Kunming School 2018: Lab assignments day 3

Rony Keppens & Jannis Teunissen

1 Write your own solver for the KdV equation

In a PRL publication from 1965, Zabusky & Kruskal demonstrated numerically how a simple $\cos(\pi x)$ initial condition to the KdV equation shows the spontaneous development of solitons, which are nonlinear pulses that can pass through one-another without losing their identity. Your task is to write your own code to do this, introducing a uniform grid of $2N$ grid points x_i on the interval $x \in [0,2]$, i.e. a fixed spatial step equal to $\Delta x = 1/N$. You will set up the initial condition such that the function $u(x,t)$ we solve for has $u(x_i,0) = \cos(\pi x_i)$. During the time stepping, employ a temporal time step Δt from time levels n to $n+1$, and compute u_i^{n+1} based on earlier values such as u_i^n or u_i^{n-1}. The $x=0$ and $x=2$ boundaries during the time stepping are easy to handle when we assume periodic boundaries: $u_i^n = u_{i+2N}^n$.

You can try a variety of the schemes we discussed in the lectures. In the original PRL, the KdV equation

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + \delta^2 \frac{\partial^3 u}{\partial x^3} = 0$$

was approximated by using a leapfrog scheme, using in it the following discretizations of the derivatives:

$$u_i^{n+1} = u_i^{n-1} - \frac{\Delta t}{\Delta x}\frac{u_{i+1}^n + u_i^n + u_{i-1}^n}{3}\left(u_{i+1}^n - u_{i-1}^n\right) - \delta^2 \frac{\Delta t}{(\Delta x)^3}\left(u_{i+2}^n - 2u_{i+1}^n + 2u_{i-1}^n - u_{i-2}^n\right) \quad (1)$$

and this for all $i \in 0,1,2,\ldots 2N-1$. Verify that this employs up to fourth order accurate FD expressions for the third derivative term. You should simulate for far enough in time, and verify the near-discontinuous behavior near the breakdown time where $t \approx 1/\pi$, and go to at least time $t = 3.6/\pi$. Check your findings versus the original PRL publication. In that publication, they used the specific value for the constant $\delta = 0.022$.

2 Use MPI-AMRVAC for Burgers and KdV in multi-D

You can continue using `MPI-AMRVAC`, but this time we will solve nonlinear scalar equations, like Burgers equation, or the nonconvex variant, or the KdV equation. You can do so in more than 1D, and explore e.g. what happens when you feed in your own picture you advected earlier. You can experiment with 1D, 2D or even 3D configurations, and get familiar with the way you prescribe predefined boundary conditions, select a particular discretization, or use AMR, through the `amrvac.par` input file.

3 MPI-AMRVAC for shallow water waves

Finally, solve the 2D shallow water equations using MPI-AMRVAC. Employ a 2D unit square domain, adopt reflective boundaries at all sides, and set up a localized initial pulse of given height. See how the waves travel through the pond, and make Schlieren plots of the evolution.