Computational Physics Project / Solitons

This project focuses on one-dimensional waves, as one might find in shallow water in a narrow canal (once a practical situation, but even now a useful idealized approximation). Much of the discussion here comes from the computational physics textbook of Landau, Páez, & Bordeianu (not *the* Landau).

The equation governing this situation is the Kortweg & de Vries (1895) equation:

$$\frac{\partial u(x,t)}{\partial t} + \epsilon u(x,t)\frac{\partial u(x,t)}{\partial x} + \mu\frac{\partial^3 u(x,t)}{\partial x^3} = 0 \tag{1}$$

where u is the height of the water, x is the direction of propagation, and t is time. The second term is a nonlinear advection term that makes a wave propagate; roughly speaking, the speed of propagation increases with the height of the wave. The third term is a dispersive term; roughly speaking, gradients in the height cause the wave to spread out.

1. Prep work

Before trying to solve this problem numerically, it is worthwhile to consider some related problems and analytic solutions.

- First, consider the simpler problem:

  $$\frac{\partial u(x,t)}{\partial t} + \epsilon\frac{\partial u(x,t)}{\partial x} = 0 \tag{2}$$

  This is the classic *advection* equation. Show that any function of the form

  $$u(x,t) = f(x - \epsilon t) \tag{3}$$

  is a solution. What is the interpretation of this form of solution? What is the interpretation of ϵ in this case?

- Given the previous result, make a qualitative prediction regarding what happens in the non-linear, but non-dispersive, case, where $\mu = 0$ in Equation 1.

- It so happens that there are solutions of the form

  $$u(x,t) = f(x - ct) \tag{4}$$

  for the full Equation 1, but that unlike the advection equation, not *any* function of that form is a solution. Show that if $\xi = x - ct$, then these solutions must satisfy the ordinary differential equation:

  $$-c\frac{\partial u}{\partial \xi} + \epsilon u\frac{\partial u}{\partial \xi} + \mu\frac{\partial^3 u}{\partial \xi^3} = 0. \tag{5}$$

- Further show that a solution to this equation is:

$$u(x,t) = -\frac{c}{2}\,\mathrm{sech}^2\left[\frac{1}{2}\sqrt{c}\,(x - ct - \xi_0)\right], \tag{6}$$

  where ξ_0 is a constant representing an initial phase. Plot this solution, which is known as a soliton. Can you interpret qualitatively why this solution exists?

2. Designing the code

We will use a finite difference technique. You will need to choose Δx and Δt. We will refer to steps in time as j and steps in space as i.

- What are the first-order central difference approximations for $\partial u/\partial t$ and $\partial u/\partial x$?
- Expand $u(x,t)$ in a Taylor series in x around x_i to third order, at fixed time t. For the four points $x_i - 2\Delta x$, $x_i - \Delta x$, $x + \Delta x$ and $x + 2\Delta x$, you have the values $u_{i-2,j}$, $u_{i-1,j}$, $u_{i+1,j}$ and $u_{i+2,j}$. You should be able to solve this set of equations for $\partial^3 u/\partial x^3$ at $x = x_i$.
- Combine the above results into an evolution scheme for this set of equations, where you use where needed:

$$u(x_i, t_j) = \frac{u_{i-1,j} + u_{i,j} + u_{i+1,j}}{3} \tag{7}$$

- Can you determine the relationship between Δx and Δt necessary for linear stability of this scheme?
- To get started, you need two time steps. Write how you do this with a forward difference in time instead of a central difference.

The above equations do not handle the boundary conditions. Thus, in the very first time step, $u_{1,2}$, $u_{2,2}$, $u_{N-1,2}$, and $u_{N,2}$ are not determined. In the implementation below, you will assume fixed values of $u_{1,j}$ and u_N, j and a zero derivative (i.e. for the purposes of the finite difference that $u_{0,j} = u_{1,j}$ and $u_{N+1,j} = u_{N,j}$).

3. Testing and running the code

- Write an implementation of the finite difference solution to these equations.
- Design a set of unit tests that will help you determine that the individual elements of your code are working as expected. Implement these unit tests.

- Set $\mu = 0$, and make the second term linear (i.e. without the prefactor u). This is just the advection equation. Show that your code successfully advects any initial waveform, by comparing quantitatively the waveform at early and late time steps (use a time interval which should propagate the wave an integer number of steps). Use boundary conditions of zero, and make the grid big enough that your test stays away from the boundary.

- Set $\mu = 0$, but use the nonlinear second term. This special case is known as *Burgers' equation*. Show what happens to an initially Gaussian wave and interpret it in the context of what you thought would happen in the prep work section. Use boundary conditions of zero again.

- For non-zero μ and ϵ, set initial conditions of a soliton wave. Show that it propagates as expected, again through a quantitative comparison with a late time step. Use boundary conditions of zero again.

- Now test the convergence of the method by increasing the number of grid points by a factor of two and then four.

- Make initial conditions of two soliton waves, one taller than and "behind" the second one. What happens? Use boundary conditions of zero again.

- Finally, try initial conditions of:

$$u(x, t = 0) = \frac{1}{2}\left[1 - \tanh\left(\frac{x - x_0}{\sigma}\right)\right]. \tag{8}$$

  Use boundary conditions of $u = 1$ on the left and $u = 0$ on the right. Watch what happens and interpret it.