Fundamentals of Simulation Methods

Exercise Sheet 8 — Hyperbolic conservation laws and finite volume scheme

Daniel Rosenblüh, Janosh Riebesell · Lecturer: Volker Springel · January 8th, 2016

1 An isothermal Riemann problem

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

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

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 time $t_0 = 0$ at a fiducial interface $x = 0$, i.e. at $t_0 = 0$, 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 $x < 0$.

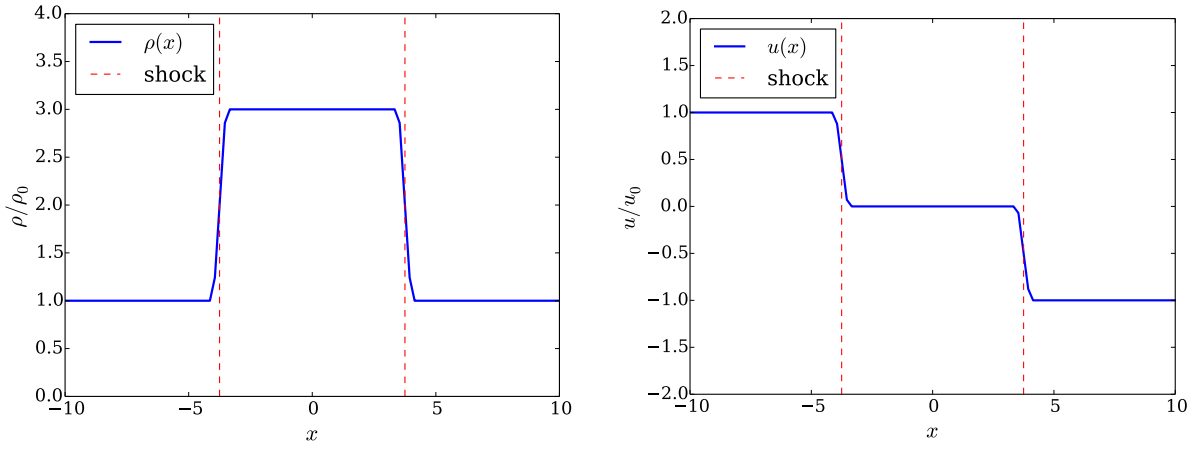
(a) Make a sketch of $\rho(x)$ and $u(x)$ at some later time $t_1 > t_0$.

(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)?

(a) Figure 1 (a) and (b) give a rough idea of how the gas density $\rho(x)$ and velocity $u(x)$ look like at time $t_1 > t_0$ after the two shocks have begun their outward propagation.



(a) Sketch of $\rho(x)$ (b) Sketch of $u(x)$

Figure 1: Density and velocity distribution in the Riemann problem some time after the initial conditions were allowed to evolve

(b) The Euler equations in their weak formulation as integral equations remain valid even at the discontinuity of a shock front, meaning that even at a shock, mass, momentum and energy must be conserved. These requirements are known as the Rankine-Hugoniot jump conditions and they govern the change in density ρ, pressure $P = \rho c_s^2$, and velocity u during a shock:

$$\rho_i u_i = \rho_o u_o \qquad \text{(mass flux in } \stackrel{!}{=} \text{ mass flux out)}, \tag{2}$$

$$\rho_i u_i^2 + P_i = \rho_o u_o^2 + P_o \qquad \text{(momentum flux in } \stackrel{!}{=} \text{ momentum flux out)}, \tag{3}$$

$$(\rho_i e_i + P_i)u_i = (\rho_o e_o + P_o)u_o \quad \text{(energy flux in } \stackrel{!}{=} \text{ energy flux out)}. \tag{4}$$

Note: An isothermal shock necessarily involves some form of heat dissipation, such as thermal radiation. No such process is included in eq. (4), as it was already absent from its root, the third Euler equation,

$$\partial_t(\rho e) + \partial_x[(\rho e + P)u] = 0. \tag{5}$$

Thus, eq. (4) is an incomplete statement about energy conservation in the context of isothermal processes. This also explains the third Euler equation's absence from the isothermal gas equations (1), which are otherwise identical to the Euler equations. While eqs. (4) and (5) could be modified to incorporate heat dissipation, it is more common to simply replace them by the requirement that the flow should return to its original temperature after the shock has passed, i.e.

$$T_0 = T_1. \tag{6}$$

Conditions (2) and (3) are derived from the integral form of eq. (1) and continue to hold without change also for isothermal systems.

To determine the shock velocity, we need to take into account that the two shock waves propagating outward in either direction from $x = 0$ form a compressed region of constant density $\rho_1 = 3\rho_0$ and, by symmetry, zero velocity $u_1 = 0$. Furthermore, gas which has not yet gone through the shock front is undisturbed, i.e. still travels with velocity u_0 towards the shock. The trick with problems such as this one is to move into the rest frame of one of the shock waves. Let us choose the right-moving one where gas travels into the shock with density $\rho_i = \rho_0$ and velocity $u_i = -u_0 - u_s$ and out again with density $\rho_o = 3\rho_0$ and velocity $u_o = -u_s$. Inserting these values into the first Rankine-Hugoniot condition, i.e. eq. (2), we obtain

$$\rho_0(-u_s - u_0) = -3\rho_0 u_s, \tag{7}$$

Solving for u_s, we get a shock velocity of

$$u_s = \frac{u_0}{2}. \tag{8}$$

(c) We use again that in the rest frame of the shock, gas flows into it with velocity $u_i = -u_0 - u_s$ and out again with $u_o = -u_s$. The in- and outgoing densities remain $\rho_i = \rho_0$ and $\rho_o = \rho_1 = 3\rho_0$, so that the second jump condition reads

$$\rho_0\left[(u_0 + u_s)^2 + c_s^2\right] = 3\rho_0(u_s^2 + c_s^2). \tag{9}$$

which can be simplified to

$$u_0^2 + 2u_0u_s + u_s^2 + c_s^2 = 3u_s^2 + 3c_s^2. \tag{10}$$

By inserting our result for the shock velocity from part (b), we obtain

$$\underbrace{u_0^2 + 2u_0\frac{u_0}{2} - 2\frac{u_0^2}{4}}_{\frac{3}{2}u_0^2} = 2c_s^2 \tag{11}$$

and thus arrive at a speed of sound of

$$c_s = \frac{\sqrt{3}}{2}u_0. \tag{12}$$

(d) The Mach number is defined as the ratio of the shock velocity to the speed of sound *in the rest frame of the gas* which is hit by the shock. Since in the rest frame of the shock, the gas flows into the shock with velocity $-u_0 - u_s$, the shock appears to move towards the gas with velocity $u_0 + u_s$ in its own rest frame. Dividing $u_0 + u_s$ by the speed of sound, we get a Mach number of

$$\mathcal{M} = \frac{u_0 + u_s}{c_s} = \frac{u_0 + \frac{u_0}{2}}{\frac{\sqrt{3}}{2} u_0} = \sqrt{3}. \tag{13}$$

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:

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

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

$$\boldsymbol{U} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \end{pmatrix}, \quad \boldsymbol{F} = \begin{pmatrix} \rho u \\ \rho(u^2 + c_s^2) \\ \rho u v \end{pmatrix}. \tag{15}$$

c_s is the fixed sound speed, and u and v describe 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

$$\boldsymbol{U}_i^{n+1} = \boldsymbol{U}_i^n + \frac{\Delta t}{\Delta x} \left[\boldsymbol{F}_{i-\frac{1}{2}} - \boldsymbol{F}_{i+\frac{1}{2}} \right]. \tag{16}$$

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

$$\boldsymbol{U}(x, 0) = \begin{cases} \boldsymbol{U}_L & \text{for } x < 0, \\ \boldsymbol{U}_R & \text{for } x > 0, \end{cases} \tag{17}$$

then Godunov's original first order method consists of taking as flux $\boldsymbol{F}_{i+\frac{1}{2}} = F\left[\boldsymbol{U}_{i+\frac{1}{2}}(0)\right]$, where $\boldsymbol{U}_{i+\frac{1}{2}}(0) = \boldsymbol{U}_{i+\frac{1}{2}}(x/t)$ 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 eq. (14) as

$$\partial_t \boldsymbol{U} + \mathbf{A}(\boldsymbol{U}) \partial_x \boldsymbol{U} = 0, \quad \text{where } \mathbf{A} = \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{U}} \tag{18}$$

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, $\tilde{\mathbf{A}} = \tilde{\mathbf{A}}(\boldsymbol{U}_L, \boldsymbol{U}_R)$. This in turn will convert eq. (18) into a linear system with constant coefficients. Roe's Riemann solver sets for this matrix

$$\tilde{\mathbf{A}} = \begin{pmatrix} 0 & 1 & 0 \\ c_s^2 - \tilde{u}^2 & 2\tilde{u} & 0 \\ -\tilde{u}\tilde{v} & \tilde{v} & \tilde{u} \end{pmatrix}, \tag{19}$$

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.

$$\tilde{u} = \frac{\sqrt{\rho_L} u_L + \sqrt{\rho_R} u_R}{\sqrt{\rho_L} + \sqrt{\rho_R}}, \tag{20}$$

and similarly for $\tilde{v}$. Note that $\tilde{\mathbf{A}}(\boldsymbol{U}, \boldsymbol{U}) = \mathbf{A}$, 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

$$\boldsymbol{K}_1 = \begin{pmatrix} 1 \\ \tilde{u} - c_s \\ \tilde{v} \end{pmatrix}, \quad \boldsymbol{K}_2 = \begin{pmatrix} 1 \\ \tilde{u} + c_s \\ \tilde{v} \end{pmatrix}, \quad \boldsymbol{K}_3 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}. \tag{21}$$

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

$$\alpha_1 = \frac{(\tilde{u} + c_s)u_1 - u_2}{2c_s}, \quad \alpha_2 = \frac{-(\tilde{u} - c_s)u_1 + u_2}{2c_s}, \quad \alpha_3 = u_3 - \tilde{v}u_1. \tag{22}$$

(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) and (ρ_R, u_R, v_R), as well as the sound speed c_s, and returns the three components $\boldsymbol{F}^\star = (f_1, f_2, f_3)$ of the flux vector, which is given by

$$\boldsymbol{F}^\star = \frac{1}{2}(\boldsymbol{F}_L + \boldsymbol{F}_R) - \frac{1}{2}\sum_i \alpha_i \, |\lambda_i| \, \boldsymbol{K}_i. \tag{23}$$

(d) Test your Riemann solver with the input values given in Table 1, 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.

(a) Direct computation confirms that the $\boldsymbol{K}_i$ are eigenvectors of $\tilde{\mathbf{A}}$.

$$\tilde{\mathbf{A}}\boldsymbol{K}_1 = \begin{pmatrix} \tilde{u} - c_s \\ c_s^2 - \tilde{u}^2 + 2\tilde{u}(\tilde{u} - c_s) \\ -\tilde{u}\tilde{v} + \tilde{v}(\tilde{u} - c_s) + \tilde{u}\tilde{v} \end{pmatrix} = (\tilde{u} - c_s)\begin{pmatrix} 1 \\ \tilde{u} - c_s \\ \tilde{v} \end{pmatrix} = \lambda_1 \boldsymbol{K}_1. \tag{24}$$

$$\tilde{\mathbf{A}}\boldsymbol{K}_2 = \begin{pmatrix} \tilde{u} + c_s \\ c_s^2 - \tilde{u}^2 + 2\tilde{u}(\tilde{u} + c_s) \\ -\tilde{u}\tilde{v} + \tilde{v}(\tilde{u} + c_s) + \tilde{u}\tilde{v} \end{pmatrix} = (\tilde{u} + c_s)\begin{pmatrix} 1 \\ \tilde{u} + c_s \\ \tilde{v} \end{pmatrix} = \lambda_2 \boldsymbol{K}_2. \tag{25}$$

$$\tilde{\mathbf{A}}\boldsymbol{K}_3 = \begin{pmatrix} 0 \\ 0 \\ \tilde{u} \end{pmatrix} = \tilde{u}\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \lambda_3 \boldsymbol{K}_3. \tag{26}$$

(b) We compute the three components of $\sum_i \alpha_i \boldsymbol{K}_i$ separately.

$$\left(\sum_i \alpha_i \boldsymbol{K}_i\right)_x = \alpha_1 + \alpha_2 = \frac{1}{2c_s}\left[(\tilde{u} + c_s)u_1 - u_2 - (\tilde{u} - c_s)u_1 + u_2\right] = u_1. \tag{27}$$

$$\left(\sum_i \alpha_i \boldsymbol{K}_i\right)_y = \alpha_1(\tilde{u} - c_s) + \alpha_2(\tilde{u} + c_s)$$
$$= \frac{1}{2c_s}\left[(\tilde{u}^2 - c_s^2)u_1 - u_2(\tilde{u} - c_s) - (\tilde{u}^2 - c_s^2)u_1 + u_2(\tilde{u} + c_s)\right] = u_2. \tag{28}$$

$$\left(\sum_i \alpha_i \boldsymbol{K}_i\right)_z = \alpha_1\tilde{v} + \alpha_2\tilde{v} + \alpha_3$$
$$= \frac{1}{2c_s}\left[(\tilde{u} + c_s)u_1\tilde{v} - u_2\tilde{v} - (\tilde{u} - c_s)u_1\tilde{v} + u_2\tilde{v} + 2c_s(u_3 - \tilde{v}u_1)\right] = u_3. \tag{29}$$

Combining eqs. (27) to (29), we indeed have

$$\sum_i \alpha_i \boldsymbol{K}_i = \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} = \Delta \boldsymbol{U}. \tag{30}$$

(c) See `roes_riemann_solver.c`.

(d) A test of our Riemann solver with the input values supplied in Table 1 was conducted, successfully reproducing the flux values in rows 1 and 2 and generating new values as given in the third row.

Table 1: Riemann solver input values and results

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	−0.5	5.5	−2.172

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 eq. (16), 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 $U^{n+1} = (q_1, q_2, q_3)$ at the end of the sweep.

(b) Write two nested loops that go over all primary cells, $i \in \{0, ..., N-1\}$, $j \in \{0, ..., M-1\}$, 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, $\boldsymbol{F}^{\star}_{i-\frac{1}{2}}$ and $\boldsymbol{F}^{\star}_{i+\frac{1}{2}}$. Calculate the new state of each cell as

$$\boldsymbol{U}^{n+1} = \boldsymbol{U}^n + \frac{\Delta t}{\Delta x}\left[\boldsymbol{F}^\star_{i-\frac{1}{2}} - \boldsymbol{F}^\star_{i+\frac{1}{2}}\right]. \tag{31}$$

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

For parts (a) to (c), see `roes_riemann_solver.c`.

4 Carrying out a multidimensional simulation

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

$$\partial_t \boldsymbol{U} + \partial_x \boldsymbol{F} + \partial_y \boldsymbol{G} = 0, \tag{32}$$

with

$$\boldsymbol{U} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \end{pmatrix}, \quad \boldsymbol{F} = \begin{pmatrix} \rho u \\ \rho(u^2 + c_s^2) \\ \rho uv \end{pmatrix}, \quad \boldsymbol{G} = \begin{pmatrix} \rho v \\ \rho uv \\ \rho(v^2 + c_s^2) \end{pmatrix}. \tag{33}$$

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

$$\partial_t \boldsymbol{U} + \partial_x \boldsymbol{F} = 0, \tag{34}$$

and

$$\partial_t \boldsymbol{U} + \partial_y \boldsymbol{G} = 0, \tag{35}$$

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 eq. (35), 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 two-dimensional isothermal problem over a certain time interval $[0, T_{\max}]$, based on your function sweep. Adopt as timestep

$$\Delta t = C_{\mathrm{CFL}} \frac{\min(\Delta x, \Delta y)}{c_s + \max(u, v)}. \tag{36}$$

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

$$\rho(x,y) = \begin{cases} 4.0 & \text{for } |x - L_x/2| < L_x/4 \text{ and } |y - L_y/2| < L_y/4, \\ 1.0 & \text{otherwise,} \end{cases} \tag{37}$$

with $u(x,y) = v(x,y) = 0$. Evolve the system with $C_{\mathrm{CFL}} = 0.4$ and $N = 60$, $M = 30$. Plot the density field at the final time $t = T_{\max}$ 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 in Figure 2.

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

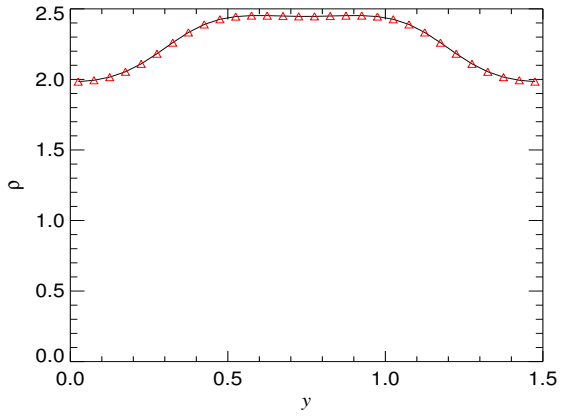
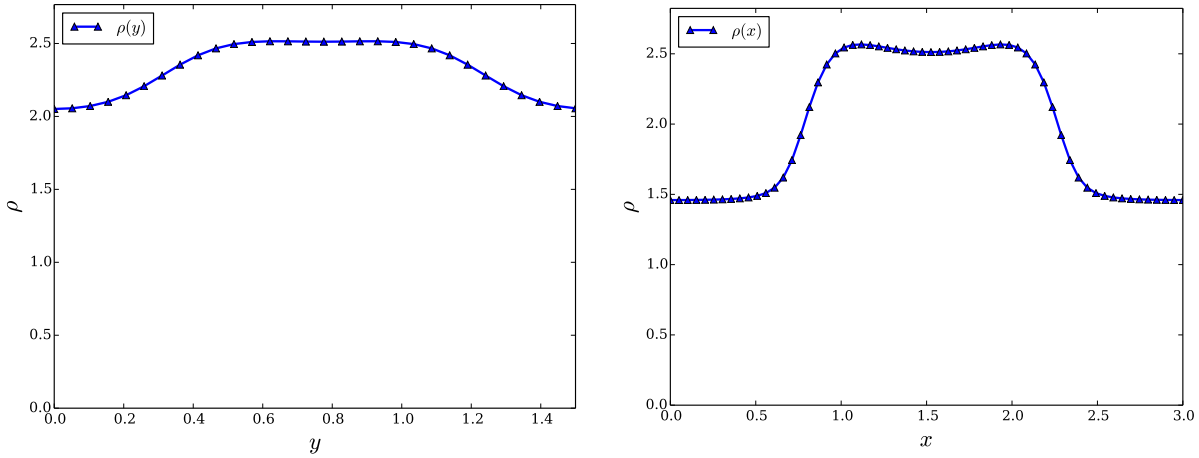


Figure 2: Expected result

For parts (a) and (b), see `roes_riemann_solver.c`.

(c) Figure 3 (a) and (b) show the density at final time $T_{\max}$ along lines of constant x and y, respectively.



(a) $\rho(y)$ for constant x

(b) $\rho(x)$ for constant y

Figure 3: Density $\rho(x, y)$ at final time $T_{\max}$ along lines parallel to the x- and y-axis through the middle of the simulation domain