Mesh Based Numerical Hydrodynamics of Ideal Gases

The Used and Implemented Equations

Mladen Ivkovic (mladen.ivkovic@hotmail.com)

Contents

1 Ideal Gases

1.1 Governing Equations

We are mostly going to concern ourselves with ideal gases, which are described by the Euler equations:

$$\frac{\partial}{\partial t}\begin{pmatrix}\rho\\ \rho\mathbf{v}\\ E\end{pmatrix}+\nabla\cdot\begin{pmatrix}\rho\mathbf{v}\\ \rho\mathbf{v}\otimes\mathbf{v}+p\\ (E+p)\mathbf{v}\end{pmatrix}=\begin{pmatrix}0\\ \rho\mathbf{a}\\ \rho\mathbf{a}\mathbf{v}\end{pmatrix} \tag{1}$$

Where

- ρ: fluid density
- $\mathbf{v}$: fluid (mean/bulk) velocity at a given point. I use the notation $\mathbf{v}=(u,v,w)$, or when indices are useful, $\mathbf{v}=(v_1,v_2,v_3)$
- p: pressure
- E: specific energy. $E=\frac{1}{2}\rho\mathbf{v}^2+\rho\varepsilon$, with $\varepsilon=$ specific internal thermal energy
- $\mathbf{a}$: acceleration due to some external force.

The outer product $\cdot\otimes\cdot$ gives the following tensor:

$$(\mathbf{v}\otimes\mathbf{v})_{ij}=\mathbf{v}_i\mathbf{v}_j \tag{2}$$

Furthermore, we have the following relations for ideal gasses:

$$p=nkT \tag{3}$$

$$p=C\rho^\gamma \qquad \text{entropy relation for smooth flow, i.e. no shocks} \tag{4}$$

$$s=c_V\ln\left(\frac{p}{\rho^\gamma}\right)+s_0 \qquad \text{entropy} \tag{5}$$

$$c=\sqrt{\left.\frac{\partial p}{\partial\rho}\right|_s}=\sqrt{\frac{\gamma p}{\rho}} \qquad \text{sound speed} \tag{6}$$

with

- n : number density
- k : Boltzmann constant
- T : temperature
- s : entropy
- γ: adiabatic index
- c_V: specific heat

and the Equation of State

$$\varepsilon = \frac{1}{\gamma - 1}\frac{p}{\rho} \tag{7}$$

The Euler equations can be written as a conservation law as

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{U}) = 0 \tag{8}$$

where we neglect any outer forces, i.e. $\mathbf{a} = 0$, and

$$\mathbf{U} = \begin{pmatrix} \rho \\ \rho\mathbf{v} \\ E \end{pmatrix}, \qquad \mathbf{F}(\mathbf{U}) = \begin{pmatrix} \rho\mathbf{v} \\ \rho\mathbf{v} \otimes \mathbf{v} + p \\ (E + p)\mathbf{v} \end{pmatrix} \tag{9}$$

1.1.1 Euler equations in 1D

In 1D, we can write the Euler equations without source terms ($\mathbf{a} = 0$) as

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}(\mathbf{U})}{\partial x} = 0 \tag{10}$$

or explicitly (remember $\mathbf{v} = (u, v, w)$)

$$\frac{\partial}{\partial t}\begin{pmatrix} \rho \\ \rho u \\ E \end{pmatrix} + \frac{\partial}{\partial x}\begin{pmatrix} \rho u \\ \rho u^2 + p \\ (E + p)u \end{pmatrix} = 0 \tag{11}$$

1.1.2 Euler equations in 2D

In 2D, we have without source terms ($\mathbf{a} = 0$)

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}(\mathbf{U})}{\partial x} + \frac{\partial \mathbf{G}(\mathbf{U})}{\partial y} = 0 \tag{12}$$

or explicitly (remember $\mathbf{v} = (u, v, w)$)

$$\frac{\partial}{\partial t}\begin{pmatrix} \rho \\ \rho u \\ \rho v \\ E \end{pmatrix} + \frac{\partial}{\partial x}\begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ (E + p)u \end{pmatrix} + \frac{\partial}{\partial y}\begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ (E + p)v \end{pmatrix} = 0 \tag{13}$$

1.2 Conserved and primitive variables

For now, we have described the Euler equation as a hyperbolic conservation law using the (conserved) state vector $\mathbf{U}$:

$$\mathbf{U} = \begin{pmatrix} \rho \\ \rho \mathbf{v} \\ E \end{pmatrix} \tag{14}$$

for this reason, the variables ρ, $\rho\mathbf{v}$, and E are referred to as "conserved variables", as they obey conservation laws.

However, this is not the only set of variables that allows us to describe the fluid dynamics. In particular, the solution of the Riemann problem (section 3) will give us a set of with so called "primitive variables" (or "physical variables") with the "primitive" state vector $\mathbf{W}$

$$\mathbf{W} = \begin{pmatrix} \rho \\ \mathbf{v} \\ p \end{pmatrix} \tag{15}$$

Using the ideal gas equations, these are the equations to translate between primitive and conservative variables:

Primitive to conservative:

$$(\rho) = (\rho) \tag{16}$$
$$(\rho\mathbf{v}) = (\rho) \cdot (\mathbf{v}) \tag{17}$$
$$(E) = \frac{1}{2}(\rho)(\mathbf{v})^2 + \frac{1}{\gamma - 1}\frac{(p)}{(\rho)} \tag{18}$$

Conservative to primitive:

$$(\rho) = (\rho) \tag{19}$$
$$(\mathbf{v}) = \frac{(\rho\mathbf{v})}{(\rho)} \tag{20}$$
$$(p) = \frac{\gamma - 1}{(\rho)}\left((E) - \frac{1}{2}\frac{(\rho\mathbf{v})^2}{(\rho)}\right) \tag{21}$$

1.3 Implementation Details

All the functions for computing gas related quantities are written in `gas.h` and `gas.c`. Every cell is represented by a struct `struct cell` written in `cell.h`. It stores both the

primitive variables/states and the conservative states in the `pstate` and `cstate` structs, respectively.

The adiabatic index γ is hardcoded as a macro in `defines.h`. If you change it, all the derived quantities stored in macros (e.g. $\gamma-1$, $\frac{1}{\gamma-1}$) should be computed automatically.

2 Notation

We are working on numerical methods. Both space and time will be discretized.

Space will be discretized in cells which will have integer indices to describe their position. Time will be discretized in fixed time steps, which may have variable lengths. Nevertheless the time progresses step by step.

The lower left corner has indices $(0, 0)$ in 2D. In 1D, index 0 also represents the leftmost cell.

We have:

- integer subscript: Value of a quantity at the cell, i.e. the center of the cell. Example: $\mathbf{U}_i$, $\mathbf{U}_{i-2}$ or $\mathbf{U}_{i,j+1}$ for 2D.
- non-integer subscript: Value at the cell faces, e.g. $\mathbf{F}_{i-1/2}$ is the flux at the interface between cell i and $i-1$, i.e. the left cell as seen from cell i.
- integer superscript: Indication of the time step. E.g. $\mathbf{U}^n$: State at timestep n
- non-integer superscript: (Estimated) value of a quantity in between timesteps. E.g. $\mathbf{F}^{n+1/2}$: The flux at the middle of the time between steps n and $n+1$.

3 Riemann Solvers

3.1 Solution Strategy

3.2 Exact Solver

3.3 HLL Solver

3.4 HLLC Solver

3.5 Two-Rarefaction Riemann Solver

3.6 Two-Shock Riemann Solver

4 Hydrodynamics Methods

4.1 Upwind Godunov

5 Slope and Flux Limiters

5.1 Slope Limiters

Slope limiters are employed because issues arise around numerical schemes because of their discrete nature. For example, a non-limited piecewise linear advection scheme will produce oscillations around jump discontinuities. So the idea is to compute the slope in way that is useful for us based on the current situation of the gas state that we're solving for.

The choice of the slope can be expressed via a function $\phi(r)$ (see eqns. 50, 51) with

$$r^n_{i-1/2} = \begin{cases} \frac{\mathbf{U}^n_{i-1} - \mathbf{U}^n_{i-2}}{\mathbf{U}^n_i - \mathbf{U}^n_{i-1}} & \text{for } \mathbf{v} \geq 0 \\ \frac{\mathbf{U}^n_{i+1} - \mathbf{U}^n_i}{\mathbf{U}^n_i - \mathbf{U}^n_{i-1}} & \text{for } \mathbf{v} \leq 0 \end{cases}$$

.

Possible limiters are:

$$\text{Minmod} \qquad \phi(r) = \text{minmod}(1, r) \tag{22}$$

$$\text{Superbee} \qquad \phi(r) = \max(0, \min(1, 2r), \min(2, r)) \tag{23}$$

$$\text{MC (monotonized cenral-difference)} \qquad \phi(r) = \max(0, \min((1 + r)/2, 2, 2r)) \tag{24}$$

$$\text{van Leer} \qquad \phi(r) = \frac{r + |r|}{1 + |r|} \tag{25}$$

where

$$\text{minmod}(a, b) = \begin{cases} a & \text{if } |a| < |b| \text{ and } ab > 0 \\ b & \text{if } |a| > |b| \text{ and } ab > 0 \\ 0 & \text{if } ab \leq 0 \end{cases} \tag{26}$$

6 Boundary Conditions

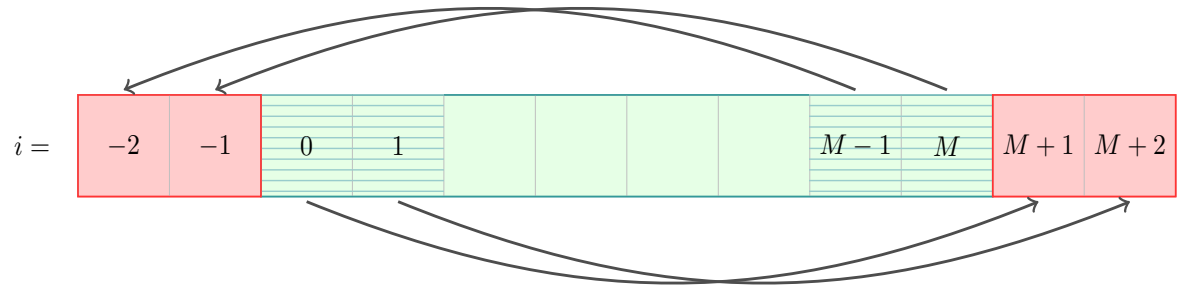

Figure 1: Method to obtain periodic boundary conditions. The ghost cells are red, the arrows show what will be copied where.

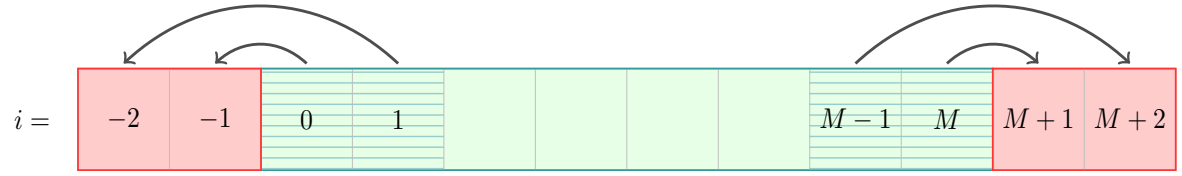

Figure 2: Method to obtain wall boundary conditions. The ghost cells are red, the arrows show what will be copied where.

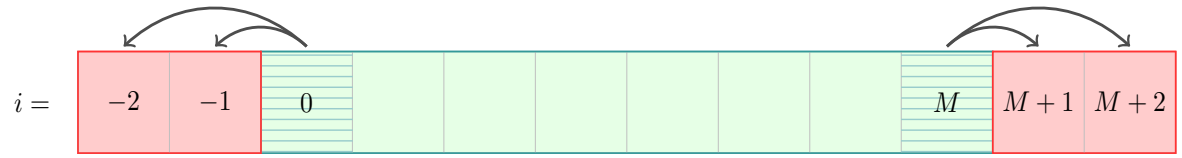

Figure 3: Method to obtain transmissive boundary conditions. The ghost cells are red, the arrows show what will be copied where.

There are tricks how to obtain different kinds of boundary conditions. In every case, we add additional cells ("*ghost cells*") in every dimension so we can simulate the desired behaviour. How many cells you need to add depends on the methods (and mostly stencils) you use. If you only take into account one neighbouring cell, then one ghost cell on every boundary suffices. In figures 1, 2, and 3, two ghost cells for a 1D grid are drawn.

Suppose we have 1D grid with M cells and require 2 ghost cells each, which will have indices -2, -1, $M+1$, and $M+2$. Then we can get:

- **periodic boundary conditions:**

what goes over the right edge, comes back in over the left edge, and vice versa. We achieve this behaviour by enforcing (fig. 1)

$$\begin{aligned}\mathbf{U}_{-2} &= \mathbf{U}_{M-1}\\ \mathbf{U}_{-1} &= \mathbf{U}_{M}\\ \mathbf{U}_{M+1} &= \mathbf{U}_{0}\\ \mathbf{U}_{M+2} &= \mathbf{U}_{1}\end{aligned}$$

- **reflective boundary conditions:**

  pretend there is a wall at the boundary. We achieve that by "mirroring" the cells next to the boundary (fig 2):

$$\begin{aligned}\mathbf{U}_{-2} &= \mathbf{U}_{1}\\ \mathbf{U}_{-1} &= \mathbf{U}_{0}\\ \mathbf{U}_{M+1} &= \mathbf{U}_{M}\\ \mathbf{U}_{M+2} &= \mathbf{U}_{M-1}\end{aligned}$$

  However, every directional component (i.e. velocities/momentum) needs to have the negative value in the ghost cell compared to the real cell.

- **transmissive boundary conditions**:

  Just let things flow out however they want. We achieve this by copying the last boundary cell over and over again, such that it looks that the fluid appears to have that state infinitely, and there are no net fluxes to interfere with the hydrodynamics inside the actual grid (fig. 3)

$$\begin{aligned}\mathbf{U}_{-2} &= \mathbf{U}_{0}\\ \mathbf{U}_{-1} &= \mathbf{U}_{0}\\ \mathbf{U}_{M+1} &= \mathbf{U}_{M}\\ \mathbf{U}_{M+2} &= \mathbf{U}_{M}\end{aligned}$$

7 Dimensional Splitting

To go from one to multiple dimensions, it is tempting to just extend the one dimensional discretisation. For example, starting with the conservation law

$$\frac{\partial}{\partial t}\mathbf{U} + \frac{\partial}{\partial x}\mathbf{F}(\mathbf{U}) + \frac{\partial}{\partial y}\mathbf{G}(\mathbf{U}) = 0 \tag{27}$$

and just apply Godunov's finite volume method:

$$\mathbf{U}_{i,j}^{n+1} = \mathbf{U}_{i,j}^{n} + \frac{\Delta t}{\Delta x}\left(\mathbf{F}_{i-1/2,j} - \mathbf{F}_{i+1/2,j}\right) + \frac{\Delta t}{\Delta x}\left(\mathbf{G}_{i,j-1/2} - \mathbf{G}_{i,j+1/2}\right) \tag{28}$$

However, this is a bit of a problem. The upwinding here is not complete. Consider the 2D advection equation

$$\frac{\partial}{\partial t}q + u\frac{\partial}{\partial x}q + v\frac{\partial}{\partial y}q = 0 \tag{29}$$

Now suppose we have advecting velocities $u = v = 1$, i.e. the advection velocity is along the diagonal. Then our method reads

$$q_{i,j}^{n+1} = q_{i,j}^{n} + u\frac{\Delta t}{\Delta x}\left(q_{i-1/2,j} - q_{i+1/2,j}\right) + v\frac{\Delta t}{\Delta x}\left(q_{i,j-1/2} - q_{i,j+1/2}\right) \tag{30}$$

This expression doesn't involve $q_{i-1,j-1}$ at all, but that's actually the value that should be advected to $q_{i,j}$ in the next timestep!

One way of doing things is to actually formulate more sophisticated unsplit methods that involve the appropriate stencils. We shan't do that here though. We make use of dimensional splitting. Instead of solving

$$\begin{cases} \text{PDE:} & \frac{\partial}{\partial t}\mathbf{U} + \frac{\partial}{\partial x}\mathbf{F}(\mathbf{U}) + \frac{\partial}{\partial y}\mathbf{G}(\mathbf{U}) = 0 \\ \text{IC:} & \mathbf{U}(x, y, t^n) = \mathbf{U}^n \end{cases} \tag{31}$$

we do it in 2 (3 for 3D) steps:

Step 1: We obtain a intermediate result $\mathbf{U}^{n+1/2}$ by solving the "x - sweep" over the full time interval Δt :

$$\begin{cases} \text{PDE:} & \frac{\partial}{\partial t}\mathbf{U} + \frac{\partial}{\partial x}\mathbf{F}(\mathbf{U}) = 0 \\ \text{IC:} & \mathbf{U}^{n} \end{cases} \tag{32}$$

and then we evolve the solution to the final $\mathbf{U}^{n+1}$ by solving the "y - sweep" over the full time interval Δt :

$$\begin{cases} \text{PDE:} & \frac{\partial}{\partial t}\mathbf{U} + \frac{\partial}{\partial y}\mathbf{G}(\mathbf{U}) = 0 \\ \text{IC:} & \mathbf{U}^{n+1/2} \end{cases} \tag{33}$$

using the 1D methods that are described.

What's even better is that it can be shown (see LeVeque 2002) that by switching the order of the sweeps every timestep (and keeping the time step interval Δt constant) leads to a second order accurate method in time. This is called "Strang splitting".

8 Advection

8.1 Analytical Equation

Advection is a bit of an exception as a hydrodynamics method because we're not actually solving the (ideal) gas equations, but these instead:

$$\frac{\partial \mathbf{U}}{\partial t} + \mathbf{v} \cdot \frac{\partial \mathbf{U}}{\partial x} = 0 \tag{34}$$

Which is still a conservation law of the form

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} = 0 \tag{35}$$

with the flux tensor

$$\mathbf{F} = \mathbf{v} \cdot \mathbf{U} \tag{36}$$

We assume $\mathbf{v} = \text{const}$.

The analytical solution is given by any function $q(x)$ with $\mathbf{U}(x,t) = q(\mathbf{x} - \mathbf{v}t)$, which is just $q(x)$ translated by $\mathbf{v}t$.

8.2 Piecewise Constant Method

We assume that the cell state within a cell is constant (fig. 4). Furthermore, we also assume that the velocity $\mathbf{v}$ is constant and positive.

$$\mathbf{U}_i^{n+1} = \mathbf{U}_i^n + \frac{\Delta t}{\Delta x}\left(\mathbf{F}_{i-1/2}^{n+1/2} - \mathbf{F}_{i+1/2}^{n+1/2}\right) \tag{37}$$

$$\mathbf{F}_{i\pm1/2}^{n+1/2} = \mathbf{v}_{i\pm1/2} \cdot \mathbf{U}_{i-1/2\pm1/2} \tag{38}$$

The method is first order accurate in time and space.

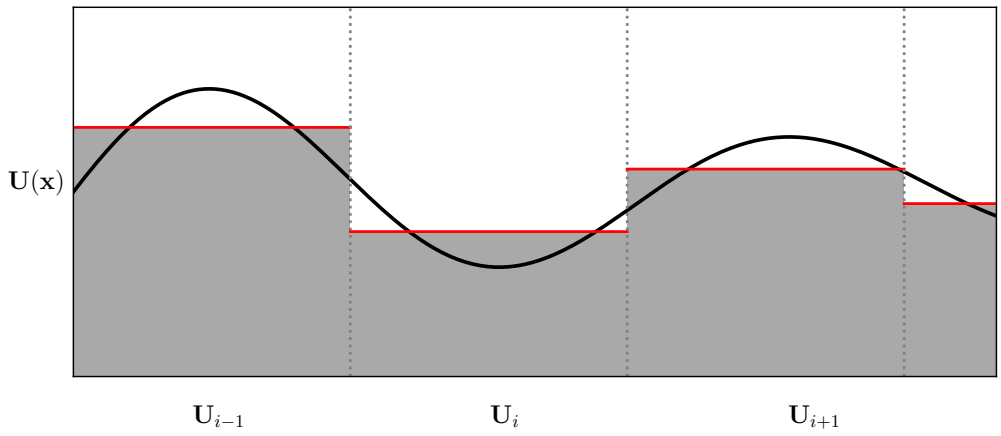

Figure 4: Piecewise constant reconstruction of the field

We assumed that the velocity is positive and constant. What if it's negative?

The important point is that we always do **downwind differencing**. To obtain a finite difference, as we do here, you must never use the value that is downstream, i.e. that is in the direction of the flux. Doing this means taking a value for your computation that won't be valid as soon as an infinitesimal time interval passes, because the ingoing flux will change the downwind state. This is unphysical and leads to violent instabilites.

So if we have negative velocity, all we need to do is change the expression 38 to

$$\mathbf{F}^{n+1/2}_{i\pm1/2} = \mathbf{v}_{i\pm1/2} \cdot \mathbf{U}_{i+1/2\pm1/2} \tag{39}$$

8.3 Piecewise Linear Method

This time, we assume that the state is not constant within a cell, but follows a piecewise linear profile with some slope $\mathbf{s}$ (fig. 5):

$$\text{For } \mathbf{x}_{i-1/2} < \mathbf{x}_i < \mathbf{x}_{i+1/2}: \qquad \mathbf{U}(\mathbf{x}, t = t_n) = \mathbf{U}^n_i + \mathbf{s}^n_i(\mathbf{x} - \mathbf{x}_i)$$

$$\text{Centered method:} \qquad \mathbf{s_i}^n = \frac{\mathbf{U}^n_{i+1} - \mathbf{U}^n_{i-1}}{2\Delta x}$$

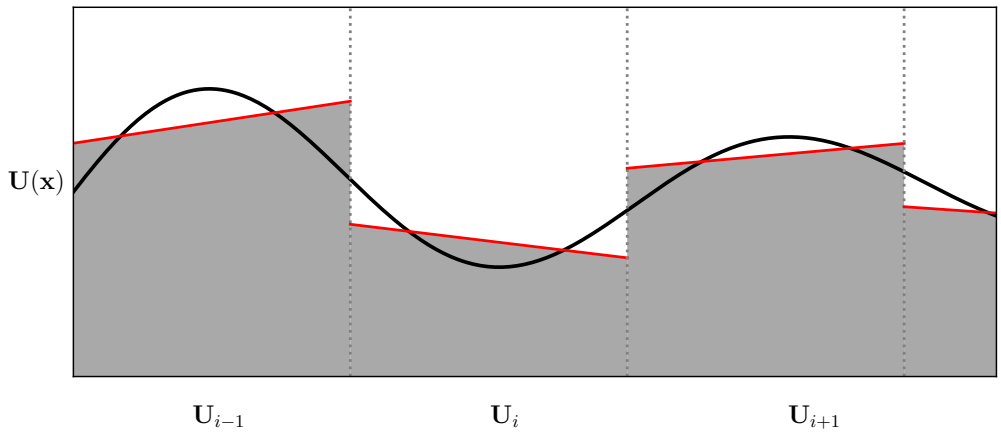

Figure 5: Piecewise linear reconstruction of the field

Other choices for the slope are possible and stable.

Assuming a positive constant velocity $\mathbf{v}$, we derive the flux $\mathbf{F}$ at the time $t^n < t < t^{n+1}$ at the interface position $i-1/2$. At time t, the cell will have been advected by a distance $\mathbf{v}(t-t^n)$, and the the current state at the interface will be

$$\begin{aligned}\mathbf{U}(x = x_{i-1/2}, t) &= \mathbf{U}^n_{i-1} + \mathbf{s}_{i-1}(\mathbf{x}_{i-1/2} - \mathbf{v}(t-t^n) - \mathbf{x}_{i-1}) \\ &= \mathbf{U}^n_{i-1} + \mathbf{s}_{i-1}(1/2\Delta x - \mathbf{v}(t-t^n))\end{aligned}$$

To understand how the $\mathbf{x}_{i-1/2} - \mathbf{v}(t-t^n)$ comes into play, imagine the state doesn't change (i.e. isn't advected), but you move the boundaries to the left instead over a distance $\mathbf{v}(t-t^n)$.

So if we have a **negative** constant velocity, the term changes to

$$\begin{aligned}\mathbf{U}(x = x_{i-1/2}, t) &= \mathbf{U}^n_{i} + \mathbf{s}_{i}(\mathbf{x}_{i-1/2} - \mathbf{v}(t-t^n) - \mathbf{x}_{i}) \\ &= \mathbf{U}^n_{i} + \mathbf{s}_{i}(-\mathbf{v}(t-t^n) - \Delta x)\end{aligned}$$

Note that the minus sign remains, and that the indices changed by one because we need

to always make sure to do upwind differencing, i.e. take only values where the flow comes from, not from the direction where it's going.

Finally, we can compute the average flux over the time step $\Delta t = t^{n+1} - t^n$:

$$\mathbf{F}_{i-1/2}^{n+1/2} = \langle \mathbf{F}_{i+1/2}(t) \rangle_{t^n}^{t^{n+1}} = \frac{1}{\Delta t} \int_{t^n}^{t^{n+1}} \mathbf{v}\mathbf{U}(\mathbf{x} = \mathbf{x}_{i-1/2}, t) \tag{40}$$

$$= \frac{1}{\Delta t} \int_{t^n}^{t^{n+1}} \mathbf{v} \left(\mathbf{U}_{i-1}^n + \mathbf{s}_{i-1}(1/2\Delta x - \mathbf{v}(t - t^n)) \right) \tag{41}$$

$$= \mathbf{v} \left(\mathbf{U}_{i-1}^n + \mathbf{s}_{i-1} \left(1/2\Delta x - \mathbf{v} \left(\left[\frac{1}{2\Delta t} t^2 \right]_{t^n}^{t^{n+1}} - t^n \right) \right) \right) \tag{42}$$

$$= \mathbf{v} \left(\mathbf{U}_{i-1}^n + \mathbf{s}_{i-1} \left(1/2\Delta x - \mathbf{v} \left[\frac{1}{2}(t^{n+1} + t^n) - t^n \right] \right) \right) \tag{43}$$

$$= \mathbf{v} \left(\mathbf{U}_{i-1}^n + 1/2\mathbf{s}_{i-1} \left(\Delta x - \mathbf{v}\Delta t \right) \right) \tag{44}$$

Finally averaging the fluxes over a time step gives:

$$\mathbf{U}_i^{n+1} = \mathbf{U}_i^n - \mathbf{v} \cdot \frac{\Delta t}{\Delta x} (\mathbf{U}_i^n - \mathbf{U}_{i-1}^n) - \mathbf{v} \cdot \frac{\Delta t}{\Delta x} \frac{1}{2} (\mathbf{s}_i^n - \mathbf{s}_{i-1}^n)(\Delta x - \mathbf{v}\Delta t) \tag{45}$$

This is the same as eq. 37 where we used

$$\begin{aligned} \mathbf{F}_{i+1/2}^{n+1/2} &= \mathbf{v}_{i+1/2} \cdot \mathbf{U}_{i+1/2}^{n+1/2} \\ &= \mathbf{v} \cdot \mathbf{U}(\mathbf{x}_{i+1/2} - 1/2\mathbf{v}\Delta t) \\ &= \mathbf{v} \cdot \left(\mathbf{U}_i^n + \mathbf{s}_i^n [(\mathbf{x}_{i+1/2} - 1/2\mathbf{v}\Delta t) - \mathbf{x}_i] \right) \\ &= \mathbf{v} \cdot (\mathbf{U}_i^n + 1/2\mathbf{s}_i^n (\Delta \mathbf{x} - \mathbf{v}\Delta t)) \end{aligned}$$

and analoguely

$$\mathbf{F}_{i-1/2}^{n+1/2} = \mathbf{v} \cdot \left(\mathbf{U}_{i-1}^n + \frac{1}{2}\mathbf{s}_{i-1}^n (\Delta \mathbf{x} - \mathbf{v}\Delta t) \right)$$

To summarize the formulae:

$$\mathbf{U}_i^{n+1} = \mathbf{U}_i^n + \frac{\Delta t}{\Delta x}\left(\mathbf{F}_{i-1/2}^{n+1/2} - \mathbf{F}_{i+1/2}^{n+1/2}\right)$$

$$\mathbf{F}_{i-1/2}^{n+1/2} = \begin{cases} \mathbf{v}_{i-1/2}\cdot\mathbf{U}_{i-1}^n + \frac{1}{2}\mathbf{v}_{i-1/2}\cdot\mathbf{s}_{i-1}^n(\Delta\mathbf{x} - \mathbf{v}_{i-1/2}\Delta t) & \text{for } \mathbf{v}\geq 0 \\ \mathbf{v}_{i-1/2}\cdot\mathbf{U}_i^n - \frac{1}{2}\mathbf{v}_{i-1/2}\cdot\mathbf{s}_i^n(\Delta\mathbf{x} + \mathbf{v}_{i-1/2}\Delta t) & \text{for } \mathbf{v}\leq 0 \end{cases}$$

$$\mathbf{F}_{i+1/2}^{n+1/2} = \begin{cases} \mathbf{v}_{i+1/2}\cdot\mathbf{U}_i^n + \frac{1}{2}\mathbf{v}_{i+1/2}\cdot\mathbf{s}_i^n(\Delta\mathbf{x} - \mathbf{v}_{i+1/2}\Delta t) & \text{for } \mathbf{v}\geq 0 \\ \mathbf{v}_{i+1/2}\cdot\mathbf{U}_{i+1}^n - \frac{1}{2}\mathbf{v}_{i+1/2}\cdot\mathbf{s}_{i+1}^n(\Delta\mathbf{x} + \mathbf{v}_{i+1/2}\Delta t) & \text{for } \mathbf{v}\leq 0 \end{cases}$$

We can now insert a more general expression for the slopes. Let

$$\theta_{i-1/2} = \begin{cases} +1 & \text{for } \mathbf{v}\geq 0 \\ -1 & \text{for } \mathbf{v}\leq 0 \end{cases} \tag{46}$$

Then

$$\Delta x_{i-\{0,1\}}\mathbf{s}_{i-\{0,1\}} = \frac{1}{2}\Delta x\left[(1+\theta_{i-1/2})\mathbf{s}_{i-1}^n + (1-\theta_{i-1/2})\mathbf{s}_i^n\right] \tag{47}$$

$$\equiv \phi(r_{i-1/2}^n)(\mathbf{U}_i^n - \mathbf{U}_{i-1}^n) \tag{48}$$

$$r_{i-1/2}^n = \begin{cases} \frac{\mathbf{U}_{i-1}^n - \mathbf{U}_{i-2}^n}{\mathbf{U}_i^n - \mathbf{U}_{i-1}^n} & \text{for } \mathbf{v}\geq 0 \\ \frac{\mathbf{U}_{i+1}^n - \mathbf{U}_i^n}{\mathbf{U}_i^n - \mathbf{U}_{i-1}^n} & \text{for } \mathbf{v}\leq 0 \end{cases} \tag{49}$$

ϕ is discussed later. Finally:

$$\begin{aligned}\mathbf{F}_{i-1/2}^{n+1/2} =& \frac{1}{2}\mathbf{v}_{i-1/2}\left[(1+\theta_{i-1/2})\mathbf{U}_{i-1}^n + (1-\theta_{i-1/2})\mathbf{U}_i^n\right] + \\ & \frac{1}{2}|\mathbf{v}_{i-1/2}|\left(1 - \left|\frac{\mathbf{v}_{i-1/2}\Delta t}{\Delta x}\right|\right)\phi(r_{i-1/2}^n)(\mathbf{U}_i^n - \mathbf{U}_{i-1}^n)\end{aligned} \tag{50}$$

$$\begin{aligned}\mathbf{F}_{i+1/2}^{n+1/2} =& \frac{1}{2}\mathbf{v}_{i+1/2}\left[(1+\theta_{i+1/2})\mathbf{U}_i^n + (1-\theta_{i+1/2})\mathbf{U}_{i+1}^n\right] + \\ & \frac{1}{2}|\mathbf{v}_{i+1/2}|\left(1 - \left|\frac{\mathbf{v}_{i+1/2}\Delta t}{\Delta x}\right|\right)\phi(r_{i+1/2}^n)(\mathbf{U}_{i+1}^n - \mathbf{U}_i^n)\end{aligned} \tag{51}$$

Depending on our choice of ϕ, we can get different slopes. Here for positive velocity only, and for $r = r_{i-1/2}$:

$$\phi(r) = 0 \rightarrow \mathbf{s}_i = 0 \qquad \text{No slopes; Piecewise constant method.}$$

$$\phi(r) = 1 \rightarrow \mathbf{s}_i = \frac{\mathbf{U}_i - \mathbf{U}_{i-1}}{\Delta x} \qquad \text{Downwind slope (Lax-Wendroff)}$$

$$\phi(r) = r \rightarrow \mathbf{s}_i = \frac{\mathbf{U}_{i-1} - \mathbf{U}_{i-2}}{\Delta x} \qquad \text{Upwind slope (Beam-Warming)}$$

$$\phi(r) = \frac{1}{2}(1 + r) \rightarrow \mathbf{s}_i = \frac{\mathbf{U}_i - \mathbf{U}_{i-2}}{2\Delta x} \qquad \text{Centered slope (Fromm)}$$

8.4 CFL Condition

To keep things stable and physical, we must not allow any flux in the simulation to go further than one single cell size. Otherwise, you're skipping interactions between fluxes on cells. This time restriction is known as the CFL condition.

In 1D, it's straightforward:

$$\Delta t_{max} = C_{cfl} \frac{\Delta x}{v_{max}} \tag{52}$$

$C_{cfl} \in [0, 1)$ is a user-set factor. The lower it is, the more precise the results, but the more computations you need to do.

In 2D, it is:

$$\Delta t_{max} = C_{cfl} \left(\frac{|v_{x,max}|}{\Delta x} + \frac{|v_{y,max}|}{\Delta y} \right)^{-1} \tag{53}$$

This condition is more strict than what one would expect from the restriction based on physical arguments, i.e. not allowing the flux to pass more than one cell, which would be $\Delta t_{max} = C_{cfl} \min \left\{ \frac{\Delta x}{|v_{x,max}|}, \frac{\Delta y}{|v_{y,max}|} \right\}$. It follows from a convergence condition in (von Neumann) stability analysis of the method.

For N dimensions, the condition translates to

$$\Delta t_{max} = C_{cfl} \left(\sum_{i=1}^{N} \frac{|v_{i,max}|}{\Delta x_i} \right)^{-1} \tag{54}$$

8.5 Implementation Details

What is implemented is the equation

$$\frac{\partial \mathbf{U}}{\partial t} + \mathbf{v} \cdot \frac{\partial \mathbf{U}}{\partial \mathbf{x}} = 0 \tag{55}$$

where we assume that the velocity $\mathbf{v}$ is constant. Therefore, the fluid velocity is never updated, but kept identical to the initial conditions.

You can change that behaviour by removing the `ADVECTION_KEEP_VELOCITY_CONSTANT` macro definition in `defines.h`

References

LeVeque, Randall J. (2002). *Finite Volume Methods for Hyperbolic Problems*. Cambridge Texts in Applied Mathematics. Cambridge University Press. DOI: 10.1017/CBO9780511791253.