Problem Set 10

Exercises for the Lecture Fundamentals of Simulation Methods (WS24/25)

Cornelis Dullemond and Michela Mapelli (Lecture Tuesday and Thursday 14h - 16h)
Lucia Haerer (Tutor Group T1 on Wednesday 11h - 13h)
Stefano Rinaldi (Tutor Group T2 on Thursday 11h - 13h)
Nicholas Storm (Tutor Group T3 on Friday 11h - 13h)

Submit the solution in electronic by the **Friday 6pm**, **January 10, 2025**.

10. Hydrodynamics and higher-order advection algorithms

10.1. Advection: Numerical diffusion (*10 points*)

We consider, again, the following 1D advection equation:

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

with $u > 0$ *constant in space and time*. We use a constant grid spacing Δx. The (unstable) *symmetric* differencing scheme is:

$$\frac{\rho_i^{n+1} - \rho_i^n}{\Delta t} = -u\frac{\rho_{i+1}^n - \rho_{i-1}^n}{2\Delta x} \tag{2}$$

The (stable) *upwind* differencing scheme (for $u > 0$) is:

$$\frac{\rho_i^{n+1} - \rho_i^n}{\Delta t} = -u\frac{\rho_i^n - \rho_{i-1}^n}{\Delta x} \tag{3}$$

The upwind scheme is a first-order (in space) advection scheme.

1. Show that one can regard the upwind differencing scheme as being the symmetric differencing scheme plus a diffusion term. Determine the diffusion coefficient D for that term. *Hint:* Refer back to the chapter on boundary value problems to see what the diffusion operator is, both in continuous form as well as (for the present purpose) in discrete form.

The *Lax-Wendroff* differencing scheme is

$$\frac{\rho_i^{n+1} - \rho_i^n}{\Delta t} = -u\frac{\rho_{i+1}^n - \rho_{i-1}^n}{2\Delta x} + \left(\frac{u^2\Delta t}{2}\right)\frac{\rho_{i+1}^n - 2\rho_i^n + \rho_{i-1}^n}{\Delta x^2} \tag{4}$$

The Lax-Wendroff scheme is a second-order (in space) advection scheme.

2. Show that also this equals the symmetric scheme plus a diffusion term, and give the diffusion coefficient D.

3. Show that for this scheme, D is smaller than for the upwind scheme. Lax-Wendroff is apparently less numerically diffusive than upwind.

4. Implement the Lax-Wendroff method and apply it to the same advection problem as last week's exercise. In which qualitative sense is the result different?

What do we learn from this: (1) unstable schemes can be stabilized by adding *artificial diffusion*, (2) even if a diffusion term is not explicitly seen in the discrete equations (like in the upwind algorithm), it may be still there: This is what is called *numerical diffusion*. All algorithms have some amount of numerical diffusion. Higher-order methods have lower numerical diffusion, but some amount of numerical diffusion is unavoidable. The more you try to reduce it, the more unstable the algorithm becomes, producing oscillations that may stay contained (if sufficient diffusion is present) or grow exponentially (if insufficient diffusion is present).

10.2. A first 1D hydrodynamics code (*14 points*)

Let us now try to program a first 1D hydrodynamics code. For simplicity we assume that the temperature stays constant, and thereby the isothermal sound speed c_s as well. The gas pressure is then linearly proportional to the density: $p = \rho c_s^2$. Let us set $c_s = 1$ for convenience. With this assumption we only have to solve the following two coupled partial differential equations:

$$\frac{\partial \rho}{\partial t} + \frac{\partial(\rho u)}{\partial x} = 0 \tag{5}$$

$$\frac{\partial(\rho u)}{\partial t} + \frac{\partial(\rho u^2 + p)}{\partial x} = 0 \tag{6}$$

where, this time, u is a function of space x and time t. Let us write this in state vector form:

$$\frac{\partial \mathbf{q}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} = 0 \tag{7}$$

where

$$\mathbf{q} = \begin{pmatrix} \rho \\ \rho u \end{pmatrix} \qquad \text{and} \qquad \mathbf{F} = \begin{pmatrix} \rho u \\ \rho u^2 + p \end{pmatrix} \tag{8}$$

1. Rewrite Eq. (7) in the form of two coupled conservative advection equations with advection velocity $u(x,t)$, and a right-hand-side source term for the second equation.

For the hydrodynamics algorithm we will use the *operator splitting* procedure: As the first operator we apply the donor-cell algorithm to advect $q_0(x,t) \equiv \rho(x,t)$ as well as $q_1(x,t) \equiv \rho(x,t)u(x,t)$. The velocity at the cell walls is computed using a simple mean of the velocities in the two neighboring cells. Then (second operator) we will add the right-hand-side term, computed as

$$-\frac{\partial p}{\partial x} \quad \rightarrow \quad -\frac{p_{i+1} - p_{i-1}}{2\Delta x} \tag{9}$$

(i.e., using a symmetric derivative). Make sure to compute the p_{i+1} and p_{i-1} from the *new* density values (after the advection step), otherwise the algorithm is unstable. We impose periodic boundary conditions using the *ghost cell* method. Make sure to impose these boundary conditions after each split operator (i.e., after the advection step and again after the addition of the pressure term), otherwise the algorithm is unstable.

2. Implement this hydrodynamics code. You can use the donor-cell algorithm provided in `advection.py` for the advection.

3. Apply it to the following problem: $-2 \leq x \leq 2$ with *periodic boundaries*, initial condition $\rho(x, 0) = 1 + \exp(-(x/w)^2/2)$ with $w = 0.5$, $u(x, 0) = 0$, isothermal sound speed $c_s = 1$, with $N = 1000$ gridpoints in x, and $6N$ time steps in time from $t = 0$ to $t = 5$.

4. Show the results at the 1000-th time step ($t = 0.833$), the 2000-th time step ($t = 1.667$), the 3000-th time step ($t = 2.500$), and the 4000-th time step ($t = 2.334$).