristics of the solution. What happens if $\Delta t = \frac{\Delta x}{u}$, $(c == 1)$?

Hint: Select Δt to have a stable scheme remembering the CFL condition.

Expected result



Julia Code

```
x0 = 0.0; x1 = 1000.0; t0 = 0.0; t1 = 2000.0; u = 0.42; dx = 0.25; dt = 0.15; tp = 500.0;
c = u * dt/dx;
nx = round(Int64,(x1-x0)/dx) + 1;
x = LinRange(x0,x1,nx);
function phi0(x)
  if x < 400.0
    return 0.0;
  elseif x >= 400.0 && x < 600.0
    return (sin(pi * (x-400.0)/200.0))^2;
  else
    return 0.0;
  end;
end;
phi = map(phi0,x);
function ftfs(phi_now)
  (1+c) * phi_now - c * circshift(phi_now,-1)
end;
function ftbs(phi_now)
  (1-c) * phi_now + c * circshift(phi_now,1)
end;
t = t0;
while (t < t1)
  if u > 0
    phi_new = ftbs(phi);
  else
    phi_new = ftfs(phi);
  end;
  global phi[:] = phi_new;
  global t = t + dt;
  if mod(t,tp) < dt
    plot!(....)
  end;
end;
```