Exercises for the lecture

Fundamentals of Simulation Methods

Friedrich Röpke & Cornelis Dullemond

Exercise sheet Chapter 10 & 11 - part 1

Numerical Hydrodynamics

1. **Numerical advection in 1-D**

To "warm up" to the solving of hydrodynamics problems in 1-D, we will first solve a very simple advection problem. Consider a 1-D domain in the coordinate x, ranging from $x = 0$ to $x = L$, for some value of L (let us take this $L = 10$). We consider a function $q(x)$ on this domain. We now allow this function to evolve in time t. It thus becomes $q(x, t)$. The way $q(x, t)$ evolves in time is according to the following advection equation:

$$\frac{\partial q(x,t)}{\partial t} + v\frac{\partial q(x,t)}{\partial x} = 0 \tag{1}$$

where $v > 0$ is a constant. Let us take $v = 1$. For this simple problem we know the analytic solution: $q(x, t > 0) = q(x - vt, t = 0)$, where we impose the periodic boundary condition implicitly. But let us try to solve this equation numerically. We set up a grid in x with $N = 100$ grid points between $x = 0$ and $x = L$. In addition we add 1 ghost cell on each side, to make the implementation of the boundary conditions easier. In total we thus have 102 grid points in x. As initial condition we set

$$q(x,0) = \begin{cases} 1 & \text{for} \quad x < L/2 \\ 0 & \text{for} \quad x \geq L/2 \end{cases} \tag{2}$$

As boundary condition we set $q(0, t) = 1$ and $q(L, t) = 0$. The boundary condition is imposed after every time step simply by (re-)setting the values of q in the ghost cells to the boundary value. We integrate in time from $t = 0$ (initial condition) to $t = 3$ using 100 time steps (i.e. with $\Delta t = 0.03$).

a) Write a program to perform this numerical integration using the *symmetric* numerical derivative operator $(q_{i+1} - q_{i-1})/2\Delta x$. You can show that this will lead to a numerical instability.

b) Now use the one-sided numerical derivative operator $(q_i - q_{i-1})/\Delta x$. Show that now it stays stable.

c) Now use the one-sided numerical derivative operator $(q_{i+1} - q_i)/\Delta x$. Show that it is unstable again.

The above exercises show that only the *upwind* advection algorithm leads to a stable solution. Now let us experiment with the upwind algorithm a bit. So from now on, only use the upwing method.

d) Put the left boundary condition to $q(0, t) = 0.5$. What happens, and why?

e) Put the right boundary condition to $q(L,t) = 0.5$. What happens, and why?

f) Now use a $10\times$ smaller time step: use 1000 time steps between $t = 0$ and $t = 3$. What is the difference in the result at $t = 3$? Does it become better or worse?

g) Now use a $10\times$ *larger* time step: use 10 time steps between $t = 0$ and $t = 3$. Explain what happens.

2. **A general-purpose 1-D advection subroutine/function**

Now let us create a general-purpose computer function for the purpose of 1-D numerical advection of any given 1-D function $q(x,t)$ (represented as a 1-D array q_i^n) for any given velocity (also represented as a 1-D array $v_{i+1/2}$). The function should receive the current values of q_i^n, the velocities $v_{i+1/2}$ and a time step size Δt. It should return the values q_i^{n+1}, i.e. the values of q at the next time step. For example, the function call in Python could look like `qnew = advect(qold,velo,dx,dt)`, which advects q_i^n one time step. Use ghost cells to implement the boundary conditions.

You will use this function next week to create a 1-D hydrodynamics code from that, so please test this function and make sure that it works correctly.

Note that the velocities $v_{i+1/2}$ are located *in between* the grid points. In the picture of grid cells: the velocities $v_{i+1/2}$ are defined on the *cell walls* while the to-be-advected function values q_i^n are located in the cell centers. That also means that the length of the velocity array is one different from that of the q array.

a) Apply your function to exercise 1 above (the exercise with a constant velocity) and convince yourself that your function reproduces the same results.

b) Now let us choose the velocity $v(x) = -2(x - L/2)/L$. This is a converging velocity field. Take as an initial condition this time $q(x,0) = 1$ for $|x - L/2| \leq L/4$ and $q(x,0) = 0$ for $|x - L/2| > L/4$. Integrate again to $t = 3$ with 100 time steps. Explain the result.