Finite Volume Methods

CCA Summer School

Romain Teyssier

07/31/2023

Finite Volume Methods

- The Euler equations
- Finite Volume scheme and Godunov Method
- Godunov Method for Linear Systems
- Godunov Method for the Euler Equations
- Approximate Riemann Solvers
- High-order Schemes and Slope Limiters
- Second-order MUSCL-Hancock Scheme
- Fourth-order Finite Volume Scheme
- Adaptive Mesh Refinement
- Multidimensional Aspects

Problem Statement

A system of conservation laws

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

is hyperbolic if the Jacobian matrix $\mathbb{A} = \dfrac{\partial \mathbf{F}}{\partial \mathbf{U}}$ has real eigenvalues.

$\mathbf{F}(\mathbf{U})$ is the flux function and $\mathbf{U}$ is the vector of conservative variables.

Traditional methods are:

1- Finite Difference, 2- Finite Volume, 3- Finite Element.

If matrix $\mathbb{A}$ has constant coefficients, the problem is linear.

If $\mathbb{A}(\mathbf{U})$ and $\mathbf{F}(\mathbf{U})$ are nonlinear functions, the problem is non-linear.

In this lecture, we consider only the Euler equations and the FV method.

Euler Equations in Lagrangian Form

The simplest form to remember for the Euler equations features the Lagrangian time derivative or the time derivative "following the flow":

$$\frac{D}{Dt} = \frac{\partial}{\partial t} + v\frac{\partial}{\partial x}$$

Mass conservation: $$\frac{D\rho}{Dt} = -\rho\frac{\partial v}{\partial x}$$

Momentum conservation: $$\frac{Dv}{Dt} = -\frac{1}{\rho}\frac{\partial P}{\partial x}$$

Energy conservation: $$\rho\frac{D\epsilon}{Dt} = -P\frac{\partial v}{\partial x}$$

where the specific internal is defined as $e = \rho\epsilon$ and the EOS gives $P = (\gamma - 1)e$.

Euler Equations in Quasi-linear Form

Using the definition of the Lagrange derivative, we can easily derive the non-conservative form or quasi-linear form:

Mass conservation: $$\frac{\partial \rho}{\partial t} + v\frac{\partial \rho}{\partial x} + \rho\frac{\partial v}{\partial x} = 0$$

Momentum conservation: $$\frac{\partial v}{\partial t} + v\frac{\partial v}{\partial x} + \frac{1}{\rho}\frac{\partial P}{\partial x} = 0$$

Energy conservation: $$\rho\frac{\partial \epsilon}{\partial t} + \rho v\frac{\partial \epsilon}{\partial x} + P\frac{\partial v}{\partial x} = 0$$

Using $\frac{\partial}{\partial t}(\rho\epsilon) = \rho\frac{\partial \epsilon}{\partial t} + \epsilon\frac{\partial \rho}{\partial t}$, we find:

$$\frac{\partial}{\partial t}(e) + v\rho\frac{\partial \epsilon}{\partial x} + P\frac{\partial v}{\partial x} + v\epsilon\frac{\partial \rho}{\partial x} + e\frac{\partial v}{\partial x} = 0$$

Finally, $\frac{\partial}{\partial t}(e) + v\frac{\partial e}{\partial x} + (P+e)\frac{\partial v}{\partial x} = 0$ or $\frac{\partial}{\partial t}(P) + v\frac{\partial P}{\partial x} + \gamma P\frac{\partial v}{\partial x} = 0$

Euler Equations in Conservative Form

We start easy with $\frac{\partial \rho}{\partial t} + v\frac{\partial \rho}{\partial x} + \rho\frac{\partial v}{\partial x} = 0$ leading to $\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x}(\rho v) = 0.$

Using $\frac{\partial}{\partial t}(\rho v) = \rho\frac{\partial v}{\partial t} + v\frac{\partial \rho}{\partial t}$, we find $\frac{\partial}{\partial t}(\rho v) + \rho v\frac{\partial v}{\partial x} + \frac{\partial P}{\partial x} + v^2\frac{\partial \rho}{\partial x} + \rho v\frac{\partial v}{\partial x} = 0$

Finally, we get $\frac{\partial}{\partial t}(\rho v) + \frac{\partial}{\partial x}(\rho v^2 + P) = 0.$

Defining $E = e + \frac{1}{2}\rho v^2$, we have $\frac{\partial}{\partial t}(E) = \frac{\partial e}{\partial t} + \rho v\frac{\partial v}{\partial t} + \frac{1}{2}v^2\frac{\partial \rho}{\partial t}$.

This leads to $\frac{\partial}{\partial t}(E) + \frac{\partial}{\partial x}(ev) + P\frac{\partial v}{\partial x} + \rho v^2\frac{\partial v}{\partial x} + v\frac{\partial P}{\partial x} + \frac{1}{2}v^2\frac{\partial}{\partial x}(\rho v) = 0.$

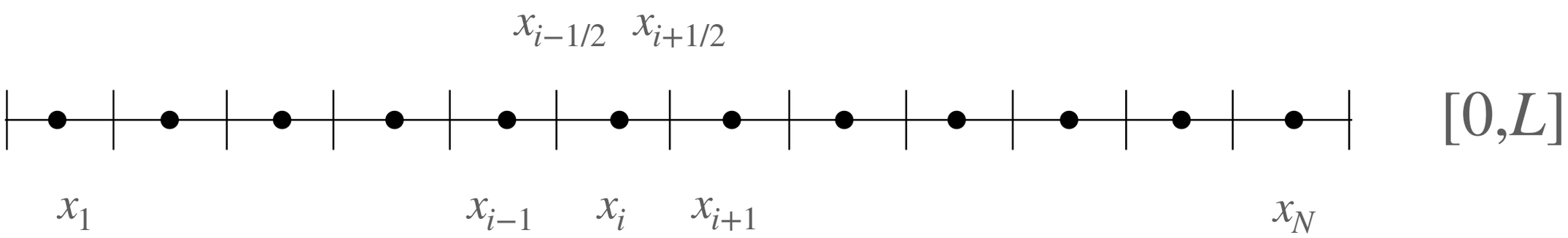
Collecting terms, we finally get:

$\frac{\partial}{\partial t}(E) + \frac{\partial}{\partial x}(ev) + \frac{\partial}{\partial x}(\frac{1}{2}\rho v^2 v) + \frac{\partial}{\partial x}(Pv) = 0$, or $\frac{\partial}{\partial t}(E) + \frac{\partial}{\partial x}(E + P)v = 0$

Finite Volume Approximation

We now introduce the finite volume (FV) method. The numerical solution is defined as the volume average of the underlying true solution at time t_n:

$$\overline{\mathbf{U}}_i^n = \frac{1}{h}\int_{x_{i-1/2}}^{x_{i+1/2}} \mathbf{U}(x,t_n)\mathrm{d}x.$$



Integrating our system of conservation laws both in time and space,

$$\int_{t_n}^{t_{n+1}}\int_{x_{i-1/2}}^{x_{i+1/2}} \left(\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x}\right)\mathrm{d}x\mathrm{d}t = 0,$$

we obtain the exact integral form: $\dfrac{\overline{\mathbf{U}}_i^{n+1} - \overline{\mathbf{U}}_i^n}{\Delta t} + \dfrac{\mathbf{F}_{i+1/2}^{n+1/2} - \mathbf{F}_{i-1/2}^{n+1/2}}{h} = 0.$

Godunov Method

The key new quantity is the numerical inter-cell flux defined by

$$\mathbf{F}_{i+1/2}^{n+1/2} = \frac{1}{\Delta t}\int_{t_n}^{t_{n+1}} \mathbf{F}(x_{i+1/2}, t)\mathrm{d}t.$$

Godunov idea is to compute this time-averaged quantity located at the interface $x_{i+1/2}$ by solving the Riemann problem defined by the initial conditions:

- $\mathbf{U}_L(x) = \overline{\mathbf{U}}_i^n$ for $x < x_{i+1/2}$
- $\mathbf{U}_R(x) = \overline{\mathbf{U}}_{i+1}^n$ for $x > x_{i+1/2}$

Let's call $\mathbf{U}_S(x,t) = \mathbf{U}_S(x/t)$ the self-similar solution to this Riemann problem.

We define the Godunov state as $\mathbf{U}_G = \mathbf{U}_S(x/t = 0)$.

We obtain $\mathbf{F}_{i+1/2}^{n+1/2} = \frac{1}{\Delta t}\int_{t_n}^{t_{n+1}} \mathbf{F}(\mathbf{U}_S(x_{i+1/2}, t))\mathrm{d}t = \mathbf{F}_G$ called the Godunov flux.

Godunov Method

If the initial solution at time t_n is piecewise constant, the Godunov method gives the exact solution at time t_{n+1}.

Note that this new solution is not piecewise constant anymore.

Assuming that the solution at t_{n+1} is also piecewise constant introduced a first-order approximation of the true solution. It however allows us to repeat the same method for the next step.

Summary for first-order Godunov method:

- Assume that the solution $\mathbf{U}_i^n(x) = \overline{\mathbf{U}}_i^n$ is piecewise constant at time t_n
- Compute the intercell flux as $\mathbf{F}_{i+1/2}^{n+1/2} = \mathrm{RP}(\overline{\mathbf{U}}_i^n, \overline{\mathbf{U}}_{i+1}^n)$
- Update the new solution by $\overline{\mathbf{U}}_i^{n+1} = \overline{\mathbf{U}}_i^n - \dfrac{\Delta t}{h}(\mathbf{F}_{i+1/2}^{n+1/2} - \mathbf{F}_{i-1/2}^{n+1/2})$
- Repeat with $\overline{\mathbf{U}}_i^{n+1}$ at time $t_{n+1} = t_n + \Delta t$

Godunov Method

The Godunov method is valid (and exact) for any nonlinear hyperbolic system of conservation laws.

The main and most expensive component of the Godunov method is the Riemann solver.

Ideally, one must compute the exact solution to the Riemann problem. These are called exact Riemann solvers.

We will see later approximate Riemann solvers for the Euler equations that are cheaper and accurate enough for most applications.

In order for the self-similar Riemann solution to be valid, the time step must satisfy the Courant stability condition $\Delta t < \dfrac{h}{|\lambda_{\max}|}$.

The Courant stability condition means here that two adjacent Riemann solutions at interface $x_{i-1/2}$ and $x_{i+1/2}$ are not allowed to interact.

Godunov Method for the Advection Equation

The advection equation is a scalar and linear hyperbolic PDE.

$$\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} = 0.$$

The finite volume scheme writes $\overline{u}_i^n = \dfrac{1}{h}\displaystyle\int_{x_{i-1/2}}^{x_{i+1/2}} u(x, t_n)\mathrm{d}x$ with solution update

$$\overline{u}_i^{n+1} = \overline{u}_i^n - \frac{\Delta t}{h}(f_{i+1/2}^{n+1/2} - f_{i-1/2}^{n+1/2})$$

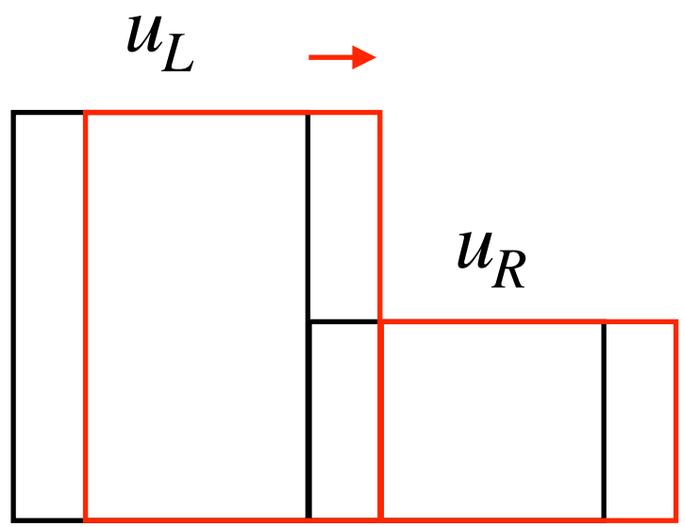
where $f_{i+1/2}^{n+1/2} = \mathrm{RP}(\overline{u}_i^n, \overline{u}_{i+1}^n)$.

The Riemann solution is quite simple with

$f_{i+1/2}^{n+1/2} = a\overline{u}_i^n$ if $a > 0$,

$f_{i+1/2}^{n+1/2} = a\overline{u}_{i+1}^n$ otherwise.

This is also called the **upwind scheme**.



$$f_G = \frac{a}{2}(\overline{u}_L + \overline{u}_R) - \frac{|a|}{2}(\overline{u}_R - \overline{u}_L).$$

Godunov Method for Linear Systems

A linear system of hyperbolic PDE can be written in matrix form as

$$\frac{\partial \mathbf{U}}{\partial t} + \mathbb{A}\frac{\partial \mathbf{U}}{\partial x} = 0.$$

The solution is updated using the conservative form with $\mathbf{F} = \mathbb{A}\mathbf{U}$:

$$\overline{\mathbf{U}}_i^{n+1} = \overline{\mathbf{U}}_i^n - \frac{\Delta t}{h}(\mathbf{F}_{i+1/2}^{n+1/2} - \mathbf{F}_{i-1/2}^{n+1/2}),$$

where $\mathbf{F}_{i+1/2}^{n+1/2} = \mathrm{RP}(\overline{\mathbf{U}}_i^n, \overline{\mathbf{U}}_{i+1}^n)$. Recall that matrix $\mathbb{A}$ has constant coefficients.

Using the eigenvector decomposition $\mathbf{V} = \mathbb{L}\mathbf{U}$, $\mathbf{U} = \mathbb{R}\mathbf{V}$ and $\mathbb{D} = \mathbb{L}\mathbb{A}\mathbb{R}$, the solution to the Riemann problem (upwinding) can be solved independently for each wave leading to the following Godunov flux:

$$\mathbf{F}_G = \frac{1}{2}\mathbb{A}(\overline{\mathbf{U}}_L + \overline{\mathbf{U}}_R) - \frac{1}{2}\mathbb{R}\,|\,\mathbb{D}\,|\,\mathbb{L}(\overline{\mathbf{U}}_R - \overline{\mathbf{U}}_L).$$

The resulting scheme is called the upwind scheme for linear systems.

Riemann Problem for Linear Systems

The Riemann problem is defined by $\mathbf{U}(x) = \overline{\mathbf{U}}_L$ for $x < 0$ and $\mathbf{U}(x) = \overline{\mathbf{U}}_R$ otherwise.

Solving $\dfrac{\partial \mathbf{U}}{\partial t} + \mathbb{A}\dfrac{\partial \mathbf{U}}{\partial x} = 0$ is equivalent to solving $\dfrac{\partial \mathbf{V}}{\partial t} + \mathbb{D}\dfrac{\partial \mathbf{V}}{\partial x} = 0$

with $\mathbf{V} = \mathbb{L}\mathbf{U}$, $\mathbf{U} = \mathbb{R}\mathbf{V}$ and $\mathbb{D} = \mathbb{L}\mathbb{A}\mathbb{R} = \mathrm{diag}(\lambda_1, \cdots, \lambda_N)$.

We have N independent Riemann problems for $\mathbf{V} = (\alpha_1, \cdots, \alpha_N)$ defined by:

- initial conditions $\alpha_j(x) = \overline{\alpha}_{j,L}$ for $x < 0$ and $\alpha_j(x) = \overline{\alpha}_{j,R}$ otherwise,
- scalar linear hyperbolic PDE: $\dfrac{\partial \alpha_j}{\partial t} + \lambda_j \dfrac{\partial \alpha_j}{\partial x} = 0.$

The Godunov state is given by the exact solution:

$\alpha_{j,G} = \overline{\alpha}_{j,L}$ if $\lambda_j > 0$, $\alpha_{j,G} = \overline{\alpha}_{j,R}$ otherwise. This gives us $\mathbf{V}_G = (\alpha_{1,G}, \cdots, \alpha_{N,G})$.

We get the final Godunov flux as: $\mathbf{F}_G = \mathbb{A}\mathbf{U}_G = \mathbb{A}\mathbb{R}\mathbf{V}_G$.

Godunov Method for Linearized Euler

For the adiabatic Euler equations, we have $\mathbf{U} = [\delta\rho, \delta v, \delta P]^T$ and the matrix:

$$\mathbb{A} = \begin{bmatrix} v_0 & \rho_0 & 0 \\ 0 & v_0 & \frac{1}{\rho_0} \\ 0 & \gamma P_0 & v_0 \end{bmatrix}$$ with eigenvalues $\lambda^+ = v_0 + c_0$, $\lambda^0 = v_0$ and $\lambda^- = v_0 - c_0$.

and $c_0^2 = \dfrac{\gamma P_0}{\rho_0}$

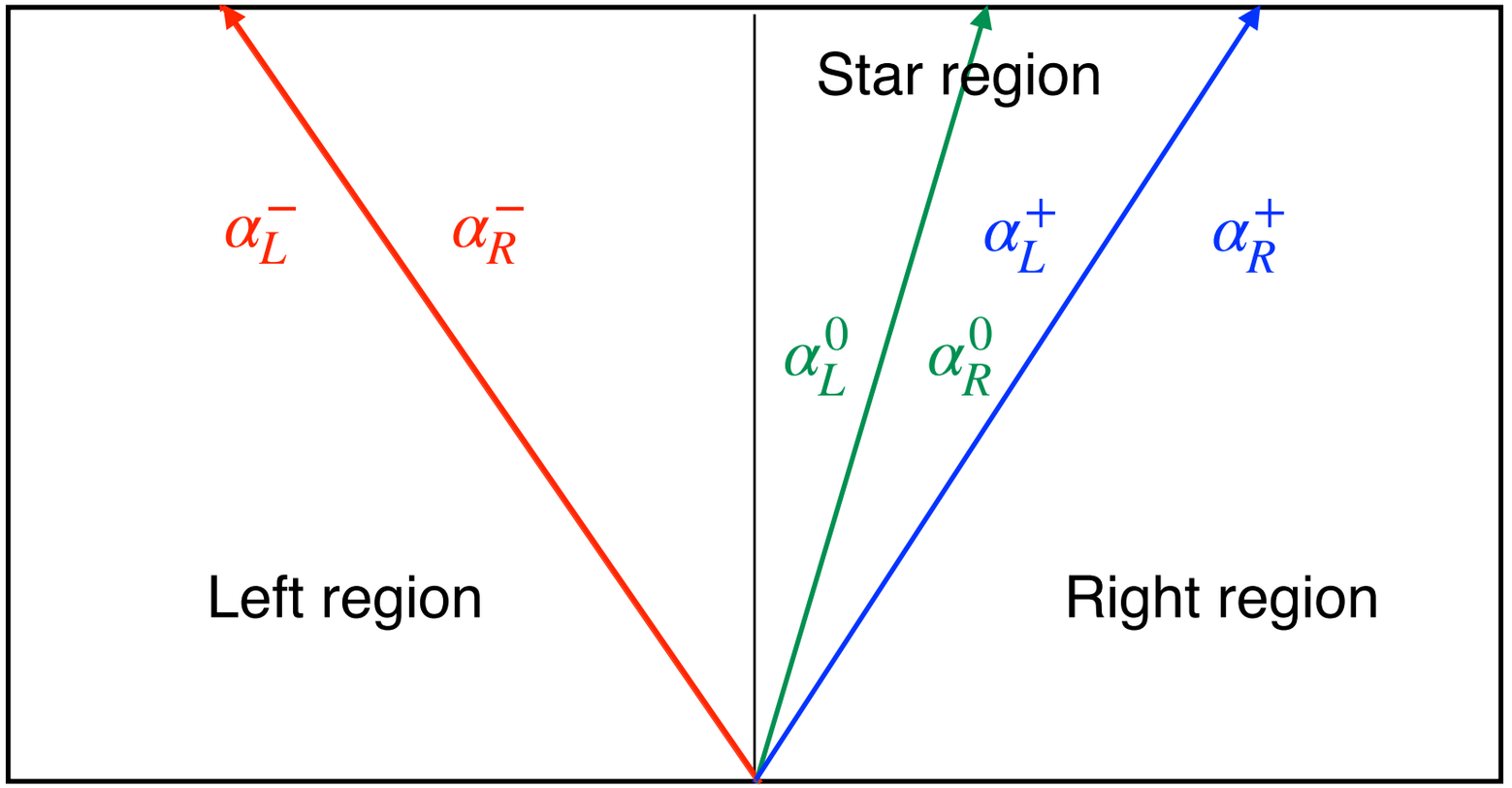
The eigenvector decomposition with $\mathbf{V} = [\alpha^+, \alpha^0, \alpha^-]^T$ leads to the relations:

$$\alpha^+ = \frac{\delta P}{2c_0^2} + \rho_0 \frac{\delta v}{2c_0}, \quad \alpha^0 = \delta\rho - \frac{\delta P}{c_0^2}, \qquad \alpha^- = \frac{\delta P}{2c_0^2} - \rho_0 \frac{\delta v}{2c_0}.$$

$$\delta\rho = \alpha^+ + \alpha^0 + \alpha^-, \quad \delta v = \frac{c_0}{\rho_0}(\alpha^+ - \alpha^-), \quad \delta P = c_0^2(\alpha^+ + \alpha^-).$$

corresponding to $\mathbb{L} = \begin{bmatrix} 0 & \frac{\rho_0}{2c_0} & \frac{1}{2c_0^2} \\ 1 & 0 & -\frac{1}{c_0^2} \\ 0 & -\frac{\rho_0}{2c_0} & \frac{1}{2c_0^2} \end{bmatrix}$ and $\mathbb{R} = \begin{bmatrix} 1 & 1 & 1 \\ \frac{c_0}{\rho_0} & 0 & -\frac{c_0}{\rho_0} \\ \frac{c_0}{\rho_0} & 0 & \frac{c_0}{\rho_0} \end{bmatrix}$.

Riemann Problem for Linearized Euler



The Godunov state can be reconstructed in this particular case as

$$\delta\rho_G = \alpha_L^+ + \alpha_L^0 + \alpha_R^-, \quad \delta v_G = \frac{c_0}{\rho_0}(\alpha_L^+ - \alpha_R^-), \quad \delta P_G = c_0^2(\alpha_L^+ + \alpha_R^-).$$

Riemann Problem for Linearized Euler

The Riemann solution has 4 different regions:

- the left unperturbed state $\mathbf{U}_L = [\delta\rho_L, \delta v_L, \delta P_L]^T$
- the left star state $\mathbf{U}_L^* = [\delta\rho_L^*, \delta v^*, \delta P^*]^T$
  - $\delta\rho_L^* = \alpha_L^+ + \alpha_L^0 + \alpha_R^-$
  - $\delta v^* = \dfrac{c_0}{\rho_0}(\alpha_L^+ - \alpha_R^-)$ and $\delta P^* = c_0^2(\alpha_L^+ + \alpha_R^-)$
- the right star state $\mathbf{U}_R^* = [\delta\rho_R^*, \delta v^*, \delta P^*]^T$
  - $\delta\rho_R^* = \alpha_L^+ + \alpha_R^0 + \alpha_R^-$
  - $\delta v^* = \dfrac{c_0}{\rho_0}(\alpha_L^+ - \alpha_R^-)$ and $\delta P^* = c_0^2(\alpha_L^+ + \alpha_R^-)$
- the right unperturbed stare $\mathbf{U}_R = [\delta\rho_R, \delta v_R, \delta P_R]^T$

The star region has uniform velocity and uniform pressure.

Star State for Linearized Euler

We can look more closely to the star pressure and star velocity.

We have $\delta v^* = \frac{c_0}{\rho_0}(\alpha_L^+ - \alpha_R^-)$ and $\delta P^* = c_0^2(\alpha_L^+ + \alpha_R^-)$,

with $\alpha_L^+ = \frac{\delta P_L}{2c_0^2} + \rho_0 \frac{\delta v_L}{2c_0}$ and $\alpha_R^- = \frac{\delta P_R}{2c_0^2} - \rho_0 \frac{\delta v_R}{2c_0}$.

We find $$\delta v^* = \frac{\delta v_L + \delta v_R}{2} + \frac{1}{\rho_0 c_0}\frac{\delta P_L - \delta P_R}{2}$$

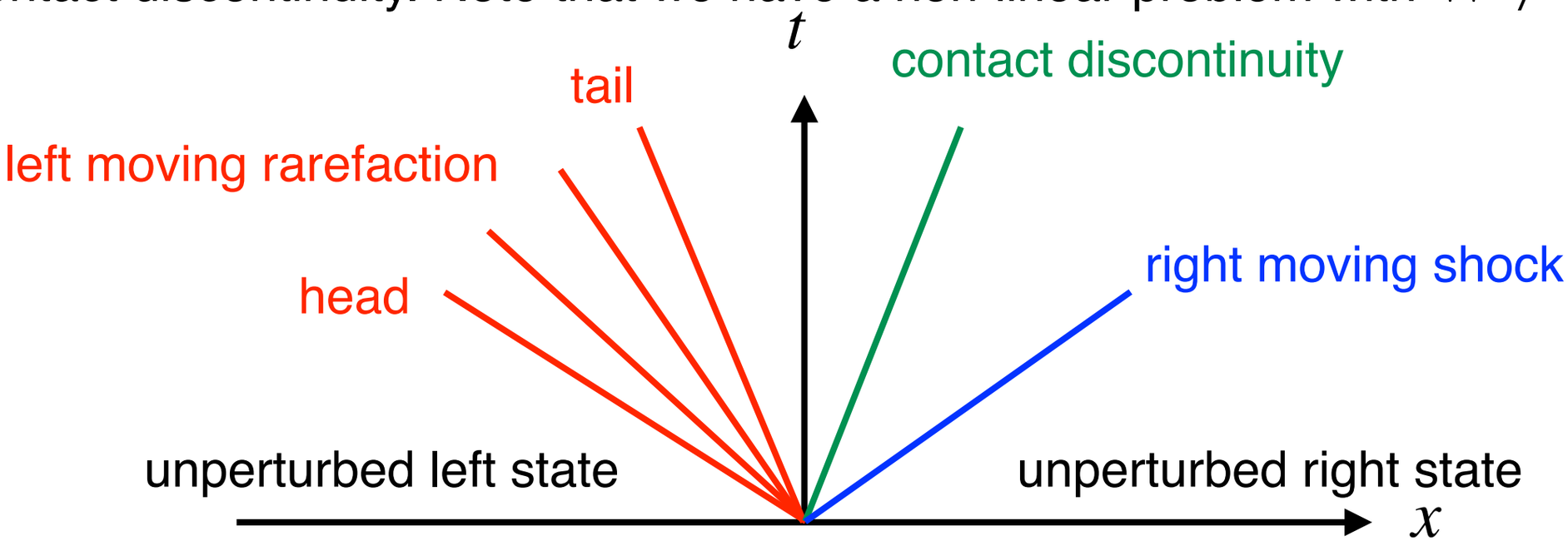
and $$\delta P^* = \frac{\delta P_L + \delta P_R}{2} + \rho_0 c_0 \frac{\delta v_L - \delta v_R}{2}.$$

Adding back the reference state v_0 and P_0, we find the acoustic star state:

$$v^* = \frac{v_L + v_R}{2} + \frac{1}{\rho_0 c_0}\frac{P_L - P_R}{2} \text{ and } P^* = \frac{P_L + P_R}{2} + \rho_0 c_0 \frac{v_L - v_R}{2}.$$

Riemann Problem for the Euler Equations

The Riemann problem for the Euler equations is defined by constant left and right states $\mathbf{W}_L = [\rho_L, v_L, P_L]^T$ and $\mathbf{W}_R = [\rho_R, v_R, P_R]^T$ separated initially by a contact discontinuity. Note that we have a non-linear problem with $\mathbf{W} \neq \mathbf{U}$.



For $t > 0$, the **exact Riemann solution** features 3 waves and 4 distinct regions:

- a shock or a rarefaction wave moving into the left unperturbed region
- a contact discontinuity separating the left and the right star regions
- a shock or a rarefaction wave moving into the right unperturbed region

Godunov Method for the Euler Equations

We now describe Godunov method for the Euler equations.

We know at time t^n the conservative variables $\overline{\mathbf{U}}_i^n = [\overline{\rho}_i^n, \overline{(\rho v)}_i^n, \overline{E}_i^n]$.

We compute the primitive variables $\overline{\mathbf{W}}_i^n = [\overline{\rho}_i^n, \overline{v}_i^n, \overline{P}_i^n]$ using:

$$\overline{v}_i^n = \overline{(\rho v)}_i^n / \overline{\rho}_i^n \text{ and } \overline{P}_i^n = (\gamma - 1)\left(\overline{E}_i^n - \frac{1}{2}\overline{\rho}_i^n (\overline{v}_i^n)^2 \right).$$

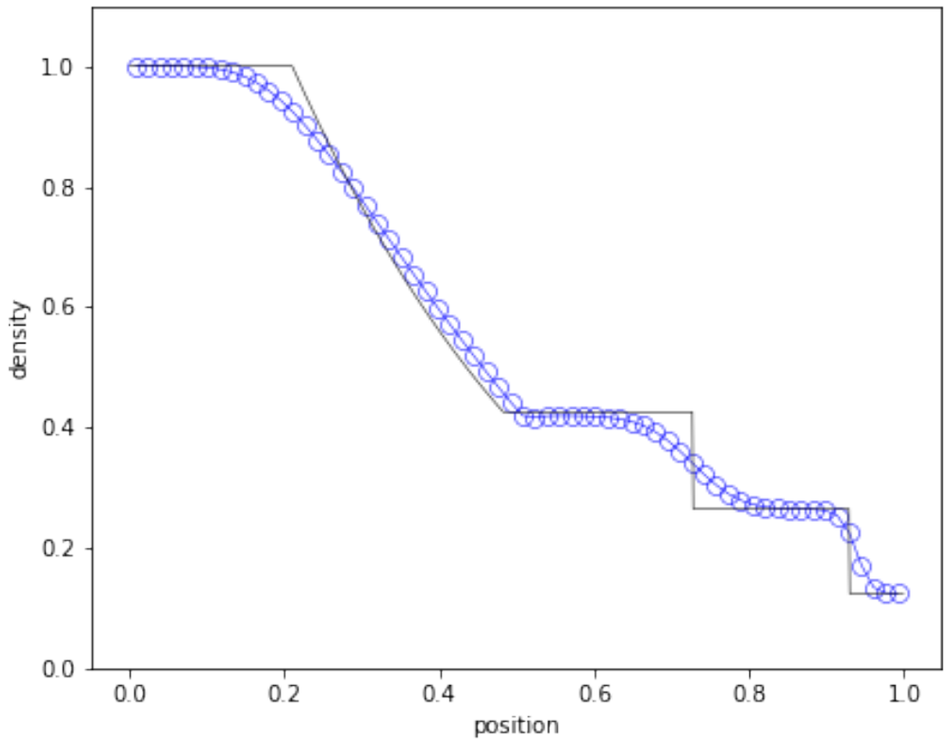
We compute the sound speed $\overline{c}_i^n = \sqrt{\dfrac{\gamma \overline{P}_i^n}{\overline{\rho}_i^n}}$.

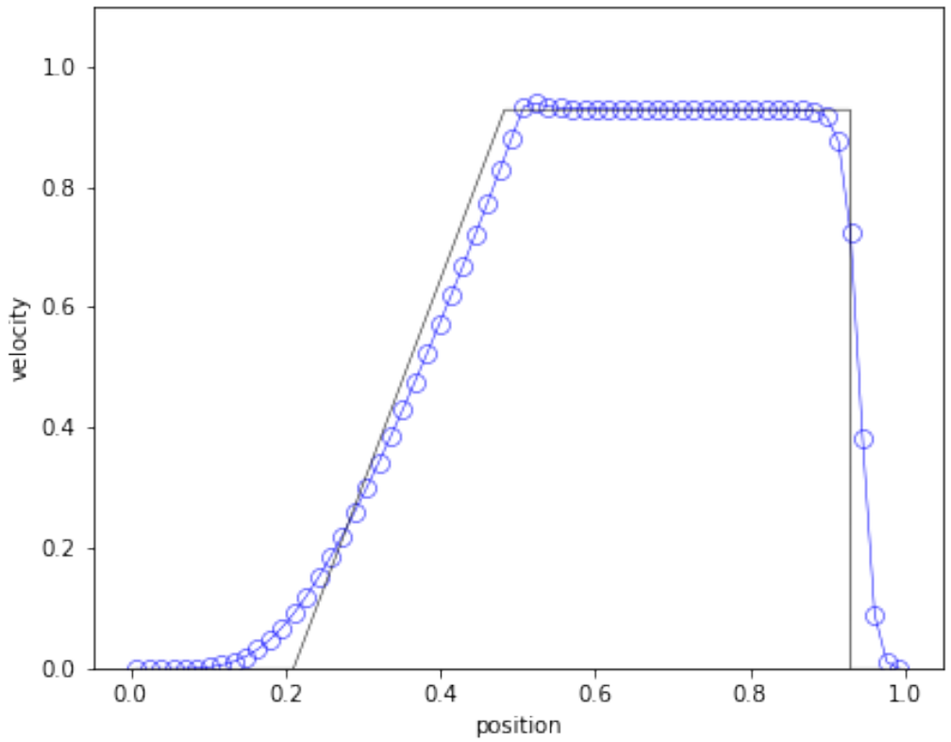
We compute the new time step as $\Delta t = C \dfrac{h}{\max(|\overline{v}_i^n| + \overline{c}_i^n)}$ (use $C = 0.8$).

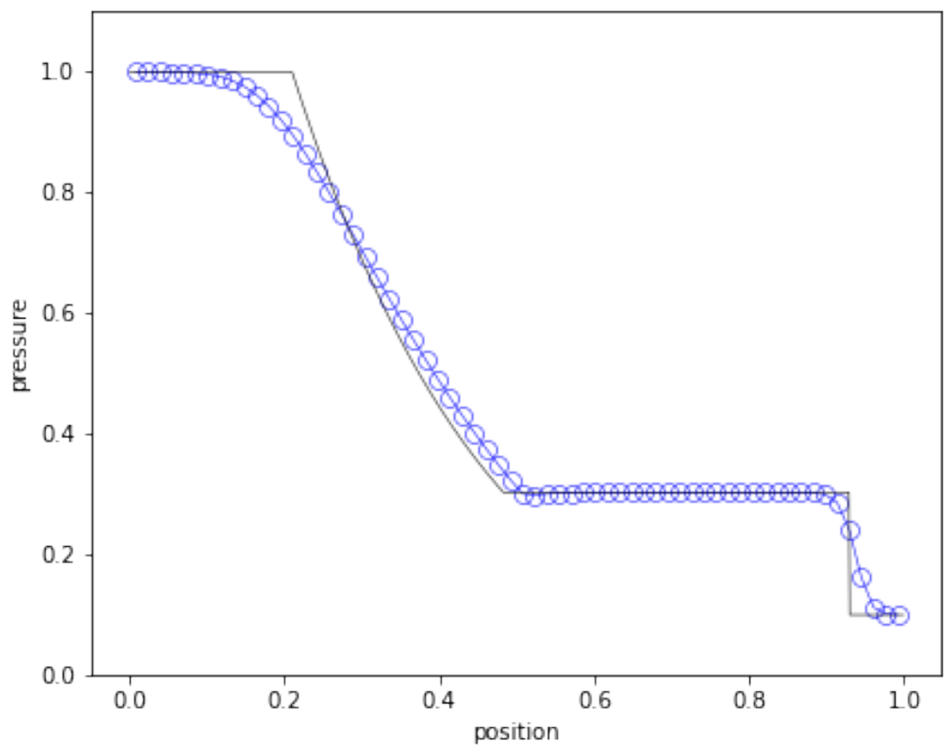
We compute the Godunov flux $\mathbf{F}_{i+1/2}^{n+1/2} = RP(\overline{\mathbf{W}}_i^n, \overline{\mathbf{W}}_{i+1}^n)$ using an exact Riemann solver (not shown here).

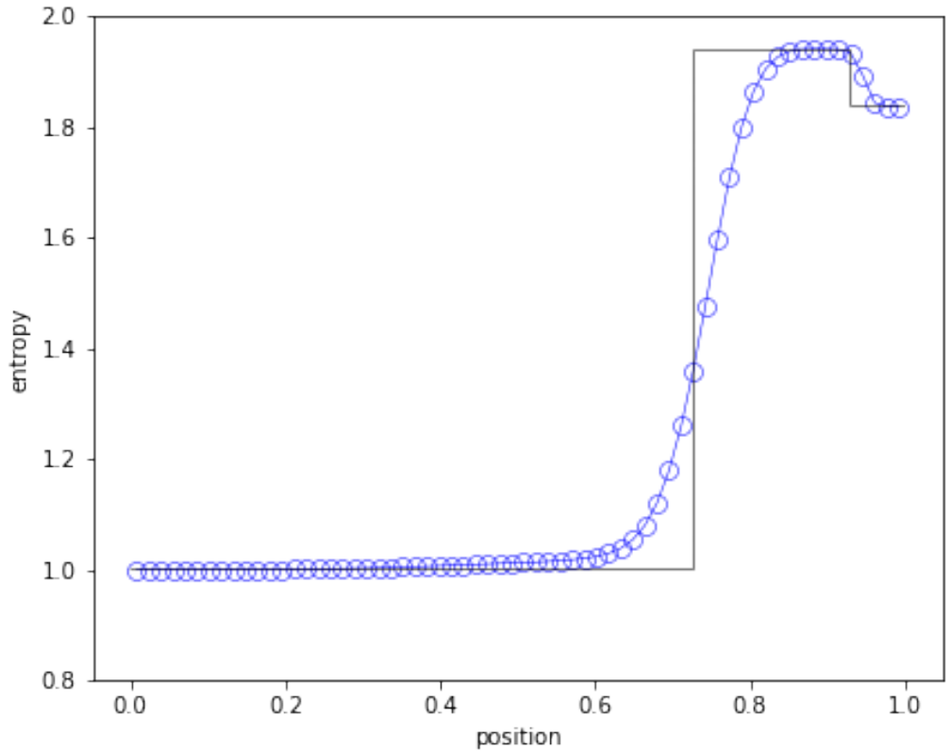
We compute the new solution $\overline{\mathbf{U}}_i^{n+1} = \overline{\mathbf{U}}_i^n - \dfrac{\Delta t}{h}\left(\mathbf{F}_{i+1/2}^{n+1/2} - \mathbf{F}_{i-1/2}^{n+1/2} \right)$

Sod Shock Tube Test









Approximate Riemann Solvers

The previous solution is quite diffusive (first order scheme) but captures all the non-linear features very well (shock waves are particularly good compared to other methods such as SPH or FD).

Most of the time is spent in the exact Riemann solver. Why solving exactly the flux if the solution is anyway only first order accurate?

We will see now several techniques to design approximate Riemann solvers:

- Linear Riemann solvers
- Harten, Lax and van Leer (HLL) Riemann solver
- Local Lax Friedrich (LLF) and Global Lax Friedrich (GLF) Riemann solvers
- HLL + Contact (HLLC) Riemann solver

Linear Riemann Solvers

By analogy to the linear case, we write the Godunov flux function as

$$\mathbf{F}_G = \frac{1}{2}(\mathbf{F}_L + \mathbf{F}_R) - \frac{1}{2}\mathbb{R}\,|\mathbb{D}|\,\mathbb{L}(\mathbf{U}_R - \mathbf{U}_L),$$

where $\mathbb{R}$ ($\mathbb{L}$) is the right (left) eigenvectors matrix and $\mathbb{D}$ is the diagonal eigenvalues matrix of the Jacobian of the flux function $\mathbb{A} = \dfrac{\partial \mathbf{F}}{\partial \mathbf{U}}$ for some reference state $\mathbf{U}_0$.

The main question is how do we evaluate $\mathbf{U}_0$?

A simple option for smooth flow is the arithmetic average: $\mathbf{U}_0 = \frac{1}{2}(\mathbf{U}_L + \mathbf{U}_R)$.

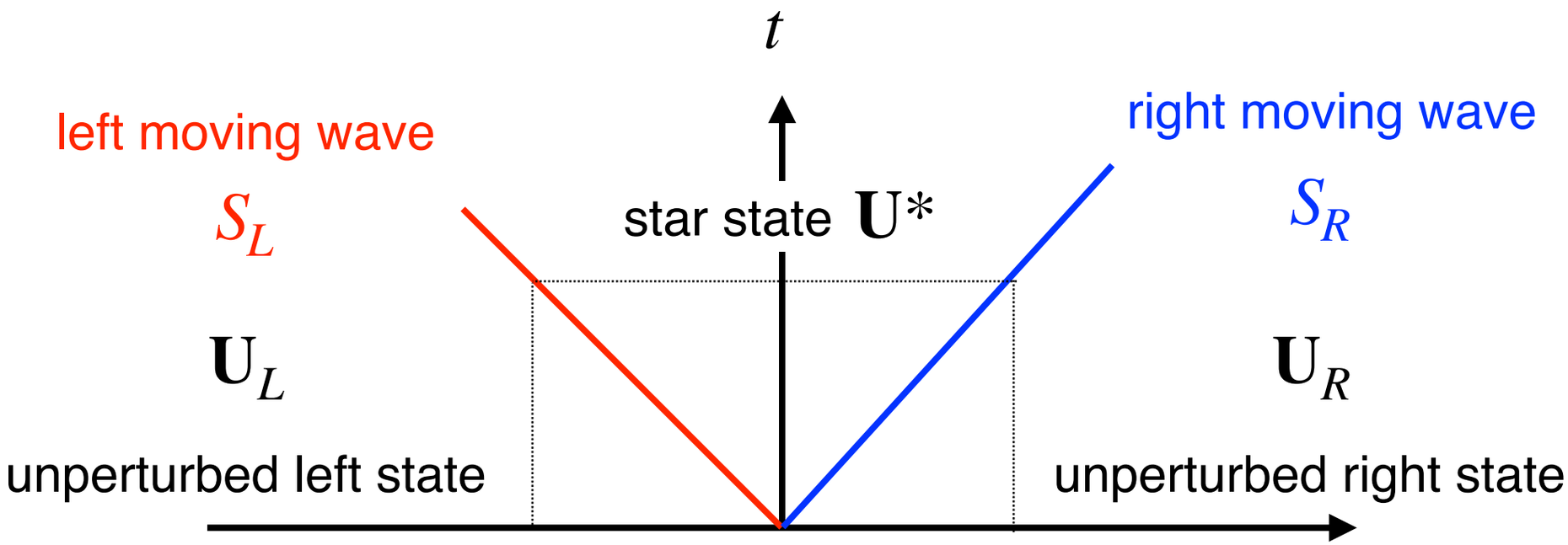
A better option when shocks are present is to use the Roe average defined by:

$$\rho_0 = \sqrt{\rho_L \rho_R} \text{ and } v_0 = \frac{\sqrt{\rho_L} v_L + \sqrt{\rho_R} v_R}{\sqrt{\rho_L} + \sqrt{\rho_R}} \text{ and } h_0 = \frac{\sqrt{\rho_L} h_L + \sqrt{\rho_R} h_R}{\sqrt{\rho_L} + \sqrt{\rho_R}}.$$

In this case, magic happens ! We have $\mathbf{F}(\mathbf{U}_R) - \mathbf{F}(\mathbf{U}_L) = \mathbb{A}(\mathbf{U}_0)(\mathbf{U}_R - \mathbf{U}_L)$ exactly.

Harten, Lax and van Leer (HLL)

This approximate Riemann solver assumes that the Riemann solution is piecewise constant with only 3 regions: the left unperturbed state, the right unperturbed state and a fully uniform star state.



Harten, Lax and van Leer (HLL)

Integrating the conservation law in the full rectangle above:

$$\int_0^t \int_{S_L t}^{S_R t} \left(\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} \right) \mathrm{d}t\mathrm{d}x = 0 \text{ leads to } \mathbf{U}^*(S_R - S_L) - \mathbf{U}_R S_R + \mathbf{U}_L S_L + \mathbf{F}_R - \mathbf{F}_L = 0.$$

We find the star state $\mathbf{U}^* = \dfrac{\mathbf{U}_R S_R - \mathbf{U}_L S_L - \mathbf{F}_R + \mathbf{F}_L}{S_R - S_L}$.

We then integrate over the left (or right) half of the rectangle:

$$\int_0^t \int_{S_L t}^{0} \left(\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} \right) \mathrm{d}t\mathrm{d}x = 0 \text{ leads to the relation } \mathbf{F}^* - \mathbf{F}_L = S_L(\mathbf{U}^* - \mathbf{U}_L)$$ (aka the Rankine-Hugoniot relation).

We find the star flux $\mathbf{F}^* = \dfrac{\mathbf{F}_L S_R - \mathbf{F}_R S_L}{S_R - S_L} + \dfrac{S_R S_L}{S_R - S_L}(\mathbf{U}_R - \mathbf{U}_L)$.

Harten, Lax and van Leer (HLL)

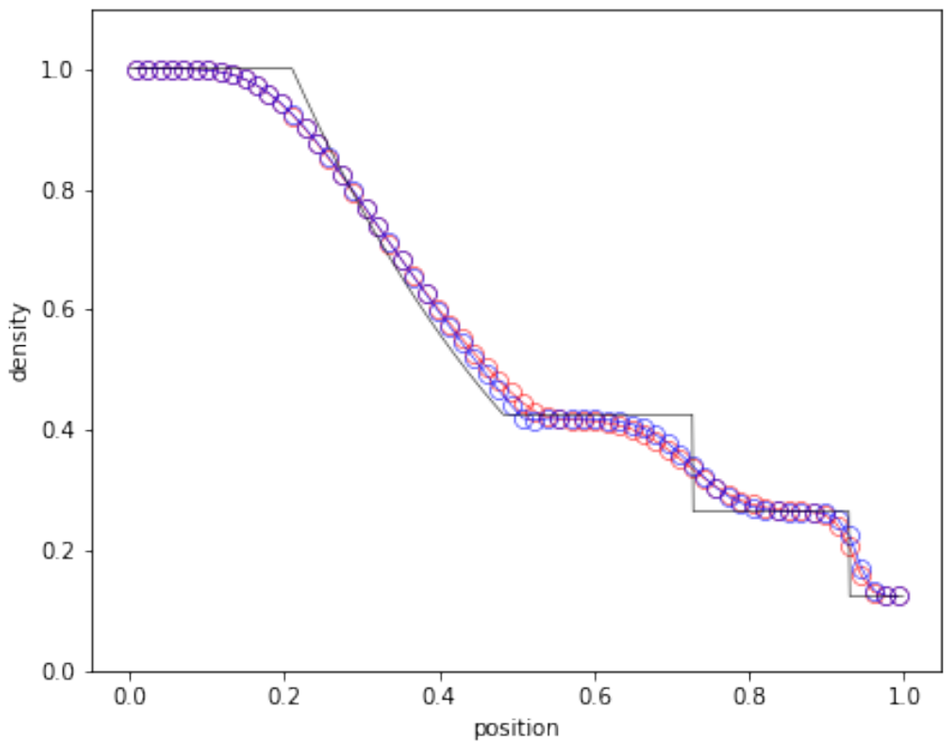
The wave speeds are computed as:

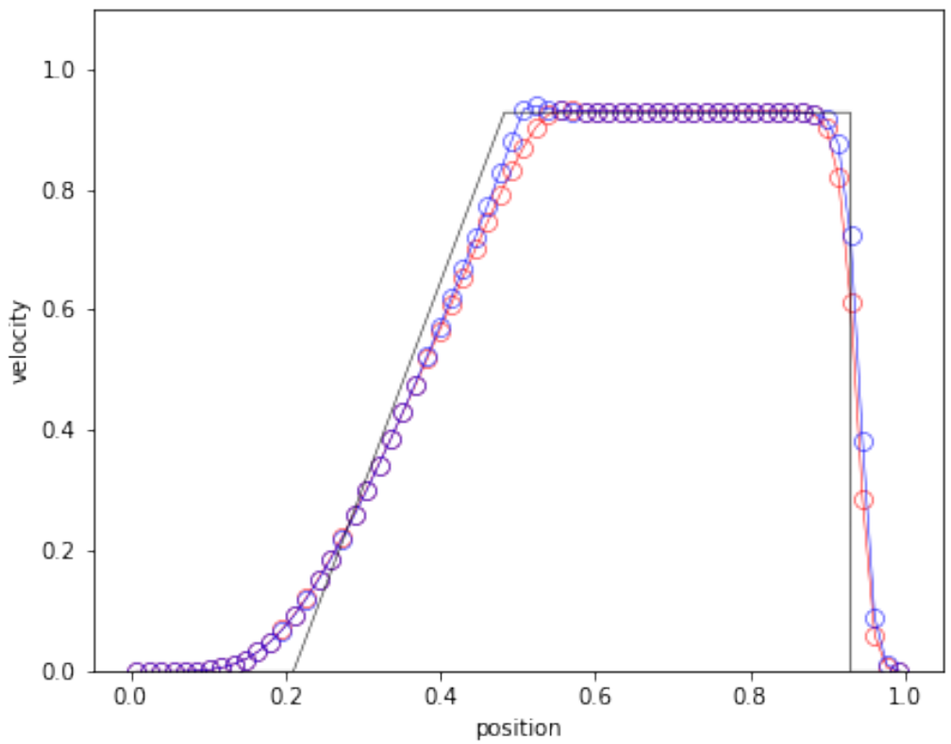
$$S_L = \min(v_L, v_R) - \max(c_L, c_R) \text{ and } S_R = \max(v_L, v_R) + \max(c_L, c_R).$$

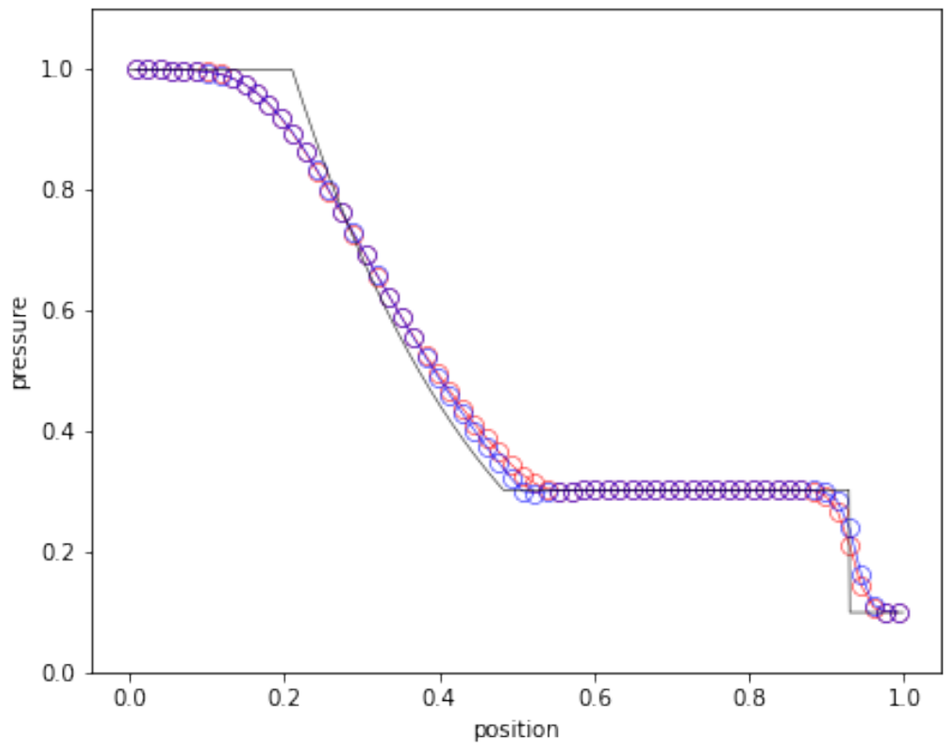
We have to consider the following 3 cases:

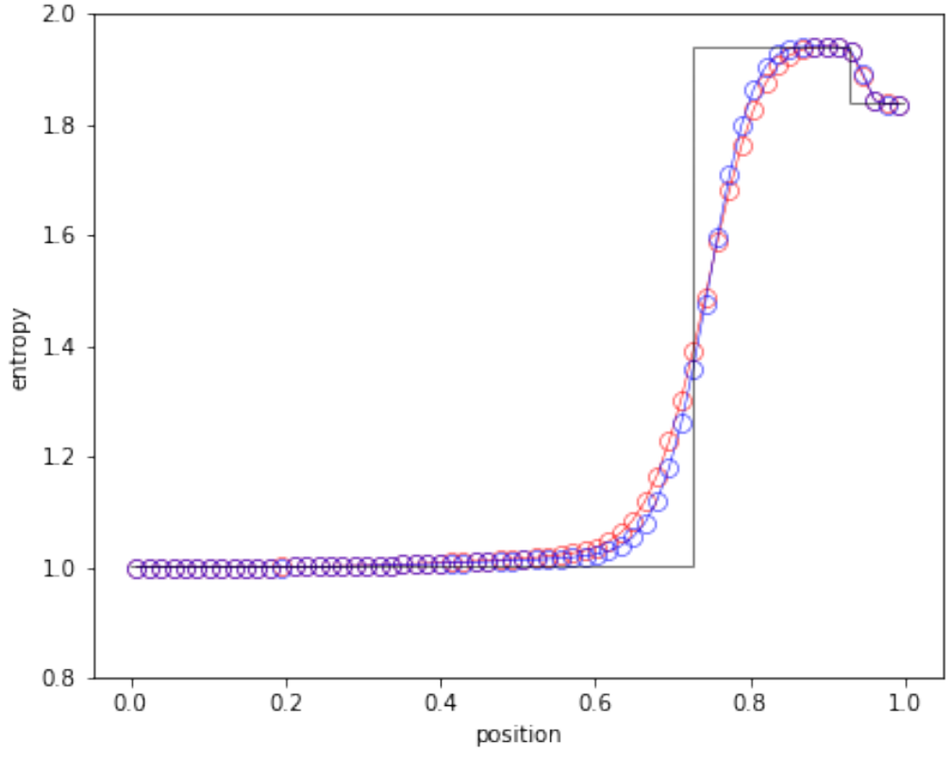
- if $S_L > 0$ and $S_R > 0$ then $\mathbf{F}_G = \mathbf{F}_L$ (supersonic to the right).
- if $S_L < 0$ and $S_R < 0$ then $\mathbf{F}_G = \mathbf{F}_R$ (supersonic to the left).
- if $S_L < 0$ and $S_R > 0$ then $\mathbf{F}_G = \dfrac{\mathbf{F}_L S_R - \mathbf{F}_R S_L}{S_R - S_L} + \dfrac{S_R S_L}{S_R - S_L}(\mathbf{U}_R - \mathbf{U}_L)$ (subsonic).

HLL versus Exact









Local or Global Lax Friedrich

A very simple variant of the HLL Riemann solver consists in taking for the wave speeds

$$S_L = -S_{\max} \text{ and } S_R = S_{\max} \text{ where } S_{\max} = \max(|v_L|, |v_R|) + \max(c_L, c_R).$$

Note that we are always in the subsonic case here and the Godunov flux writes

$$\text{LLF Riemann solver: } \mathbf{F}_G = \frac{\mathbf{F}_L + \mathbf{F}_R}{2} - \frac{S_{\max}}{2}(\mathbf{U}_R - \mathbf{U}_L).$$

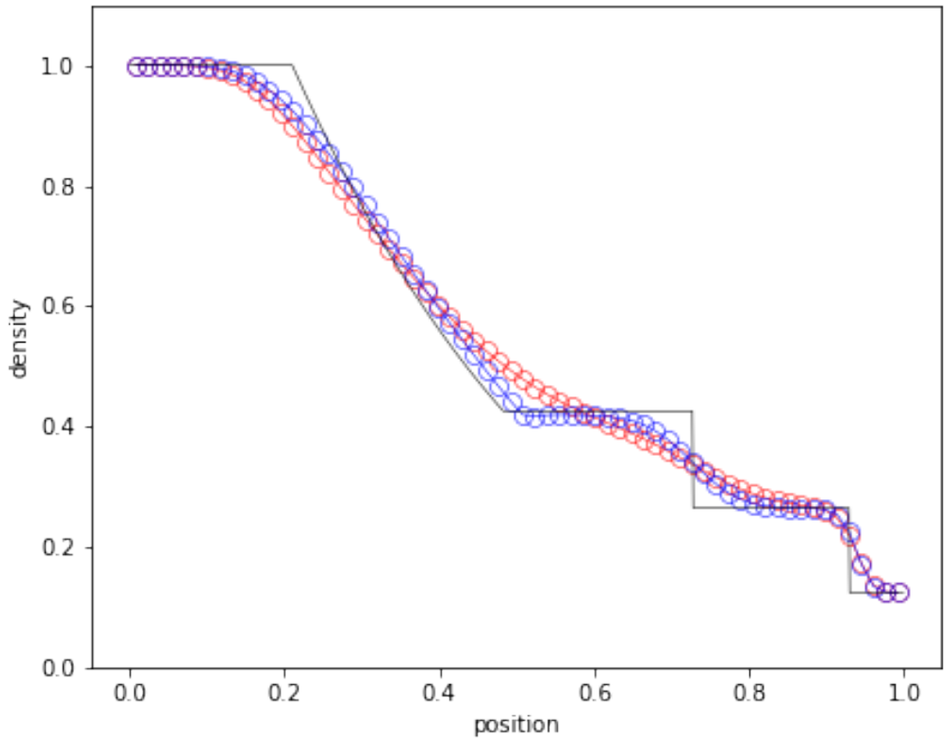
We can easily interpret this flux function as the unstable central scheme combined with a diffusive term that stabilizes the scheme. Note that here the maximum wave speed $S_{\max}$ is different for each interface.

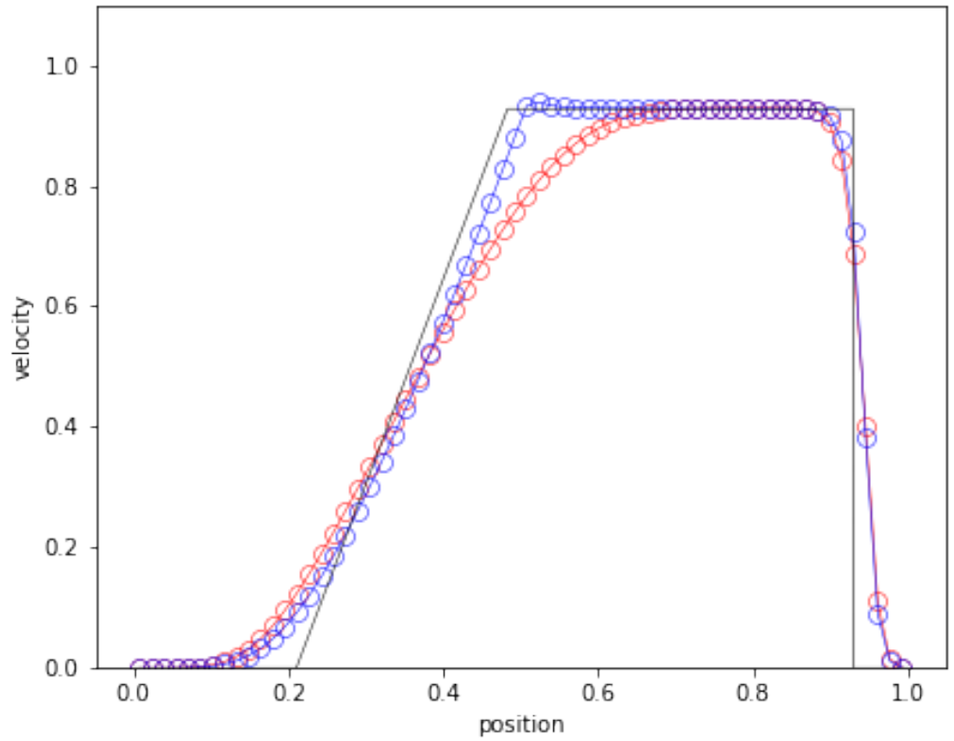
We introduce an even simpler variant where the max is now global over all cells.

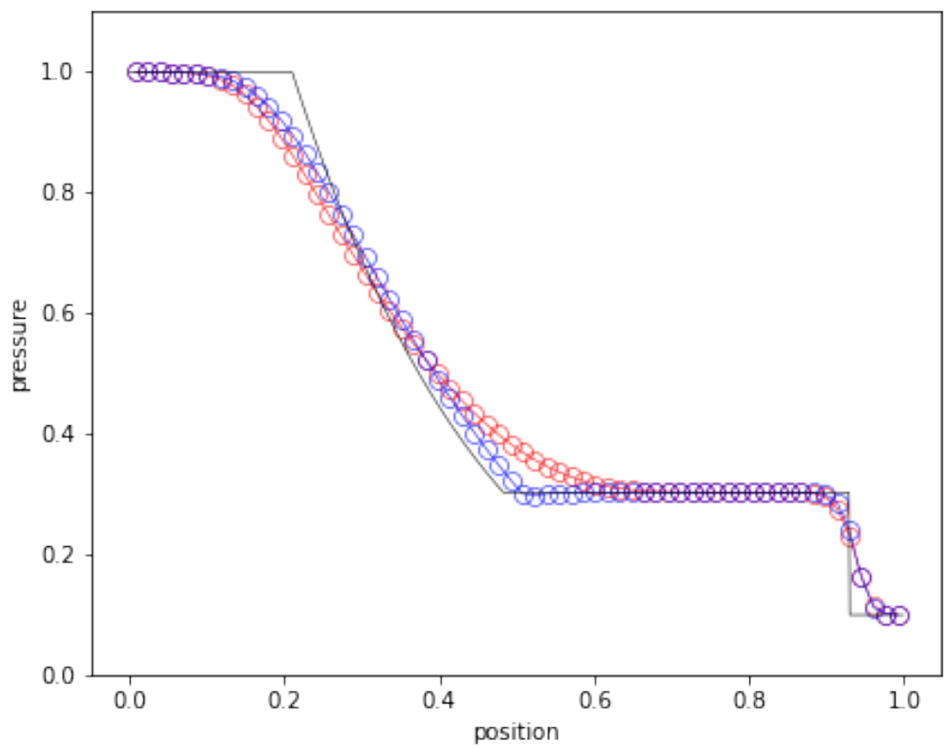
Because the time step is given by $\Delta t = C\dfrac{h}{S_{\max}}$, the global Lax Friedrich scheme writes

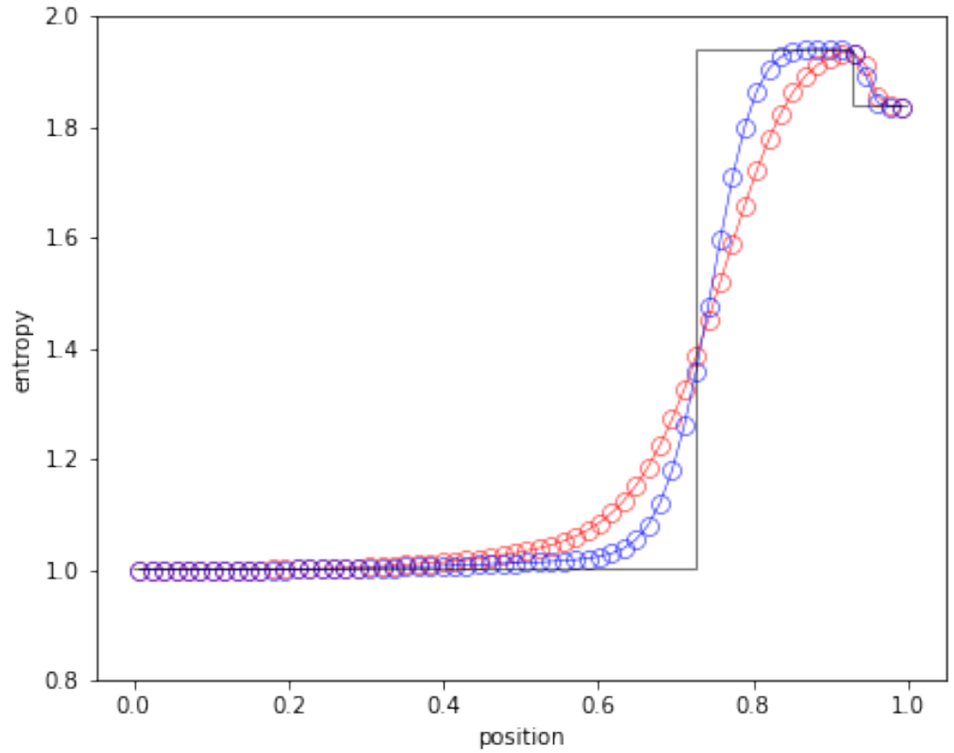
$$\text{GLF Riemann solver: } \mathbf{F}_G = \frac{\mathbf{F}_L + \mathbf{F}_R}{2} - \frac{Ch}{2\Delta t}(\mathbf{U}_R - \mathbf{U}_L).$$

Local Lax Friedrich versus Exact



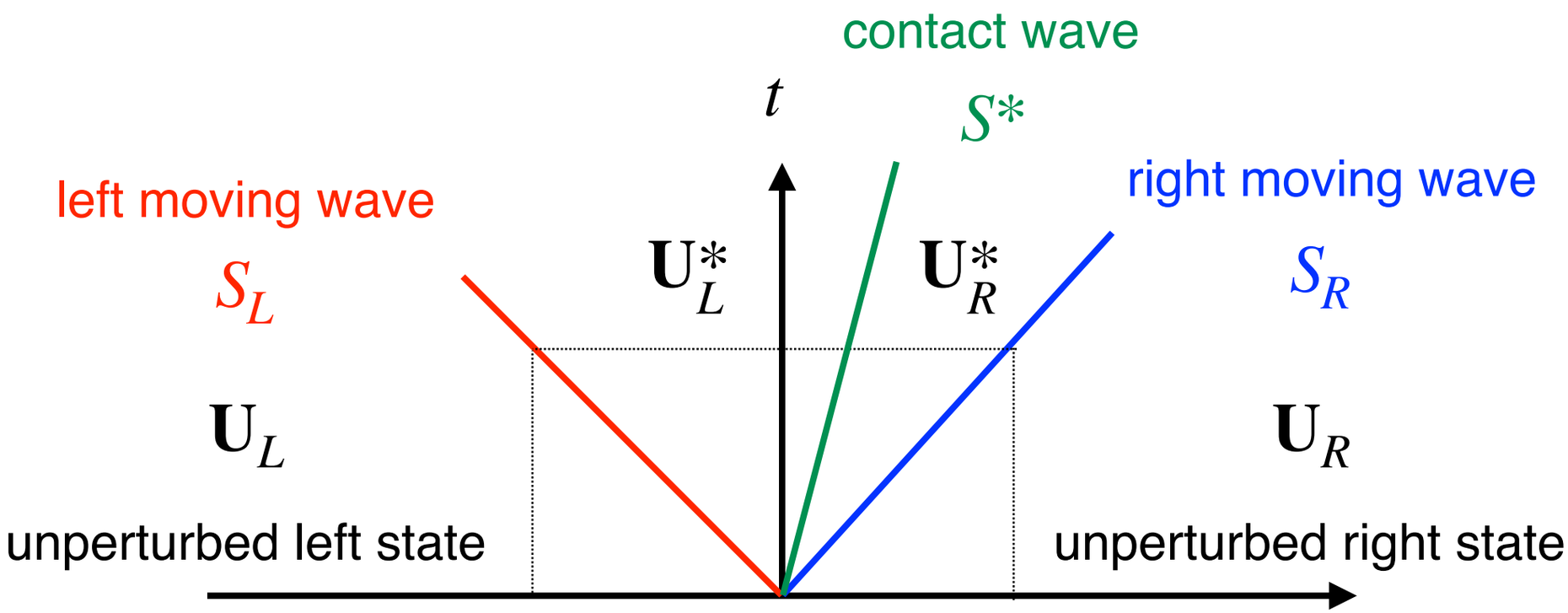






The HLLC Riemann Solver

The HLLC Riemann solver introduces the contact discontinuity as a third wave within the HLL methodology. This corresponds now to 4 regions.



The HLLC Riemann Solver

We use the properties of the exact Riemann solution for the contact wave:

$$v_L^* = v_R^* = S^* \text{ and } P_L^* = P_R^* = P^*.$$

Conservation relations for the left wave give: $\mathbf{F}_L^* - \mathbf{F}_L = S_L(\mathbf{U}_L^* - \mathbf{U}_L)$.

$\rho_L^* S^* - \rho_L v_L = S_L(\rho_L^* - \rho_L)$ or $\rho_L^* = \rho_L \dfrac{v_L - S_L}{S^* - S_L}$.

$\rho_L^*(S^*)^2 + P^* - \rho_L(v_L)^2 - P_L = S_L(\rho_L^* S^* - \rho_L v_L)$ or $P^* = P_L + \rho_L(S_L - v_L)(S^* - v_L)$.

Conservation relations for the right wave give: $\mathbf{F}_R^* - \mathbf{F}_R = S_R(\mathbf{U}_R^* - \mathbf{U}_R)$.

$\rho_R^* = \rho_R \dfrac{v_R - S_R}{S^* - S_R}$ and $P^* = P_R + \rho_R(S_R - v_R)(S^* - v_R)$.

Combining the two gives $S^* = \dfrac{P_R - P_L + \rho_L v_L(S_L - v_L) - \rho_R v_R(S_R - v_R)}{\rho_L(S_L - v_L) - \rho_R(S_R - v_R)}$.

The last two RH equations give us the total energies E_L^* and E_R^*.

The HLLC Riemann Solver

We have to consider here the following 4 cases:

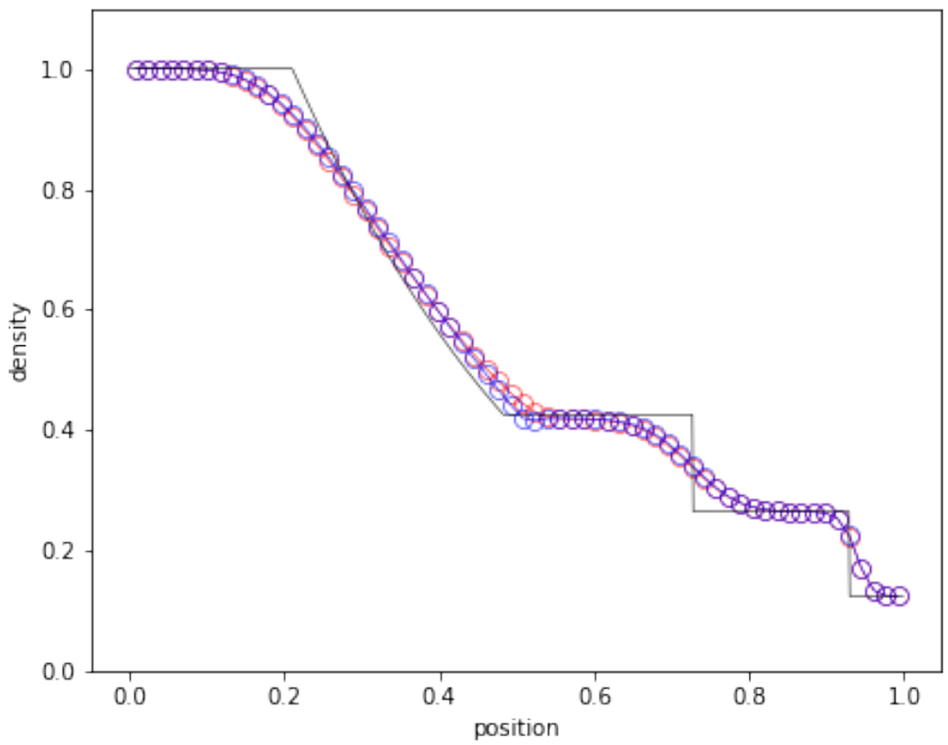
- if $S_L > 0$ and $S_R > 0$ then $\mathbf{F}_G = \mathbf{F}_L$ (supersonic to the right).
- if $S_L < 0$ and $S_R < 0$ then $\mathbf{F}_G = \mathbf{F}_R$ (supersonic to the left).
- if $S_L < 0$ and $S^* > 0$ then (subsonic to the left).

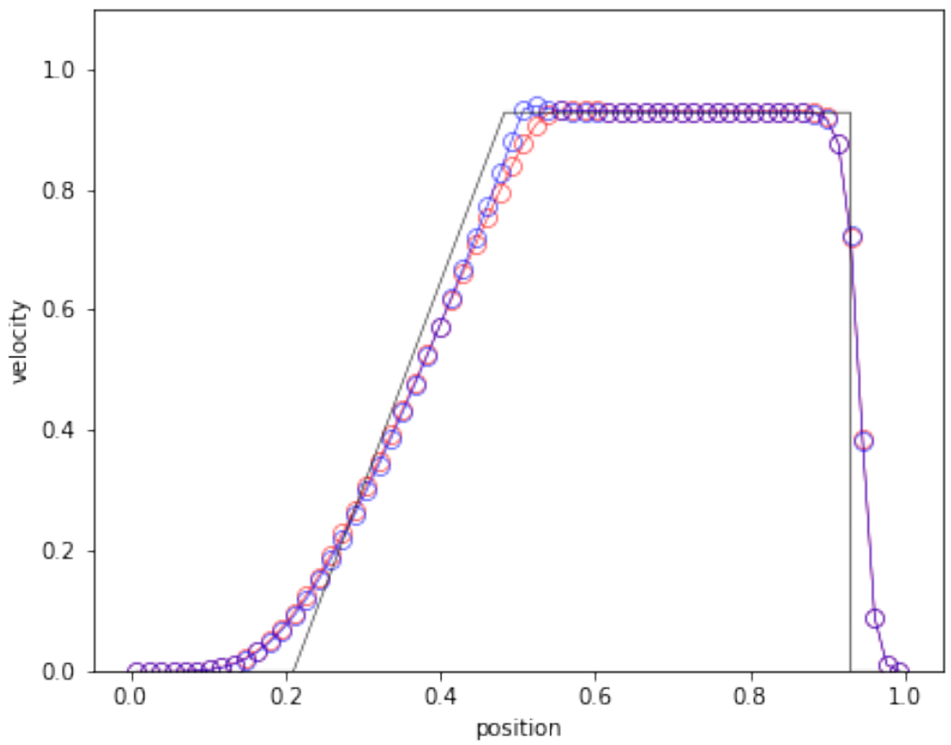
$$\rho_G = \rho_L^*, v_G = S^*, P_G = P^*, E_G = E_L^*$$

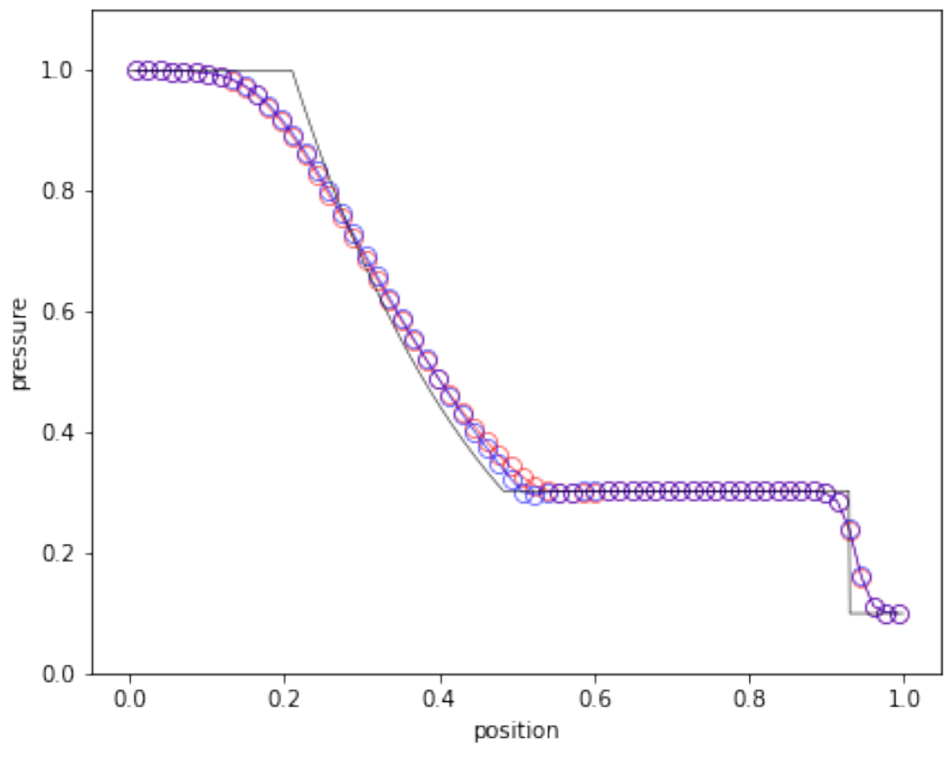
- if $S^* < 0$ and $S_R > 0$ then (subsonic to the right).

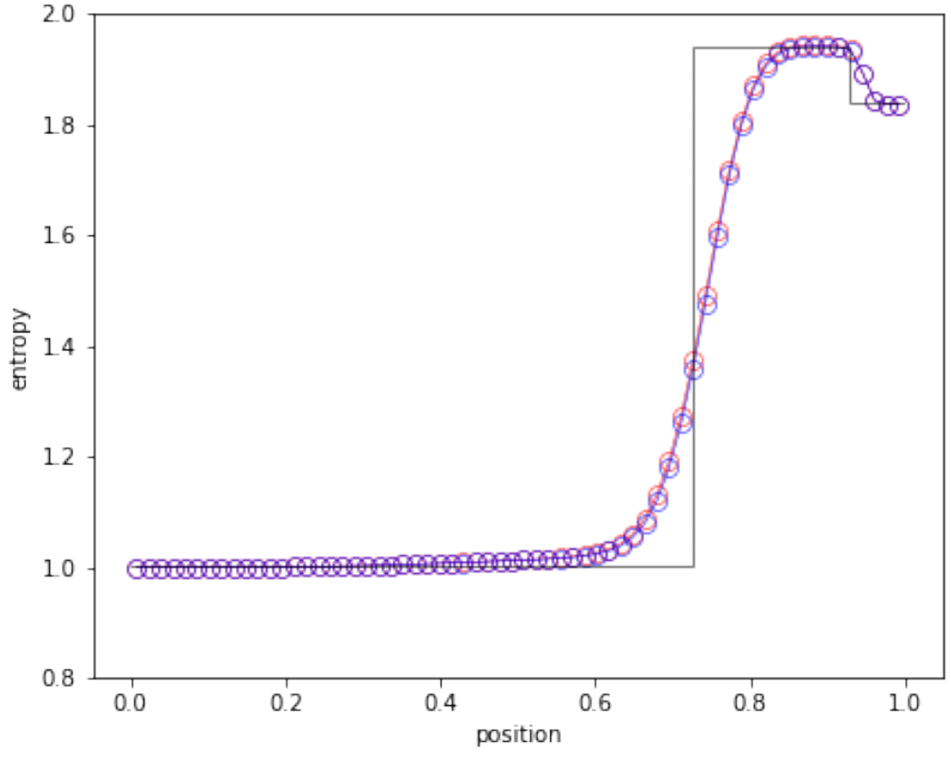
$$\rho_G = \rho_R^*, v_G = S^*, P_G = P^*, E_G = E_R^*$$

HLLC versus Exact

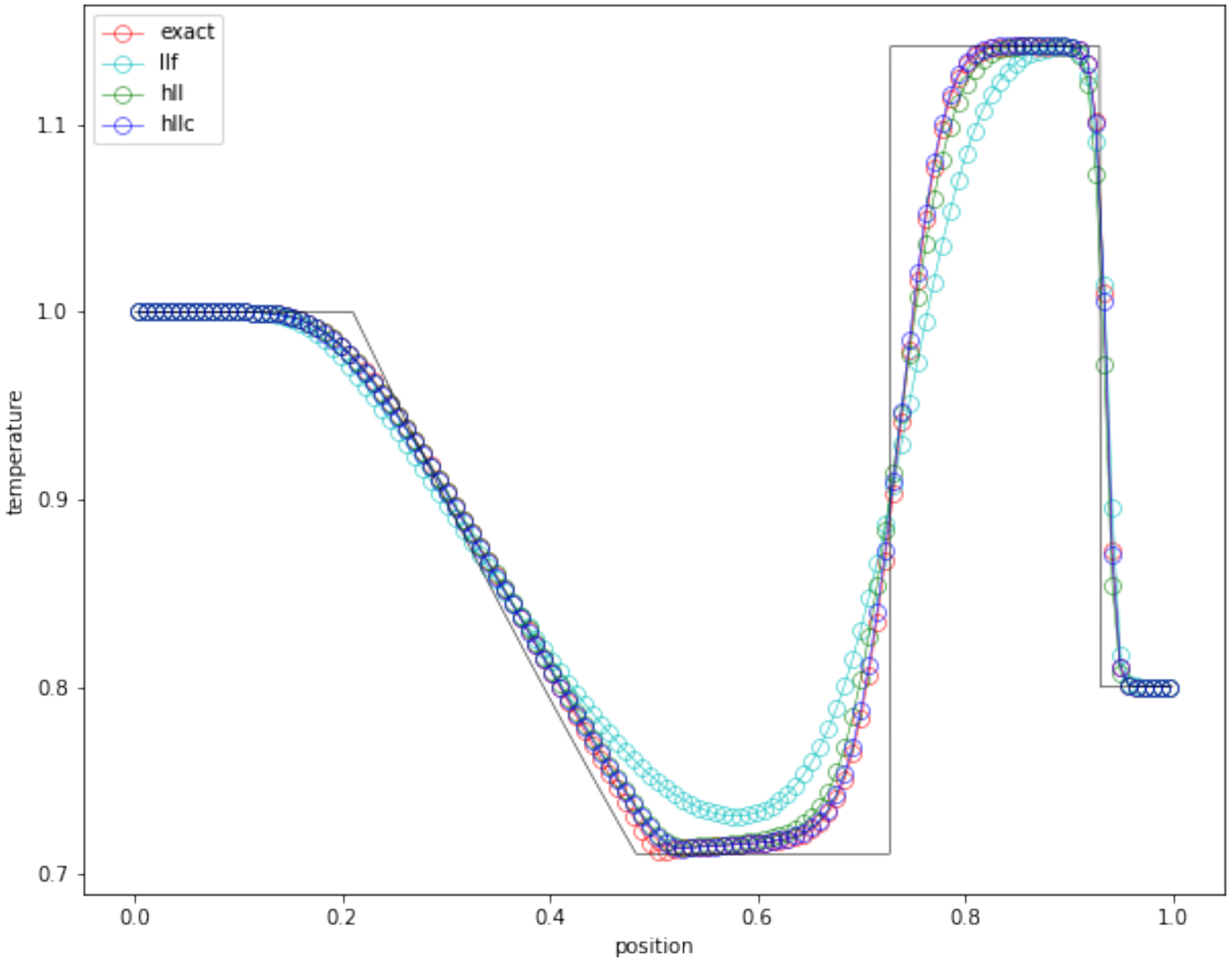








Different Riemann Solvers



Conclusions for the Euler Equations

Godunov method provides a successful framework for solving nonlinear systems of conservation laws.

We explored in detail the numerical methods for the Euler equations, but other nonlinear hyperbolic systems can be solved with exactly the same methodology:

- ideal MHD
- relativistic HD
- radiative transfer with M1 closure
- shallow water equations
- traffic flow equations

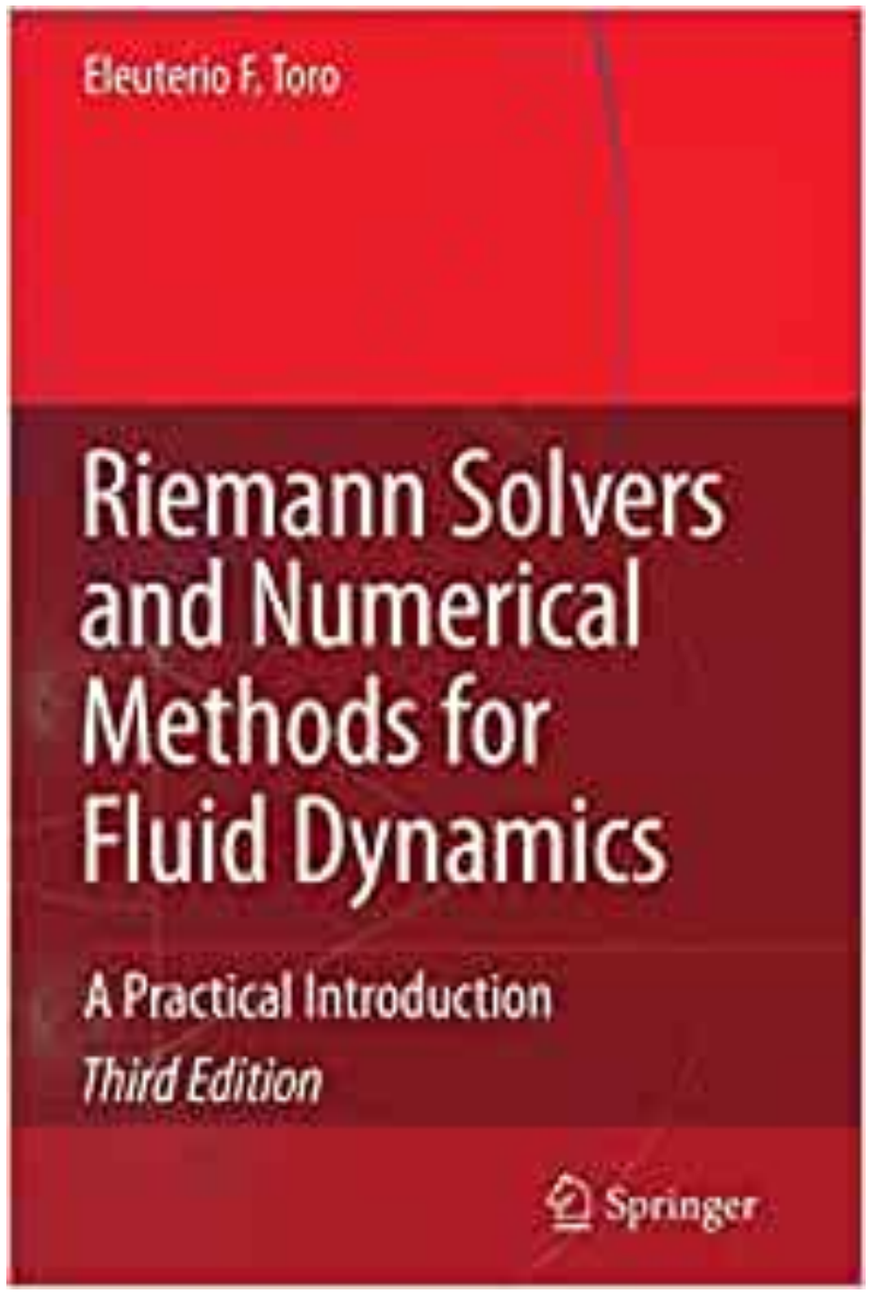
Approximate Riemann solvers can replace advantageously expensive exact Riemann solvers. Popular ones are HLLC (Euler), HLLD (ideal MHD) and variants.

Key properties of the Godunov methods are:

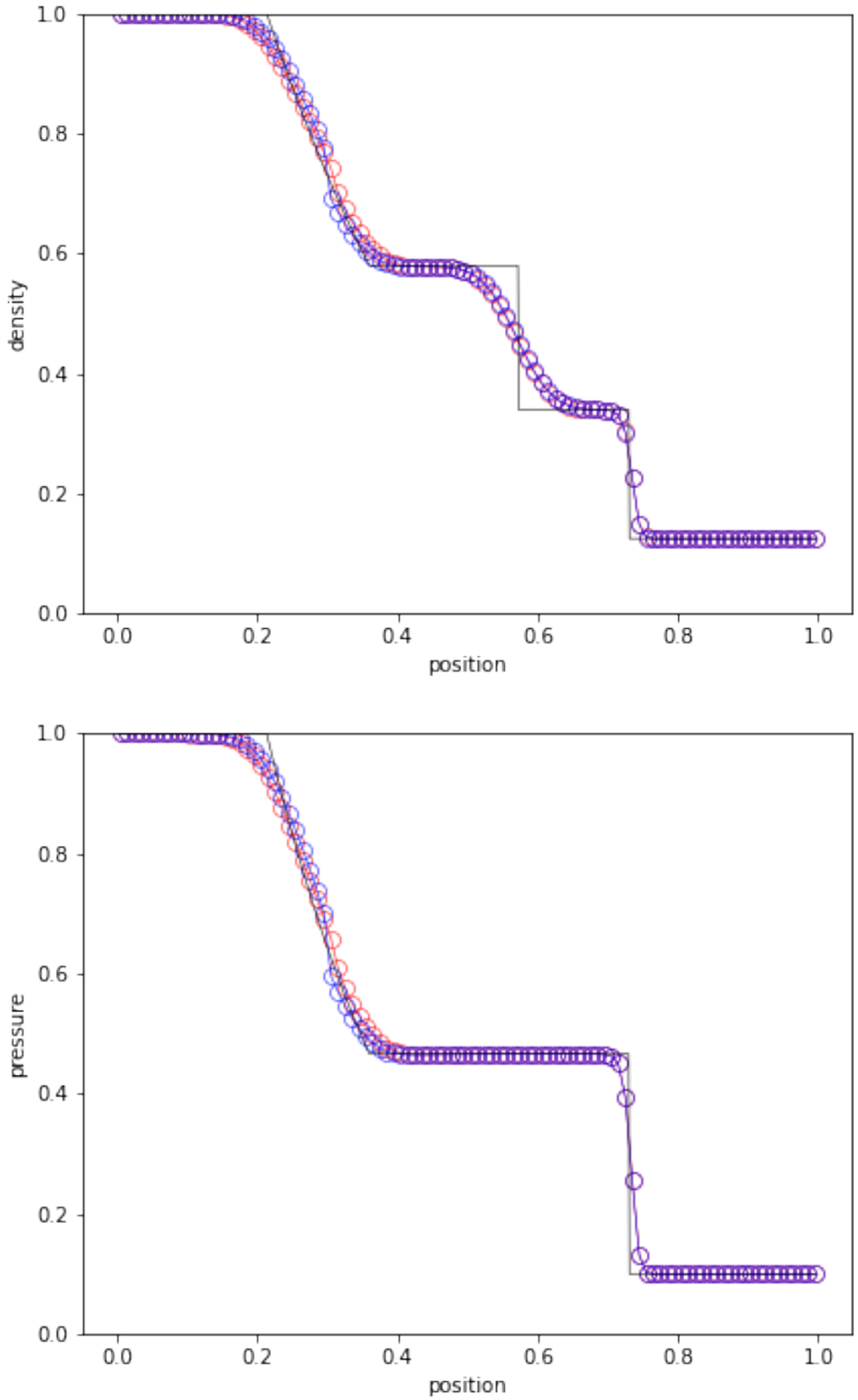
- conservative scheme allowing very good shock capturing
- explicit time integration with Courant stability condition
- easy generalization to multiD and higher order

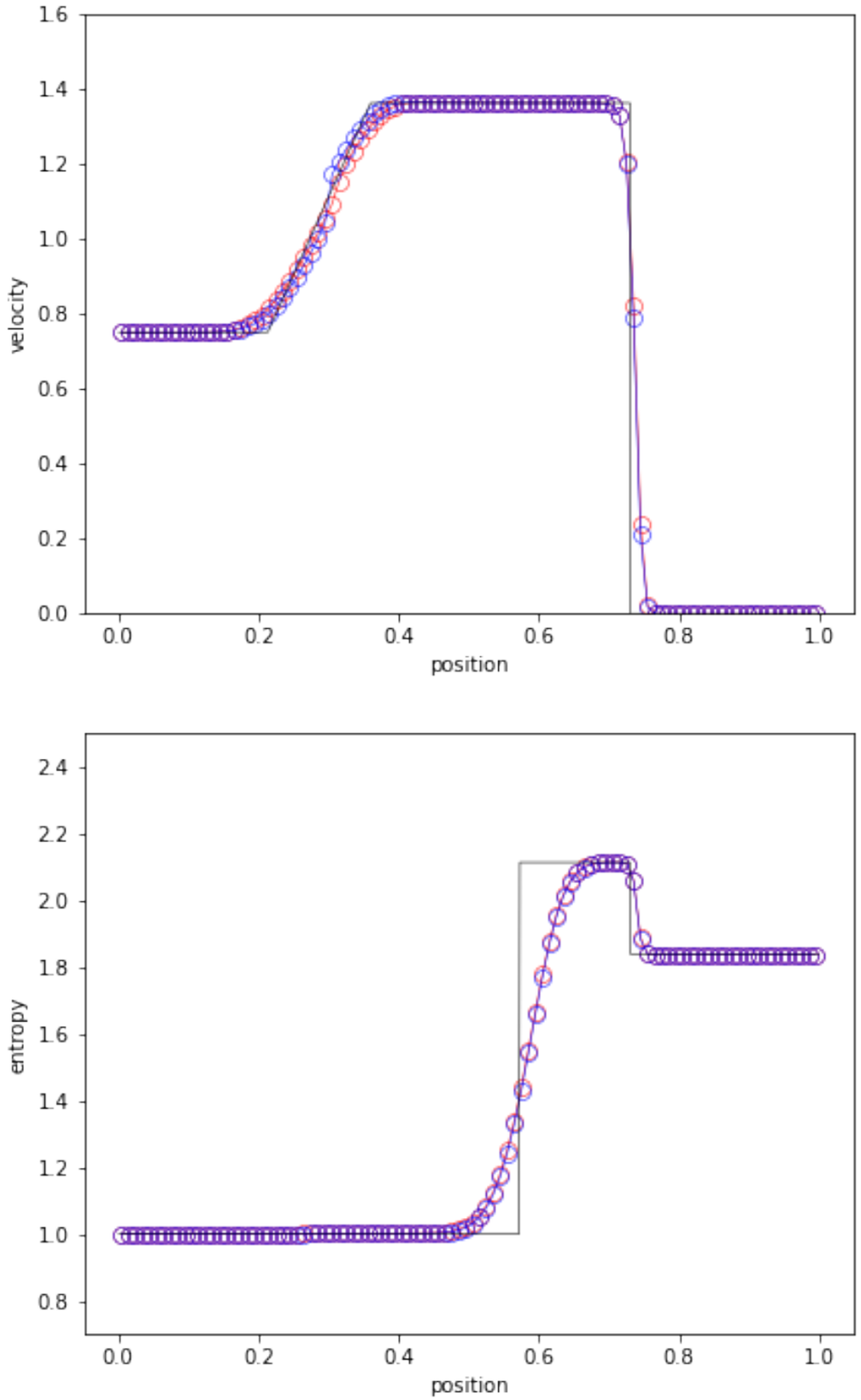
Conclusions for the Euler Equations

Much more details in Toro's book.

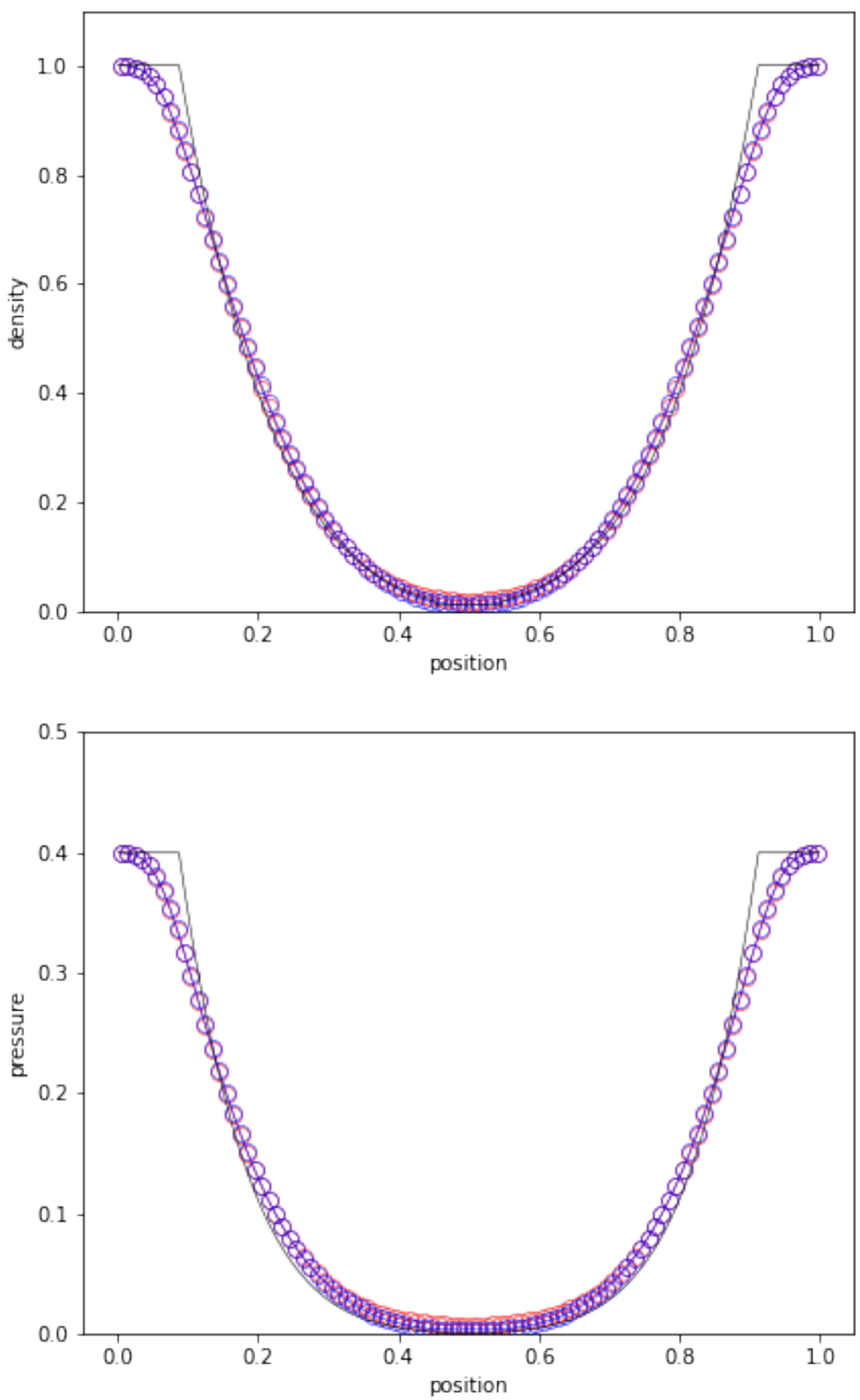


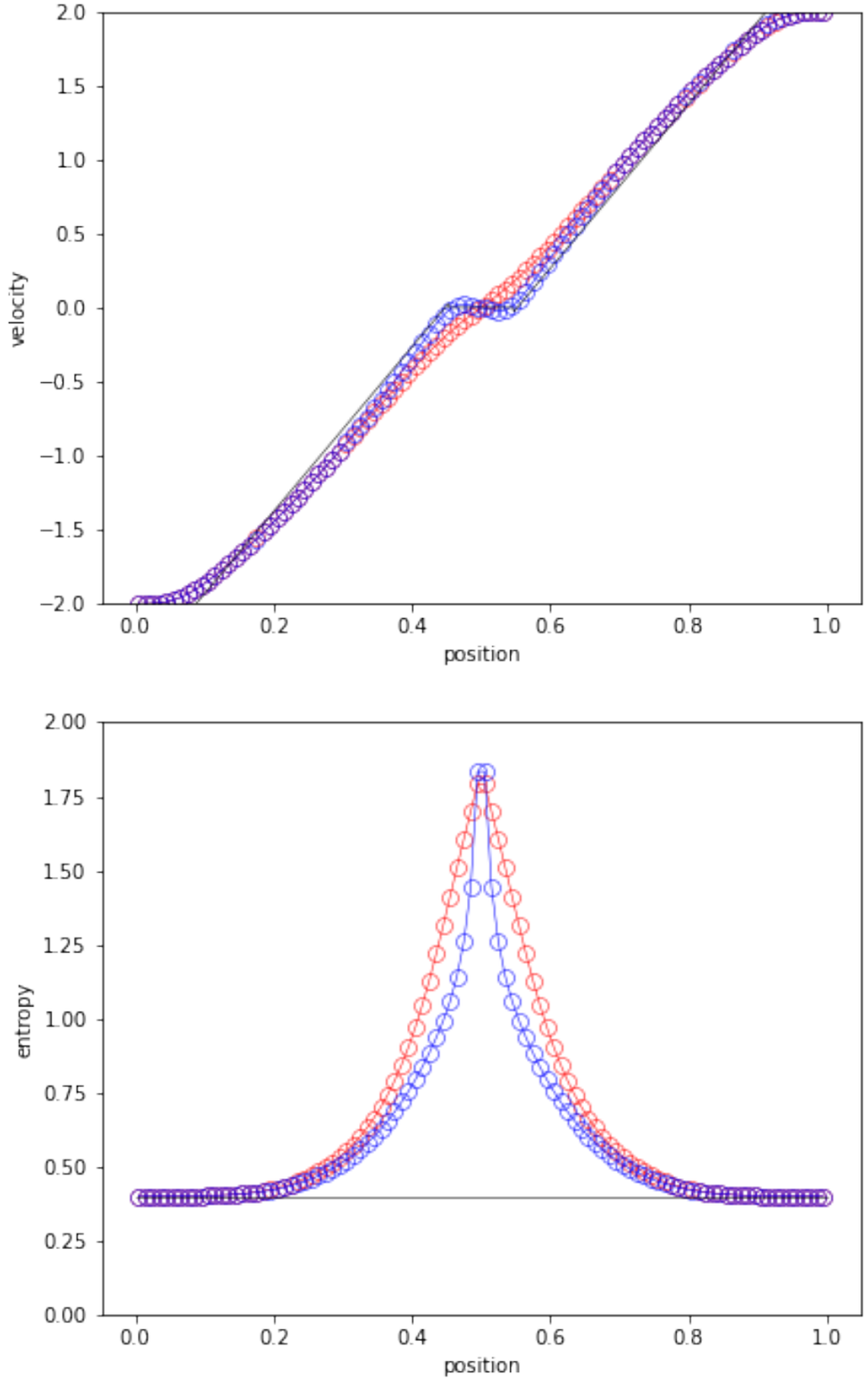
Toro Test 1 (HLLC versus exact)



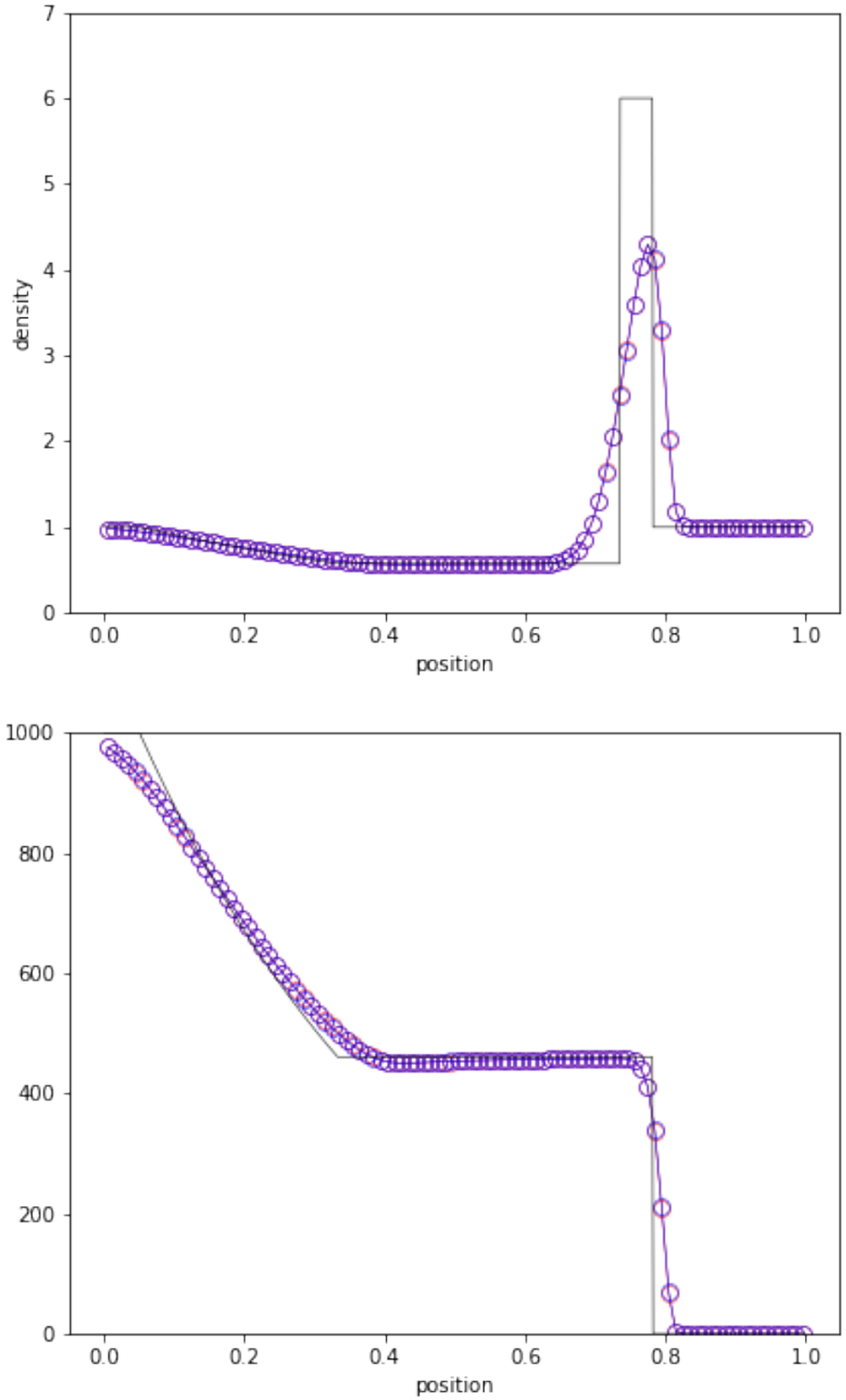


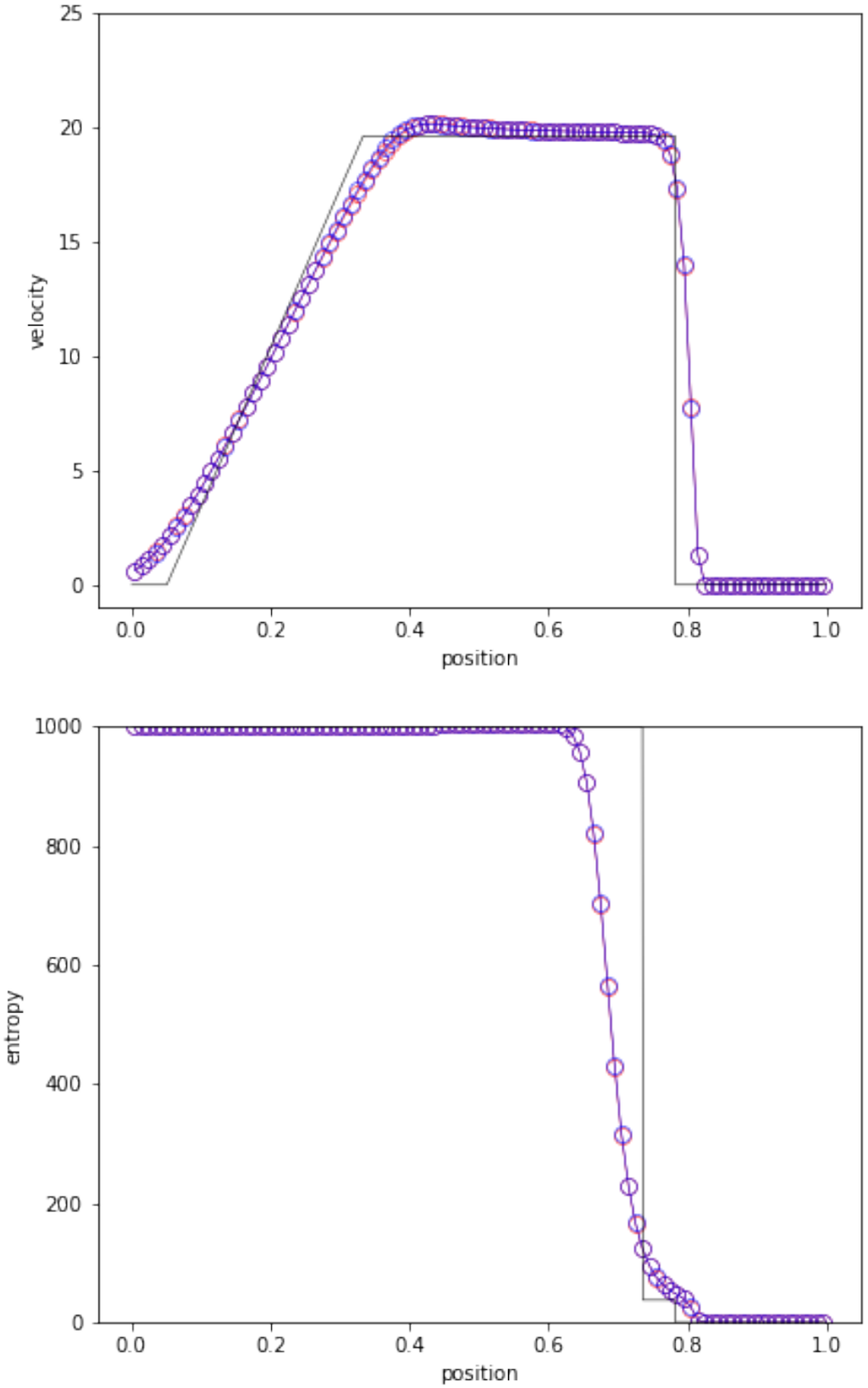
Toro Test 2 (HLLC versus exact)





Toro Test 3 (HLLC versus exact)





Second-order FV schemes

We do not assume the solution is piecewise constant but piecewise linear.

We introduce the slope Δu for the linear reconstruction of the underlying solution.

$$u(x) = \overline{u}_i + (x - x_i)\frac{\Delta u}{h}.$$

This slope will be particularly important later. It can be computed using different approximations of the first-order space derivative $\dfrac{\partial u}{\partial x} \simeq \dfrac{\Delta u}{h}$:

- the right (downwind) slope $\Delta u_R = \overline{u}_{i+1} - \overline{u}_i$ (Lax-Wendroff),
- the left (upwind) slope $\Delta u_L = \overline{u}_i - \overline{u}_{i-1}$ (Beam-Warming),
- the central slope $\Delta u_C = \dfrac{\Delta u_L + \Delta u_R}{2} = \dfrac{\overline{u}_{i+1} - \overline{u}_{i-1}}{2}$ (Fromm).

The Fromm slope is both accurate and stable (best compromise).

Fourth-order FV scheme

For fourth-order FV schemes, we can use either a right-biased 4-point kernel $\{x_{i-1}, x_i, x_{i+1}, x_{i+2}\}$ or a left-biased 4-point kernel $\{x_{i-2}, x_{i-1}, x_i, x_{i+1}\}$.

The first one is only marginally stable, while the second one is stable (upwind).

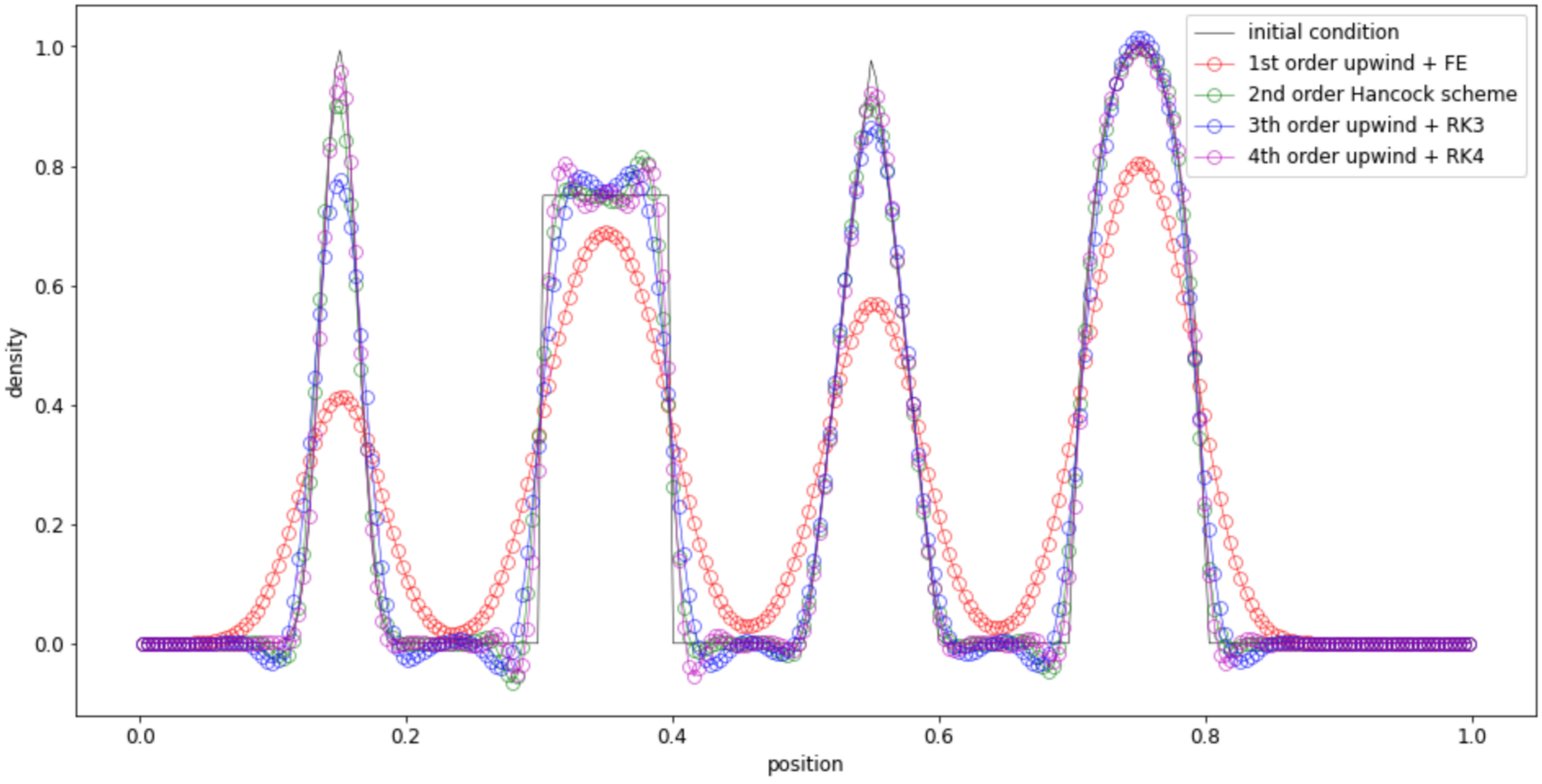
Here again, the best compromise is obtained by averaging the 2 schemes (left and right), leading to a 5-point symmetric kernel $\{x_{i-2}, x_{i-1}, x_i, x_{i+1}, x_{i+2}\}$.

At the right interface: $u(x^-_{i+1/2}) = \dfrac{\overline{u}_{i-2} - 6\overline{u}_{i-1} + 20\overline{u}_i + 10\overline{u}_{i+1} - \overline{u}_{i+2}}{24}$.

At the left interface: $u(x^+_{i-1/2}) = \dfrac{-\overline{u}_{i-2} + 10\overline{u}_{i-1} + 20\overline{u}_i - 6\overline{u}_{i+1} + \overline{u}_{i+2}}{24}$.

- Note that going to fourth-order requires a much wider stencil.
- High-order schemes also require higher-order time integration.
- High-order schemes are also oscillatory (Godunov theorem).

Numerical Example: Composite Profile



Slope Limiters

Godunov barrier theorem: all linear schemes of order > 1 are oscillatory.
The key assumption in Godunov's theorem is a linear scheme with constant coefficients. Idea: design non-linear numerical schemes that preserve monotonicity.

In all high-order FV schemes, we need to reconstruct the solution at the cell interfaces.

Second-order schemes use a linear function $u(x) = \overline{u}_i + (x - x_i)\dfrac{\Delta u_i}{h}$.

If the slope is computed using constant coefficients such as the Fromm slope:

$$\Delta u_i = \frac{\overline{u}_{i+1} - \overline{u}_{i-1}}{2},$$

the scheme will not be monotonicity preserving and will produce oscillations (and also possibly negative values).

Slope limiting consists in introducing a non-linear limiter function $\phi(\overline{u}_{i-1}, \overline{u}_i, \overline{u}_{i+1})$ with $0 \leq \phi \leq 1$ that will preserve monotonicity, if one uses instead:

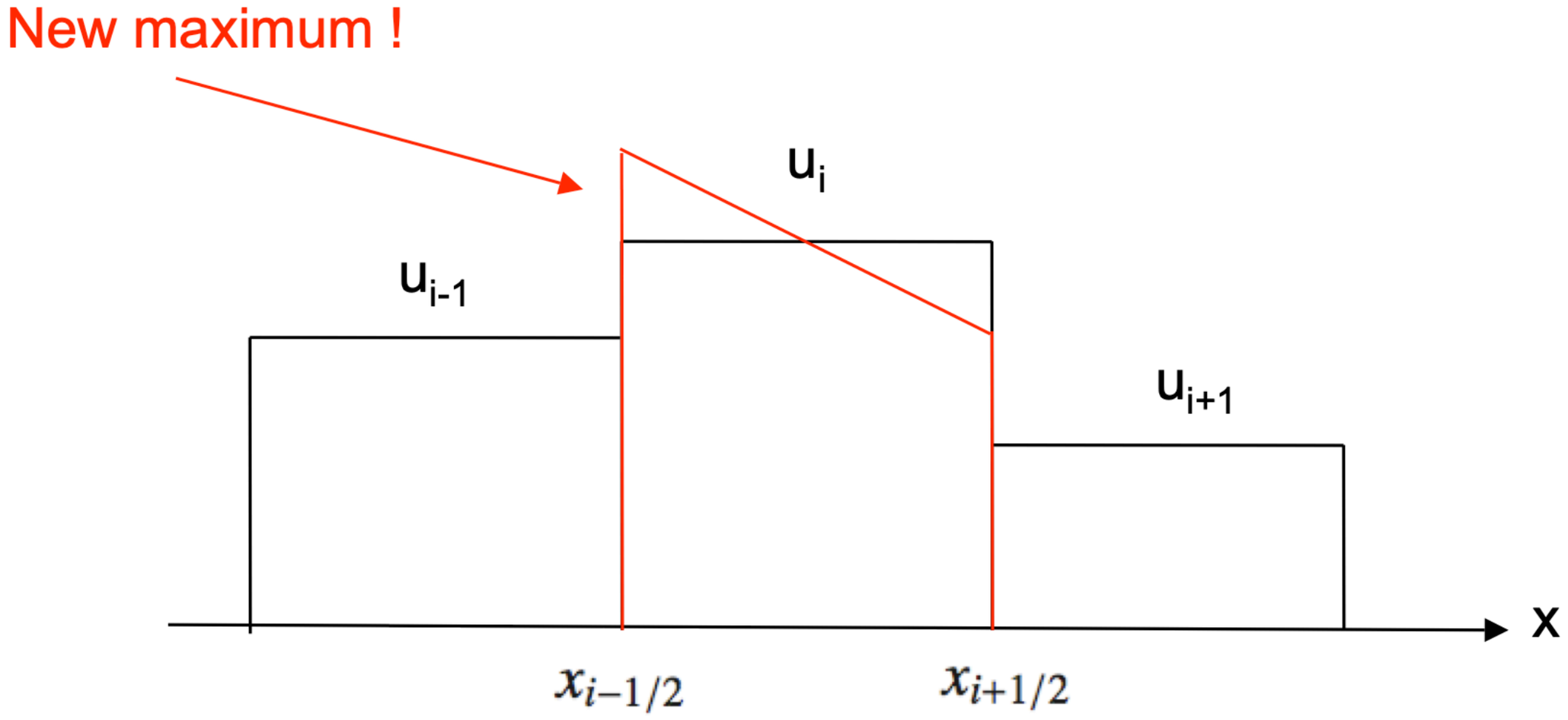
$$\Delta u_i = \phi(\overline{u}_{i-1}, \overline{u}_i, \overline{u}_{i+1})\frac{\overline{u}_{i+1} - \overline{u}_{i-1}}{2}$$

Slope Limiters

We have defined earlier 3 different slopes.:

$$\Delta u_L = \overline{u}_i - \overline{u}_{i-1},\ \Delta u_R = \overline{u}_{i+1} - \overline{u}_i, \text{ and } \Delta u_C = \frac{\overline{u}_{i+1} - \overline{u}_{i-1}}{2}.$$

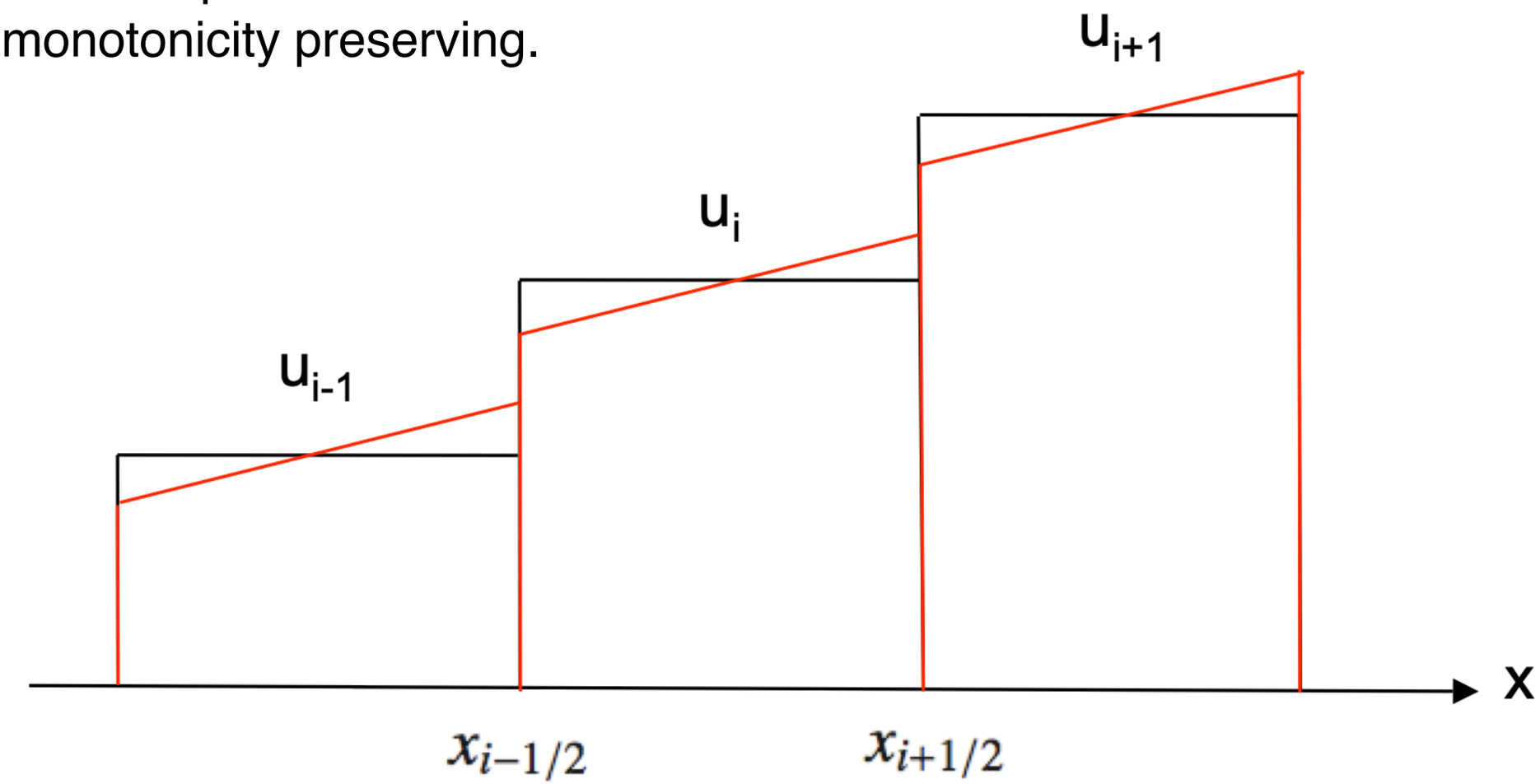
The first requirement is to avoid introducing a new extremum in the solution.



This translates in the rather strict condition: if $\Delta u_L \Delta u_R < 0$, then set $\Delta u_i = 0$.

The minmod Slope Limiter

The minmod slope limiter enforces that the reconstructed solution at time t^n is strictly monotonicity preserving.



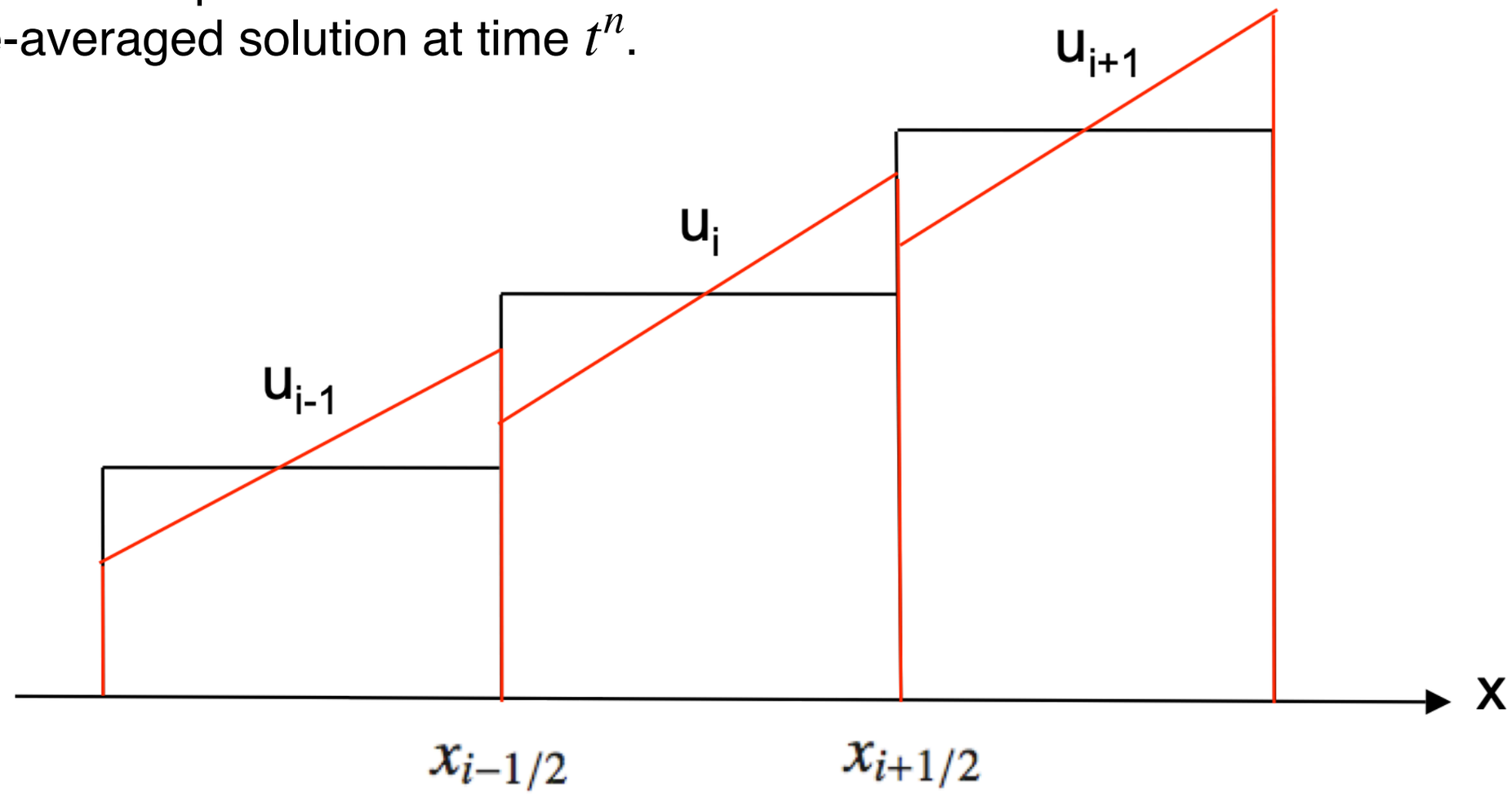
Inside cell $[x_{i-1/2}, x_{i+1/2}]$, we have $u^+_{i-1/2} = \overline{u}^n_i - \dfrac{\Delta u_i}{2}$ and $u^-_{i+1/2} = \overline{u}^n_i + \dfrac{\Delta u_i}{2}$.

Monotonicity condition $u^-_{i+1/2} \le u^+_{i+1/2}$ translates into $\dfrac{\Delta u_i + \Delta u_{i+1}}{2} \le \overline{u}_{i+1} - \overline{u}_i$.

This is satisfied only if we choose $\Delta u_i = \min(\Delta u_L, \Delta u_R)$.

The moncen Slope Limiter

The moncen slope limiter enforces that the reconstructed values are bounded by the volume-averaged solution at time t^n.



Inside cell $[x_{i-1/2}, x_{i+1/2}]$, we have $u^+_{i-1/2} = \overline{u}^n_i - \dfrac{\Delta u_i}{2}$ and $u^-_{i+1/2} = \overline{u}^n_i + \dfrac{\Delta u_i}{2}$.

Monotonicity means now $\overline{u}^n_{i-1} \le u^+_{i-1/2} \le \overline{u}^n_i$ and $\overline{u}^n_i \le u^-_{i+1/2} \le \overline{u}^n_{i+1}$.

This is satisfies if we choose $\Delta u_i = \min(2\Delta u_L, \Delta u_C, 2\Delta u_R)$.

This way, we recover exactly the Fromm scheme when $\phi = 1$.

Smooth Extrema Detection

Both moncen and minmod set the slope to zero in case the solution is near a maximum or a minimum.

For smooth solutions, this is an undesirable effect. Indeed, for smooth solutions, we generally don't see oscillations so we can keep high-order accuracy.

The fix is to detect these smooth extrema, for which the first derivative is zero but varies smoothly across the extremum, and turn off the limiter there.

We therefore design a discontinuity detection strategy for the first derivative.

We compute the first derivative using the FD approximation $u'_i = \dfrac{\overline{u}_{i+1} - \overline{u}_{i-1}}{2h}$.

We then compute the left, right and central slopes of the derivative as

$$\Delta u'_L = u'_i - u'_{i-1},\ \Delta u'_R = u'_{i+1} - u'_i, \text{ and } \Delta u'_C = \frac{u'_{i+1} - u'_{i-1}}{2}.$$

Smooth Extrema Detection

The smoothness indicator is based on the moncen slope limiter but now applied to the derivative of the solution.

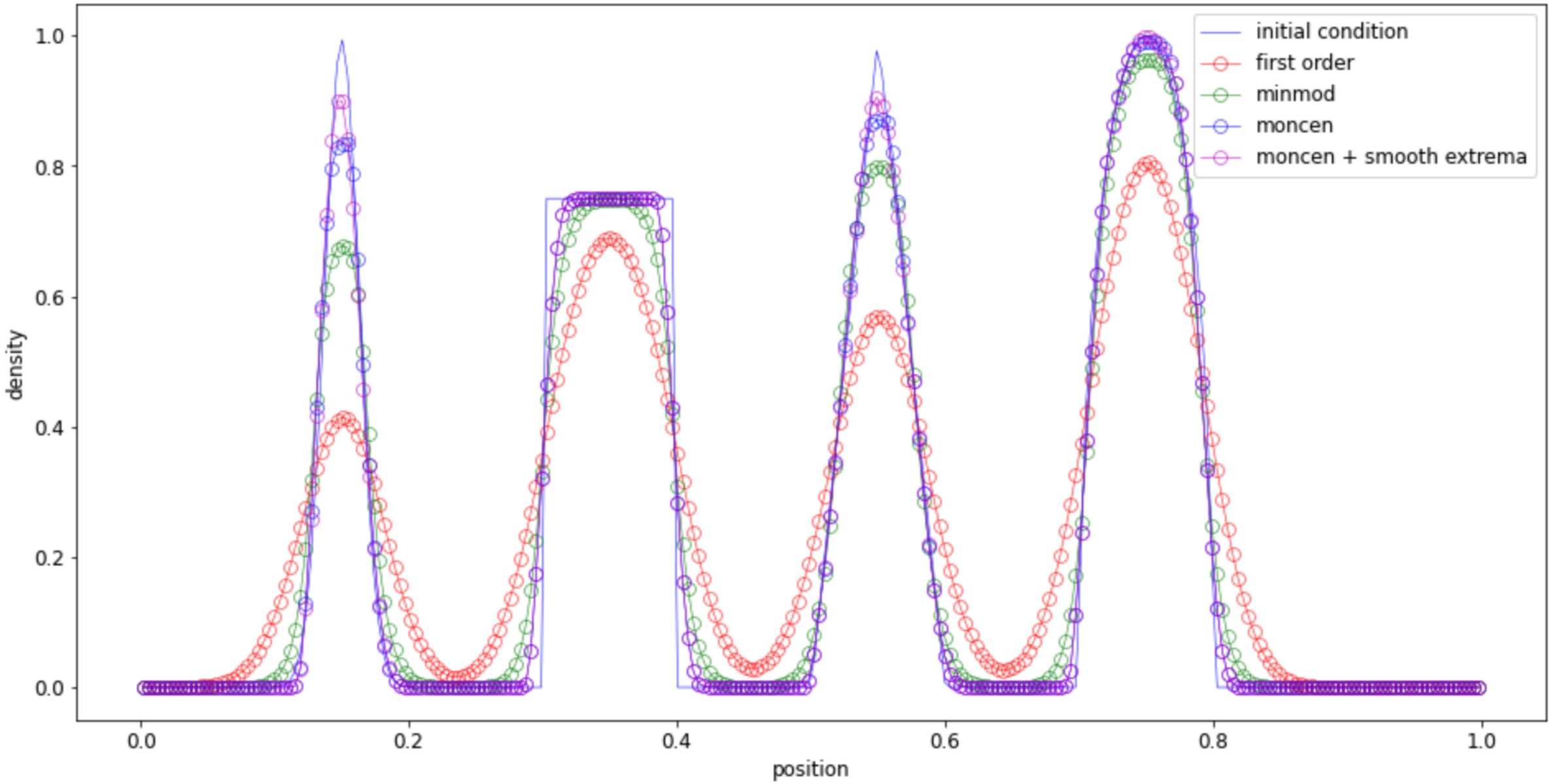
If the central slope is smaller than twice the left (or right) slope, then the current cell has a smooth first order derivative and we don't need to apply a slope limiter.

For this, we compute $\alpha_{L,R} = \begin{cases} \min\left(1, \frac{\min\left(2\Delta u'_{L,R}, 0\right)}{\Delta u'_C}\right) & \text{if } \Delta u'_C < 0 \\ 1 & \text{if } \Delta u'_C = 0. \\ \min\left(1, \frac{\max\left(2\Delta u'_{L,R}, 0\right)}{\Delta u'_C}\right) & \text{if } \Delta u'_C > 0 \end{cases}$

We define the smoothness indicator in each cell as $\alpha_i = \min(\alpha_L, \alpha_R)$.

We apply the slope limiter only if $\min(\alpha_{i-1}, \alpha_i, \alpha_{i+1}) < 1$, otherwise we consider we have a smooth extremum and we adopt the unlimited central (Fromm) slope.

Numerical Examples: Composite Profile



Time Integration for high-order FV schemes

In case of a high-order spatial reconstruction of the solution inside the cell, the Riemann problem at each cell interface is not self-similar anymore.

The Godunov scheme writes as before

$$\overline{u}_i^{n+1} = \overline{u}_i^n + \int_{t^n}^{t^{n+1}} \frac{d\overline{u}_i}{dt} \mathrm{d}t = \overline{u}_i^n - \frac{\Delta t}{h}(f_{i+1/2}^{n+1/2} - f_{i-1/2}^{n+1/2})$$

but now we have to compute the time average using a high-order quadrature

$$f_{i+1/2}^{n+1/2} = \frac{1}{\Delta t}\int_{t^n}^{t^{n+1}} f(u(x_{i+1/2}, t))\mathrm{d}t = \sum_{k=1}^{s} w_k f(u_G(x_{i+1/2}, t^k)).$$

For this, we can use Runge-Kutta schemes of various orders. The choice of the order will be determined by a stability condition or by the desired accuracy.

Note at each stage of the RK scheme, we need to solve a Riemann problem.

MUSCL Scheme for the Euler Equations

We describe step by step the MUSCL-Hancock scheme for the Euler equations.

We know at time t^n the cell-averaged conservative variables:

$$\overline{\mathbf{U}}_i^n = [\overline{\rho}_i^n, \overline{\rho v}_i^n, \overline{E}_i^n].$$

We compute the cell-averaged primitive variables:

$$\overline{\mathbf{W}}_i^n = [\overline{\rho}_i^n, \overline{v}_i^n, \overline{P}_i^n]$$

using $\overline{v}_i^n = \overline{\rho v}_i^n / \overline{\rho}_i^n$ and $\overline{P}_i^n = (\gamma - 1)\left(\overline{E}_i^n - \frac{1}{2}\overline{\rho}_i^n (\overline{v}_i^n)^2 \right)$.

We compute the cell-averaged sound speed $\overline{c}_i^n = \sqrt{\frac{\gamma \overline{P}_i^n}{\overline{\rho}_i^n}}$.

We compute the new time step as $\Delta t = C \frac{h}{\max\left(|\overline{v}_i^n| + \overline{c}_i^n \right)}$.

MUSCL Scheme for the Euler Equations

For each primitive variable $\overline{\mathbf{W}}_i^n = [\overline{\rho}_i^n, \overline{v}_i^n, \overline{P}_i^n]$, we compute the smooth extrema detector $\alpha_i(\rho),\ \alpha_i(v),\ \alpha_i(P)$ as detailed in the previous slides.

For each primitive variable, we compute the limited slope $\Delta\mathbf{W}_i = [\Delta\rho_i, \Delta v_i, \Delta P_i]$ only if $\min(\alpha_{i-1}, \alpha_i, \alpha_{i+1}) < 1$, using the minmod or moncen slope limiters as described in the previous slides.

We perform the non-conservative predictor step using the quasi-linear form:

$$\overline{\mathbf{W}}_i^{n+1/2} = \overline{\mathbf{W}}_i^n - \frac{\Delta t}{2h}\mathbb{A}\Delta W_i.$$

We reconstruct the predicted left and right interface values as:

$$\mathbf{W}_{i-1/2}^{+} = \overline{\mathbf{W}}_i^{n+1/2} - \frac{1}{2}\Delta W_i,$$

$$\mathbf{W}_{i+1/2}^{-} = \overline{\mathbf{W}}_i^{n+1/2} + \frac{1}{2}\Delta W_i.$$

MUSCL Scheme for the Euler Equations

Using primitive variables as explained previously is the most popular version of the MUSCL-Hancock scheme. It preserves nicely contact discontinuities.

Other variants exist, among which:

- predictor step and limited slopes are computed using the conservative variables

$$\overline{\mathbf{U}}_i^n = [\overline{\rho}_i^n, \overline{\rho v}_i^n, \overline{E}_i^n]$$

- predictor step and limited slopes are computed using characteristic variables.

$$\overline{\mathbf{V}}_i^n = [\overline{\alpha_-}_i^n, \overline{\alpha_0}_i^n, \overline{\alpha_+}_i^n]$$

The Riemann solver can be any of the Riemann solver we have discussed in the previous lecture (LLF, HLL, HLLC, Roe, exact…)

MUSCL Scheme for the Euler Equations

The entire reconstruction step can be summarized by the compact form:

$$\mathbf{W}^{+}_{i-1/2} = \overline{\mathbf{W}}^{n}_{i} - \left(\mathbb{I} + \frac{\Delta t}{h} \mathbb{A} \right) \frac{\Delta W_i}{2}$$

$$\mathbf{W}^{-}_{i+1/2} = \overline{\mathbf{W}}^{n}_{i} + \left(\mathbb{I} - \frac{\Delta t}{h} \mathbb{A} \right) \frac{\Delta W_i}{2}.$$

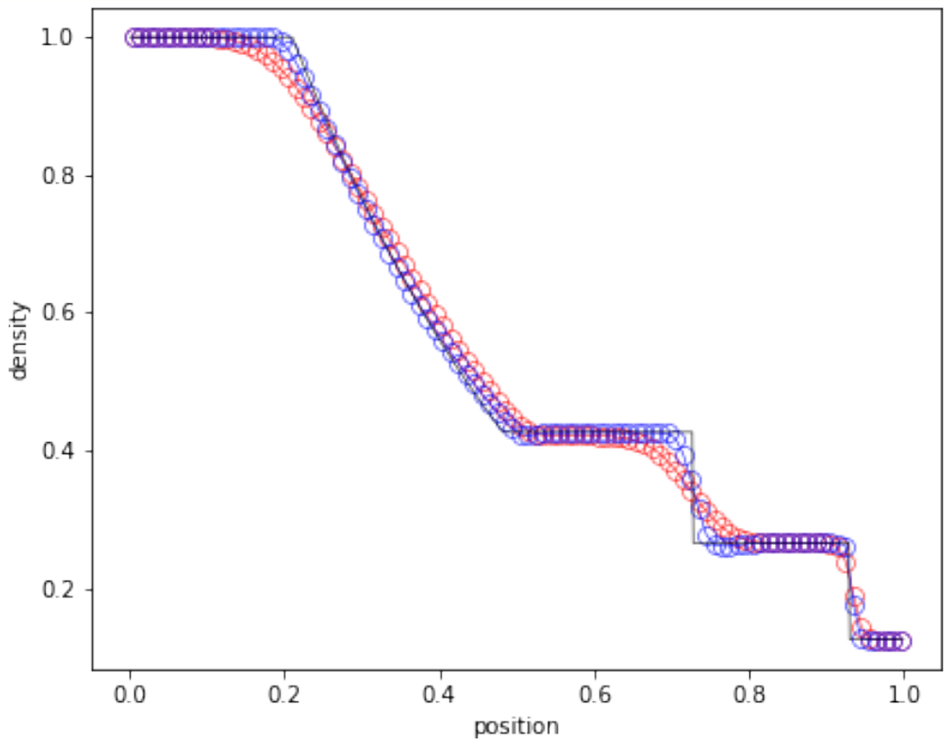
We compute the Godunov flux by solving the Riemann problem at each interface:

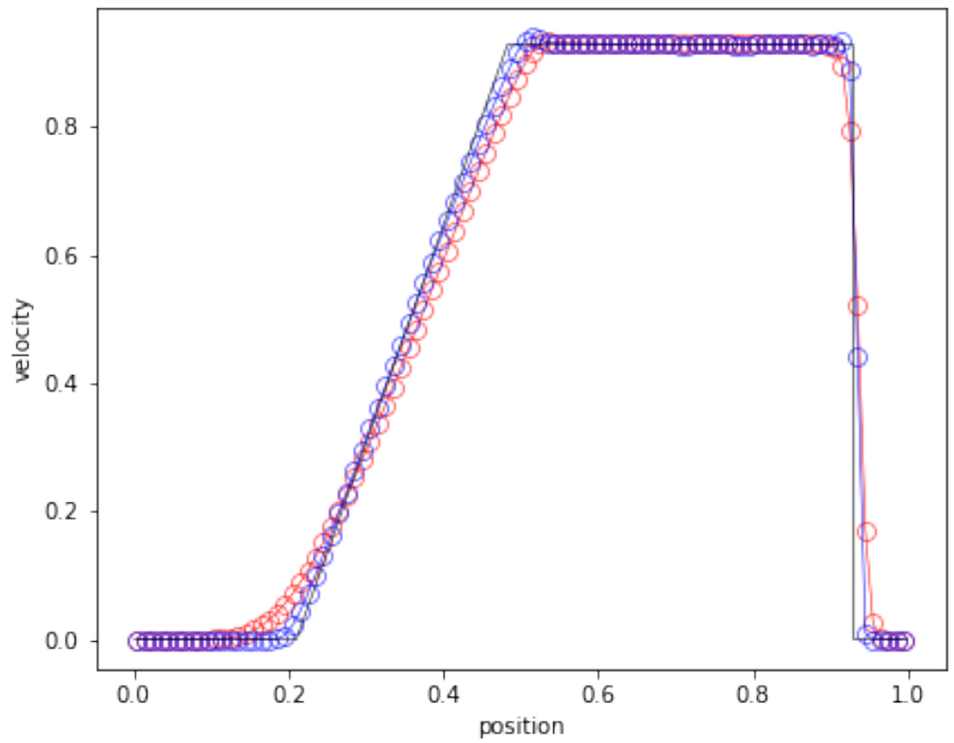
$$\mathbf{F}^{n+1/2}_{i+1/2} = RP(\mathbf{W}^{-}_{i+1/2}, \mathbf{W}^{+}_{i+1/2}).$$

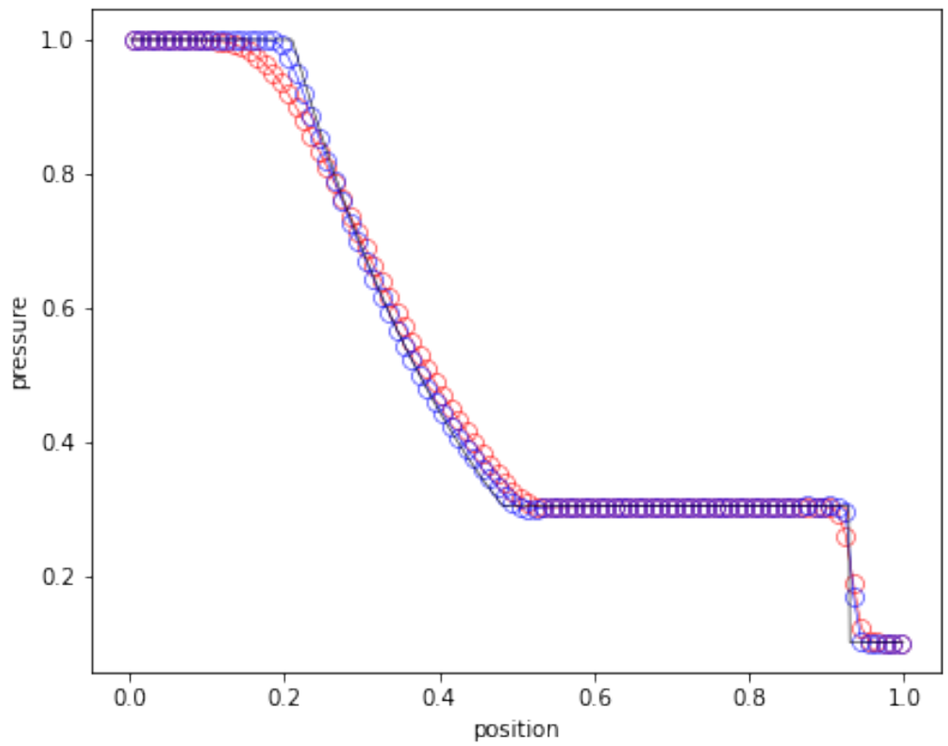
We finally compute the new solution using the conservative update:

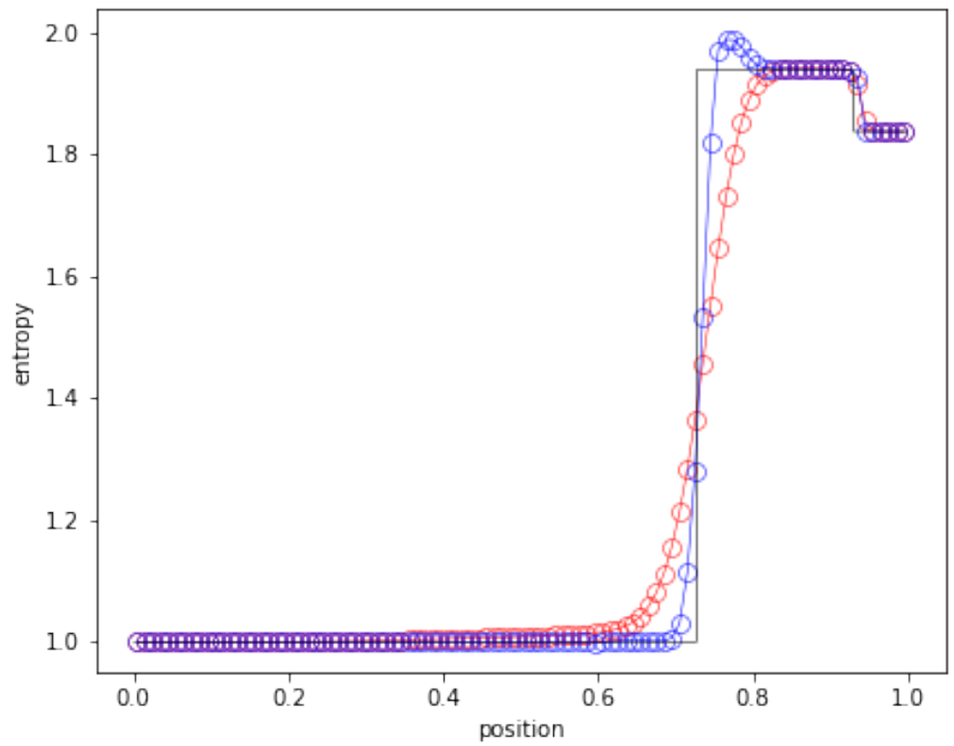
$$\overline{\mathbf{U}}^{n+1}_{i} = \overline{\mathbf{U}}^{n}_{i} - \frac{\Delta t}{h} \left(\mathbf{F}^{n+1/2}_{i+1/2} - \mathbf{F}^{n+1/2}_{i-1/2} \right).$$

Numerical Examples: MUSCL Sod Test



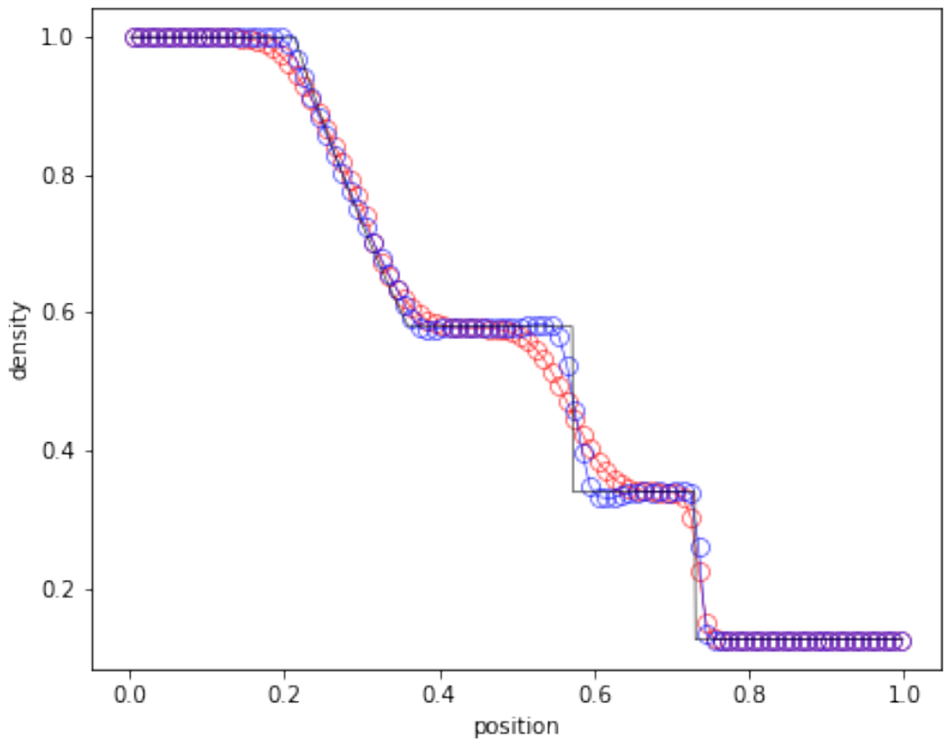


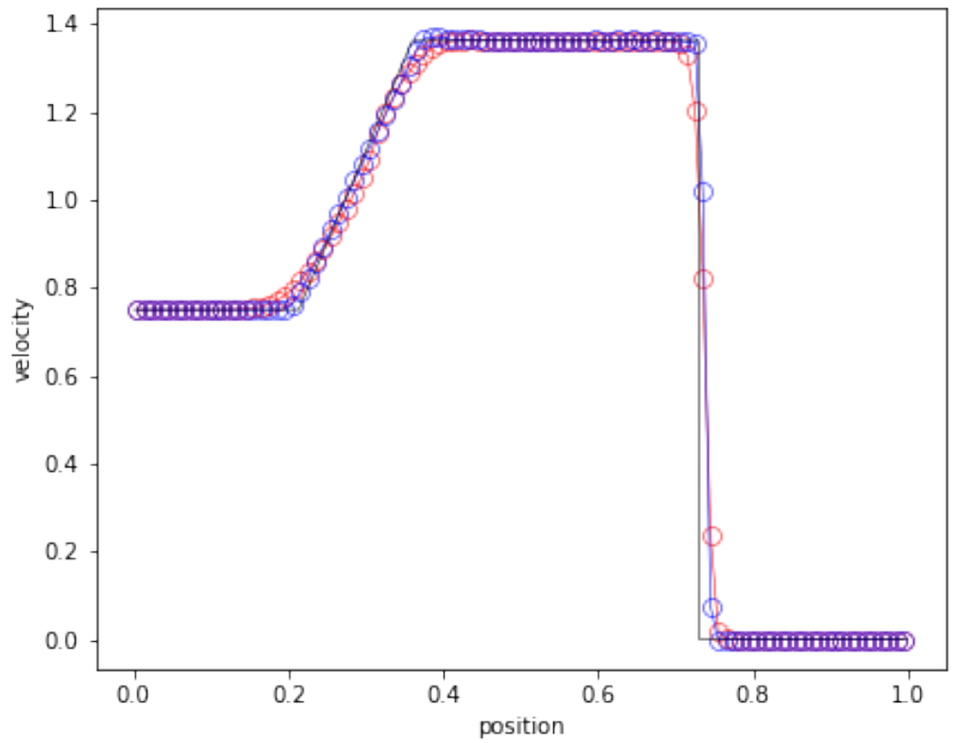


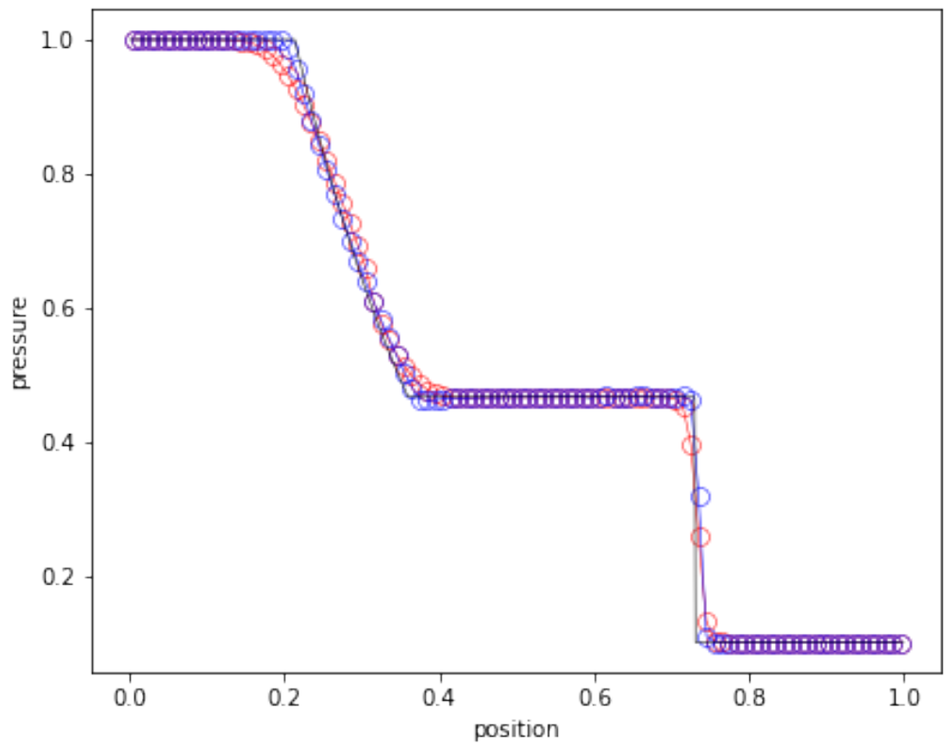


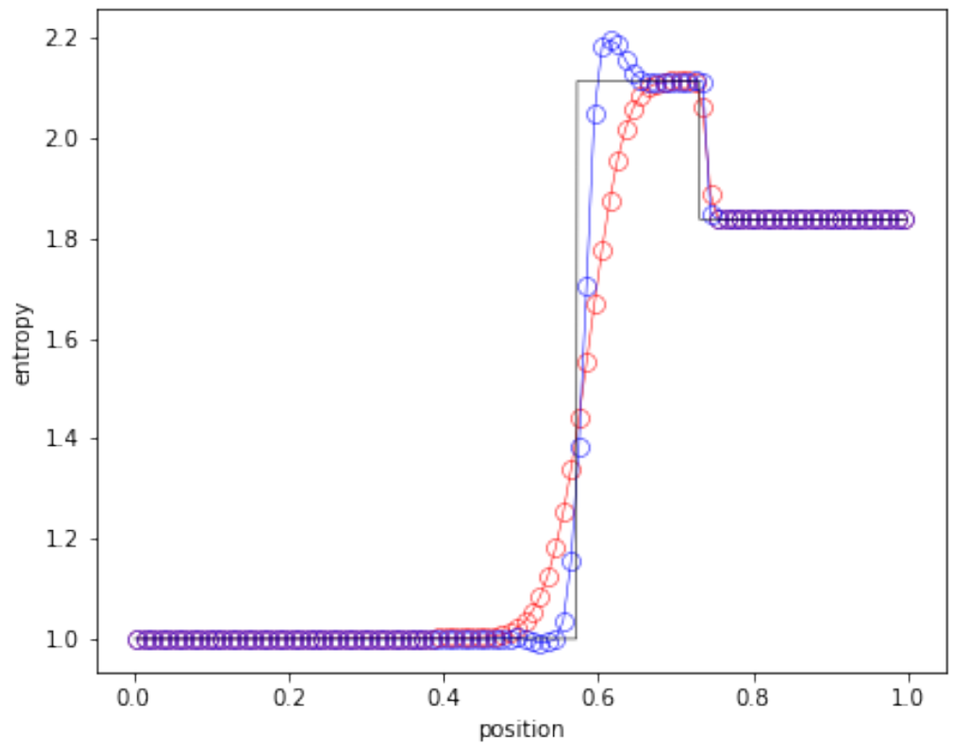
Using 100 cells and the HLLC Riemann solver, we compare first order Godunov in red with MUSCL-Hancock and the moncen slope in blue.

Numerical Examples: MUSCL Toro Test 1



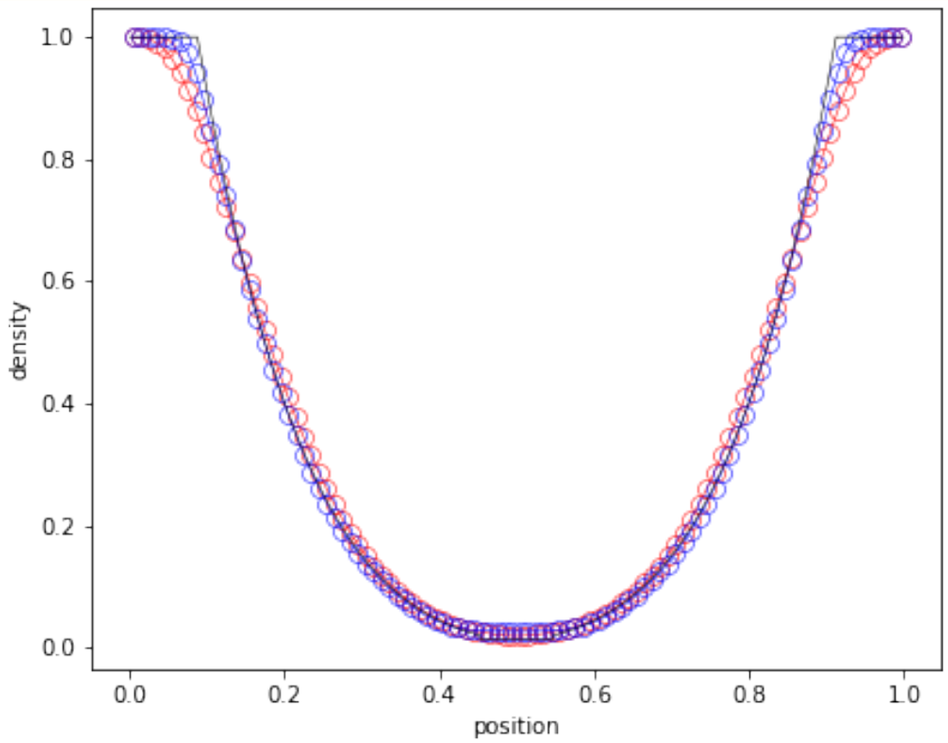


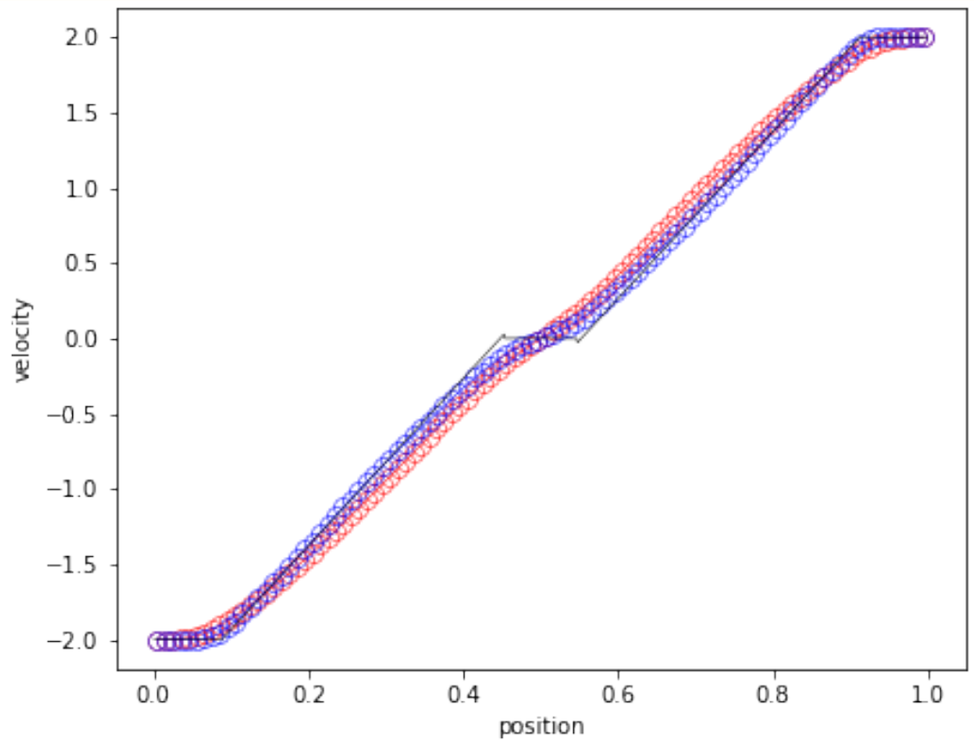


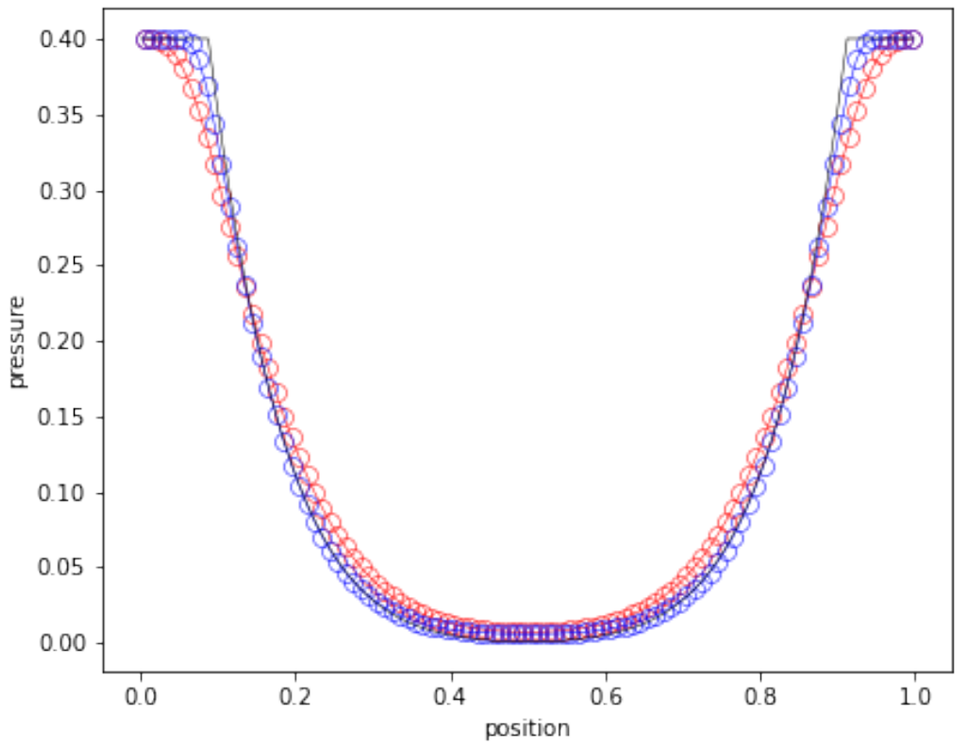


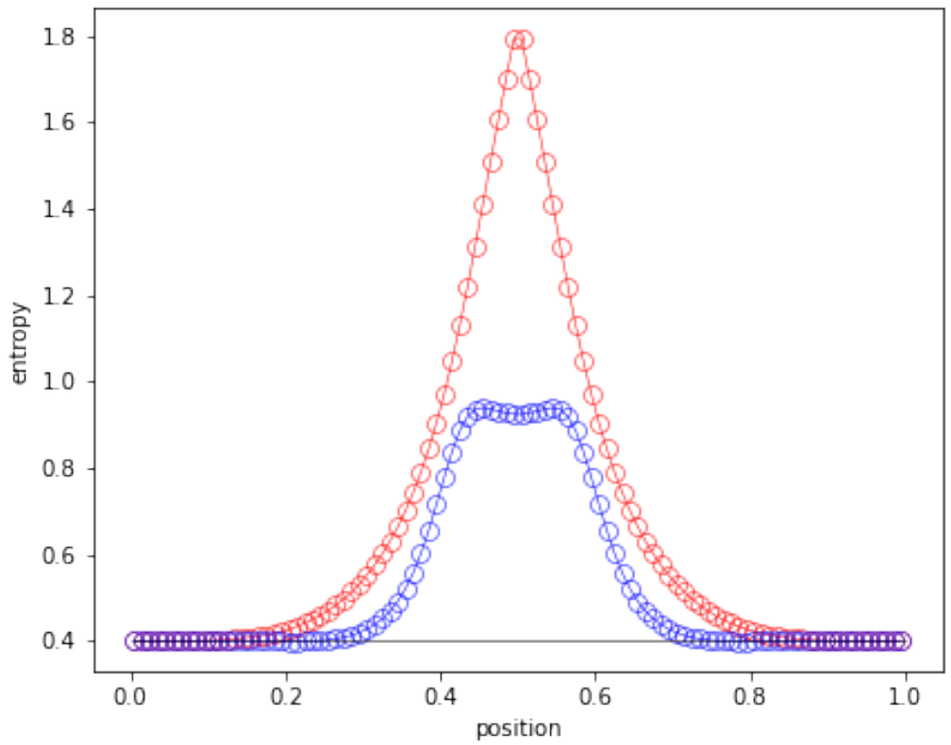
Using 100 cells and the HLLC Riemann solver, we compare first order Godunov in red with MUSCL-Hancock and the moncen slope in blue.

Numerical Examples: MUSCL Toro Test 2



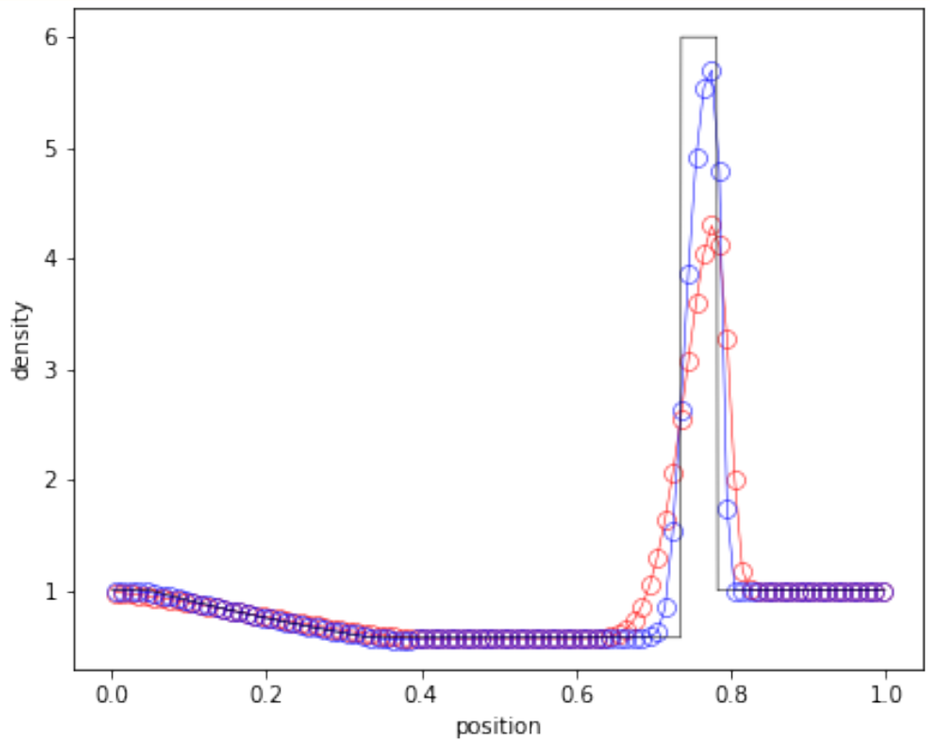


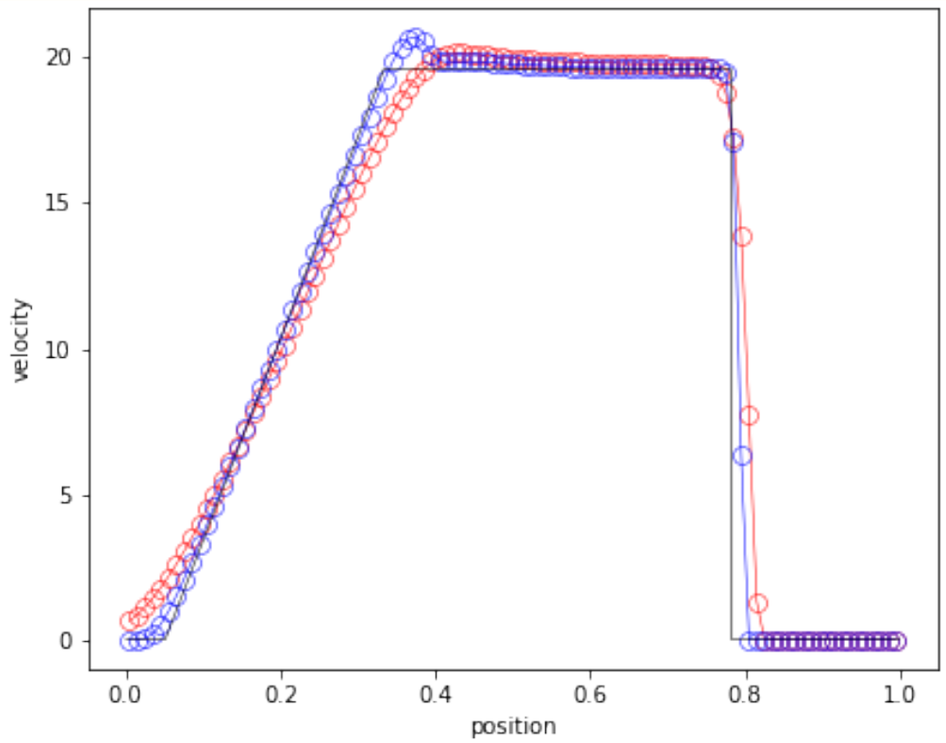


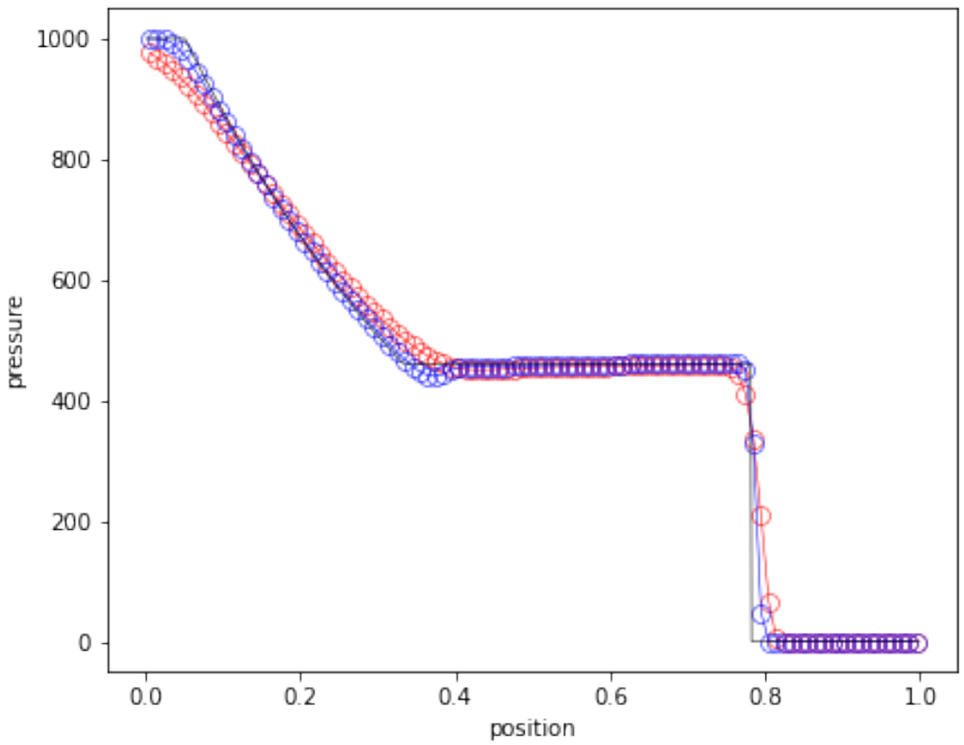


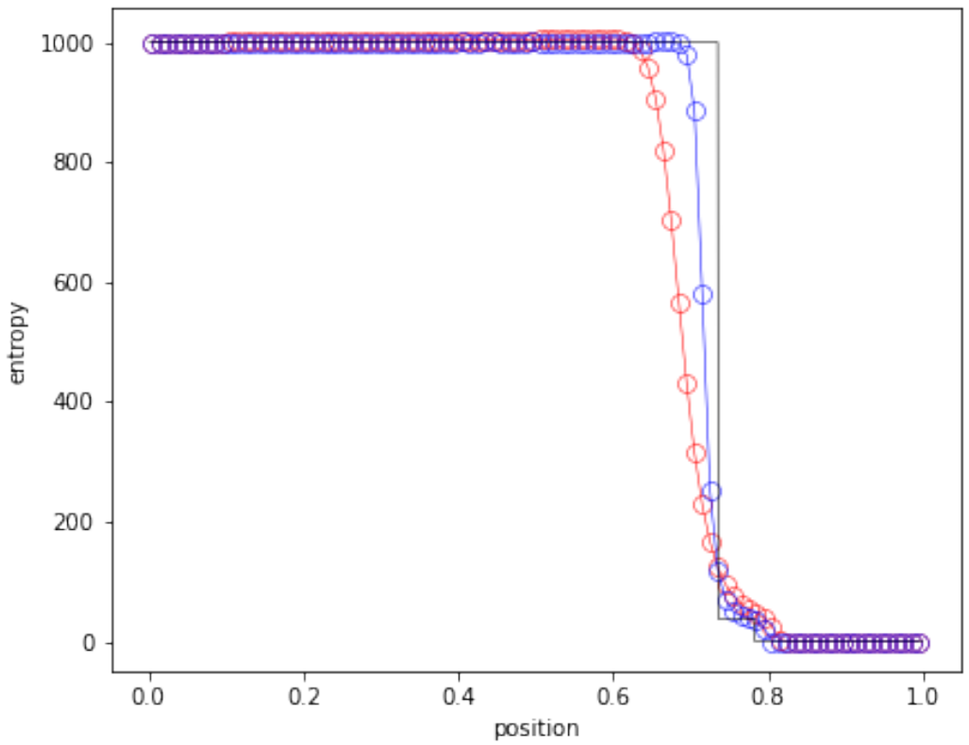
Using 100 cells and the HLLC Riemann solver, we compare first order Godunov in red with MUSCL-Hancock and the moncen slope in blue.

Numerical Examples: MUSCL Toro Test 3



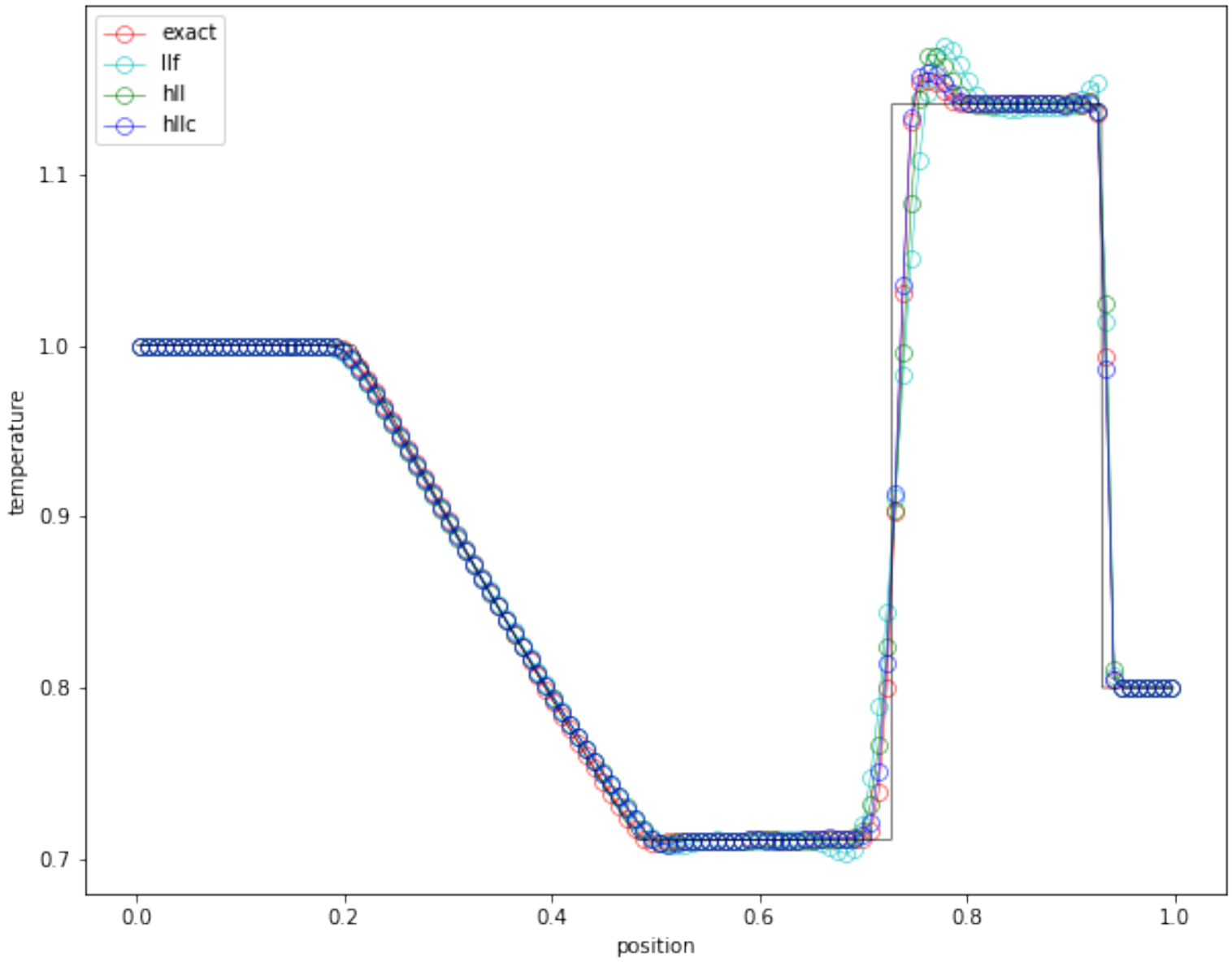






Using 100 cells and the HLLC Riemann solver, we compare first order Godunov in red with MUSCL-Hancock and the moncen slope in blue.

Effect of different Riemann solvers



Fourth-order FV Scheme for the Euler Equations

We describe step by step the fourth order FV scheme for the Euler equation.
It starts very much like the MUSCL-Hancock scheme.

We know at time t^n the cell-averaged conservative variables:

$$\overline{\mathbf{U}}_i^n = [\overline{\rho}_i^n, \overline{\rho v}_i^n, \overline{E}_i^n].$$

We compute the cell-averaged primitive variables:

$$\overline{\mathbf{W}}_i^n = [\overline{\rho}_i^n, \overline{v}_i^n, \overline{P}_i^n]$$

using $\overline{v}_i^n = \overline{\rho v}_i^n / \overline{\rho}_i^n$ and $\overline{P}_i^n = (\gamma - 1)\left(\overline{E}_i^n - \frac{1}{2}\overline{\rho}_i^n (\overline{v}_i^n)^2 \right)$.

We compute the cell-averaged sound speed $\overline{c}_i^n = \sqrt{\frac{\gamma \overline{P}_i^n}{\overline{\rho}_i^n}}$.

We compute the new time step as $\Delta t = C \frac{h}{\max(|\overline{v}_i^n| + \overline{c}_i^n)}$.

Fourth-order FV Scheme for the Euler Equations

For each primitive variable $\overline{w}_i^n = (\overline{\rho}_i^n, \overline{v}_i^n, \overline{P}_i^n)$, we compute the smooth extrema detector $\alpha_i(\rho),\ \alpha_i(v),\ \alpha_i(P)$ like for MUSCL-Hancock.

For each primitive variable, we compute the left interface, central point and right interface values using our degree 3 polynomials.

This gives

$$w_{i+1/2} = \frac{\overline{w}_{i-2}^n - 6\overline{w}_{i-1}^n + 20\overline{w}_i^n + 10\overline{w}_{i+1}^n - \overline{w}_{i+2}^n}{24}$$

$$w_i = \frac{-\overline{w}_{i-1}^n + 26\overline{w}_i^n - \overline{w}_{i+1}^n}{24} = \overline{w}_i^n - \frac{h^2}{24}\Delta\overline{w}_i^n$$

$$w_{i-1/2}^+ = \frac{-\overline{w}_{i-2}^n + 10\overline{w}_{i-1}^n + 20\overline{w}_i^n - 6\overline{w}_{i+1}^n + \overline{w}_{i+2}^n}{24}$$

Fourth-order FV Scheme for the Euler Equations

For each primitive variable, we compute the maximum and the minimum as

$M = \max(\overline{w}_{i-1}^n, \overline{w}_i^n, \overline{w}_{i+1}^n)$ and $m = \min(\overline{w}_{i-1}^n, \overline{w}_i^n, \overline{w}_{i+1}^n)$

and the reconstructed maximum and minimum as

$M_i = \max(w_{i-1/2}^+, w_i, w_{i+1/2}^-)$ and $m_i = \min(w_{i-1/2}^+, w_i, w_{i+1/2}^-)$.

The limiter function is defined as $\theta_i = \min\left(1, \left|\frac{M - \overline{w}_i^n}{M_i - \overline{w}_i^n}\right|, \left|\frac{m - \overline{w}_i^n}{m_i - \overline{w}_i^n}\right|\right)$.

For each primitive variable, we compute limited interface values only if $\min(\alpha_{i-1}, \alpha_i, \alpha_{i+1}) < 1$ using the limiter function θ_i as follows:

$$\widetilde{w}_{i-1/2}^+ = \theta_i(w_{i-1/2}^+ - \overline{w}_i^n) + \overline{w}_i^n$$

$$\widetilde{w}_{i+1/2}^- = \theta_i(w_{i+1/2}^- - \overline{w}_i^n) + \overline{w}_i^n$$

Fourth-order FV Scheme for the Euler Equations

We compute the Godunov flux by solving the Riemann problem at each interface:

$$\mathbf{F}^n_{i+1/2} = RP(\widetilde{\mathbf{W}}^-_{i+1/2}, \widetilde{\mathbf{W}}^+_{i+1/2}).$$

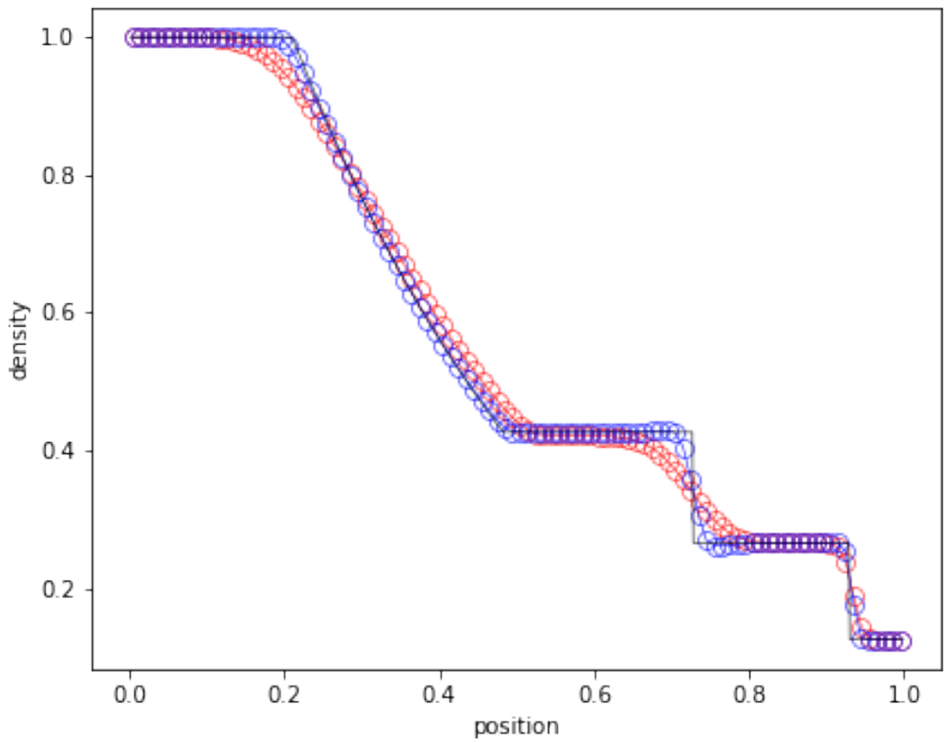
We compute the new solution using the SSP RK3 scheme with 3 stages:

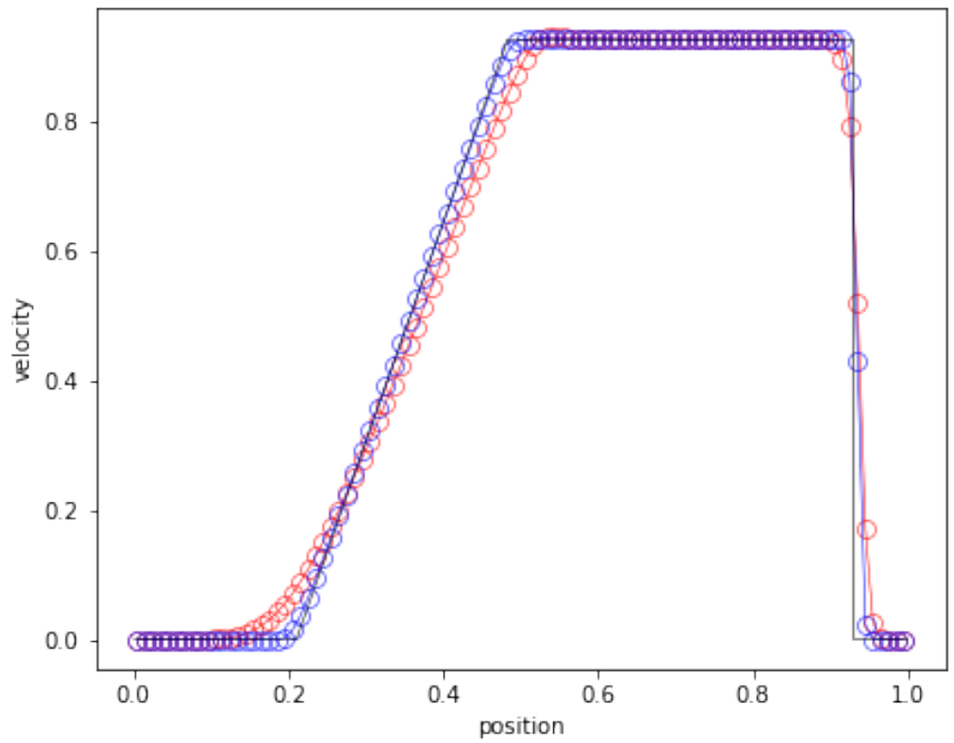
$$\overline{\mathbf{U}}^2_i = \overline{\mathbf{U}}^n_i - \Delta t \frac{\mathbf{F}^n_{i+1/2} - \mathbf{F}^n_{i-1/2}}{h},$$

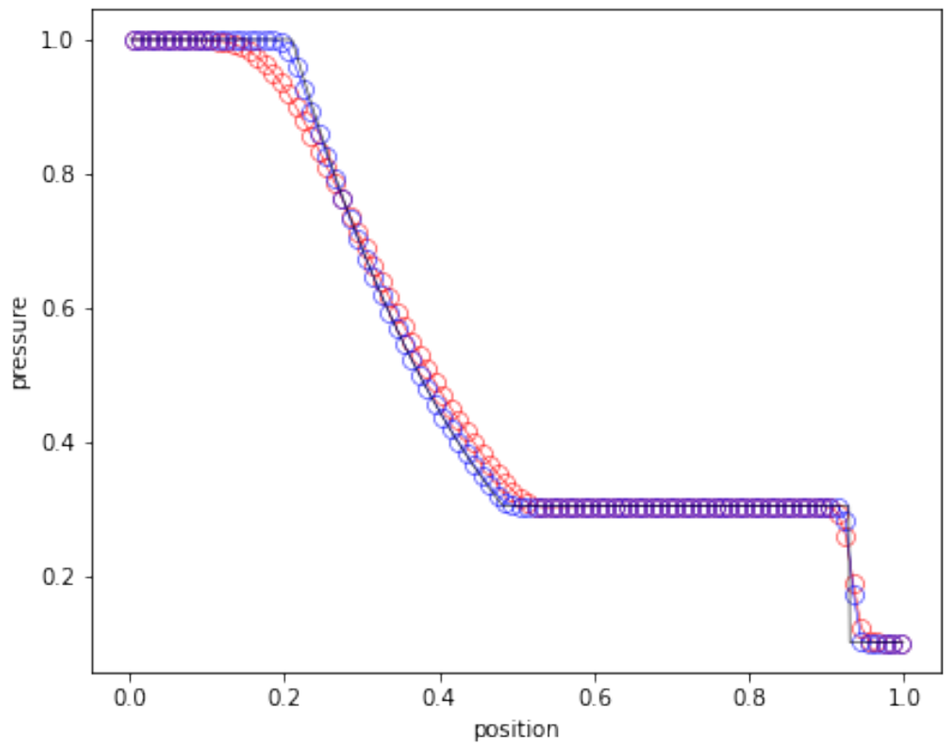
$$\overline{\mathbf{U}}^3_i = \frac{3}{4}\overline{\mathbf{U}}^n_i + \frac{1}{4}\left(\overline{\mathbf{U}}^2_i - \Delta t \frac{\mathbf{F}^2_{i+1/2} - \mathbf{F}^2_{i-1/2}}{h}\right)$$

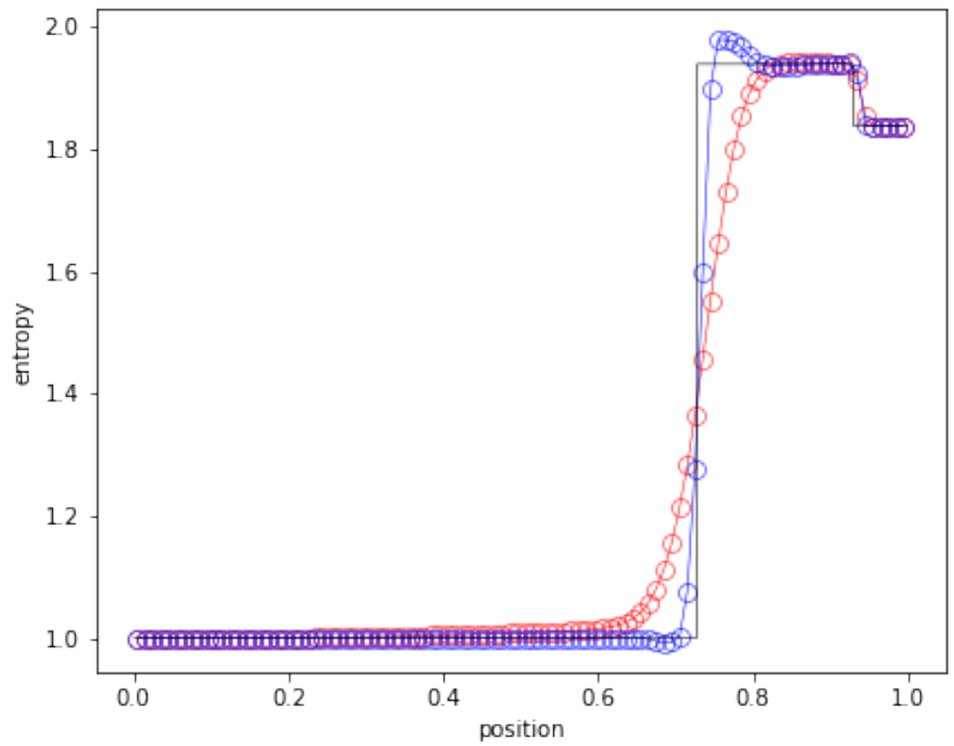
$$\overline{\mathbf{U}}^{n+1}_i = \frac{1}{3}\overline{\mathbf{U}}^n_i + \frac{2}{3}\left(\overline{\mathbf{U}}^3_i - \Delta t \frac{\mathbf{F}^3_{i+1/2} - \mathbf{F}^3_{i-1/2}}{h}\right)$$

Numerical Examples: 4th Order Sod Test



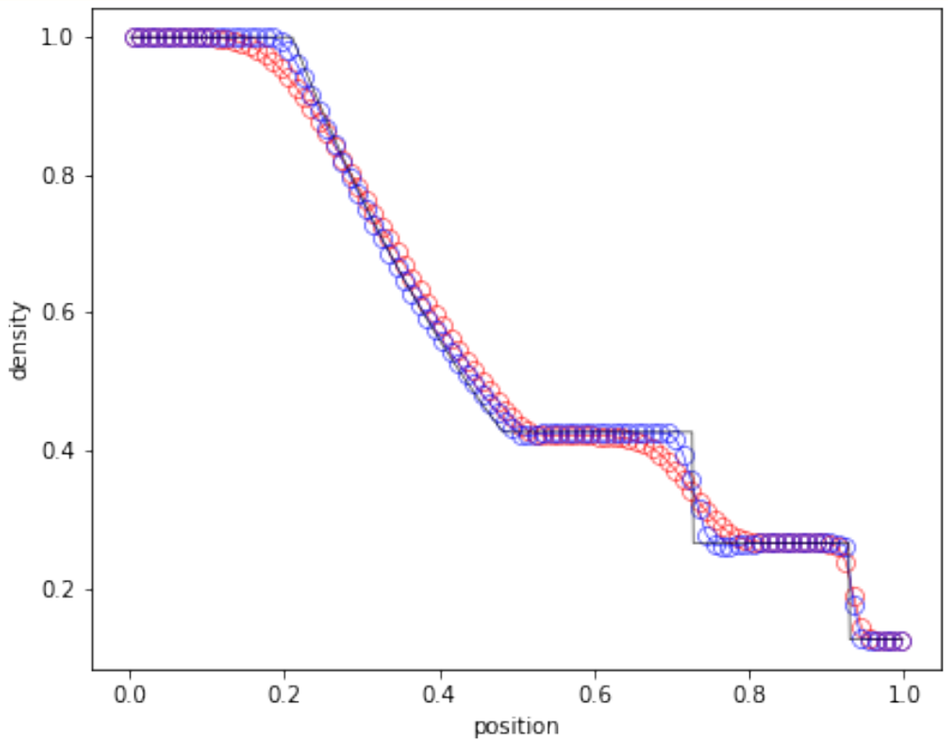


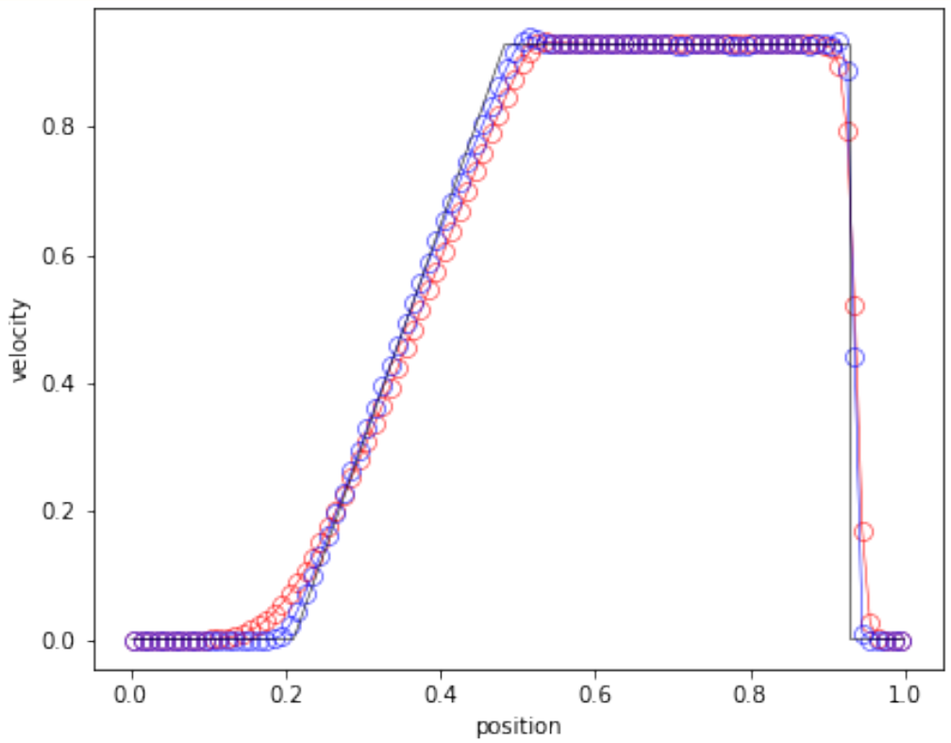


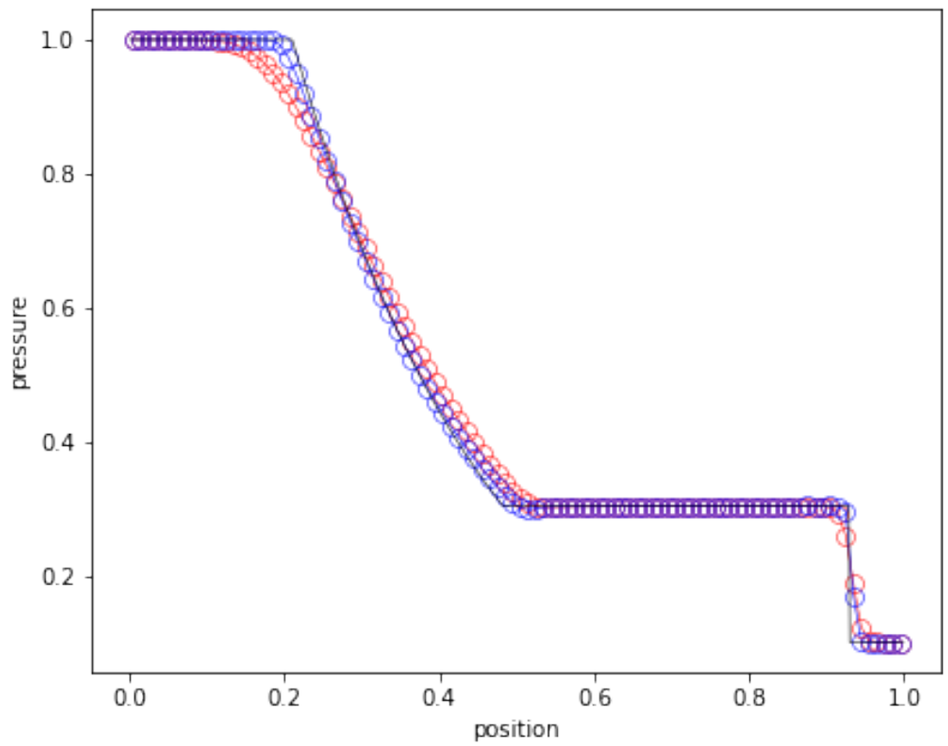


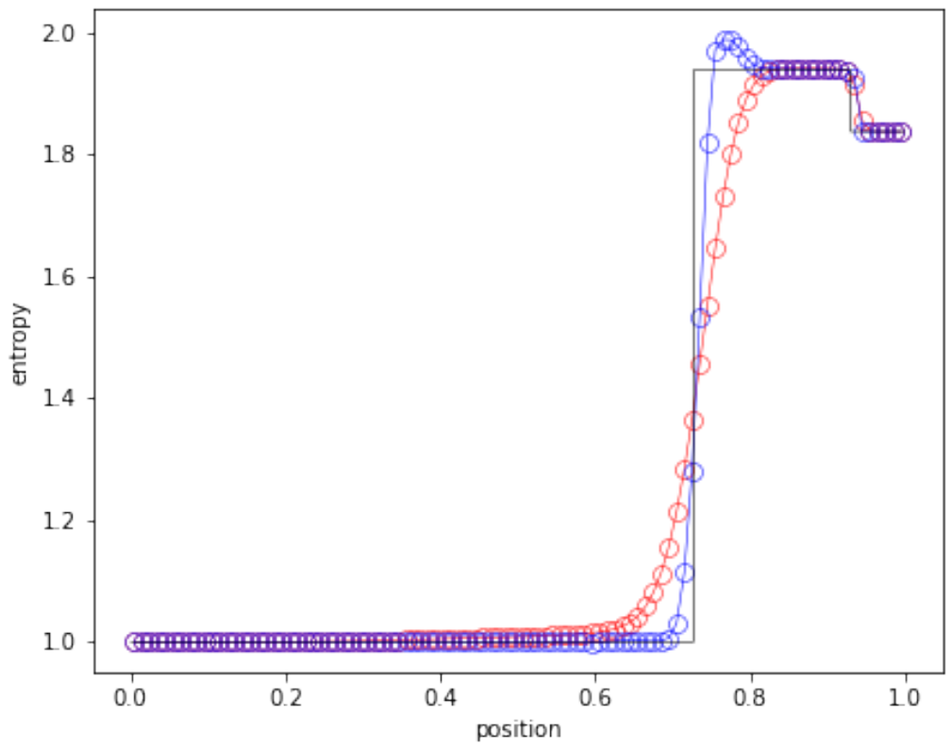
Using 100 cells and the HLLC Riemann solver, we compare first order Godunov in red with our fourth-order FV scheme in blue.

Numerical Examples: MUSCL Sod Test



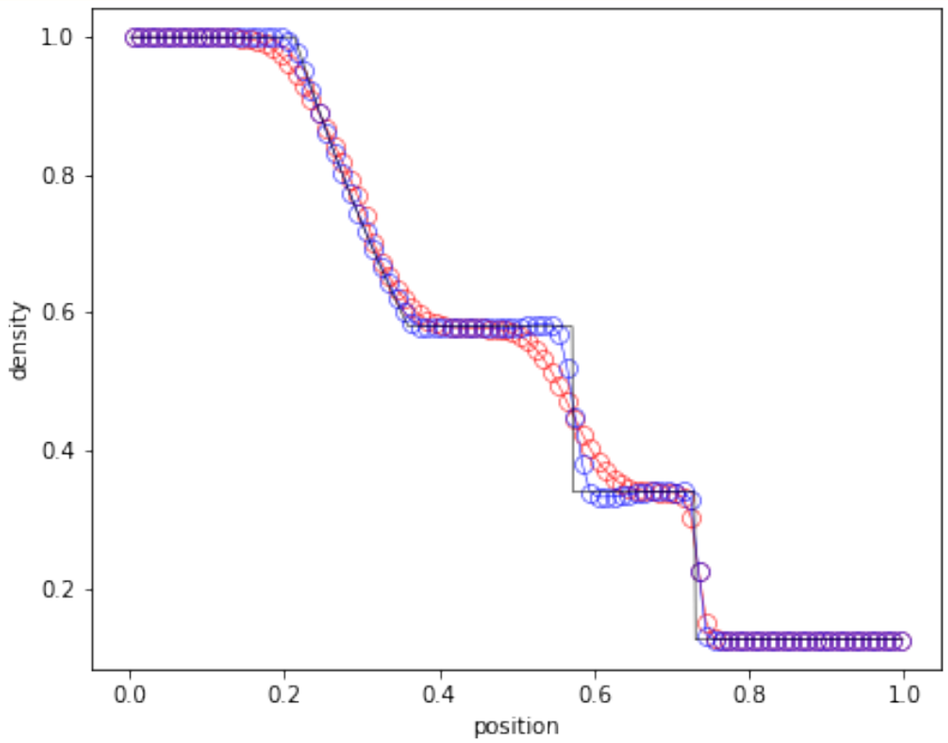


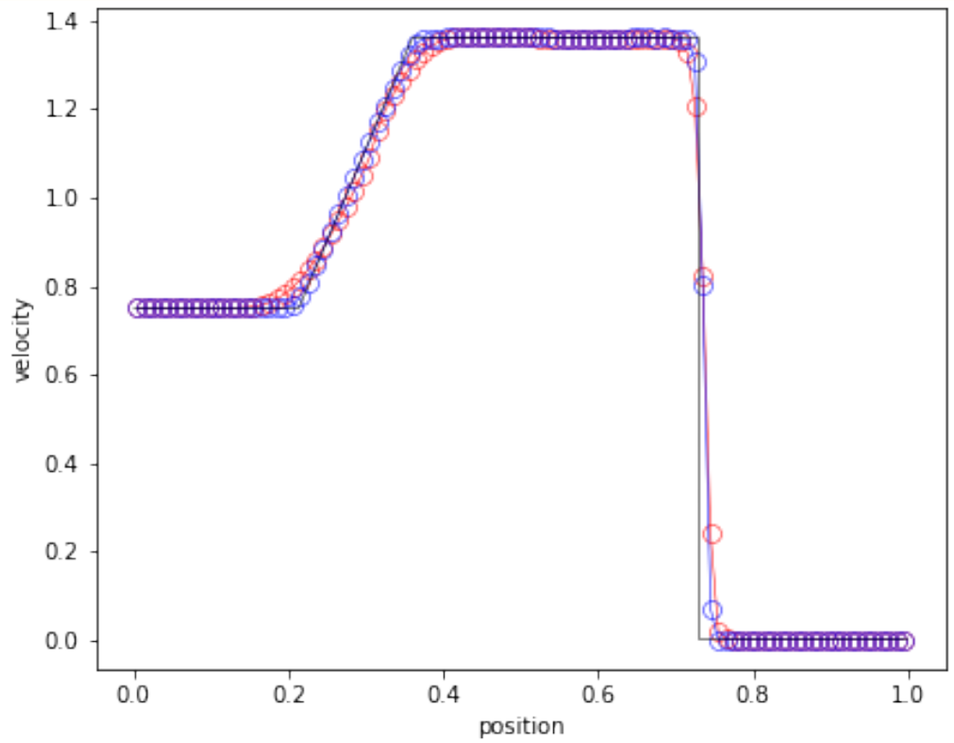


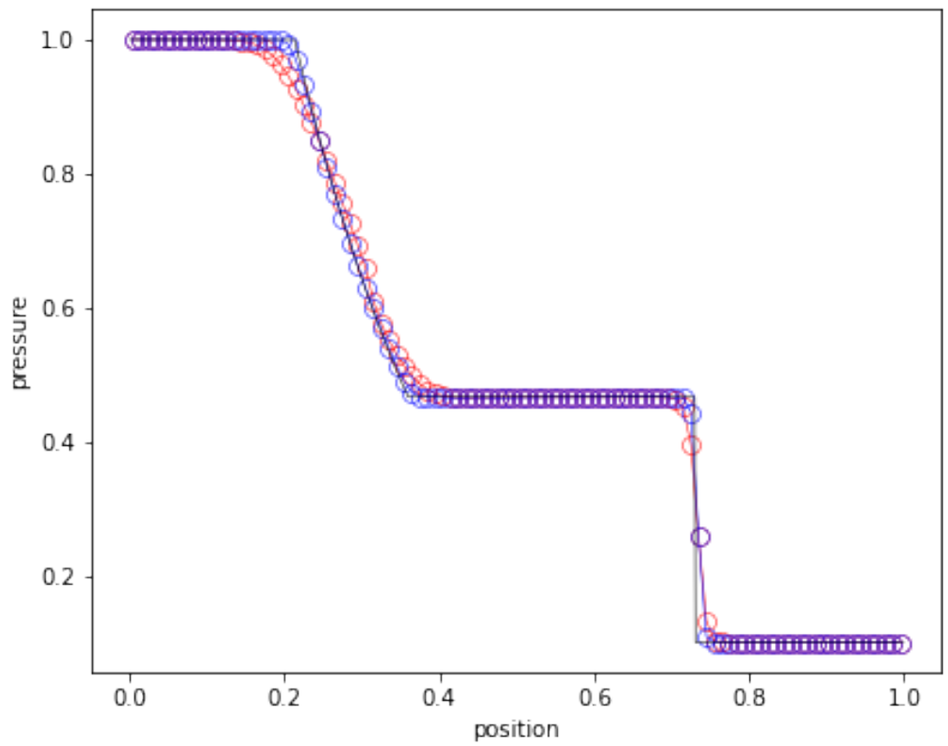


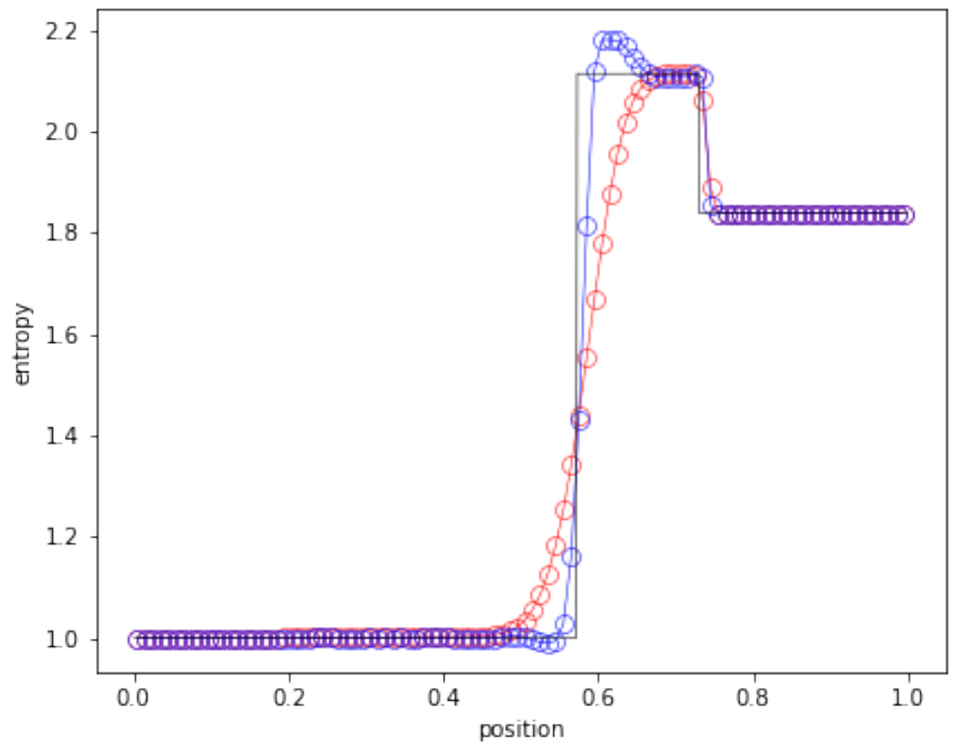
Using 100 cells and the HLLC Riemann solver, we compare first order Godunov in red with MUSCL-Hancock and the moncen slope in blue.

Numerical Examples: 4th Order Toro Test 1



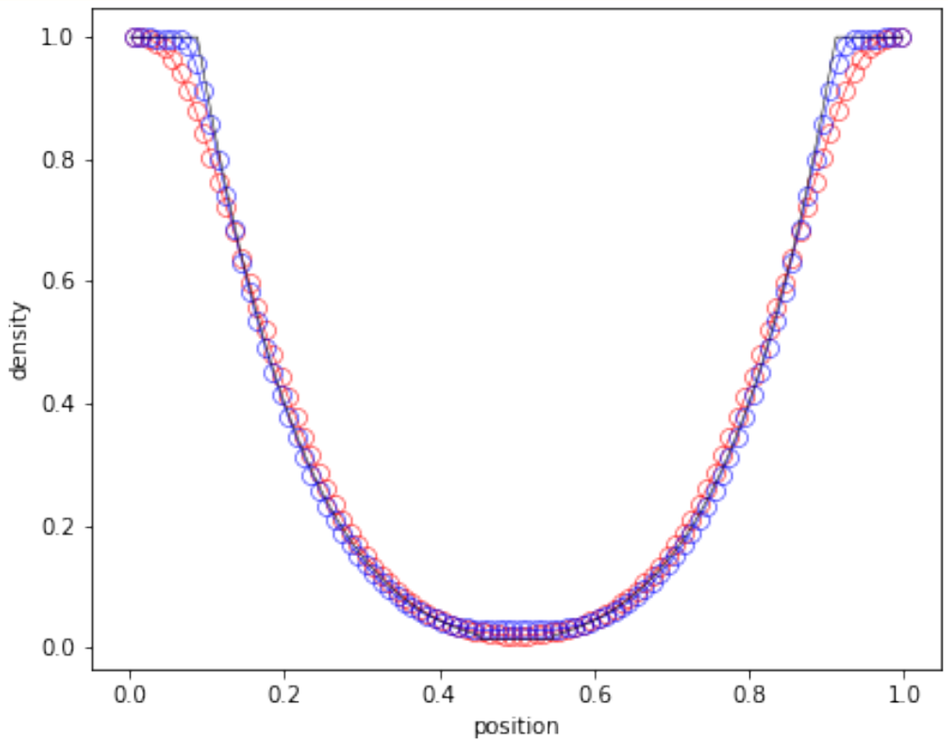


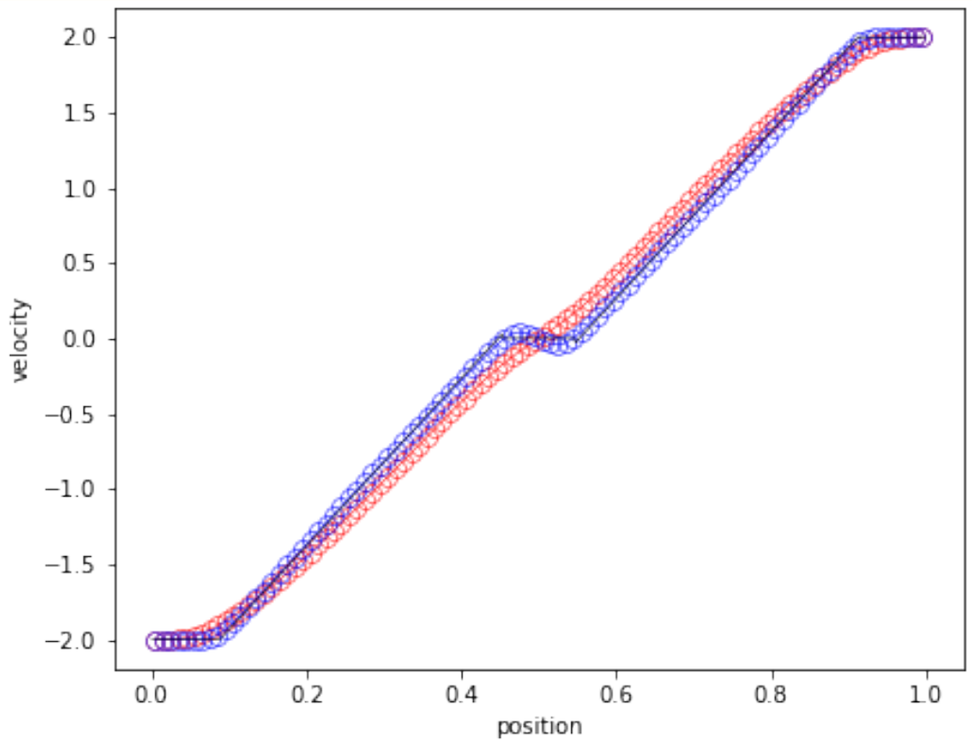


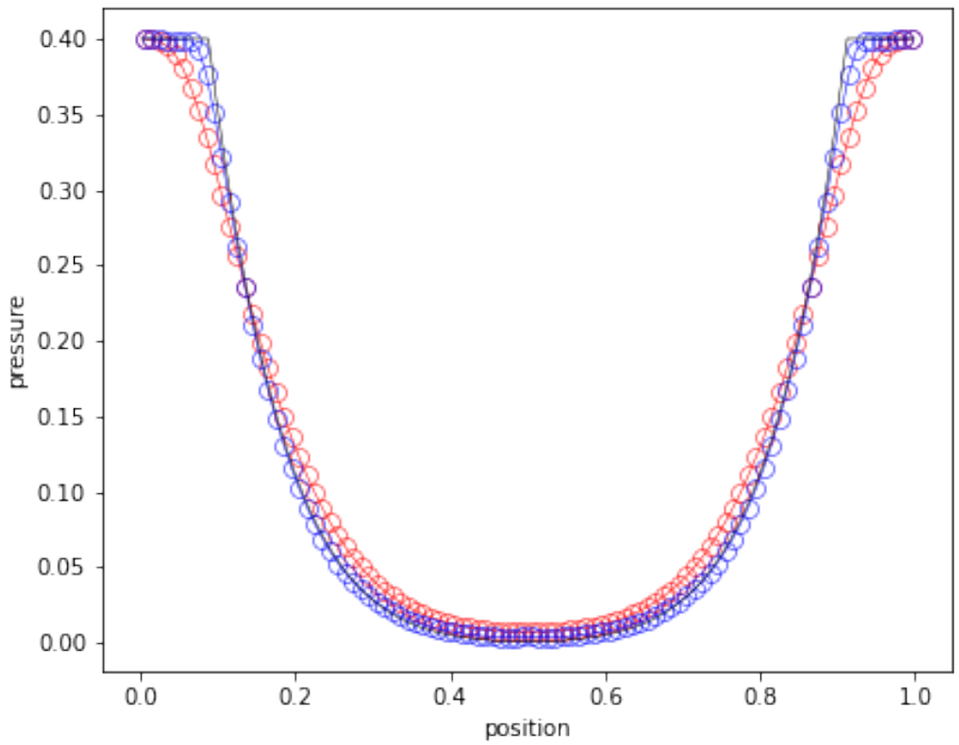


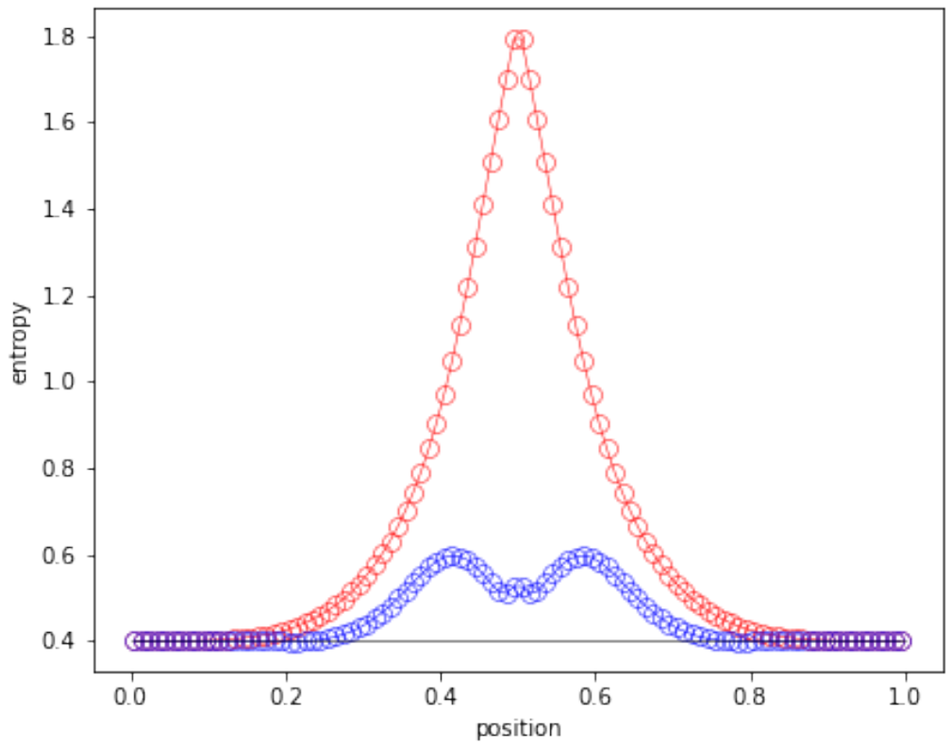
Using 100 cells and the HLLC Riemann solver, we compare first order Godunov in red with our fourth-order FV scheme in blue.

Numerical Examples: 4th Order Toro Test 2



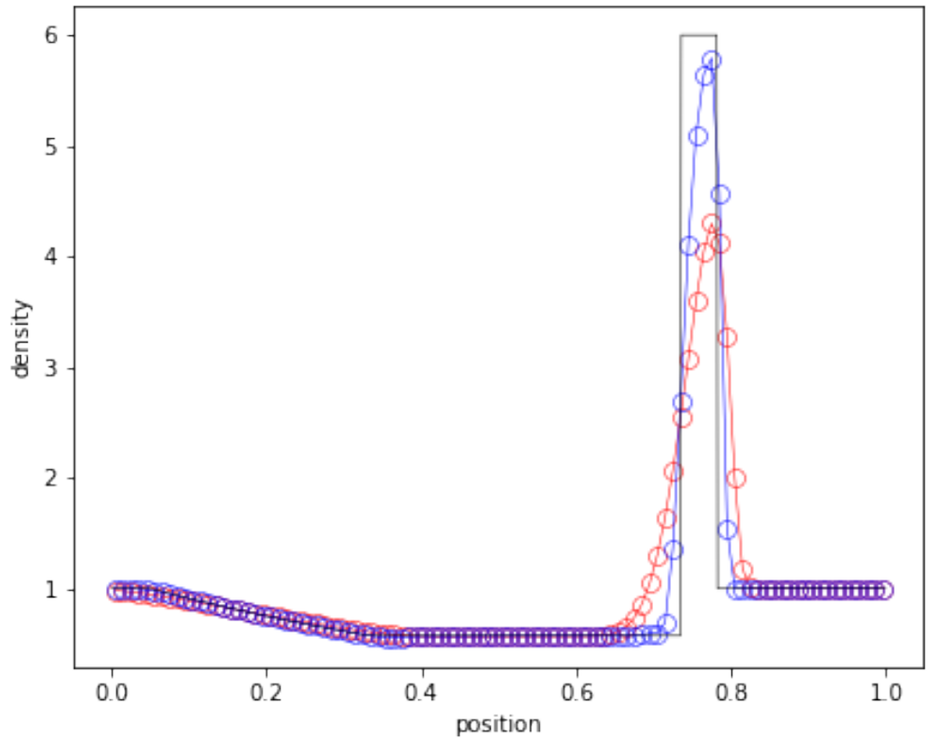


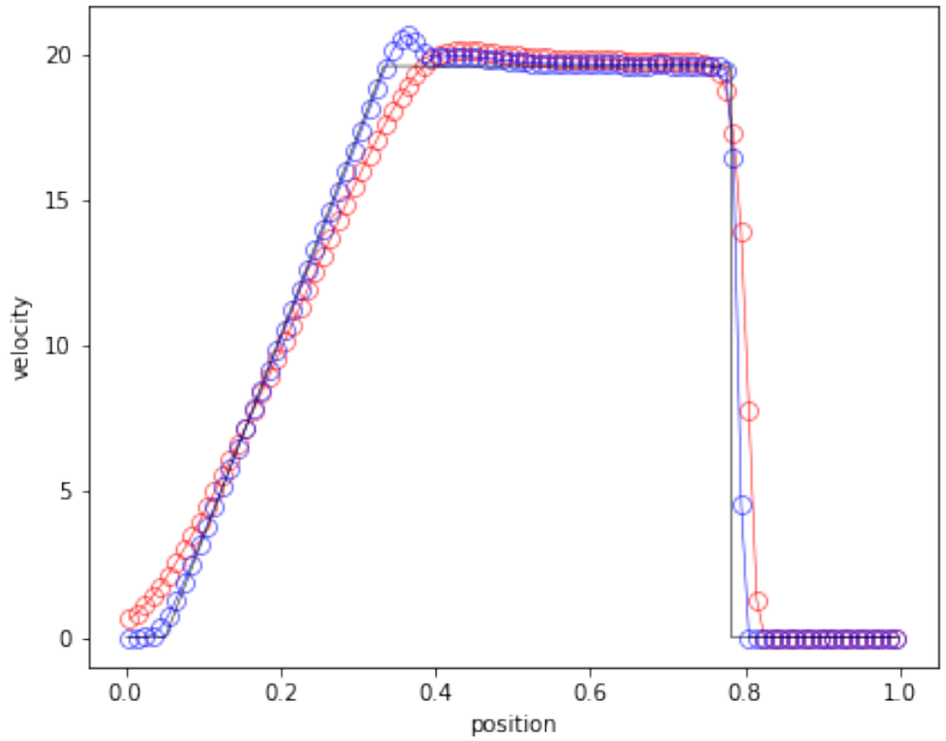


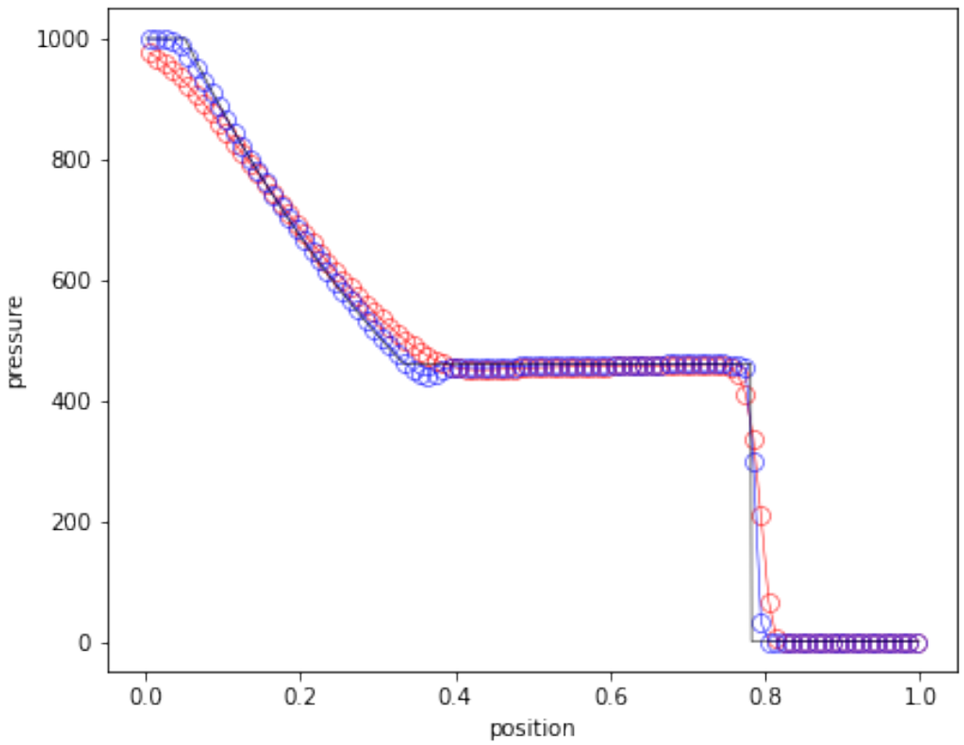


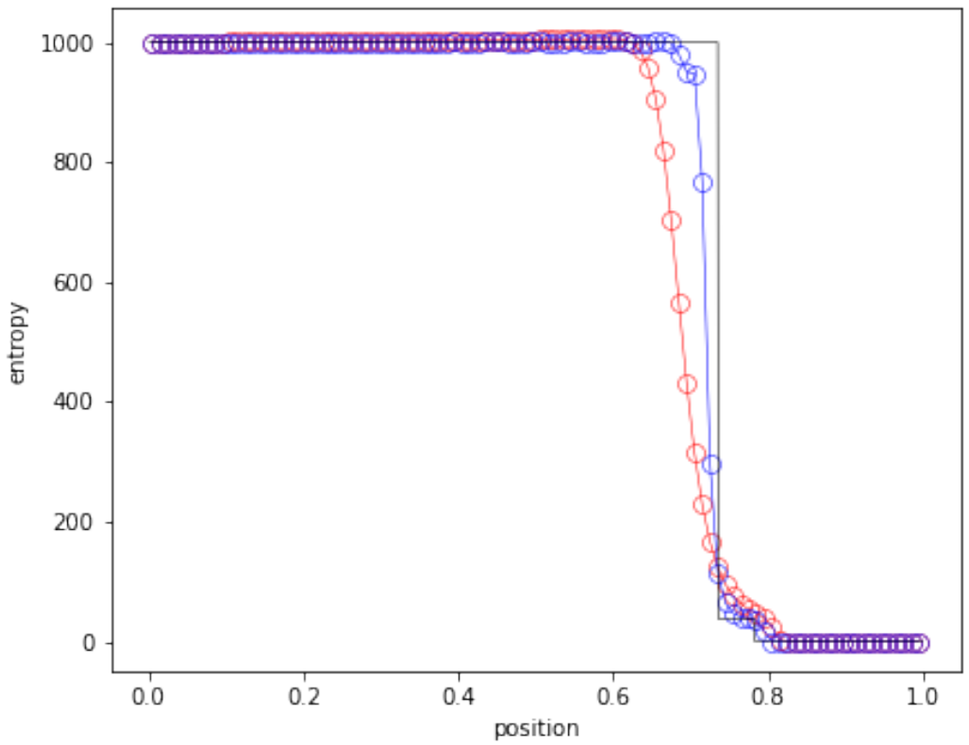
Using 100 cells and the HLLC Riemann solver, we compare first order Godunov in red with our fourth-order FV scheme in blue.

Numerical Examples: 4th Order Toro Test 3



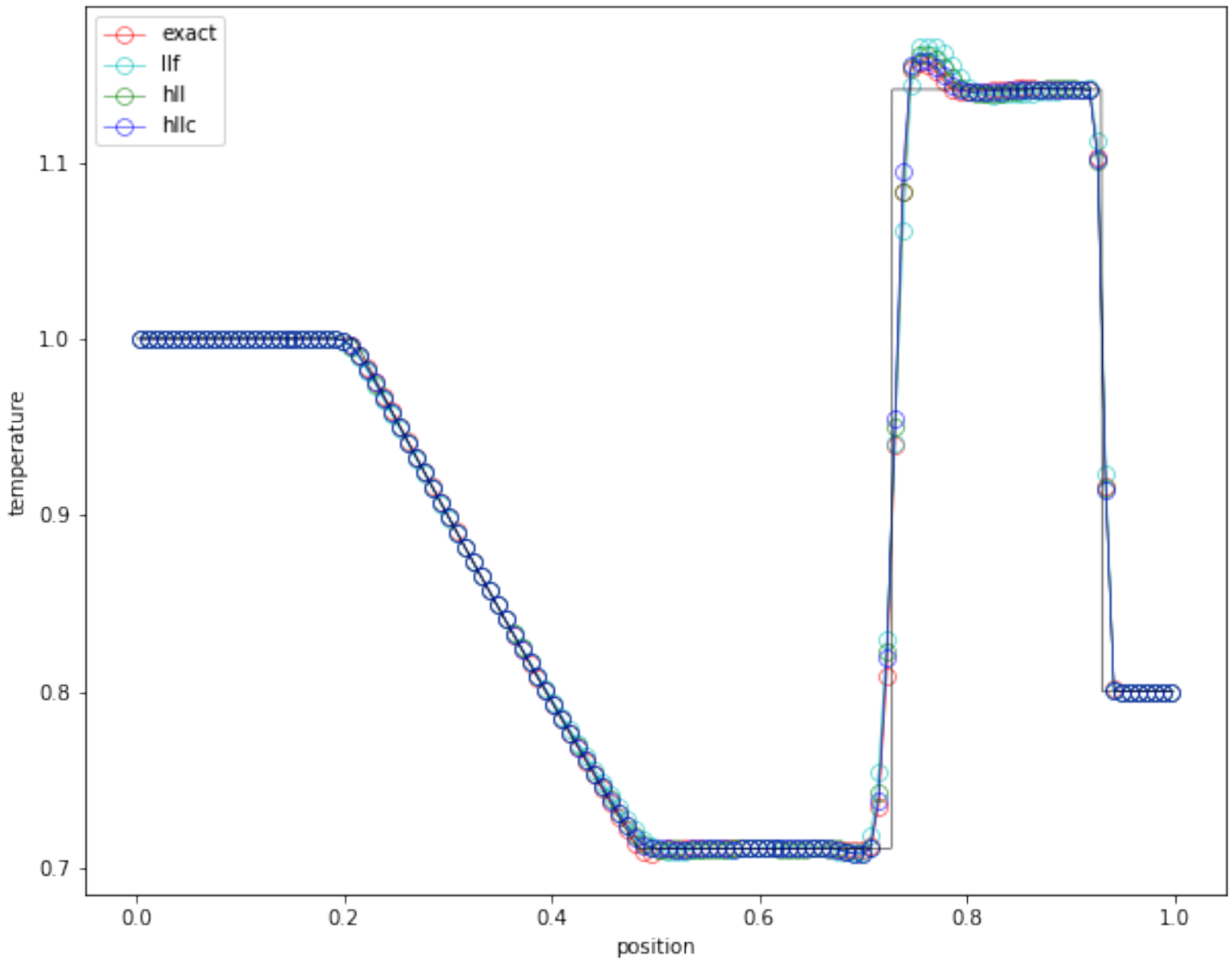




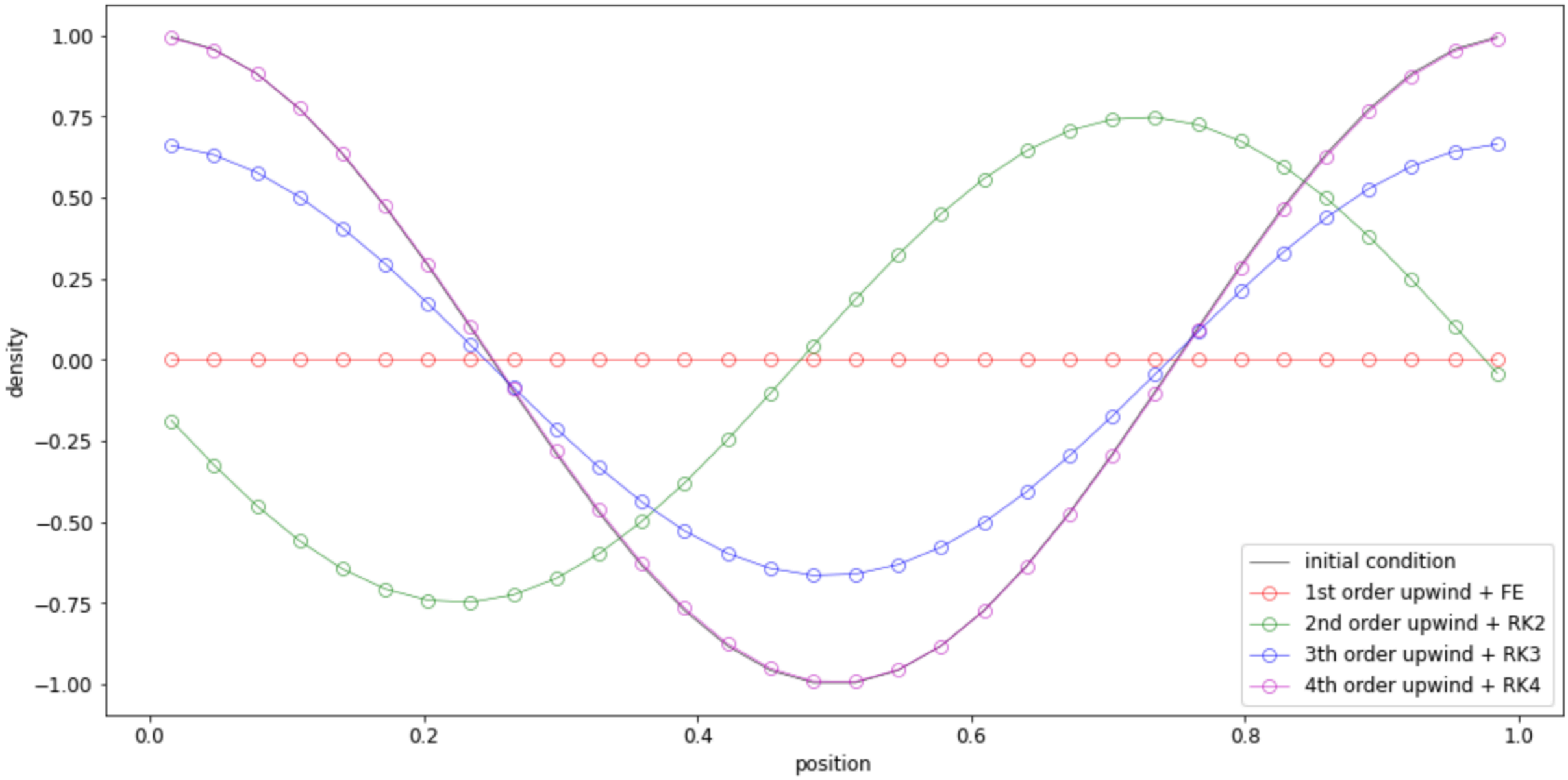


Using 100 cells and the HLLC Riemann solver, we compare first order Godunov in red with our fourth-order FV scheme in blue.

Effect of different Riemann solvers



Numerical Examples: Advection of a Sine Wave



Long time integration with $t_{\text{end}} = 100$.

Adaptive Mesh Refinement

A powerful technique to resolve efficiently discontinuities is based on adaptively refining the mesh in regions of the flow that require higher resolution.

The adaptive mesh geometry can be quite complex and the data structure need to be flexible. The traditional solution is to use an octree.

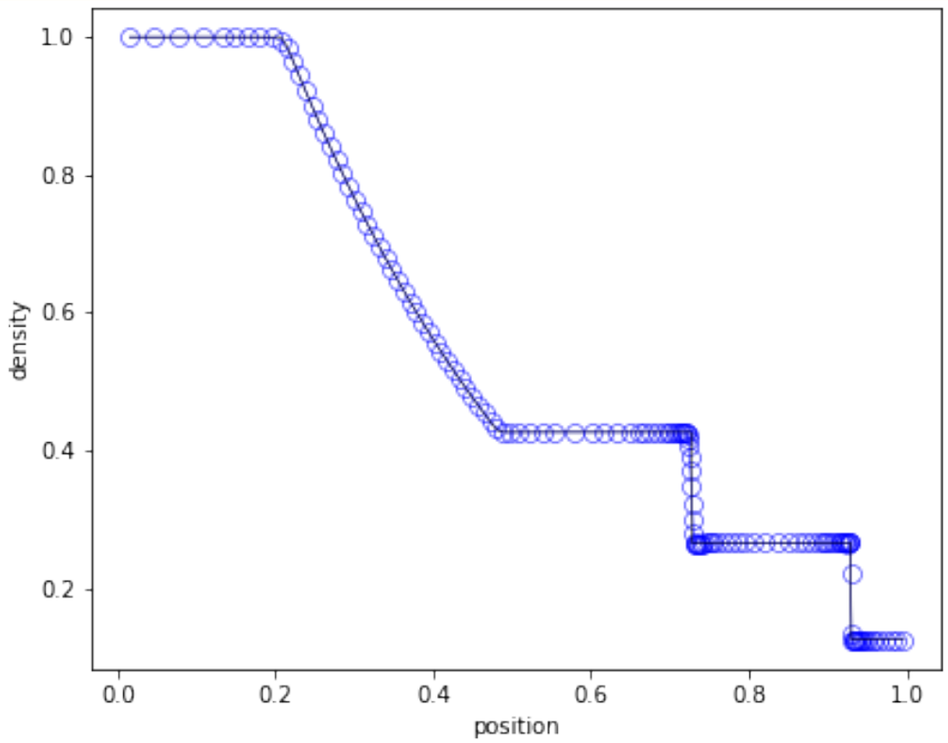
The algorithm is similar to multigrid, with proper prolongation and restriction operators. Time steps are also adaptively refined because the Courant stability condition depends on the local cell size.

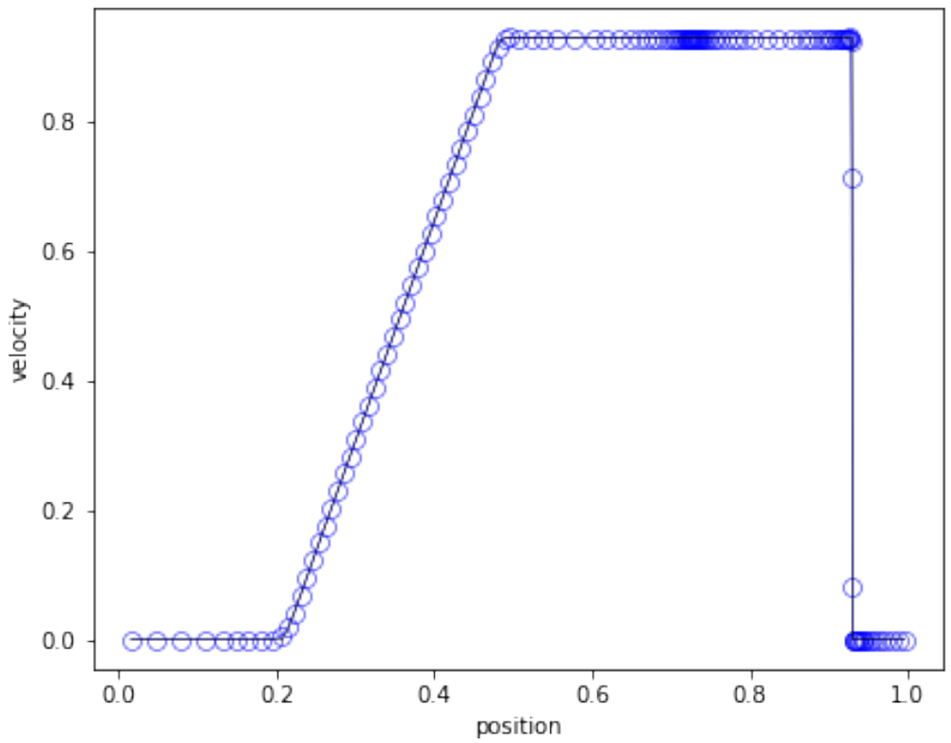
As we will see now, AMR is not the solution to all problems.
Ideally, one would like to use full h-p adaptivity:

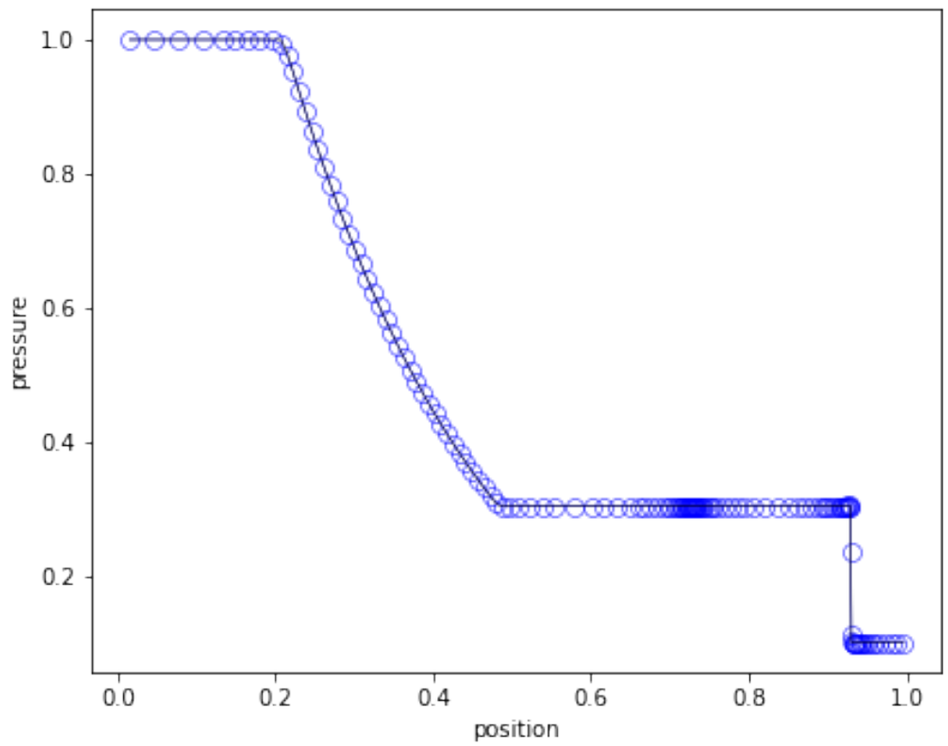
- h-adaptivity stands for cell refinement (AMR)
- p-adaptivity stands for polynomial degree (high order schemes)

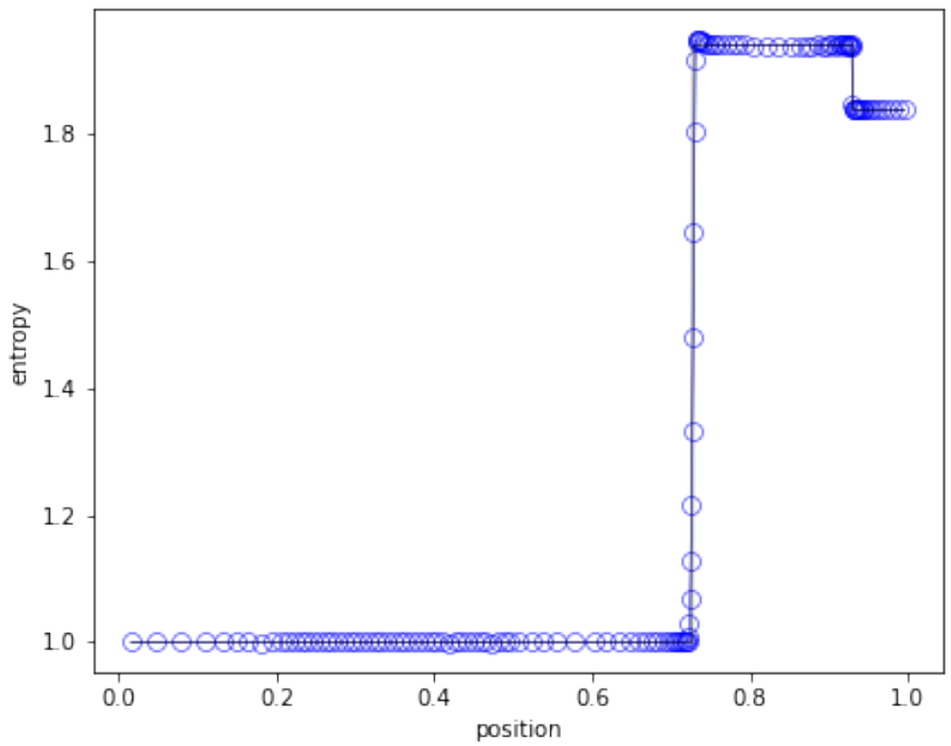
We see now some numerical examples obtained using the RAMSES code.

Numerical Examples: AMR Sod Test



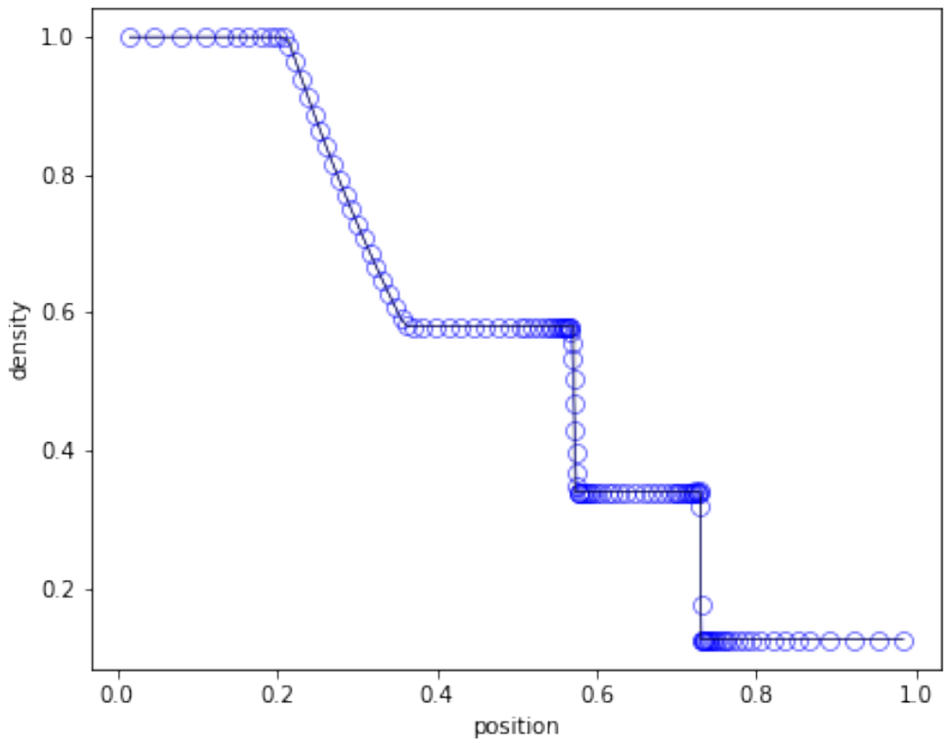


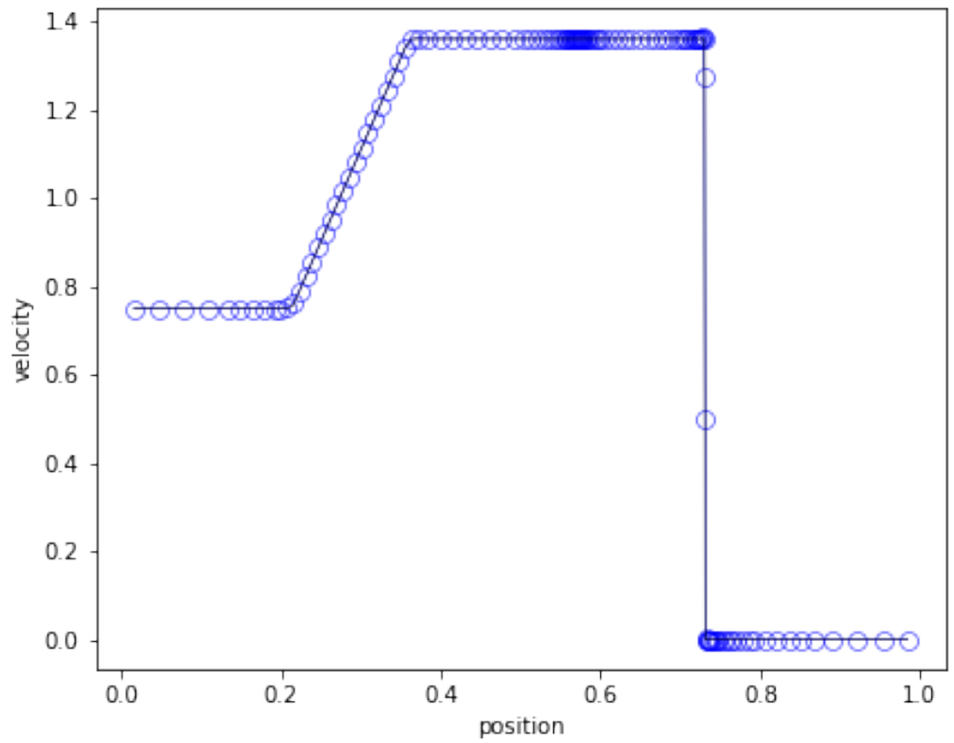


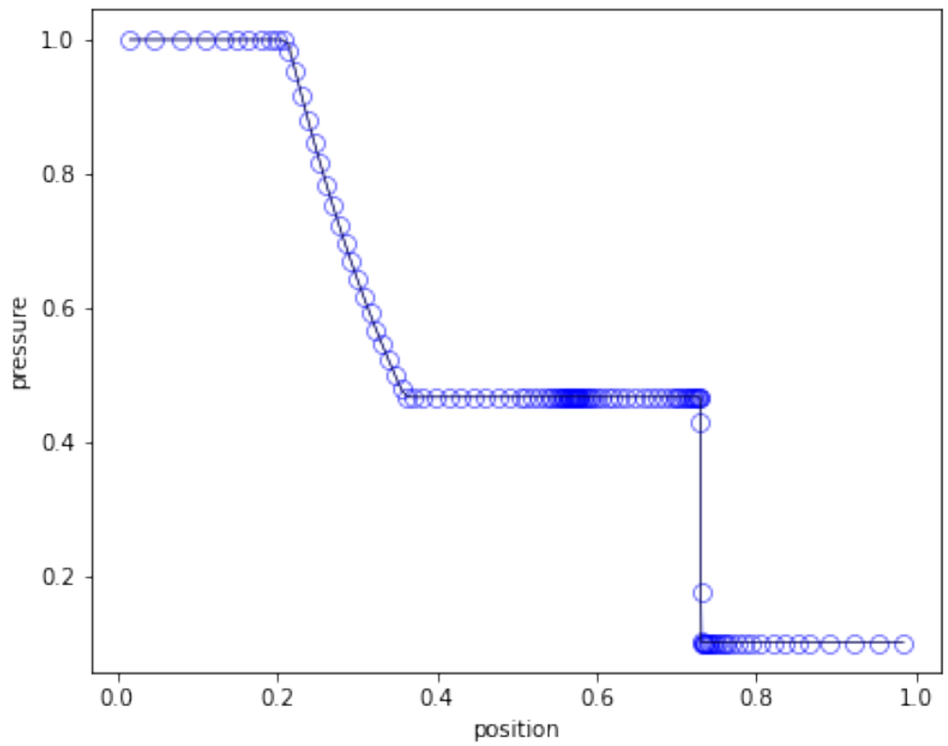


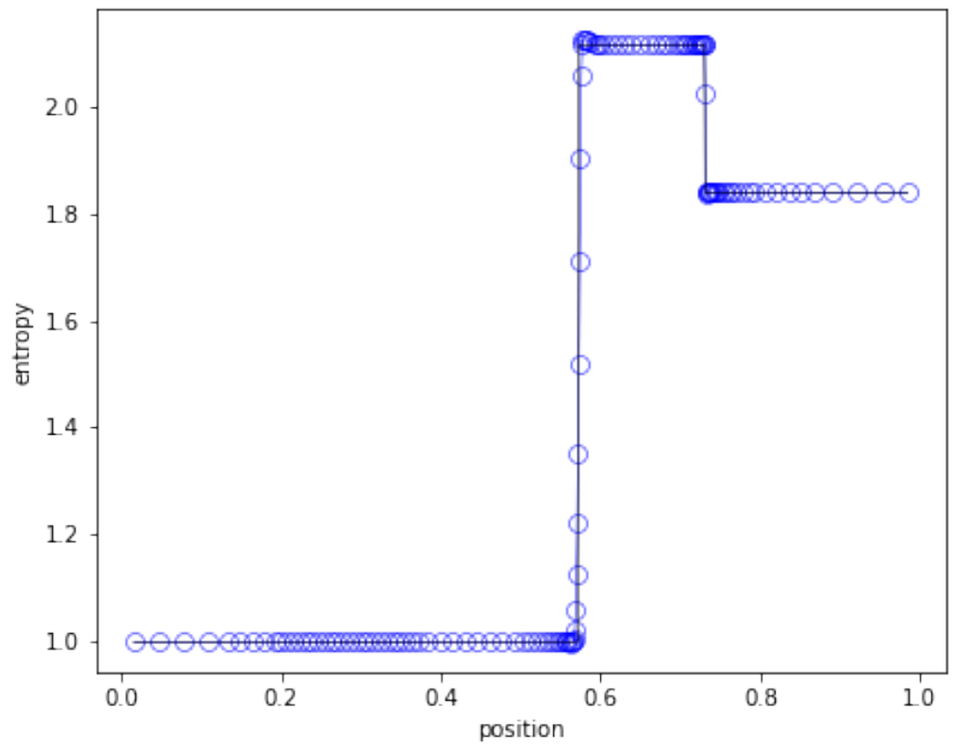
Using a refinement criterion based on the gradient of the primitive variables, we compare AMR with MUSCL + HLLC to the exact solution.

Numerical Examples: AMR Toro Test 1



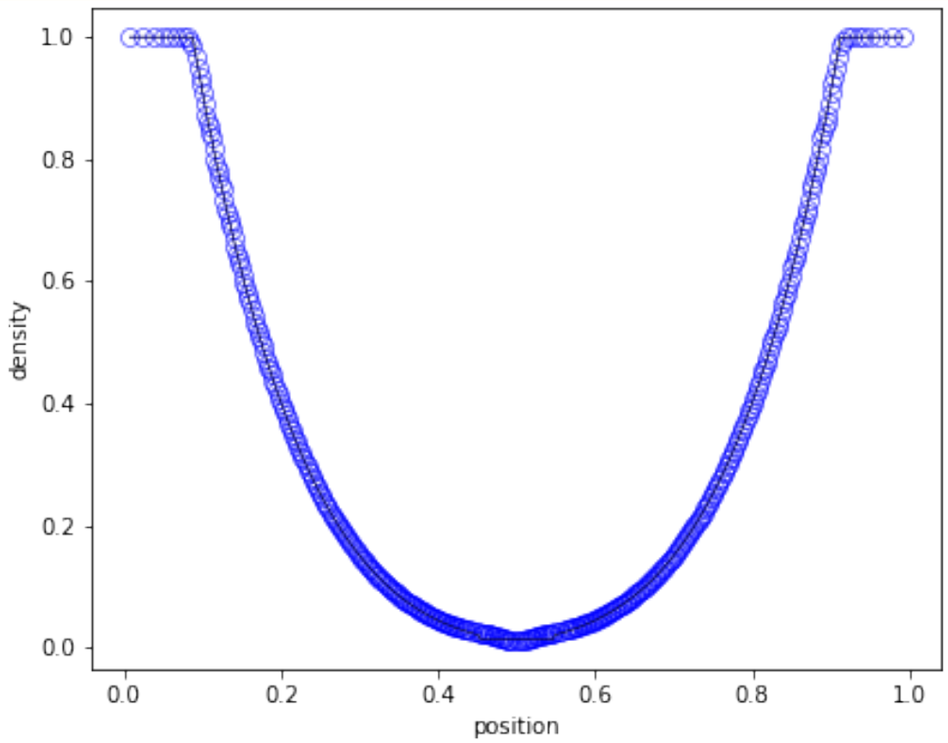


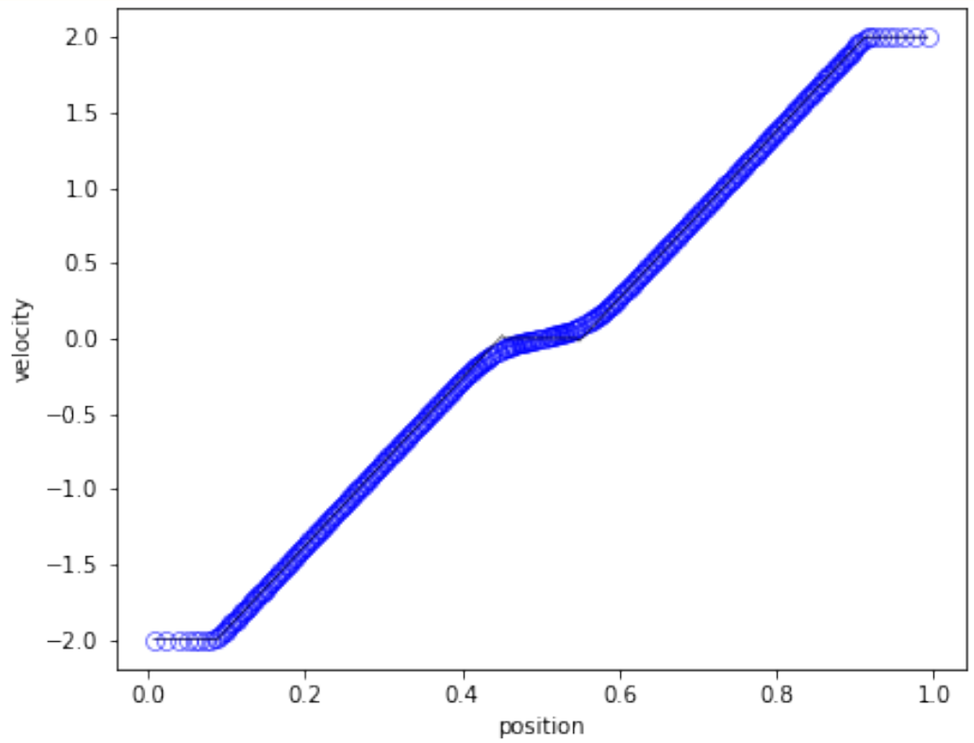


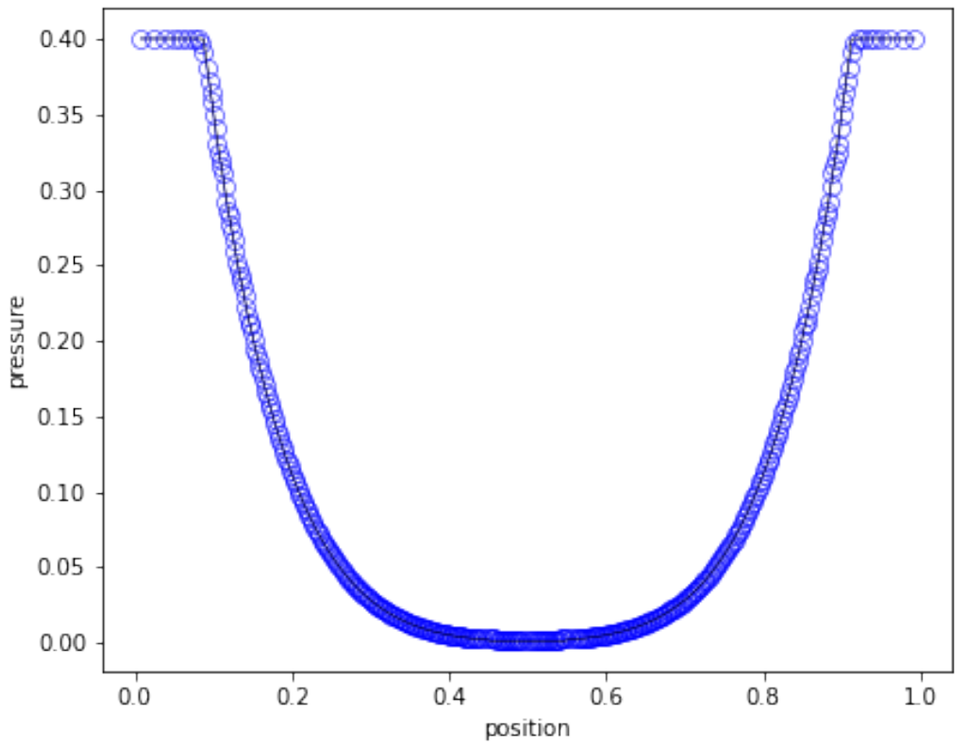


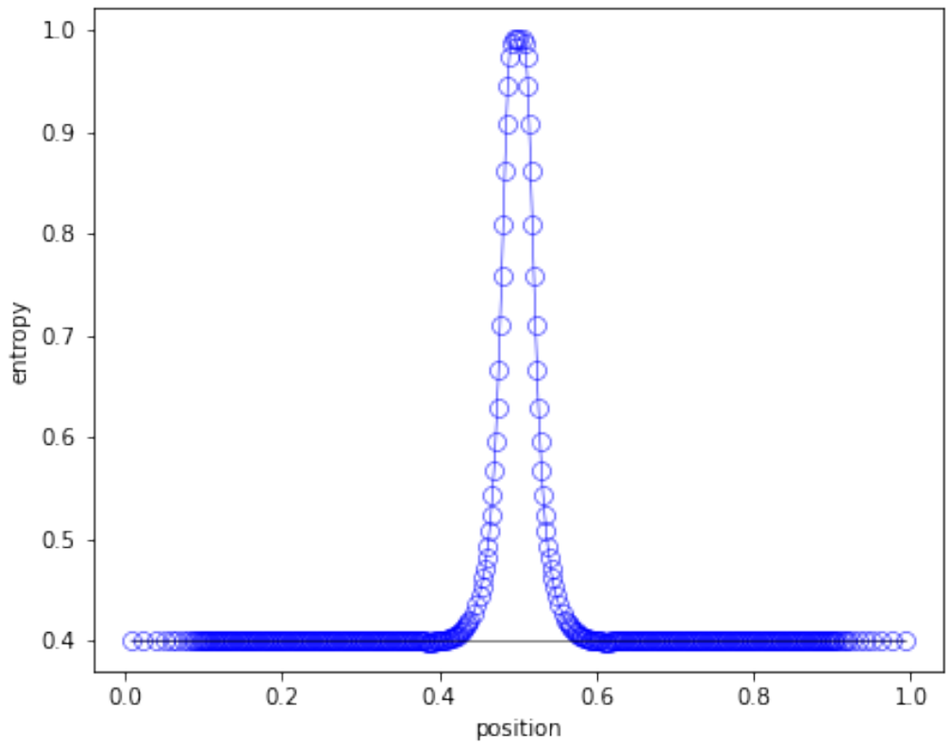
Using a refinement criterion based on the gradient of the primitive variables, we compare AMR with MUSCL + HLLC to the exact solution.

Numerical Examples: AMR Toro Test 2



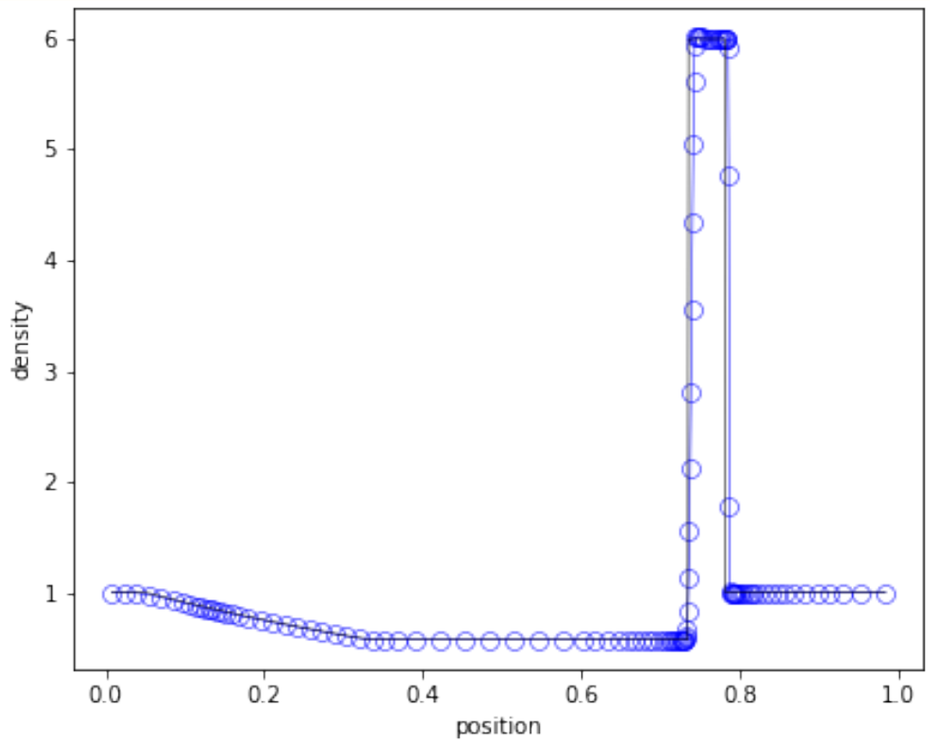


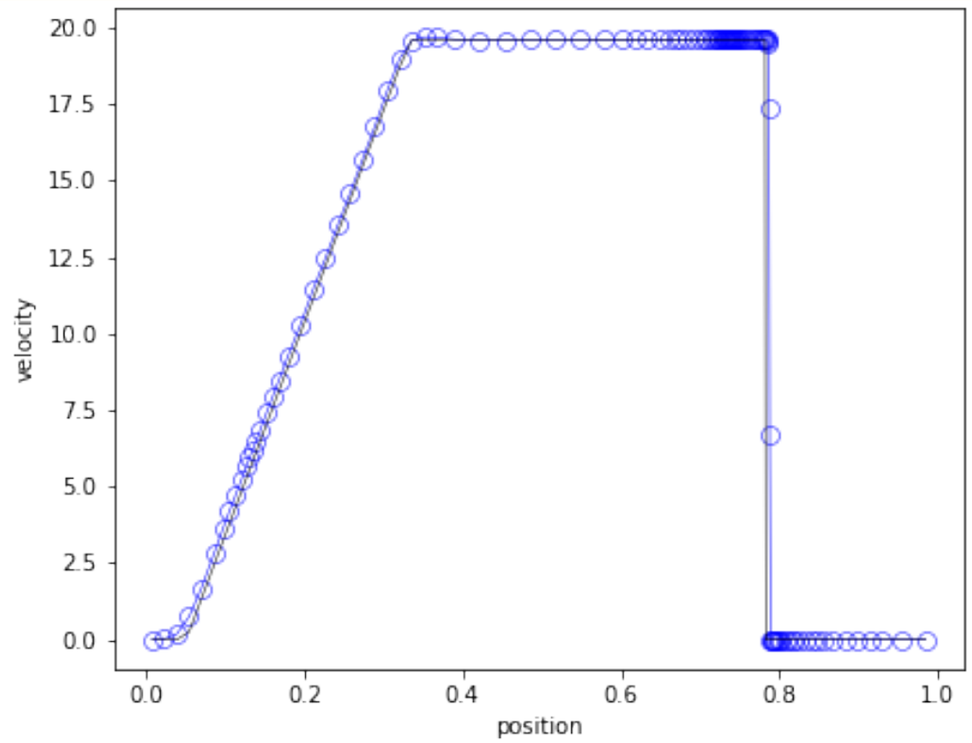


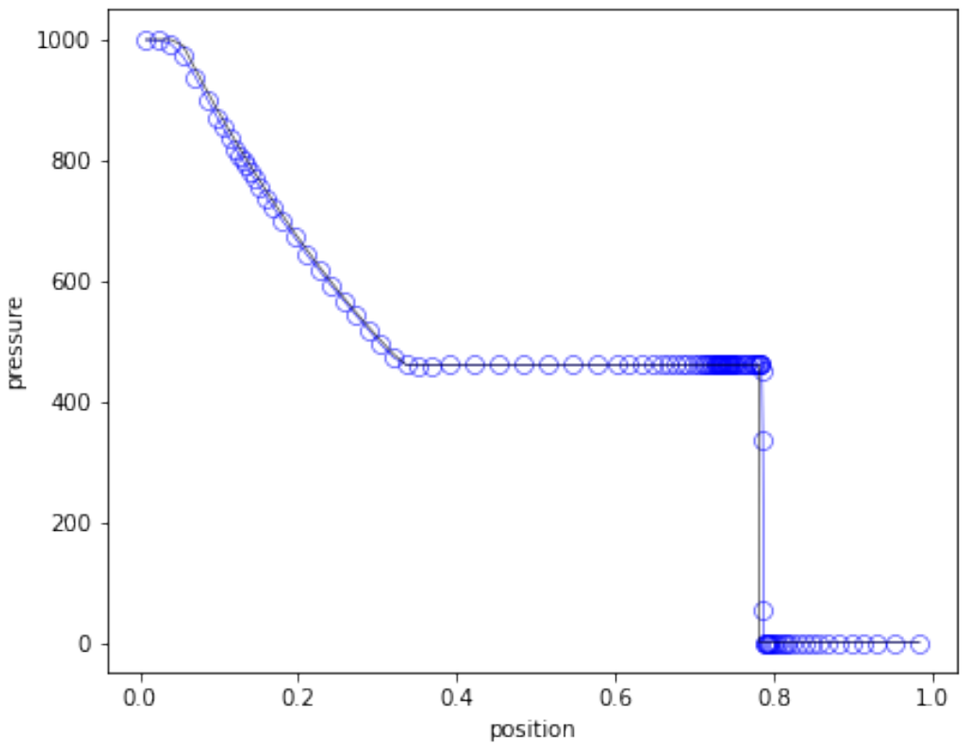


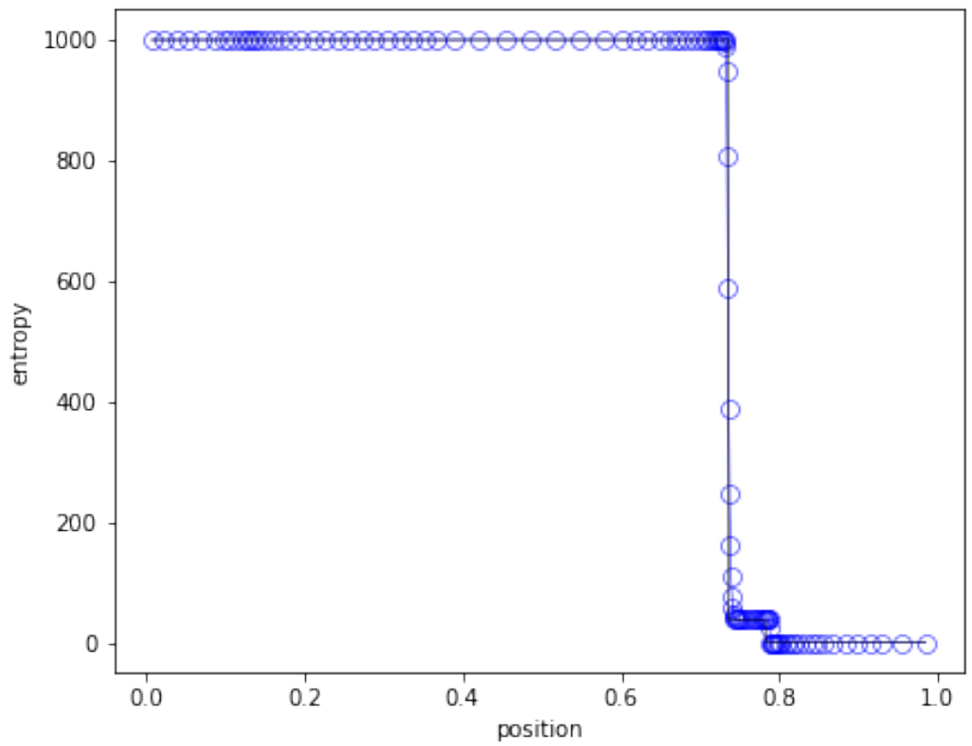
Using a refinement criterion based on the gradient of the primitive variables, we compare AMR with MUSCL + HLLC to the exact solution.

Numerical Examples: AMR Toro Test 3



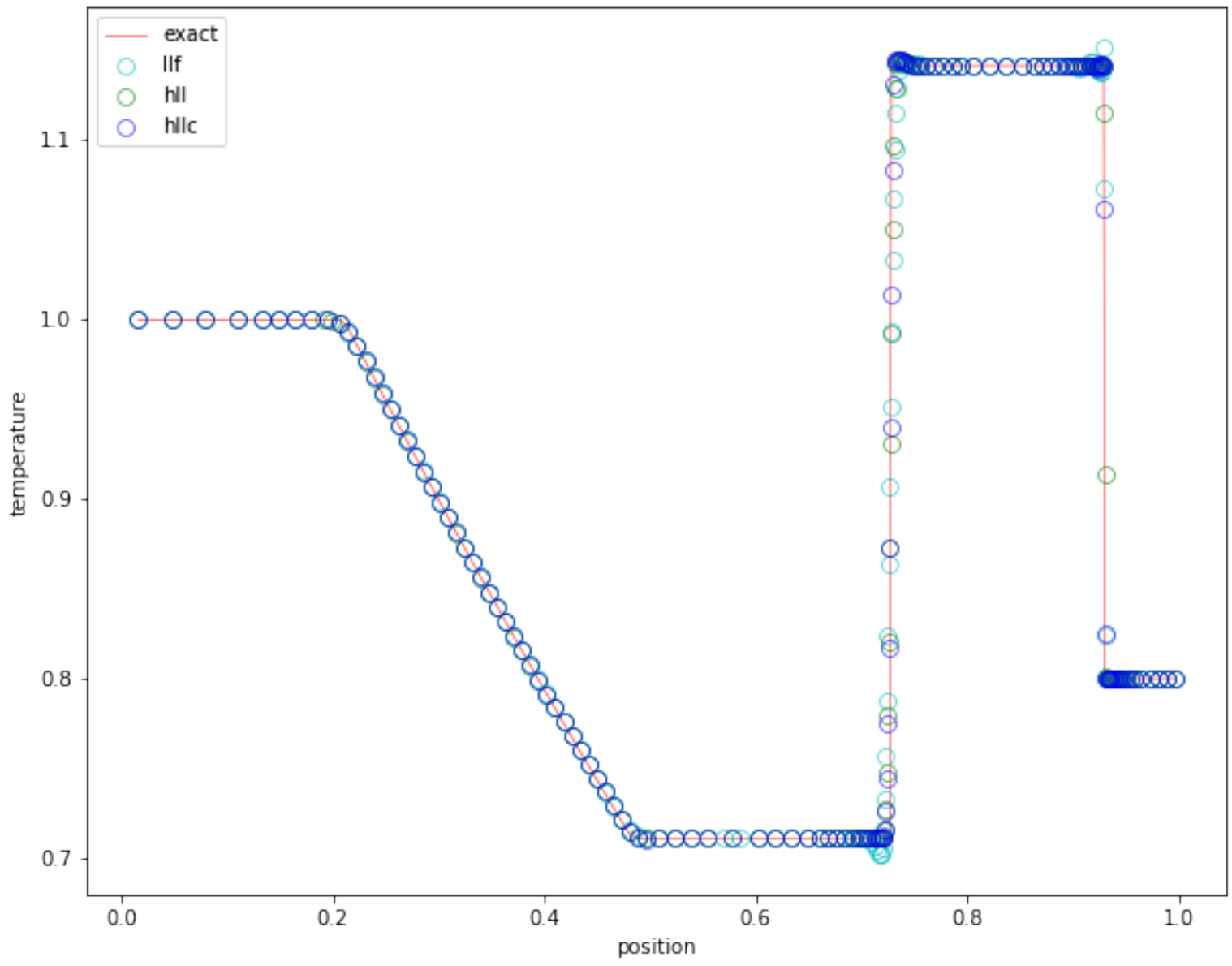






Using a refinement criterion based on the gradient of the primitive variables, we compare AMR with MUSCL + HLLC to the exact solution.

Effect of different Riemann solvers



Multidimensional Finite Volume Schemes

We discuss here the solution to the advection equation in 2D. The extension to nonlinear systems is quite straightforward, at least conceptually.

The advection equation in 2D writes (assuming here $a > 0$ and $b > 0$):

$$\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} + b\frac{\partial u}{\partial y} = 0.$$

We can very easily generalize everything we did in 1D to 2D or 3D.

The numerical solution is defined as the volume average within a square cell

$$\overline{\overline{u}}_{i,j} = \frac{1}{\Delta x \Delta y}\int_{x_{i-1/2}}^{x_{i+1/2}}\int_{y_{j-1/2}}^{y_{j+1/2}} u(x,y)\mathrm{d}x\mathrm{d}y.$$

Note the double bar notation.

Multidimensional Finite Volume Schemes

The conservative update in 2D is obtained using the divergence theorem:

$$\overline{\overline{u}}_{i,j}^{n+1} = \overline{\overline{u}}_{i,j}^{n} - a\frac{\Delta t}{\Delta x}(\overline{u}_{i+1/2,j}^{n+1/2} - \overline{u}_{i-1/2,j}^{n+1/2}) - b\frac{\Delta t}{\Delta y}(\overline{u}_{i,j+1/2}^{n+1/2} - \overline{u}_{i,j-1/2}^{n+1/2})$$

where the intercell face-averaged and time-averaged values are defined as

$$\overline{u}_{i+1/2,j}^{n+1/2} = \frac{1}{\Delta t \Delta y}\int_{t^n}^{t^{n+1}}\int_{y_{j-1/2}}^{y_{j+1/2}} u(x_{i+1/2}, y, t)\mathrm{d}y\mathrm{d}t$$

and

$$\overline{u}_{i,j+1/2}^{n+1/2} = \frac{1}{\Delta t \Delta x}\int_{t^n}^{t^{n+1}}\int_{x_{i-1/2}}^{x_{i+1/2}} u(x, y_{j+1/2}, t)\mathrm{d}x\mathrm{d}t.$$

Note that this 2D integral form is exact.

Multidimensional Finite Volume Schemes

We can generalize what we have learned in 1D and apply it here:

- we perform the time integration using a quadrature in time. We use the RK scheme that delivers the desired stability and/or order of accuracy

- we perform the interface integration using a quadrature in space. The Gauss-Legendre quadrature is the most efficient. It requires N+1 points in order to integrate exactly polynomials of degree 2N+1

  - for p=0 and p=1 we need only one quadrature point at $x = \frac{}{2}$

  - for p=2 and p=3 we need two quadrature points at $x = \frac{1 \pm 1/\sqrt{3}}{2}$

- at each quadrature point, we have discontinuous left and right states. We compute the flux using our favorite Riemann solver

First-Order Godunov Scheme in 2D

The first order method assumes that the solution is piecewise constant inside each cell. This corresponds to a polynomial degree $p = 0$.

For the quadrature in space, we use a single point at the center of the interface.

We use the upwind state as the solution to the Riemann problem there.

We have $$\overline{u}^{n+1/2}_{i+1/2,j} = \frac{1}{\Delta t}\int_{t^n}^{t^{n+1}} u_G(x_{i+1/2}, y_j, t)\mathrm{d}t = \overline{\overline{u}}^n_{i,j} \text{ because } a > 0$$

and $$\overline{u}^{n+1/2}_{i,j+1/2} = \frac{1}{\Delta t}\int_{t^n}^{t^{n+1}} u_G(x_i, y_{j+1/2}, t)\mathrm{d}t = \overline{\overline{u}}^n_{i,j} \text{ because } b > 0.$$

The fully discrete scheme writes:

$$\overline{\overline{u}}^{n+1}_{i,j} = \overline{\overline{u}}^n_{i,j} - a\frac{\Delta t}{\Delta x}(\overline{\overline{u}}^n_{i,j} - \overline{\overline{u}}^n_{i-1,j}) - b\frac{\Delta t}{\Delta y}(\overline{\overline{u}}^n_{i,j} - \overline{\overline{u}}^n_{i,j-1}).$$

First-Order Godunov Scheme in 2D

Regrouping terms, we obtain the 2D first order Godunov scheme :

$$\overline{\overline{u}}_{i,j}^{n+1} = \overline{\overline{u}}_{i,j}^{n}(1 - a\frac{\Delta t}{\Delta x} - b\frac{\Delta t}{\Delta y}) + a\frac{\Delta t}{\Delta x}\overline{\overline{u}}_{i-1,j}^{n} + b\frac{\Delta t}{\Delta y}\overline{\overline{u}}_{i,j-1}^{n}.$$

The maximum principle preserving condition writes: $a\frac{\Delta t}{\Delta x} + b\frac{\Delta t}{\Delta y} < 1.$

This is also the Courant stability condition for this scheme.

The modified equation for the 2D first-order Godunov scheme writes:

$$\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} + b\frac{\partial u}{\partial y} = \frac{a\Delta x}{2}(1 - a\frac{\Delta t}{\Delta x})\frac{\partial^2 u}{\partial x^2} + \frac{b\Delta y}{2}(1 - b\frac{\Delta t}{\Delta y})\frac{\partial^2 u}{\partial y^2} - ab\Delta t\frac{\partial^2 u}{\partial x \partial y}$$

First-Order Godunov Scheme in 2D

It can be written in matrix form as:

$$\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} + b\frac{\partial u}{\partial y} = \nabla u^T \mathbb{M} \nabla u \text{ where } \nabla u = [\frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}]^T.$$

$$\text{where matrix } \mathbb{M} = \begin{bmatrix} \frac{a\Delta x}{2}(1 - a\frac{\Delta t}{\Delta x}) & -\frac{ab}{2}\Delta t \\ -\frac{ab}{2}\Delta t & \frac{b\Delta y}{2}(1 - b\frac{\Delta t}{\Delta y}) \end{bmatrix}.$$

We have $$\det(\mathbb{M}) = \frac{ab\Delta x\Delta y}{4}(1 - a\frac{\Delta t}{\Delta x} - b\frac{\Delta t}{\Delta y}) > 0$$

and $$\text{tr}(\mathbb{M}) = \frac{a\Delta x}{2}(1 - a\frac{\Delta t}{\Delta x}) + \frac{b\Delta y}{2}(1 - b\frac{\Delta t}{\Delta y}) > 0$$

iff $a\frac{\Delta t}{\Delta x} + b\frac{\Delta t}{\Delta y} < 1.$

The 2D Courant condition guarantees positive (anisotropic) diffusion coefficients.

Higher-Order Godunov Scheme in 2D

We can also generalize our 1D high-order FV schemes to 2D and 3D equivalent schemes.

The surface average flux is computed using a spatial Gauss quadrature at each time t^s of the different RK stages:

$$\overline{u}^s_{i+1/2,j} = \sum_{k=1}^{q} w_k u_G(x_{i+1/2}, y^k_j, t^s) \text{ and } \overline{u}^s_{i,j+1/2} = \sum_{k=1}^{q} w_k u_G(x^k_i, y_{j+1/2}, t^s)$$

For the limiter function, we use a larger set to compute the maximum and the minimum of the solution:

$$M = \max(\overline{\overline{w}}^n_{i-1,j-1}, \overline{\overline{w}}^n_{i,j-1}, \overline{\overline{w}}^n_{i+1,j-1}, \overline{\overline{w}}^n_{i-1,j}, \overline{\overline{w}}^n_{i,j}, \overline{\overline{w}}^n_{i+1,j}, \overline{\overline{w}}^n_{i-1,j+1}, \overline{\overline{w}}^n_{i,j+1}, \overline{\overline{w}}^n_{i+1,j+1})$$

$$m = \min(\overline{\overline{w}}^n_{i-1,j-1}, \overline{\overline{w}}^n_{i,j-1}, \overline{\overline{w}}^n_{i+1,j-1}, \overline{\overline{w}}^n_{i-1,j}, \overline{\overline{w}}^n_{i,j}, \overline{\overline{w}}^n_{i+1,j}, \overline{\overline{w}}^n_{i-1,j+1}, \overline{\overline{w}}^n_{i,j+1}, \overline{\overline{w}}^n_{i+1,j+1})$$

For the maximum and minimum of the reconstructed solution, we use all relevant interface points and interior points of the corresponding Gauss-Lobatto quadratures.

Advanced Methods for Hyperbolic PDE

We have presented the MUSCL-Hancock scheme for the Euler equations

- conservative
- very good dispersion properties (equivalent to the Fromm scheme)
- moncen slope limiter
- smooth extrema detection
- HLLC Riemann solver is better

Advanced Methods for Hyperbolic PDE

We have presented the 4th order FV scheme for the Euler equations

- conservative
- very high order (good for long time integration of smooth solutions)
- RK3 is enough but RK4 is better
- maximum principle preserving limiter
- smooth extrema detection
- LLF Riemann solver is usually enough

Advanced Methods for Hyperbolic PDE

We have presented the Adaptive Mesh Refinement scheme for the Euler equations

- conservative
- very high resolution locally
- based on MUSCL-Hancock
- HLLC Riemann solver is better
- not so good for smooth solutions
- used in the RAMSES code