Theoretical and Computational Star Formation SoSe 2022

Assignment sheet 10

Hand-out	21.06.2022
Hand-in	28.06.2022
Discussion	05.07.2022

Total points: 30 Points

General Information

All exercises combined over all sheets sum up to a final 100%-score value. In order to pass the course one has to reach a combined score of 50%.

Please note that we can only give points for WORKING code! If your code does not compile or does not run, we will have to give 0 points for the coding part. Please consider this when submitting your exercises.

Handing in your exercises:

- You should submit in groups of 2 - 4 students.
- The assignment sheet has to be handed in directly before the start of the tutorial.
- All sheets will be handed out electronically via Ilias.
- You have to hand in your exercises electronically by sending them to the email address **lectures@ph1.uni-koeln.de**. In particular your programs should be handed in as text documents with a corresponding suffix (example: *.py for Python programs). If required, text has to be scanned in or submitted as a PDF document. Additional output of your programs has to be submitted as well. **All** of your files should be contained in **one** archive file format (example.: *.tar, *zip).
- You should name all authors in all programs (as a comment) and PDFs.
- Please write 'readable' code. Comment your thoughts and steps.

1 Sedov explosion [10 Points]

Using the 1D code you developed in the last exercise, model a Sedov explosion. For this use the following initial conditions: Flat density $\rho = 1$, zero velocity ($\vec{v} = \vec{0}$), and $\gamma = 1.4$. The box size is $x \in [-10, 10]$. Use the same boundary conditions as for last week's exercise (isolated, or "outflow" boundaries). We trigger the explosion by depositing a huge amount of thermal energy in the center of the computational domain, which is here done by increasing the pressure in the central region:

$$p = \begin{cases} 100.0 & |x| \leq 1.0, \text{ and} \\ 1.0 & \text{elsewhere} \end{cases} \tag{1}$$

1. Run the simulation for different resolutions and plot the results for the density, velocity and pressure at a late stage ($t = 0.6$).
2. For the highest resolution used, show the time evolution of the aforementioned quantities.
3. Compare in detail the results with the analytical solution derived in the previous tutorial. For this make yourself clear how the pressure relates to the amount of explosion energy.

2 Two-dimensional simulation code [20 Points]

Consider the two-dimensional partial differential equation

$$\frac{\partial}{\partial t}\vec{q} + \frac{\partial}{\partial x}\vec{f_x} + \frac{\partial}{\partial y}\vec{f_y} = 0, \tag{2}$$

where $\vec{f_x}$ is the already known from your one-dimensional solver and $\vec{f_y}$ is the flux in y-direction,

$$\vec{f_y} = \begin{bmatrix} \rho v_y \\ \rho v_y v_x \\ \rho v_y^2 + p \\ \rho v_y v_z \\ v_y\left(\frac{1}{2}\rho\|\vec{v}\|^2 + \frac{\gamma p}{\gamma - 1}\right) \end{bmatrix}. \tag{3}$$

The Cartesian grid allows us to directly extend our 1D ideas to 2D.

1. Extend your one-dimensional "grid" of N cells to a two-dimensional grid of size $N \times M$ as shown in Fig. 1. For now, we have $\Delta y = \Delta x$ and $M = N$, however, your code should be flexible enough to be able to change this in a future exercise. You will need to extend your existing loop (over all N cells) to a nested two-dimensional loop (over all $N \times M$ cells).
2. Implement isolated boundary conditions as before for cells with indices on the edges of your computational domain.

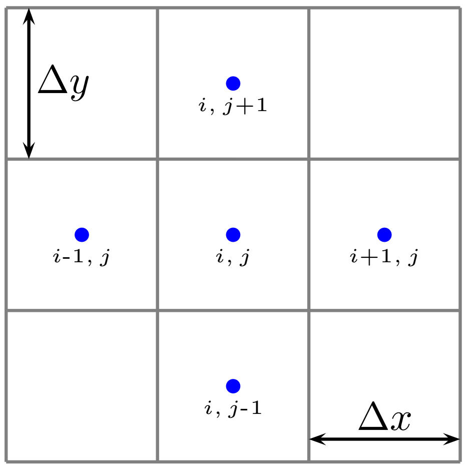

Figure 1: Sketch of two dimensional grid illustrating how we count indices in two dimensions.

3. As before, create a Riemann solver than computes the interface fluxes in y direction. The physical fluxes in y-direction are given by (3). The fastest eigenvalue of (3) is $\lambda_{\max,y} = |v| + c_s$.

4. Extend your already existing one-dimensional update procedure to the corresponding two-dimensional equivalent:

$$\vec{q}_i^{\,n+1} = \vec{q}^{\,n} - \frac{\Delta t}{\Delta x}(\vec{f}^{\,n}_{x,i+1/2,j} - \vec{f}^{\,n}_{x,i-1/2,j}) - \frac{\Delta t}{\Delta y}(\vec{f}^{\,n}_{y,i,j+1/2} - \vec{f}^{\,n}_{y,i,j-1/2}) \tag{4}$$

5. Extend the CFL condition to two spatial dimensions. There exists no ultimate solution for this, possible choices are

$$\Delta t = c \cdot \min\left(\frac{\Delta x}{\lambda_{\max,x}}, \frac{\Delta y}{\lambda_{\max,y}}\right), \qquad c \in (0,1] \tag{5}$$

or

$$\Delta t = c \cdot \frac{\min(\Delta x, \Delta y)}{\max(\lambda_{\max,x}, \lambda_{\max,y})}. \tag{6}$$

What is the physical interpretation of (5) and (6)? Is one of them preferable?