Numerical Methods in ESP

Numerical Methods II

Graziano Giuliani

International Centre for Theoretical Physics

Second Semester 2022-23

/afs/ictp.it/public/g/ggiulian/WORLD/num2_lesson4.pdf

The linear advection equation

Linear advection problem:

$$
\begin{aligned}
\frac{\partial \phi}{\partial t} &+ u\frac{\partial \phi}{\partial x} = 0 \\
\phi(x,t) &= F(x-ut)
\end{aligned}
\tag{1}
$$

Numerical solutions so far:

$$
\begin{aligned}
\phi_j^{n+1} &= (1+c)\phi_j^n - c\phi_{j+1}^n \\
\phi_j^{n+1} &= (1-c)\phi_j^n + c\phi_{j-1}^n \\
\phi_j^{n+1} &= \phi_j^{n-1} - c(\phi_{j+1}^n - \phi_{j-1}^n)
\end{aligned}
$$

where:

$$
c = u\frac{\Delta t}{\Delta x}
$$

The semi-Lagrangian scheme 1

- The advection equation can be written either in Eulerian or Lagrangian form:

$$\begin{aligned} \frac{D\phi}{Dt} &= 0 \quad \textit{Lagrangian} \\ \frac{\partial \phi}{\partial t} + u\frac{\partial \phi}{\partial x} &= 0 \quad \textit{Eulerian} \end{aligned} \tag{2}$$

- The semi-Lagrangian technique discretizes the Lagrangian formulation with finite differences:

$$\frac{\phi(x_j, t^{n+1}) - \phi(depx_j, t^n)}{\Delta t} = 0$$

The semi-Lagrangian scheme 2

$$\phi(x_j, t^{n+1}) = \phi(depx_j, t^n) \tag{3}$$

- The value of the function ϕ in a point x_j at time $t^n + \Delta t$ is the value of the function ϕ at time t^n at a point $depx_j$ which is called the departure point of the point x_j.
- the departure point $depx_j$ of the point x_j is the location at time t^n of the fluid parcel that reaches location x_j (arrival point) at time t^{n+1}.
- In general $depx_j$ is not a grid point.
- As a result, within the semi-Lagrangian framework [3], the advection problem reduces to two steps, for each point of the grid x_j:
  - computation of the departure $depx_j$ of x_j
  - interpolation of the values ϕ_j^n of the function ϕ at the computed departure point $depx_j$.

Departure point computation

- Let us assume the velocity u is constant. Then the departure point of a point x_j is:

$$depx_j = x_j - u\Delta t \tag{4}$$

- after introducing the integer and fractional part of the Courant number, m and α respectively:

$$u\Delta t = (j - m + \alpha)\Delta x, \quad m \in \mathbb{Z}, \quad \alpha \in (0,1)$$

We can write:

$$\begin{aligned} depx_j &= x_m - \alpha\Delta x \\ \alpha &= \frac{x_m - depx_j}{\Delta x} \end{aligned} \tag{5}$$

Advected field interpolation at departure point

From the definition of α in [5], two possible interpolation schemes at the departure point are (with a small change to the notation):

- Linear interpolation

$$\phi^n(depx_j) = \alpha\phi^n(x_{m-1}) + (1-\alpha)\phi^n(x_m) \tag{6}$$

- Cubic interpolation

$$\begin{aligned}\phi^n(depx_j) = & -\frac{\alpha(1-\alpha^2)}{6}\phi^n(x_{m-2}) \\ & +\frac{\alpha(1+\alpha)(2-\alpha)}{2}\phi^n(x_{m-1}) \\ & +\frac{(1-\alpha^2)(2-\alpha)}{2}\phi^n(x_m) \\ & -\frac{\alpha(1-\alpha)(2-\alpha)}{6}\phi^n(x_{m+1})\end{aligned} \tag{7}$$

Stability analysis for SL scheme

- Using the two timelevel and the linear interpolation scheme, we have:

$$\phi_j^{n+1} = \phi^n(depx_j) = (1-\alpha)\phi^n(x_m) + \alpha\phi^n(x_{m-1}) \tag{8}$$

- Assuming a solution of the form $\phi_j^n = A^n e^{ikj\Delta x}$, substituting in the [8] we get:

$$|A|^2 = \sqrt{1 - 2\alpha(1-\alpha)\left[1 - cos(k\Delta x)\right]} \tag{9}$$

  therefore $|A| \leq 1$ if $\alpha(1-\alpha) \geq 0$ i.e. $0 \leq \alpha \leq 1$.

- The scheme is stable if the interpolation points are the two nearest ones to the departure point, but it is neutral only if $\alpha = 0$ or $\alpha = 1$, i.e. when no interpolation is needed.

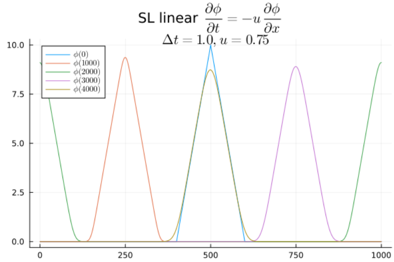
Exercise on SL scheme

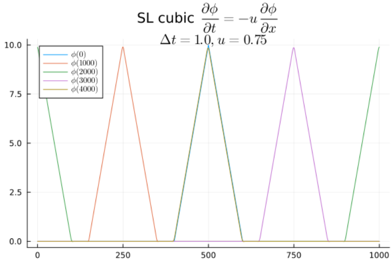
- Write a program to integrate the linear advection equation in [1] using the SL scheme with linear interpolation in the domain $0 \le x \le 1000m$ with advection velocity $u = 0.75m/s$. Let $\Delta x = 0.5m$ and assume periodic boundary conditions. Assume the initial shape to be:

$$\phi(x,0) = \begin{cases} 0.0 & for \quad x < 400 \\ 0.1(x - 400.0) & for \quad 400 \le x < 500 \\ 20.0 - 0.1(x - 400.0) & for \quad 500 \le x \le 600 \\ 0.0 & for \quad x > 600 \end{cases} \tag{10}$$

- Integrate forward and show solutions from $t = 0s$ up to $t = 2000s$ every $250s$ and explain the characteristics of the solution.
- Repeat the exercise using the cubic interpolation
- Do we need to worry about the CFL condition? Explain why or why not.

Expected result





Julia Code

```
phi_new = Array(phi_now);
while (t < t1)
  for (j,xp) in enumerate(x)
    xdep = x0 + (xp-u*dt)%(x1-x0);
    if xdep < x0
      xdep = x1 + xdep;
    end;
    m = Int(floor(xdep/dx));
    alpha = (xdep/dx)-m;
    m = m + 1;
    mp = m + 1;
    if mp == nx
      mp = 2;
    end;
    phi_new[j] = (1-alpha)*phi_now[m] + alpha*phi_now[mp];
  end;
  phi_now[:] = phi_new[:];
  global t = t + dt;
  if mod(t,tp) < dt
    it = Int(round(t));
    plot!(...)
  end;
end;
```