Exercises for the lecture

Fundamentals of Simulation Methods

WS 2015/16

Lecturer: Volker Springel

Exercise sheet 8 ***(due date: Jan 8, 2016, 11am)***

Hyperbolic conservation laws / finite volume scheme

1) An isothermal Riemann problem

Consider the isothermal gas equations in one dimension, governed by the set of conservation laws

$$\partial_t \left(\begin{array}{c} \rho \\ \rho u \end{array} \right) + \partial_x \left(\begin{array}{c} \rho u \\ \rho u^2 + \rho c_s^2 \end{array} \right) = 0, \tag{1}$$

where ρ and u are the gas density and velocity, respectively, and c_s is the constant speed of sound. We now consider two streams of gas of equal density ρ_0 that collide at $t = 0$ at a fiducial interface $x = 0$, i.e. at time $t = 0$ the density is $\rho(x) = \rho_0$, and the velocity field is $u(x) = -u_0$ for $x > 0$ and $u(x) = u_0$ for $x < 0$.

(a) Make a sketch of $\rho(x)$ and $u(x)$ at some later time $t_1 > t_0$.

(b) Now suppose the density measured at $x = 0$ at this later time is $\rho_1 = 3\,\rho_0$, and a shock propagating towards the right, starting at $x = 0$, has been identified. Calculate the shock velocity in units of u_0. (Note that another shock is moving with equal but opposite velocity to the left.)

(c) Determine the sound speed in units of u_0.

(d) What is the Mach number of the two shocks (i.e. their velocity *relative* to the pre-shock gas, divided by the sound-speed)?

2) Roe's approximate Riemann solver for the isothermal problem

We begin by considering the one-dimensional isothermal Riemann problem, *augmented* with an advection of the y-momentum:

$$\partial_t \mathbf{U} + \partial_x \mathbf{F} = 0, \tag{2}$$

where the state vector $\mathbf{U}$ and flux vector $\mathbf{F}(\mathbf{U})$ are given as

$$\mathbf{U} = \left(\begin{array}{c} \rho \\ \rho u \\ \rho v \end{array} \right), \quad \mathbf{F} = \left(\begin{array}{c} \rho u \\ \rho u^2 + \rho c_s^2 \\ \rho u v \end{array} \right). \tag{3}$$

c_s is the fixed sound speed, and (u, v) describes the velocity field in the x- and y-directions, respectively.

We now seek an (approximate) solution for the flux across the interfaces of cells of width Δx, allowing us to update the state of the cells as

$$\mathbf{U}_i^{n+1} = \mathbf{U}_i^n + \frac{\Delta t}{\Delta x}\left[\mathbf{F}_{i-\frac{1}{2}} - \mathbf{F}_{i+\frac{1}{2}}\right]. \tag{4}$$

If we consider each cell interface as an initial value problem with initial conditions

$$\mathbf{U}(x,0) = \begin{cases} \mathbf{U}_L & \text{for } x < 0 \\ \mathbf{U}_R & \text{for } x > 0 \end{cases} \tag{5}$$

then Godunov's original first order method consists of taking as flux $\mathbf{F}_{i+1/2} = \mathbf{F}(\mathbf{U}_{i+1/2}(0))$, where $\mathbf{U}_{i+1/2}(0) = \mathbf{U}_{i+1/2}(x/t)$ is the *exact* self-similar solution of the Riemann problem on the characteristic emanating from the interface.

Instead of seeking the exact solution of the isothermal Riemann problem (which is not too difficult actually, but requires the iterative solution of a non-linear equation), we shall here calculate an estimate of the flux based on an approximate solution of the Riemann problem, following the approach of Roe. To this end, we linearize the PDE. We can first rewrite equation (2) as:

$$\partial_t \mathbf{U} + \mathbf{A}(\mathbf{U})\,\partial_x \mathbf{U} = 0, \tag{6}$$

where $\mathbf{A} = \frac{\partial \mathbf{F}}{\partial \mathbf{U}}$ is a Jacobian matrix. The central idea for linearizing is now to replace $\mathbf{A}$ with a matrix $\tilde{\mathbf{A}}$ that only depends on the left and right states, $\tilde{\mathbf{A}} = \tilde{\mathbf{A}}(\mathbf{U}_L, \mathbf{U}_R)$. This in turn will convert equation (6) into a linear system with constant coefficients. Roe's Riemann solver sets

$$\tilde{\mathbf{A}} = \begin{pmatrix} 0 & 1 & 0 \\ c_s^2 - \tilde{u}^2 & 2\tilde{u} & 0 \\ -\tilde{u}\tilde{v} & \tilde{v} & \tilde{u} \end{pmatrix} \tag{7}$$

for this matrix, where $\tilde{u}$ and $\tilde{v}$ are weighted averages of the velocities of the left and right state, with the weights chosen as square roots of the density, i.e. $\tilde{u} = (\sqrt{\rho_L}u_L + \sqrt{\rho_R}u_R)/(\sqrt{\rho_L} + \sqrt{\rho_R})$ and similarly for $\tilde{v}$. Note that $\tilde{\mathbf{A}}(\mathbf{U},\mathbf{U}) = \mathbf{A}$, i.e. this average is consistent with the original Jacobian. The linearized system can now be diagonalized and solved as a set of linear advection problems.

(a) Verify that $\lambda_1 = \tilde{u} - c_s$, $\lambda_2 = \tilde{u} + c_s$ and $\lambda_3 = \tilde{u}$ are eigenvalues of $\tilde{\mathbf{A}}$ with the eigenvectors

$$\mathbf{K}_1 = \begin{pmatrix} 1 \\ \tilde{u} - c_s \\ \tilde{v} \end{pmatrix}, \quad \mathbf{K}_2 = \begin{pmatrix} 1 \\ \tilde{u} + c_s \\ \tilde{v} \end{pmatrix}, \quad \mathbf{K}_3 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}. \tag{8}$$

(b) Check that the difference vector $\Delta\mathbf{U} = (u_1, u_2, u_3) = \mathbf{U}_R - \mathbf{U}_L$ between the states left and right can be expressed in terms of the eigenvectors as $\Delta\mathbf{U} = \sum_i \alpha_i \mathbf{K}_i$, with coefficients

$$\alpha_1 = \frac{(\tilde{u} + c_s)u_1 - u_2}{2c_s}, \quad \alpha_2 = \frac{-(\tilde{u} - c_s)u_1 + u_2}{2c_s}, \quad \alpha_3 = u_3 - \tilde{v}u_1. \tag{9}$$

(c) Write a subroutine `riemann_roe_isothermal` that takes as input arguments the state left and right of an interface, i.e. ρ_L, u_L, v_L, ρ_R, u_R, v_R, as well as the sound speed c_s, and returns the three components $\mathbf{F}^\star = (f_1, f_2, f_3)$ of the flux vector, which is given as

$$\mathbf{F}^\star = \frac{1}{2}(\mathbf{F}_L + \mathbf{F}_R) - \frac{1}{2}\sum_i \alpha_i\,|\lambda_i|\,\mathbf{K}_i \tag{10}$$

(d) Test your Riemann solver with the input values given in the table below, assuming $c_s = 2.0$. The first two expected results are included in the table already, verify those and determine the one for the third row.

ρ_L	u_L	v_L	ρ_R	u_R	v_R	f_1	f_2	f_3)
1.0	1.0	2.0	3.0	1.0	0.0	0.0	6.0	1.268
2.5	2.0	3.0	1.0	-3.0	-2.0	2.6243	24.602	12.475
2.0	-1.0	-2.0	1.0	-1.0	2.0			

3) Doing a one-dimensional sweep

We now consider a two-dimensional periodic domain of extension $[L_x, L_y]$, subdivided into $N \times M$ cells. We want to carry out on this domain one timestep according to equation (4), with Roe's approximate flux as derived above.

To this end, write a subroutine `sweep` that takes as input parameters the density field $\rho[i, j]$ and the velocity fields $u[i, j]$ and $v[i, j]$. Further, the routine should accept the timestep Δt and the mesh-spacing Δx, as well as the mesh dimensions N and M as input. Finally, pass two small relative offset vectors to the routine which tell it which cell should be considered "left" and which "right" of any current cell. For example, if the x-direction is evolved, this offset vector would be $(-1, 0)$ for the left cell and $(+1, 0)$ for the right cell, since $\rho_{i-1,j}$ is then considered "left" of $\rho_{i,,j}$. In terms of output, the subroutine should automatically update the input fields $\rho[i, j]$, $u[i, j]$ and $v[i, j]$ with the new values at the end of the timestep.

Implement the following steps in the routine `sweep`:

(a) Create three empty arrays for the components of the new state $\mathbf{U}^{n+1} = (q_1, q_2, q_3)$ at the end of the sweep.

(b) Write two nested loops that go over all primary cells, $i = 0 \ldots N - 1$, $j = 0 \ldots M - 1$, and which identify the left and right cells of the current cell (here you need to observe the periodic boundary conditions by wrapping around if needed). Call the routine `riemann_roe_isothermal` that you wrote for both the left and the right interface of the cell, obtaining two flux vectors, $\mathbf{F}^\star_{i-1/2}$ and $\mathbf{F}^\star_{i+1/2}$. Calculate the new state of each cell as

$$\mathbf{U}^{n+1} = \mathbf{U}^n + \frac{\Delta t}{\Delta x}\left[\mathbf{F}^\star_{i-1/2} - \mathbf{F}^\star_{i+1/2}\right]. \tag{11}$$

(c) Based on $q_1[i, j]$, $q_2[i, j]$, and $q_3[i, j]$, calculate the new values for the density field $\rho[i, j]$, and the velocity fields $u[i, j]$ and $v[i, j]$, which forms the output of the routine.

4) Carrying out a multidimensional simulation

We now consider the full two-dimensional isothermal problem, which takes the form

$$\partial_t \mathbf{U} + \partial_x \mathbf{F} + \partial_y \mathbf{G} = 0, \tag{12}$$

with

$$\mathbf{U} = \begin{pmatrix} \rho \\ \rho\, u \\ \rho\, v \end{pmatrix}, \quad \mathbf{F} = \begin{pmatrix} \rho u \\ \rho u^2 + \rho c_s^2 \\ \rho u v \end{pmatrix}, \quad \mathbf{G} = \begin{pmatrix} \rho v \\ \rho u v \\ \rho v^2 + \rho c_s^2 \end{pmatrix}. \tag{13}$$

As we discussed in the lecture, one possibility to solve this system lies in operator splitting, essentially by achieving a full time advance over a timestep Δt by solving the two problems

$$\partial_t \mathbf{U} + \partial_x \mathbf{F} = 0 \tag{14}$$

and

$$\partial_t \mathbf{U} + \partial_y \mathbf{G} = 0 \tag{15}$$

independently one after the other. To linear order in time, this can be simply done by carrying out sweeps for these equations one after the other.

(a) Verify that our subroutine `sweep` can be used without change to carry out a sweep in the y-direction, corresponding to a step of equation (15), provided the velocity fields are appropriately passed. The sequence of calls

`sweep`, ρ, u, v, Δt, Δx, c_s, N, M, -1, 0, 1, 0

`sweep`, ρ, v, u, Δt, Δy, c_s, N, M, 0, -1, 0, 1

can hence be used to first do a sweep in the x-direction, followed by one in the y-direction.

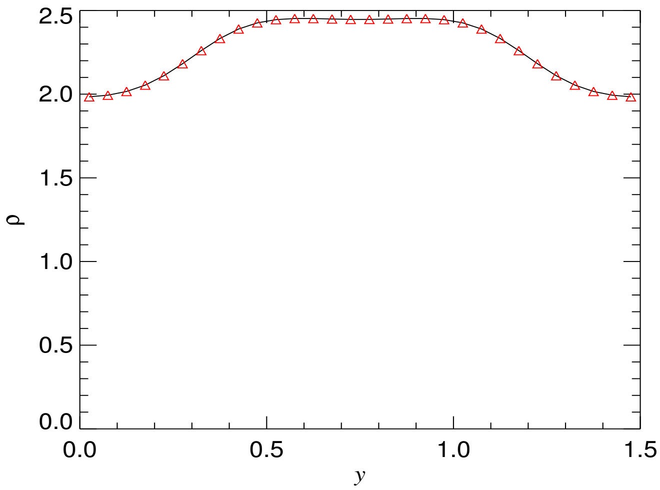
(b) Write a short code that integrates the 2D isothermal problem over a certain time interval $[0, T_{\rm max}]$, based on your function `sweep`. Adopt as timestep

$$\Delta t = C_{\rm CFL} \frac{\min(\Delta x, \Delta y)}{c_s + \max(u, v)}. \tag{16}$$

(c) For definiteness, take $L_x = 3.0$, $L_y = 1.5$, $c_s = 2.0$, $T_{\rm max} = 1.5$, and the initial conditions

$$\rho(x, y) = \begin{cases} 4.0 & \text{for } |x - L_x/2| < L_x/4 \text{ and } |y - L_y/2| < L_y/4 \\ 1.0 & \text{otherwise,} \end{cases} \tag{17}$$

with $u(x, y) = v(x, y) = 0$. Evolve the system with $C_{\rm CFL} = 0.4$ and $N = 60$, $M = 30$. Plot the density field at the final time $t = T_{\rm max}$ along the x-axis and along the y-axis through the mid-point of the box. If everything works, the result for the y-axis should not be too different from the result below:



(Optional: You may also produce a movie of the time evolution of the density field, and try things out with higher grid resolution.)