

# Mesh Hydrodynamics Results

Mladen Ivkovic

Laboratory of Astrophysics  
École Polytechnique Fédérale de Lausanne

April 21, 2020

# Contents

<b>1 Notation</b>	<b>3</b>
<b>2 Advection</b>	<b>4</b>
2.1 Piecewise Constant . . . . .	4
2.2 Piecewise Linear . . . . .	14
2.3 Order of Convergence . . . . .	22
2.4 Conclusions . . . . .	33
2.5 Future Debugging Hints . . . . .	38
2.5.1 When you're using downwind differencing . . . . .	38
2.5.2 When the CFL condition is violated . . . . .	44
<b>3 Slope and Flux Limiters</b>	<b>49</b>
3.0.1 Effects on linear advection . . . . .	49
<b>4 Riemann Solvers</b>	<b>51</b>
4.1 Approximate Riemann Solutions . . . . .	51
4.2 Conclusions . . . . .	54
<b>5 Godunov's Method</b>	<b>55</b>
5.1 Solving Riemann Problems using Godunov's Method . . . . .	55
5.2 Solving Vacuum Riemann Problems using Godunov's Method . . . . .	57
5.2.1 Left Vacuum State . . . . .	57
5.2.2 Vacuum Generating ICs . . . . .	59
5.3 Order of Convergence Study . . . . .	60
5.4 Conclusions . . . . .	62
<b>6 WAF Method</b>	<b>64</b>
6.1 Solving Riemann Problems using the WAF Method . . . . .	64
6.2 Solving Vacuum Riemann Problems using WAF Method . . . . .	66
6.2.1 Left Vacuum State . . . . .	66
6.2.2 Vacuum Generating ICs . . . . .	67
6.3 Order of Convergence Study . . . . .	68
6.4 Conclusions . . . . .	70
<b>7 MUSCL-Hancock Method</b>	<b>72</b>
7.1 Solving Riemann Problems using the MUSCL-Hancock Method . . . . .	72
7.2 Solving Vacuum Riemann Problems using MUSCL-Hancock Method . . . . .	74
7.3 Order of Convergence Study . . . . .	76
7.4 Conclusions . . . . .	78
<b>References</b>	<b>80</b>

# 1 Notation

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

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

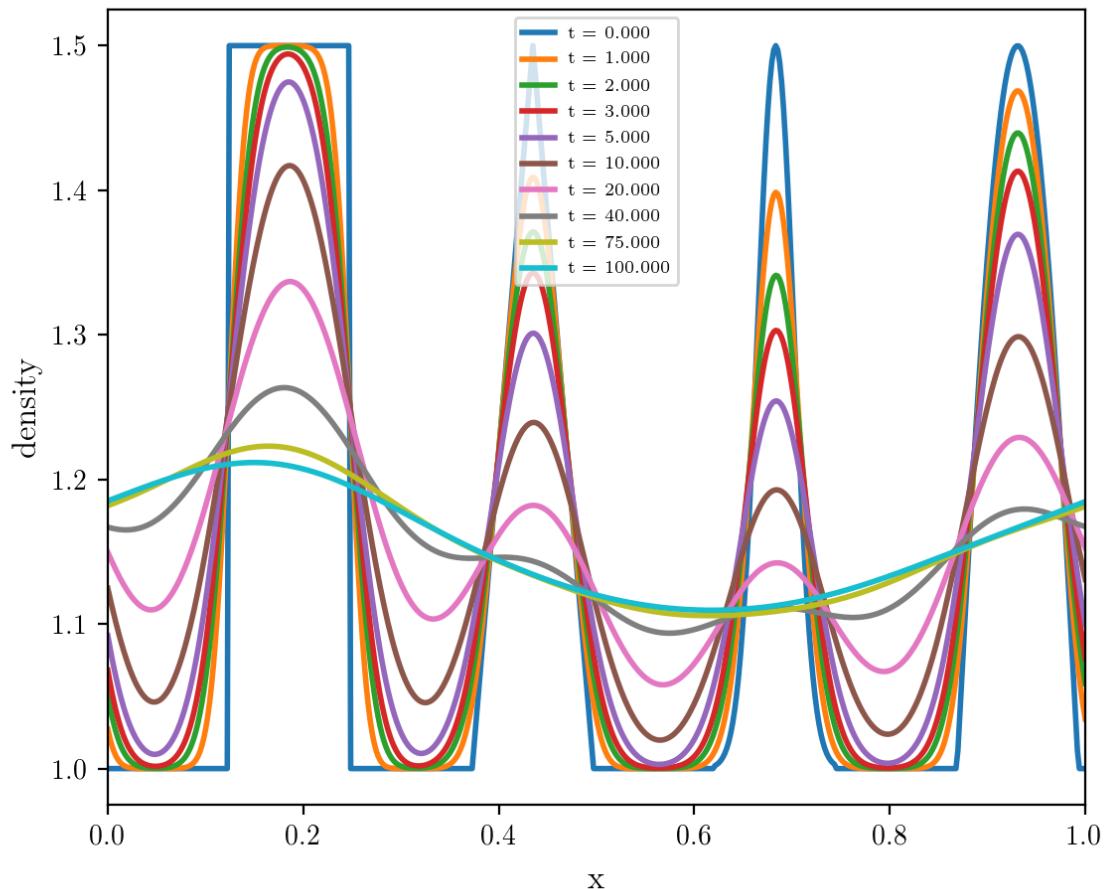
The lower left corner has indices  $(0, 0)$  in 2D. In 1D, index 0 also represents the leftmost cell.

We have:

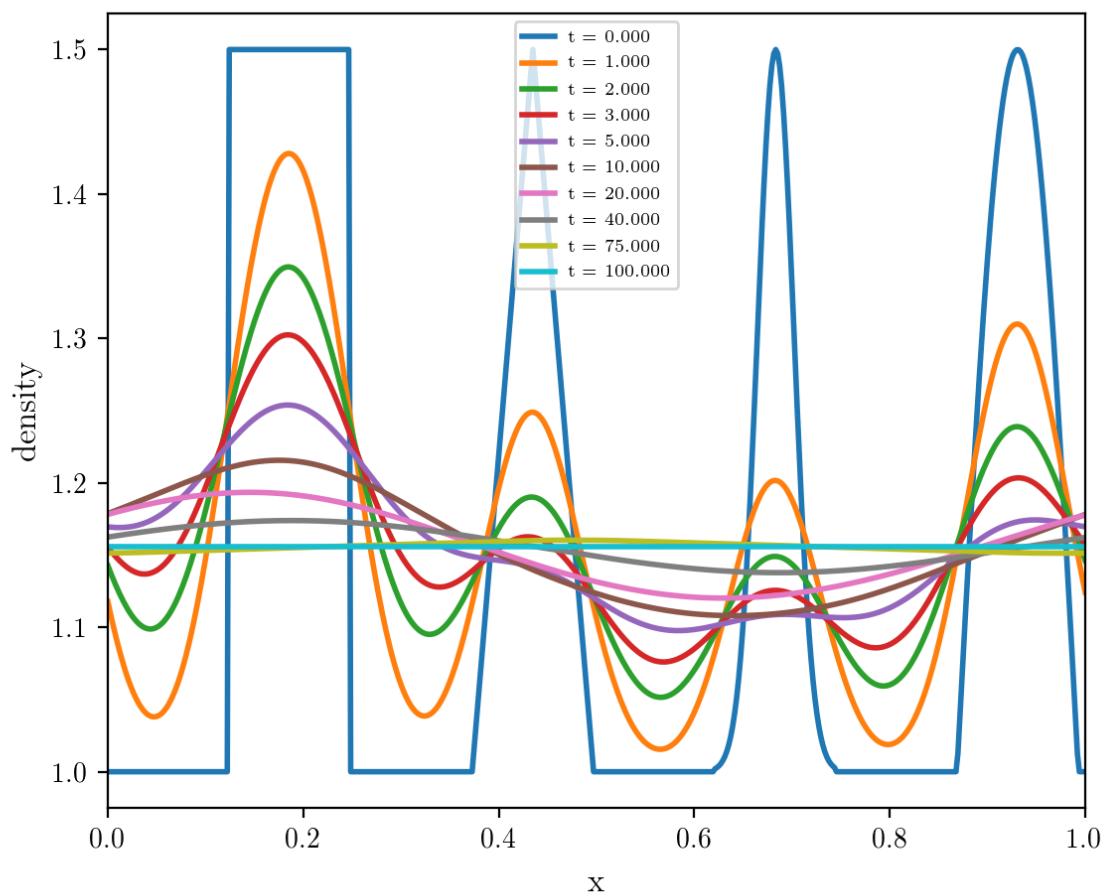
- integer subscript: Value of a quantity at the cell, i.e. the center of the cell. Example:  $\mathbf{U}_i$ ,  $\mathbf{U}_{i-2}$  or  $\mathbf{U}_{i,j+1}$  for 2D.
- non-integer subscript: Value at the cell faces, e.g.  $\mathbf{F}_{i-1/2}$  is the flux at the interface between cell  $i$  and  $i -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

## 2 Advection

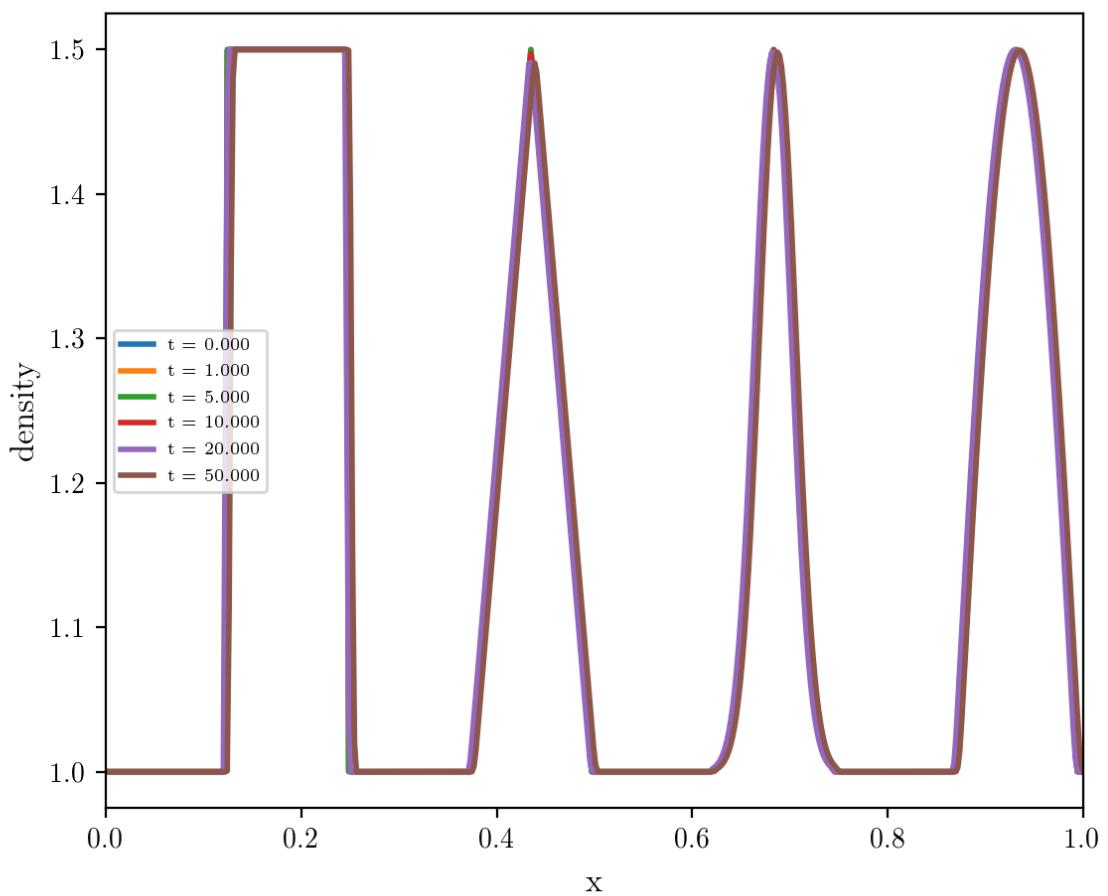
### 2.1 Piecewise Constant



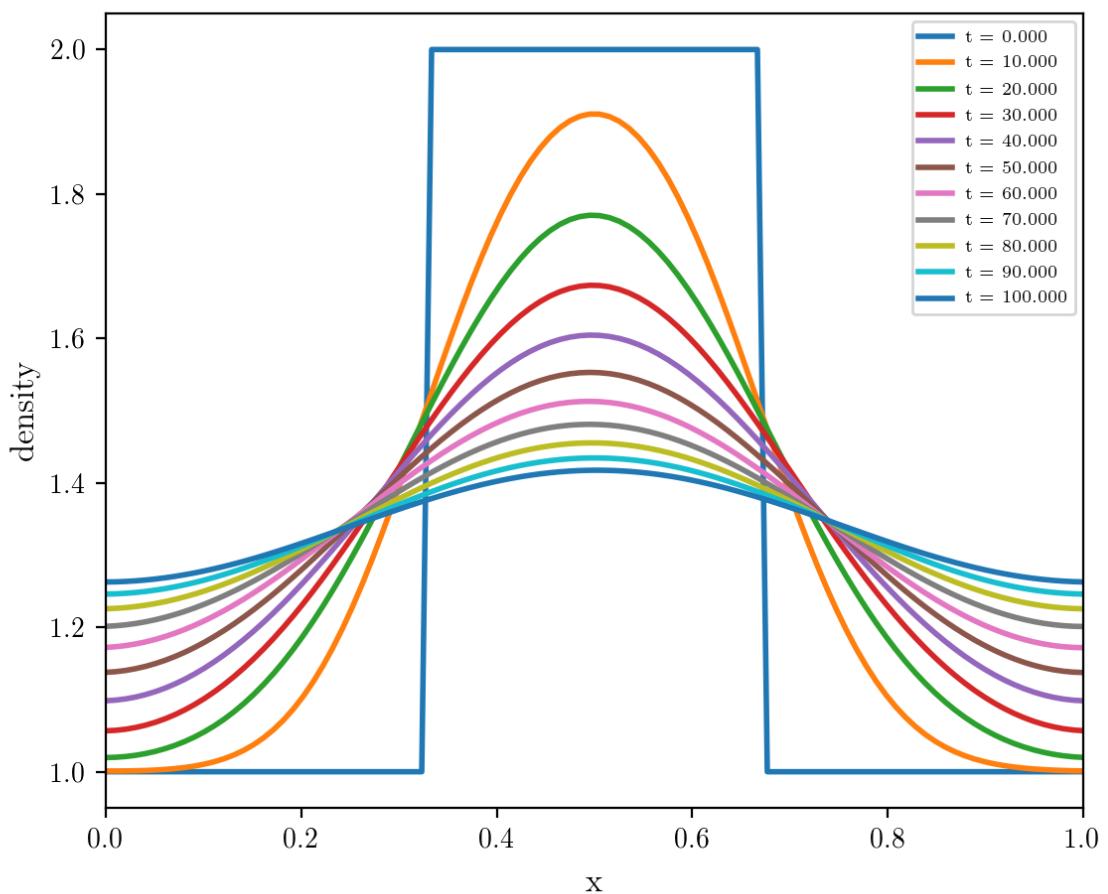
**Figure 1:** Piecewise constant advection with positive fixed global velocity  $v_x = 1$ .  
 $C_{CFL} = 0.9$ ,  $nx = 500$



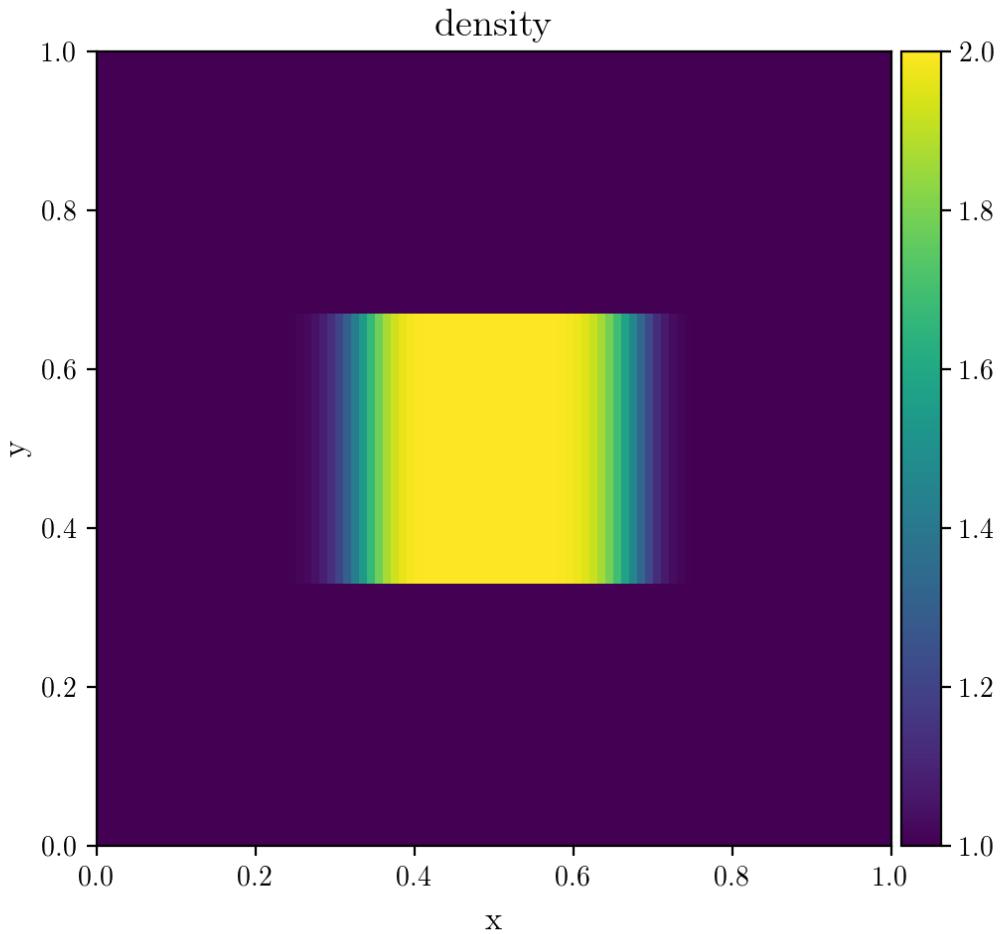
**Figure 2:** Piecewise constant advection with positive fixed global velocity  $v_x = 1$ .  
 $C_{CFL} = 0.1$ ,  $nx = 500$



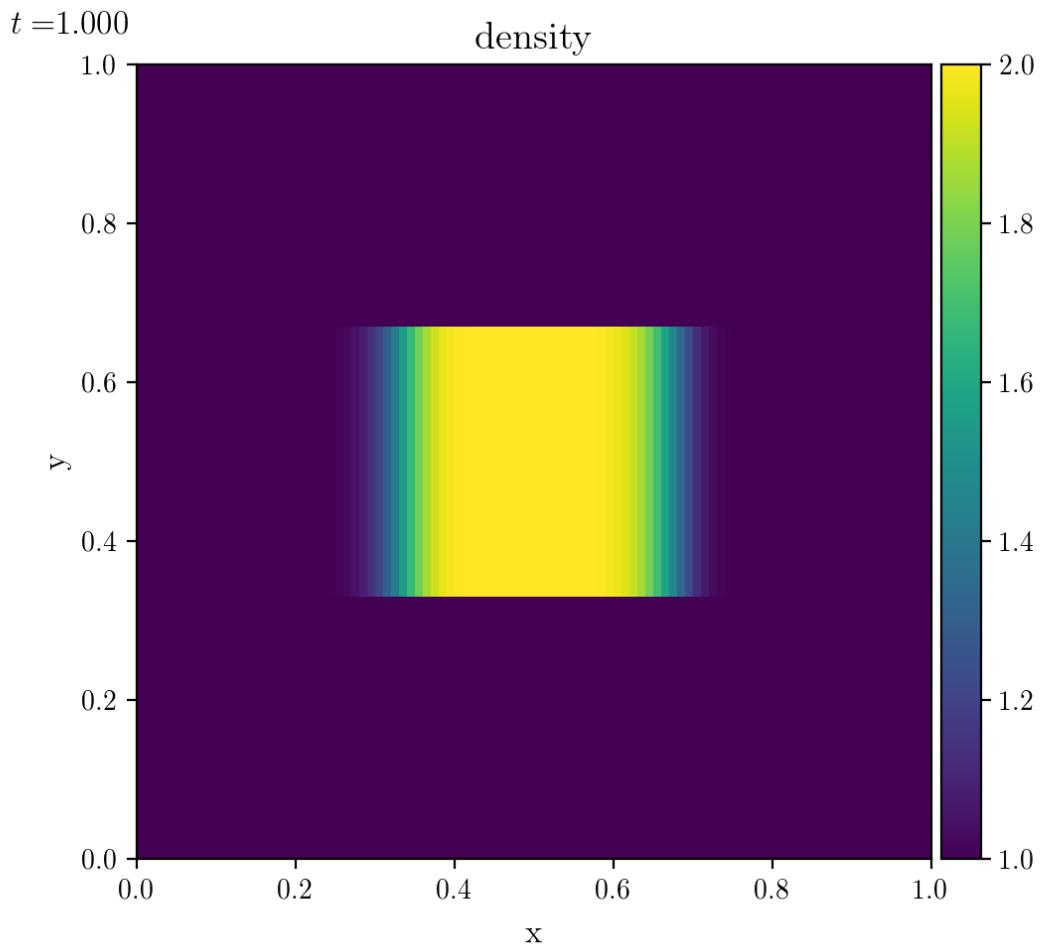
**Figure 3:** Piecewise constant advection with positive fixed global velocity  $v_x = 1$ .  
 $C_{CFL} = 1.0$ ,  $nx = 500$



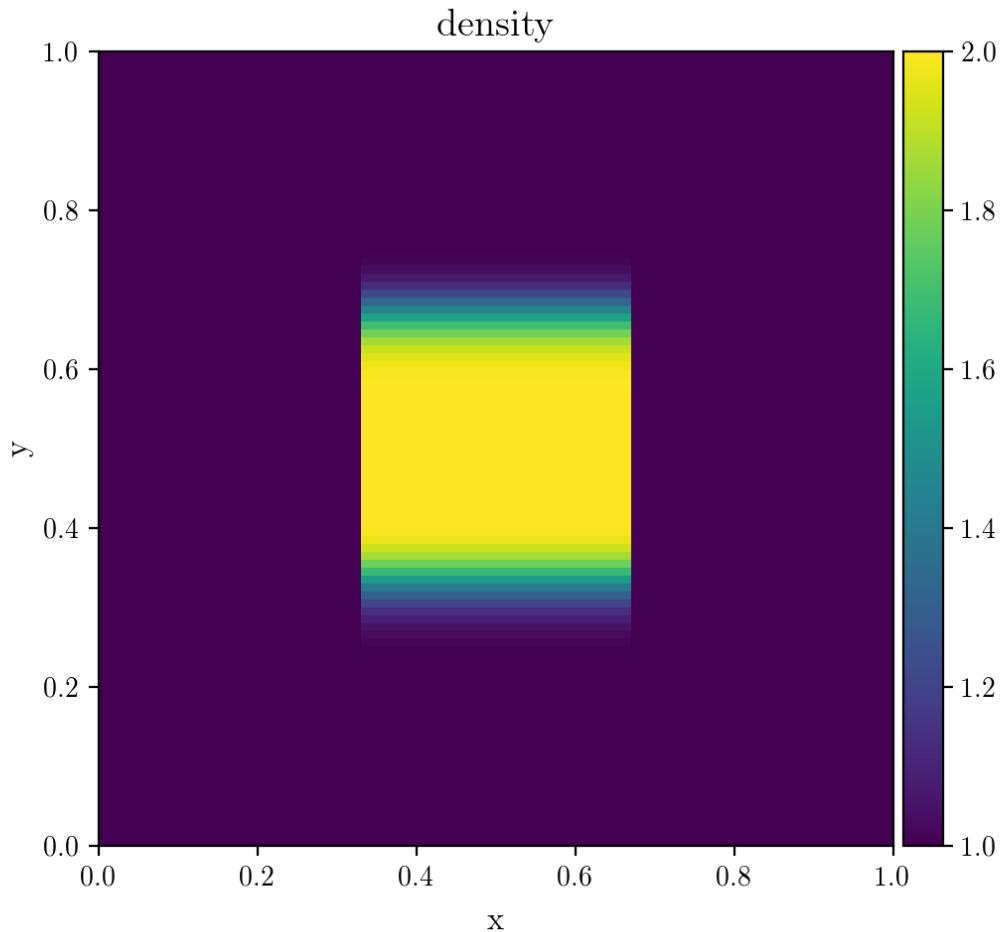
**Figure 4:** Piecewise constant advection with NEGATIVE fixed global velocity  $v_x = -1$ .  
 $C_{CFL} = 0.9$ ,  $nx = 100$



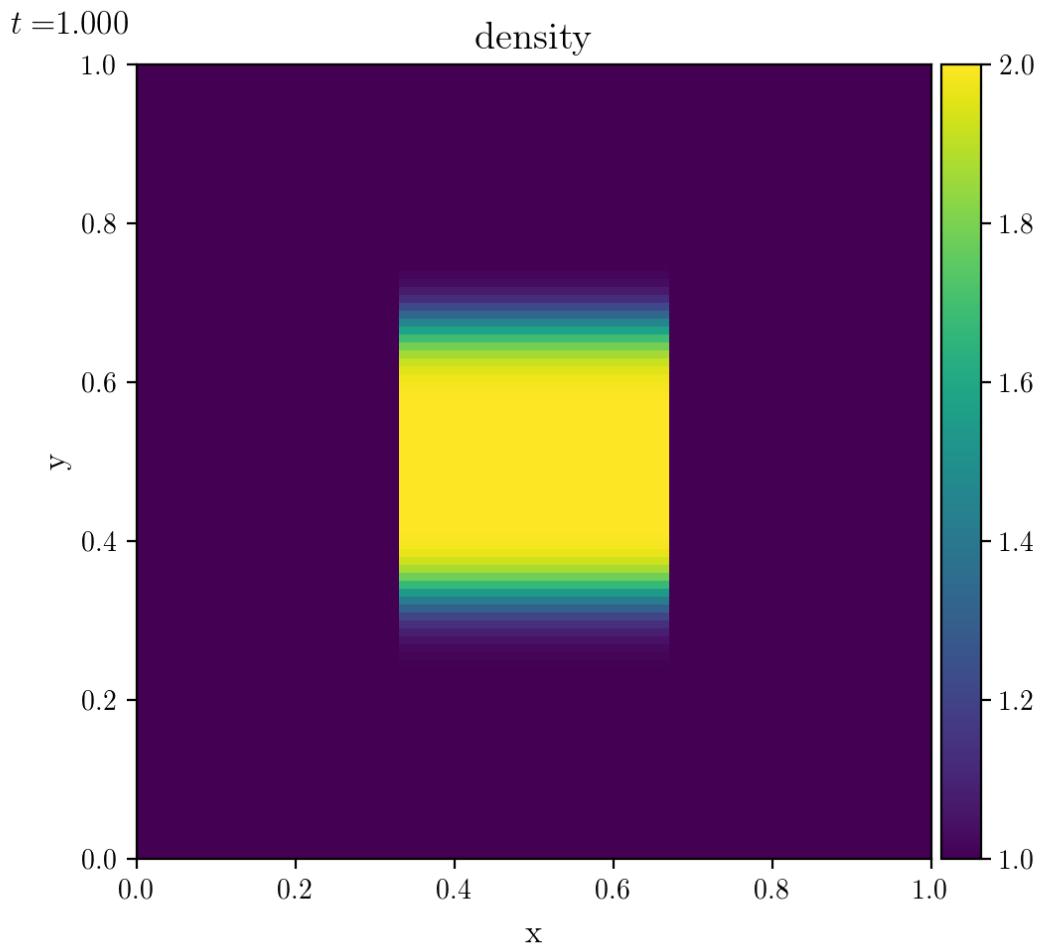
**Figure 5:** Piecewise constant advection with fixed global velocity  $v_x = 1, v_y = 0$ .  $C_{CFL} = 0.9$ ,  $nx = 100$ . ICs were a step function. This is 2D **without** Strang splitting, done naively.



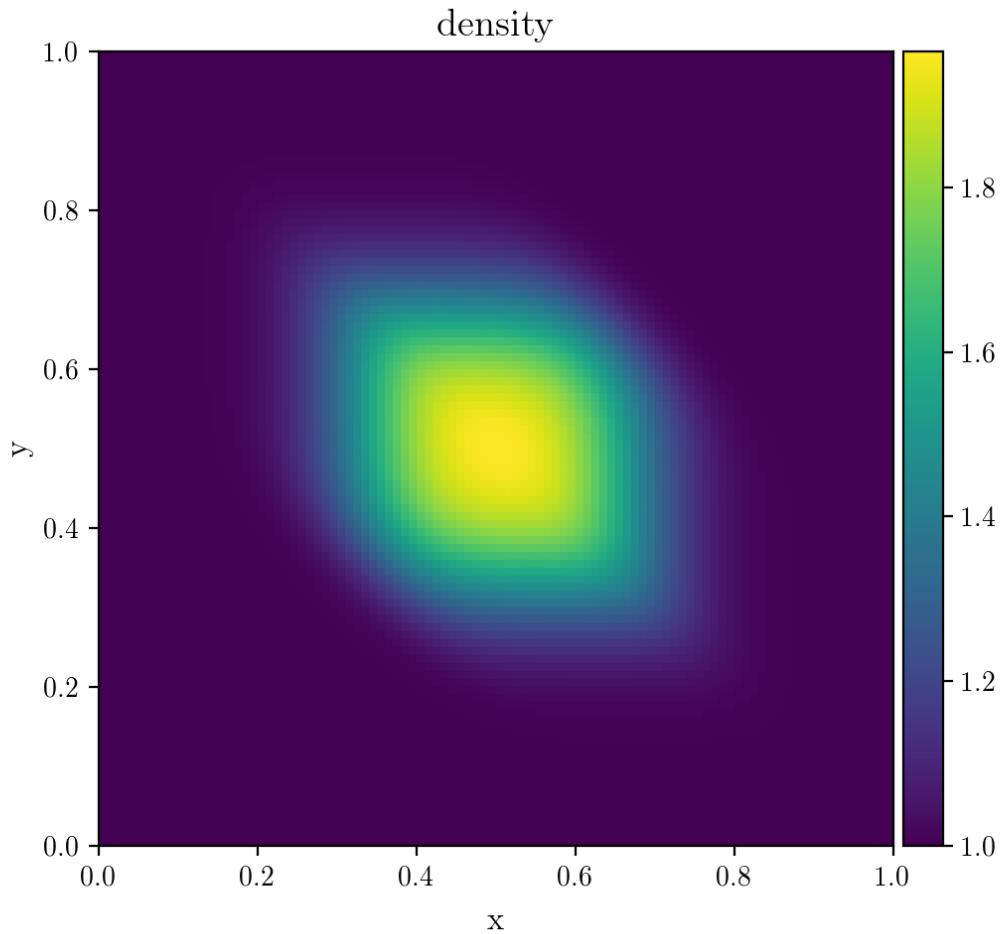
**Figure 6:** Piecewise constant advection with fixed global velocity  $v_x = 1, v_y = 0$ .  $C_{CFL} = 0.9$ ,  $nx = 100$ . ICs were a step function. **With** Strang splitting.



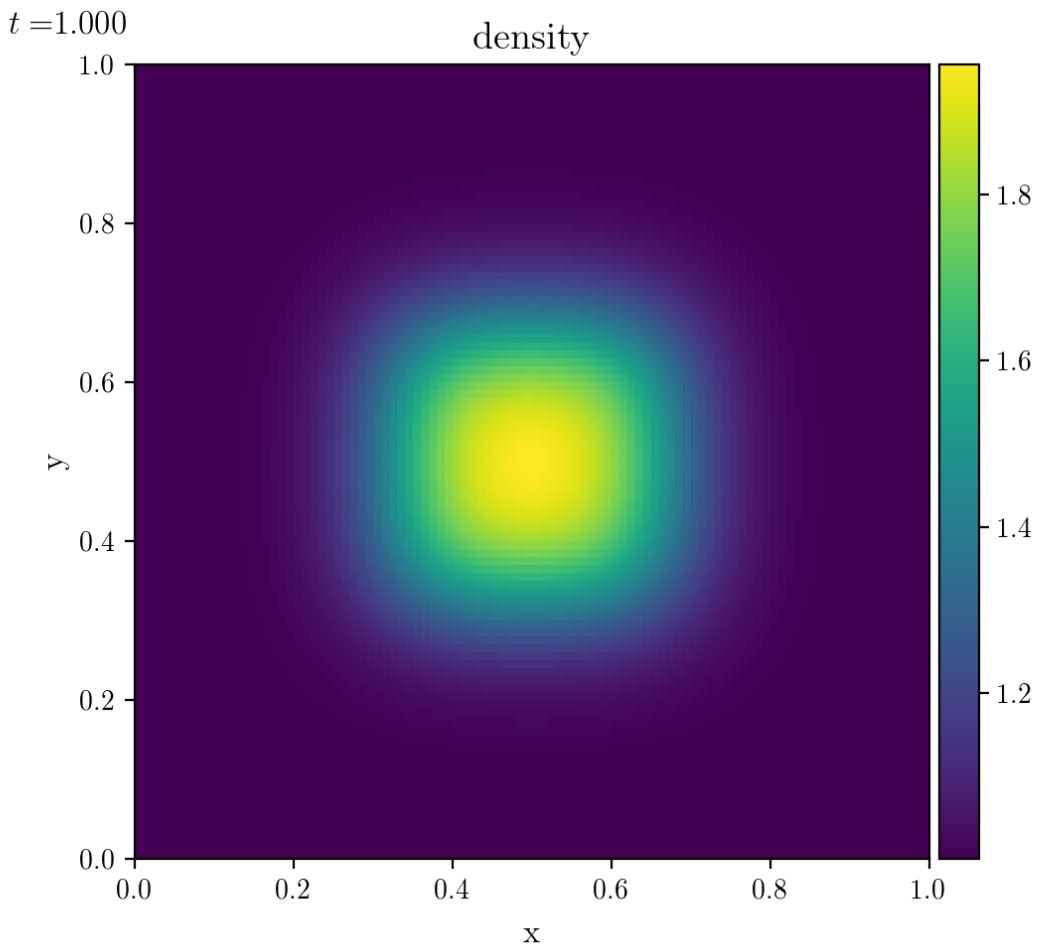
**Figure 7:** Piecewise constant advection with fixed global velocity  $v_x = 0, v_y = 1$ .  $C_{CFL} = 0.9$ ,  $nx = 100$ ,  $t = 1$ . ICs were a step function. **Without** Strang splitting, done naively.



**Figure 8:** Piecewise constant advection with fixed global velocity  $v_x = 0, v_y = 1$ .  $C_{CFL} = 0.9$ ,  $nx = 100$ ,  $t = 1$ . ICs were a step function. **With** Strang splitting.

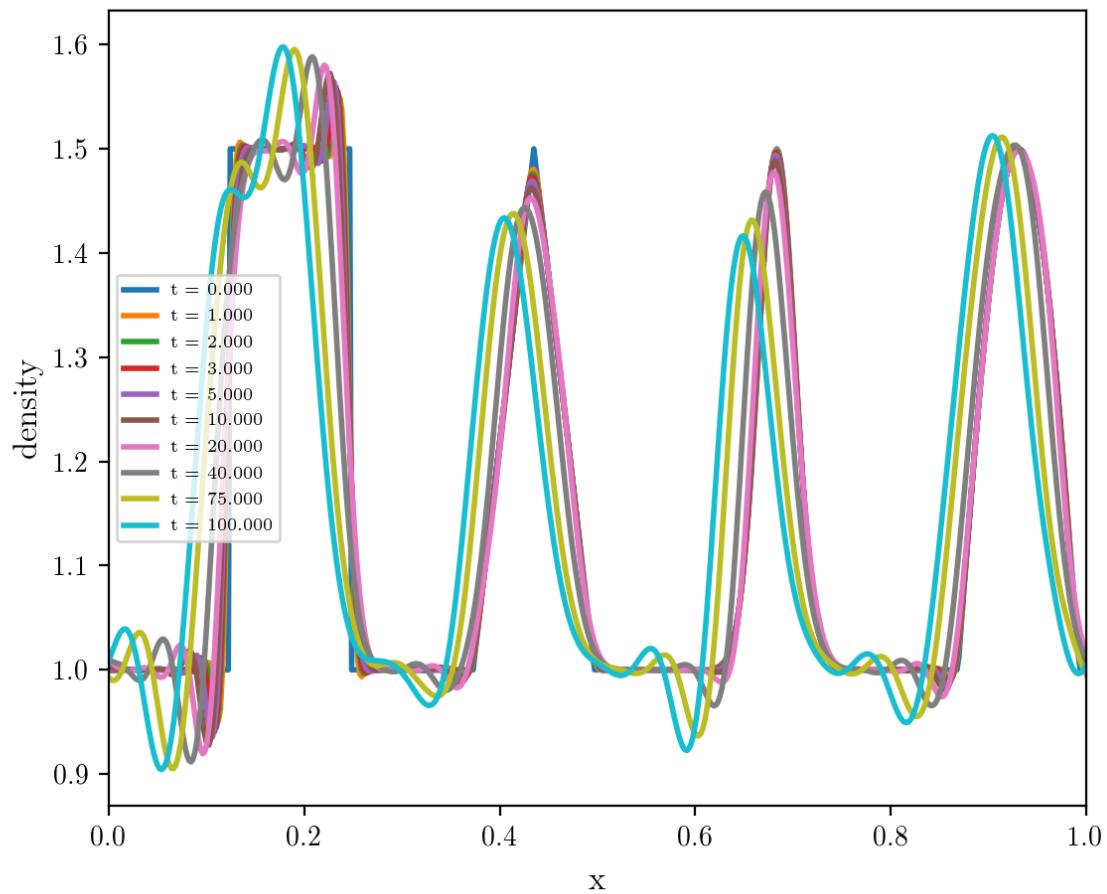


**Figure 9:** Piecewise constant advection with fixed global velocity  $v_x = v_y = 1$ .  $C_{CFL} = 0.9$ ,  $nx = 100$ ,  $t = 1$ . ICs were a step function. This is 2D **without** Strang splitting, done naively.

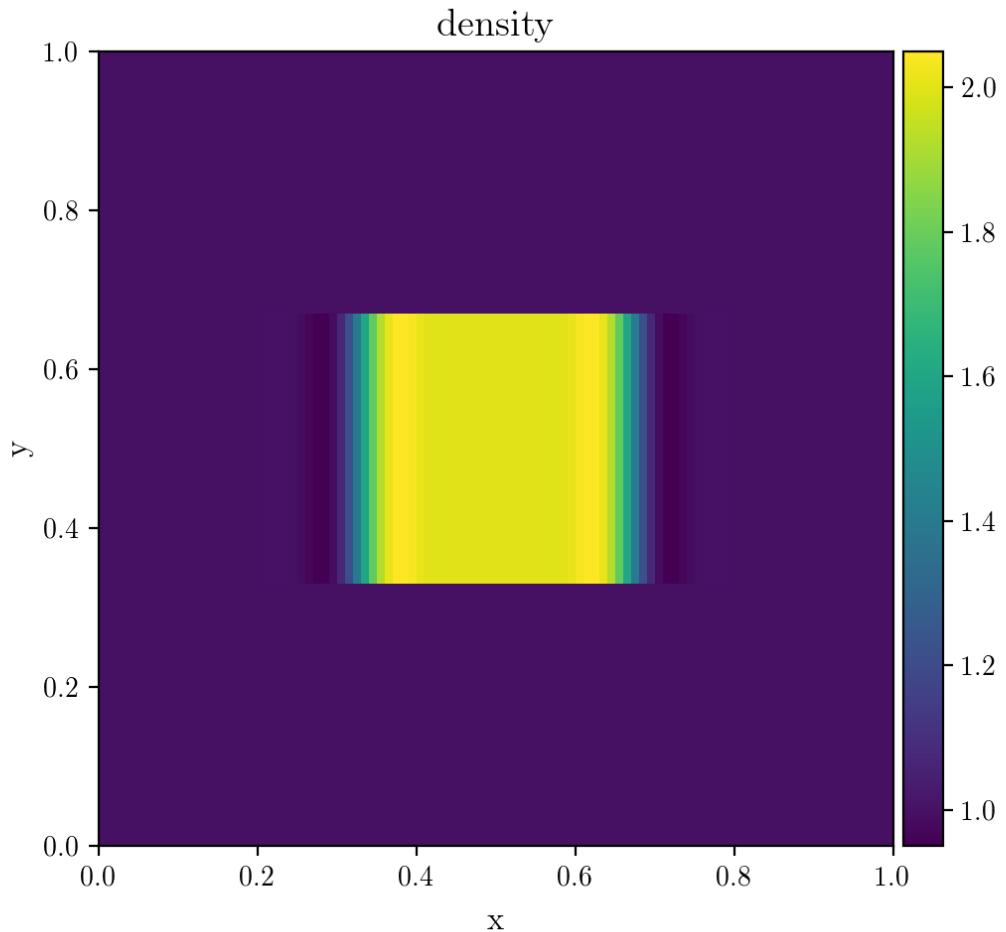


**Figure 10:** Piecewise constant advection with fixed global velocity  $v_x = v_y = 1$ .  $C_{CFL} = 0.9$ ,  $nx = 100$ ,  $t = 1$ . ICs were a step function. This is 2D **with** Strang splitting.

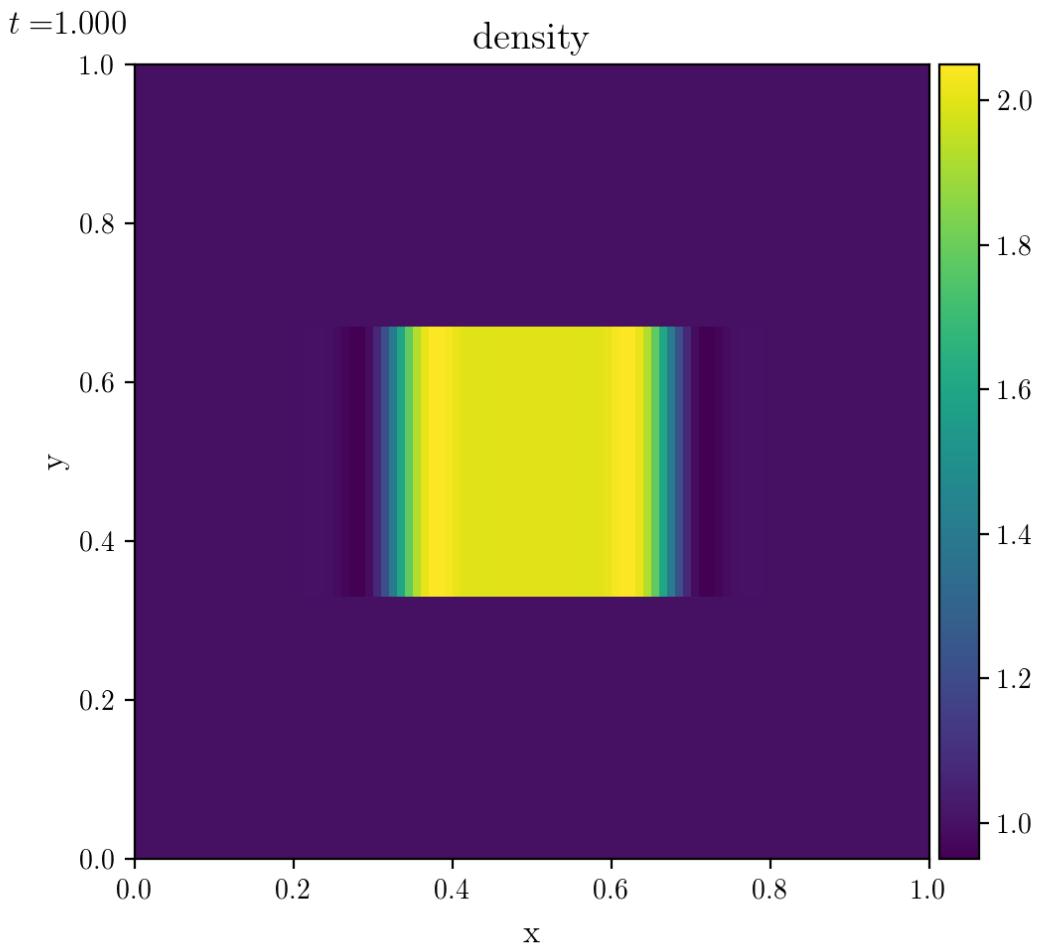
## 2.2 Piecewise Linear



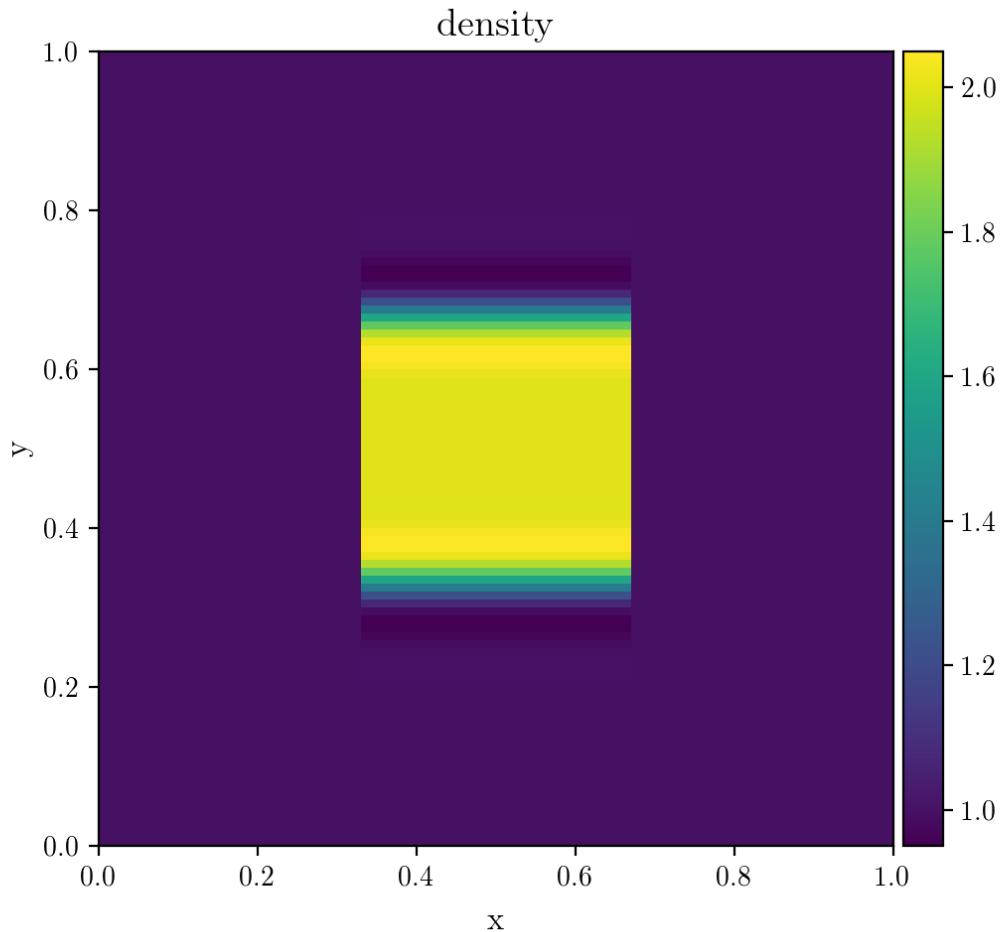
**Figure 11:** Piecewise linear advection with positive fixed global velocity  $v_x = 1$ .  
 $C_{CFL} = 0.9$ ,  $nx = 100$



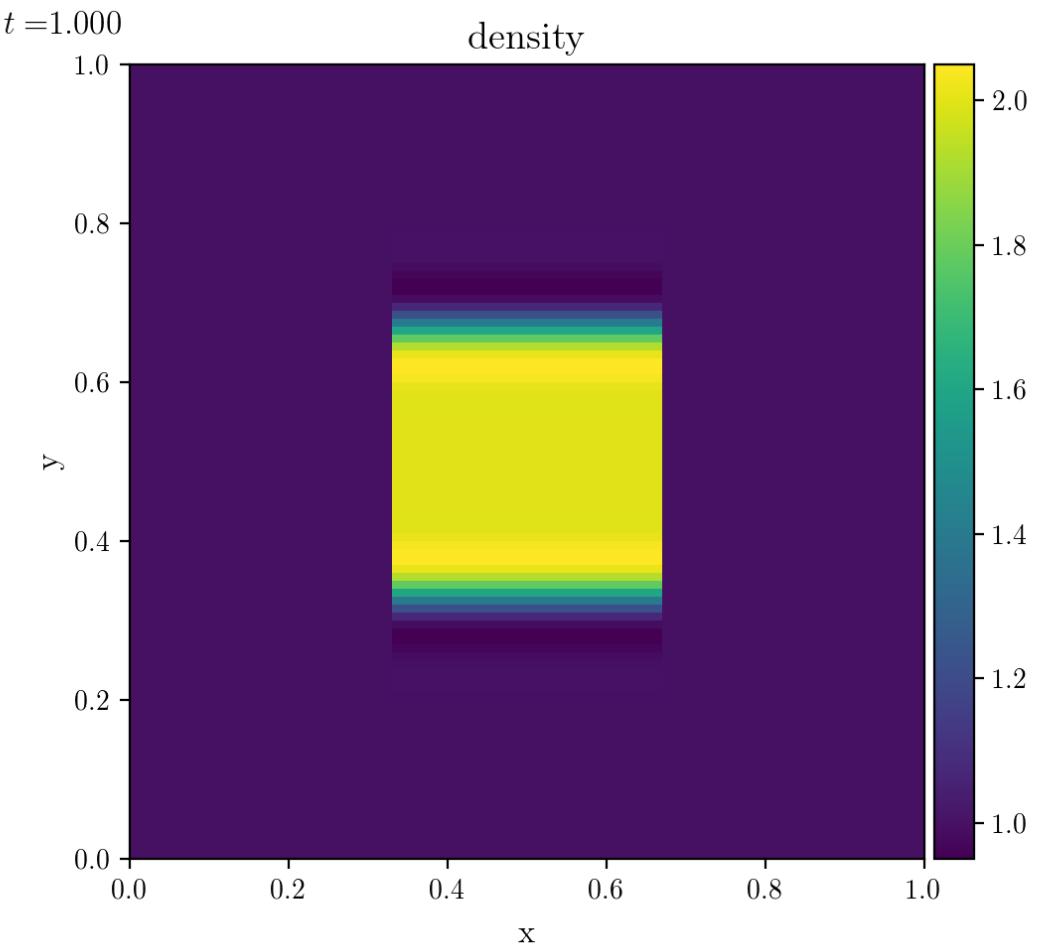
**Figure 12:** Piecewise linear advection with fixed global velocity  $v_x = 1, v_y = 0$ .  $C_{CFL} = 0.9$ ,  $nx = 100$ . ICs were a step function. **Without** Strang splitting.



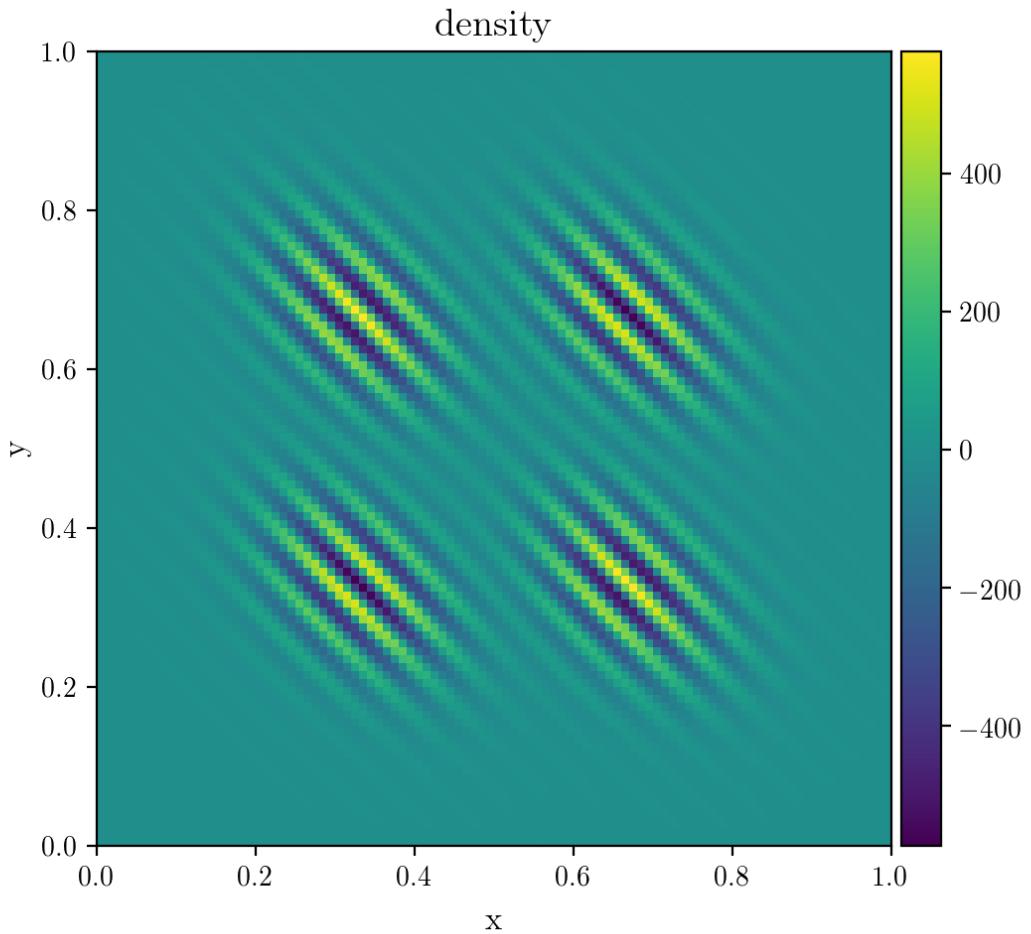
**Figure 13:** Piecewise linear advection with fixed global velocity  $v_x = 1, v_y = 0$ .  $C_{CFL} = 0.9$ ,  $nx = 100$ . ICs were a step function. **With** Strang splitting.



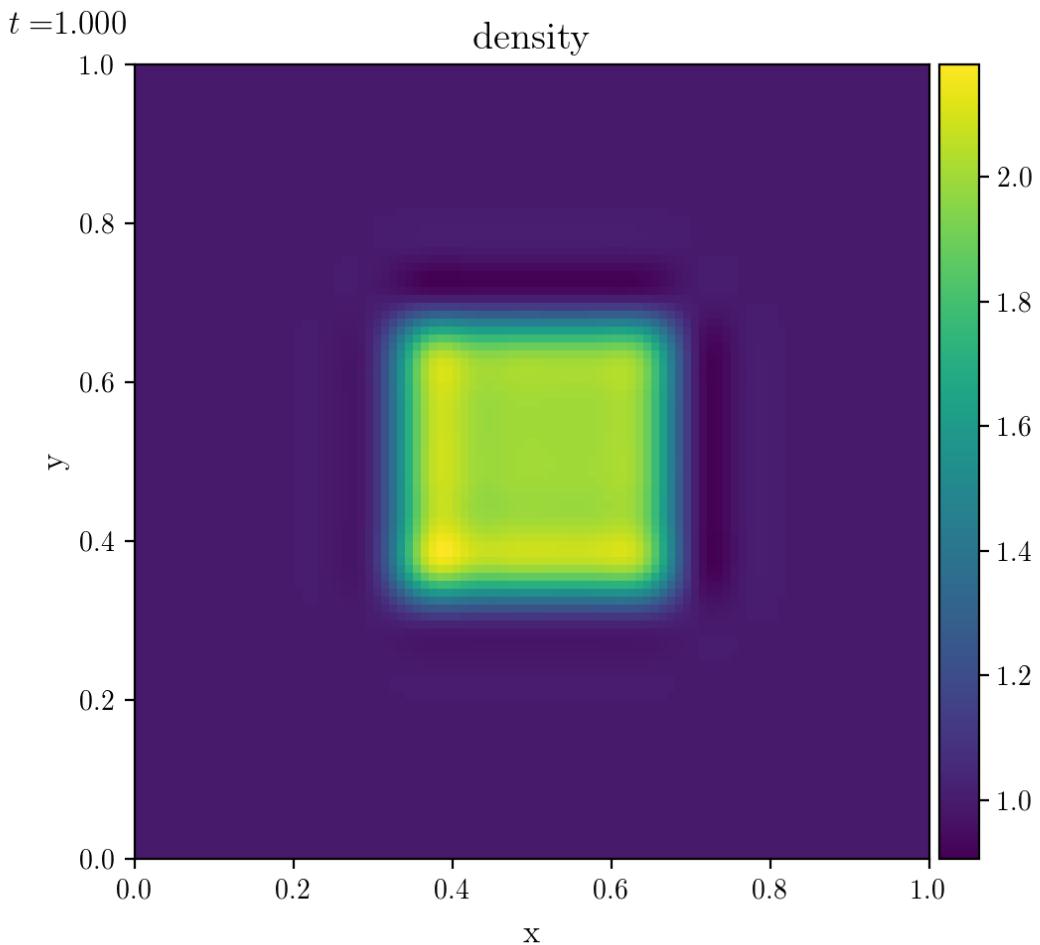
**Figure 14:** Piecewise linear advection with fixed global velocity  $v_x = 0, v_y = 1$ .  $C_{CFL} = 0.9$ ,  $nx = 100$ . ICs were a step function. **Without** Strang splitting.



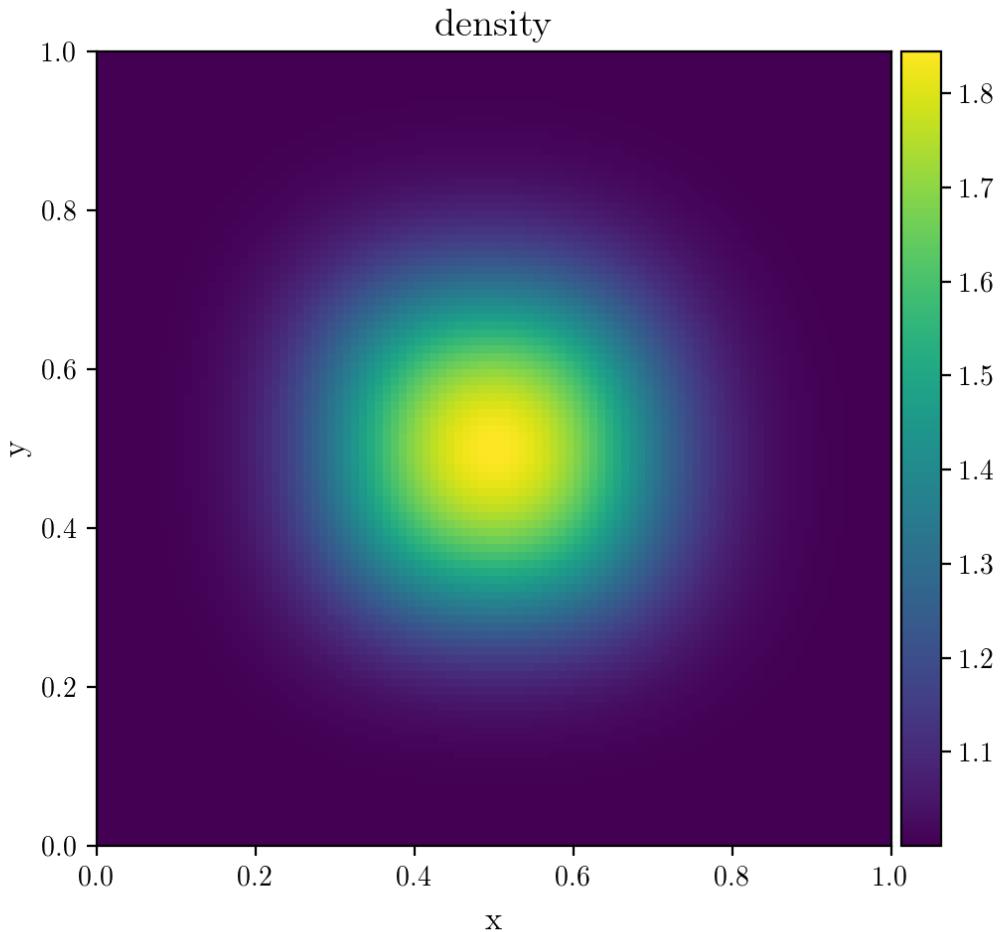
**Figure 15:** Piecewise linear advection with fixed global velocity  $v_x = 0, v_y = 1$ .  $C_{CFL} = 0.9$ ,  $nx = 100$ . ICs were a step function. **With** Strang splitting.



**Figure 16:** Piecewise linear advection with fixed global velocity  $v_x = v_y = 1$ .  $C_{CFL} = 0.9$ ,  $nx = 100$ . ICs were a step function. Note that despite of the strong oscillations, the total density is conserved! (Unless the oscillations get too big for floats to handle). **Without** Strang splitting.



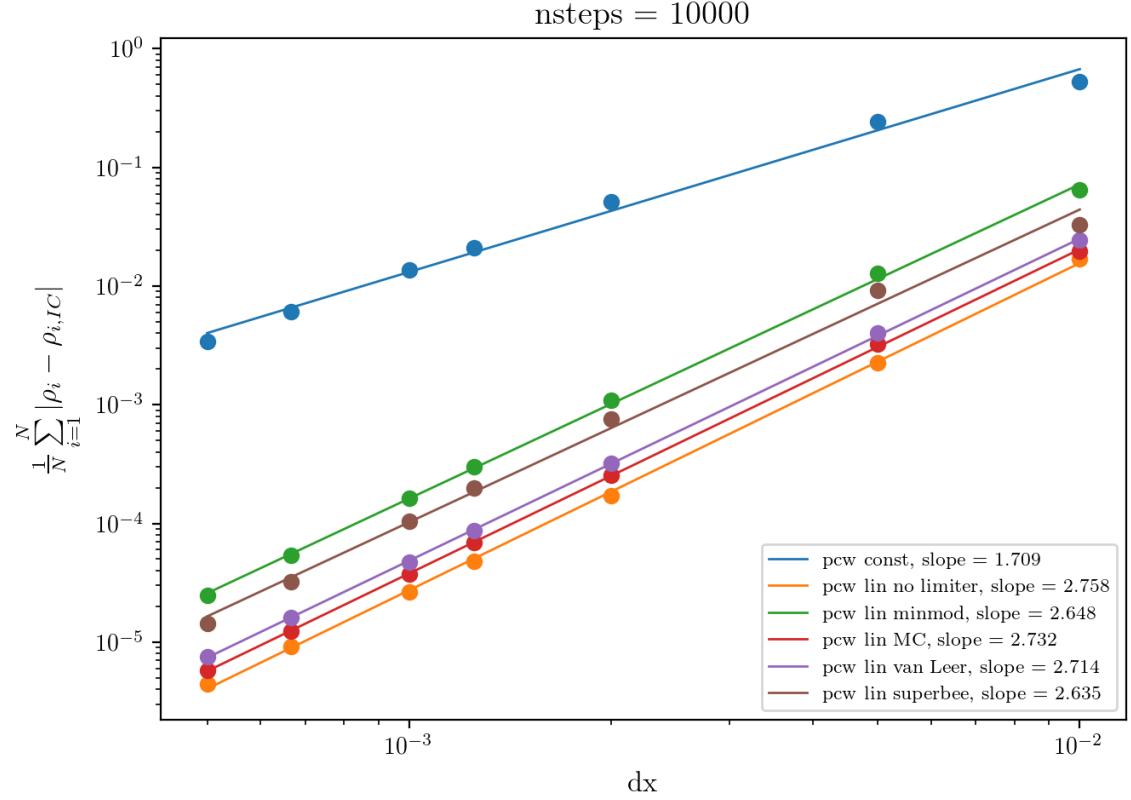
**Figure 17:** Piecewise linear advection with fixed global velocity  $v_x = v_y = 1$ .  $C_{CFL} = 0.9$ ,  $nx = 100$ . ICs were a step function. Note that despite of the strong oscillations, the total density is conserved! (Unless the oscillations get too big for floats to handle) **With** Strang splitting.



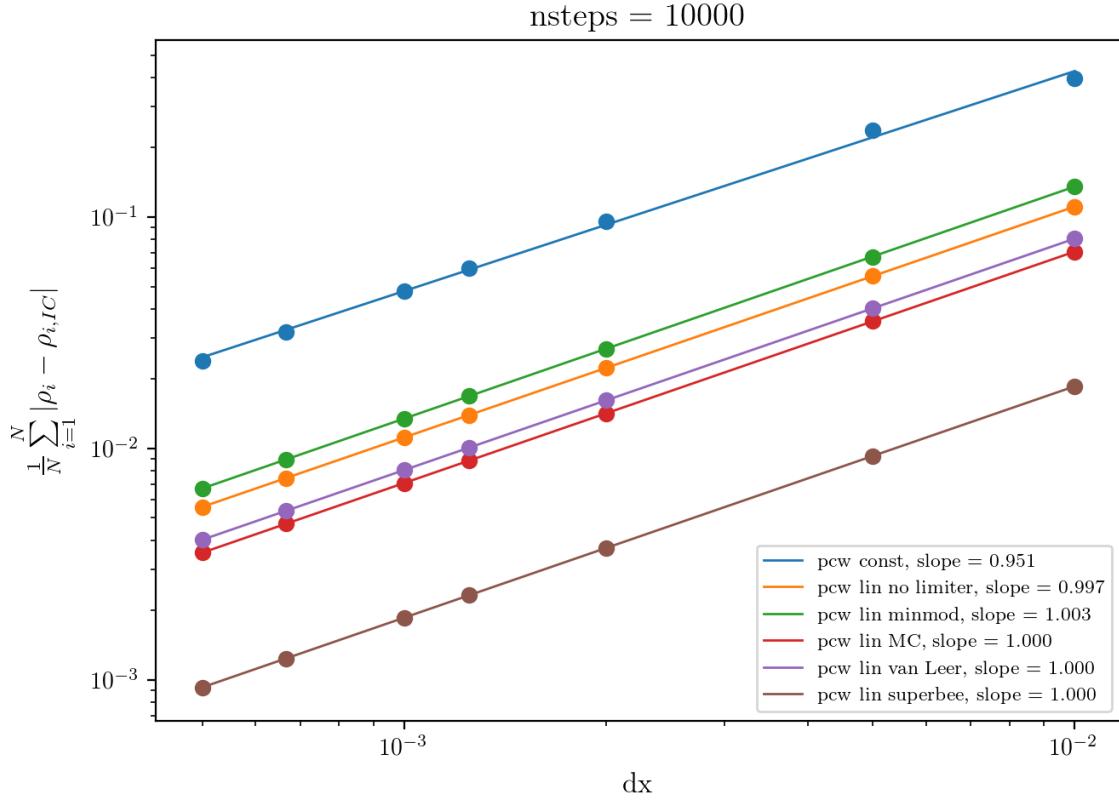
**Figure 18:** Piecewise linear advection with fixed global velocity  $v_x = v_y = 1$ .  $C_{CFL} = 0.1$ ,  $nx = 100$ . ICs were a step function. Note the lower CFL condition compared to fig 16.

**Without** Strang splitting.

## 2.3 Order of Convergence

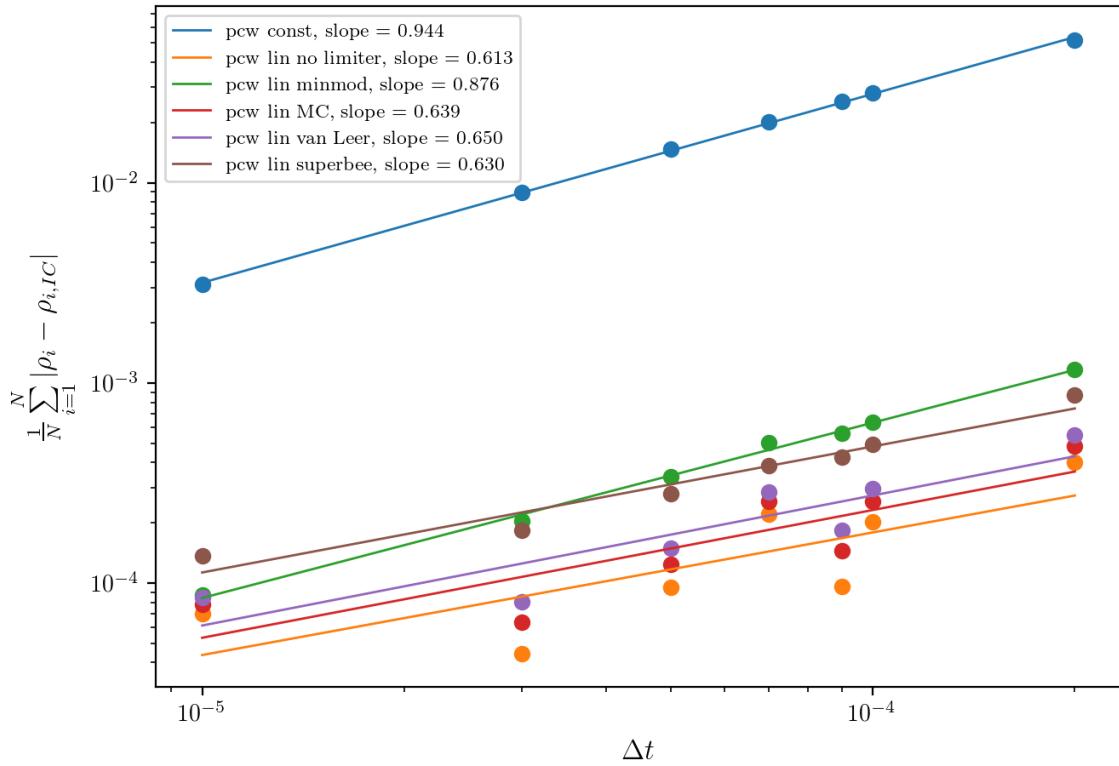


**Figure 19:** Convergence Study with respect to cell size  $\Delta x$  using the  $L1$  norm. Points are measurements, the straight lines are fitted curves, with their slope given in the legend. For an accurate comparison, the simulations are stopped after the same number of steps. This way, the  $C_{cfl}$  can also be kept constant throughout the different  $\Delta x$ . Initial conditions was a Gaussian profile. Here for piecewise linear advection.



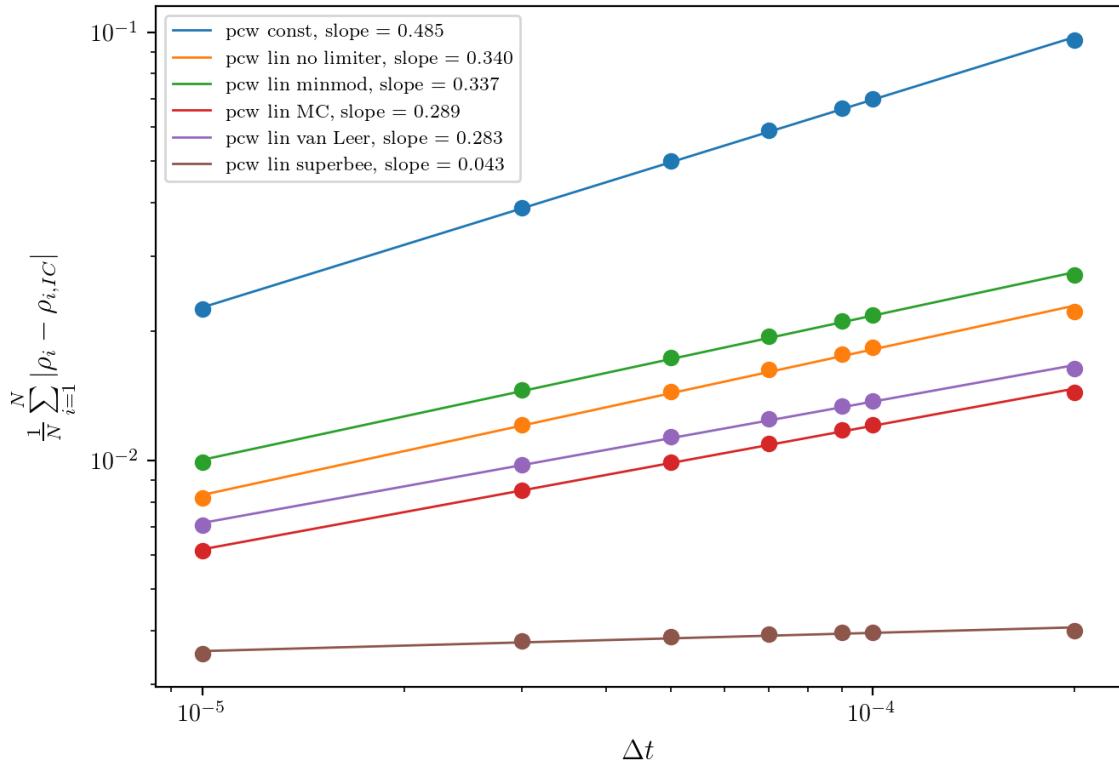
**Figure 20:** Convergence Study with respect to cell size  $\Delta x$  using the  $L1$  norm. Points are measurements, the straight lines are fitted curves, with their slope given in the legend. For an accurate comparison, the simulations are stopped after the same number of steps. This way, the  $C_{cfl}$  can also be kept constant throughout the different  $\Delta x$ . Initial conditions was a step function. Here for piecewise linear advection.

$nx = 500, nstep = 10000$



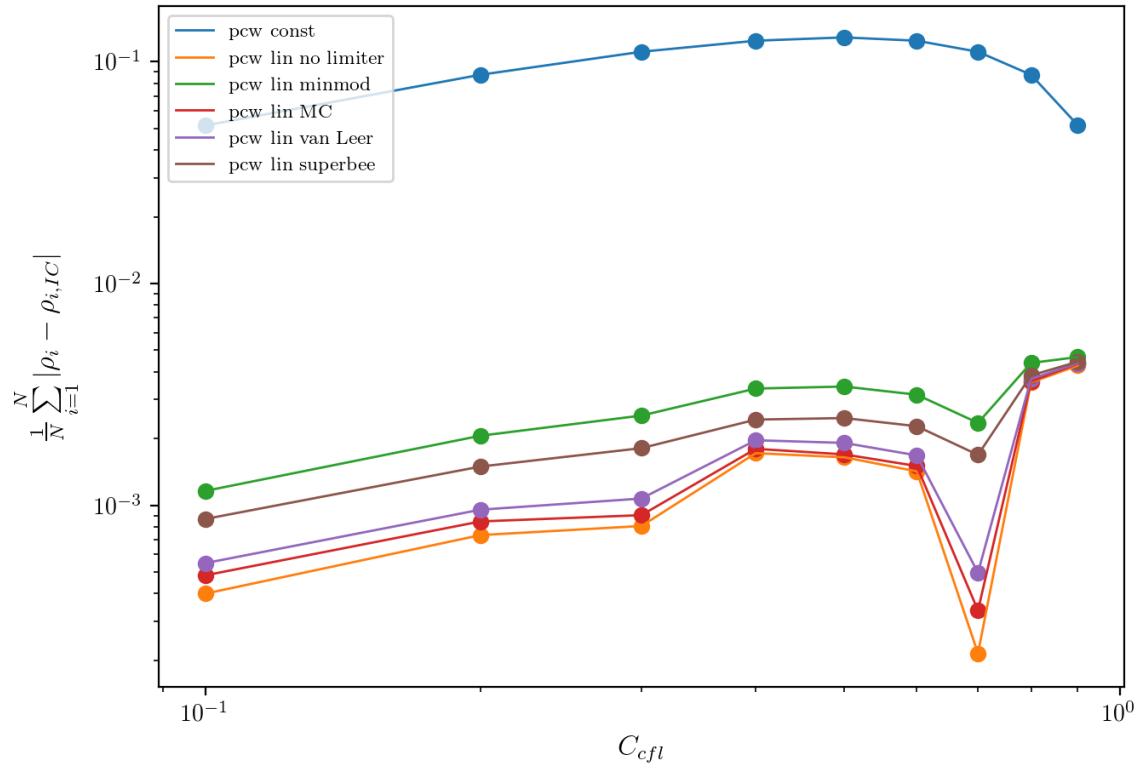
**Figure 21:** Convergence Study with respect to time step size  $\Delta t$  using the  $L1$  norm. Points are measurements, the straight lines are fitted curves, with their slope given in the legend. For an accurate comparison, the simulations are stopped after the same number of steps, and  $\Delta x$  is kept constant. Highest  $\Delta t$  corresponds to  $C_{cfl} = 0.001$ . Initial conditions was a Gaussian profile. Here for piecewise linear advection.

$nx = 500, nstep = 10000$



**Figure 22:** Convergence Study with respect to time step size  $\Delta t$  using the  $L1$  norm. Points are measurements, the straight lines are fitted curves, with their slope given in the legend. For an accurate comparison, the simulations are stopped after the same number of steps, and  $\Delta x$  is kept constant. Highest  $\Delta t$  corresponds to  $C_{cfl} = 0.001$ . Initial conditions was a step function. Here for piecewise linear advection.

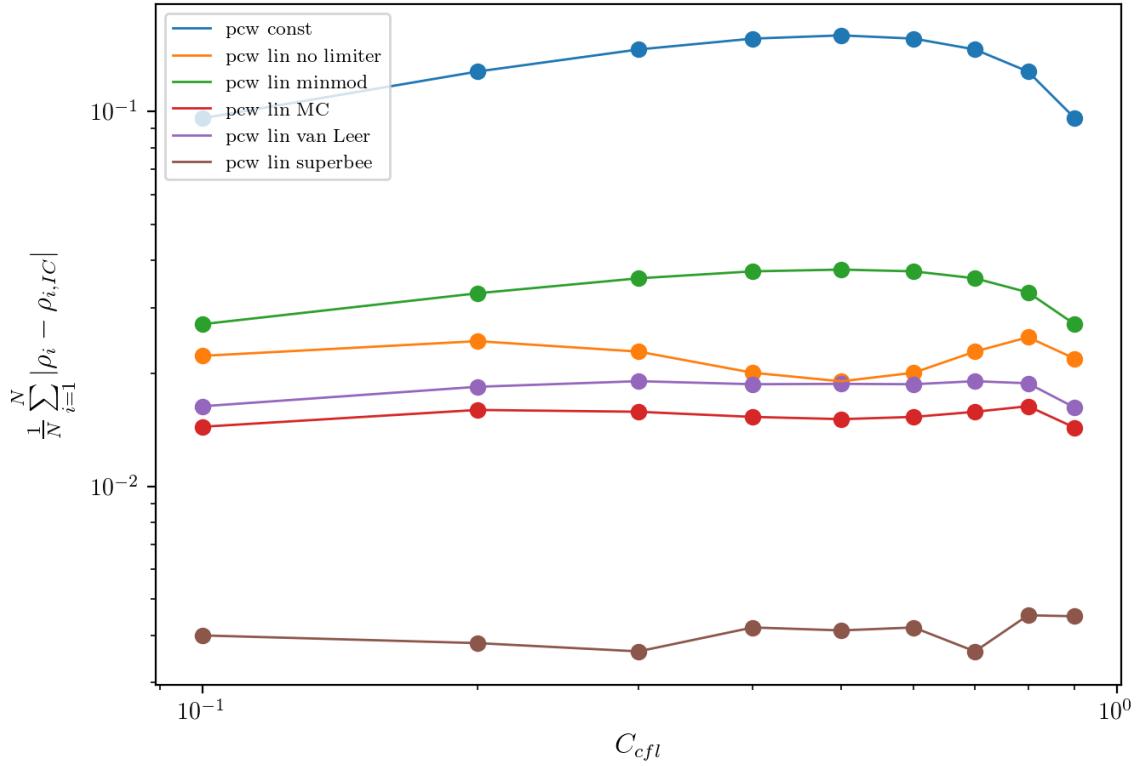
$nx = 500, nsteps = 10000$



**Figure 23:** Convergence Study with respect to the Courant number  $C_{cfl}$  using the  $L1$  norm.

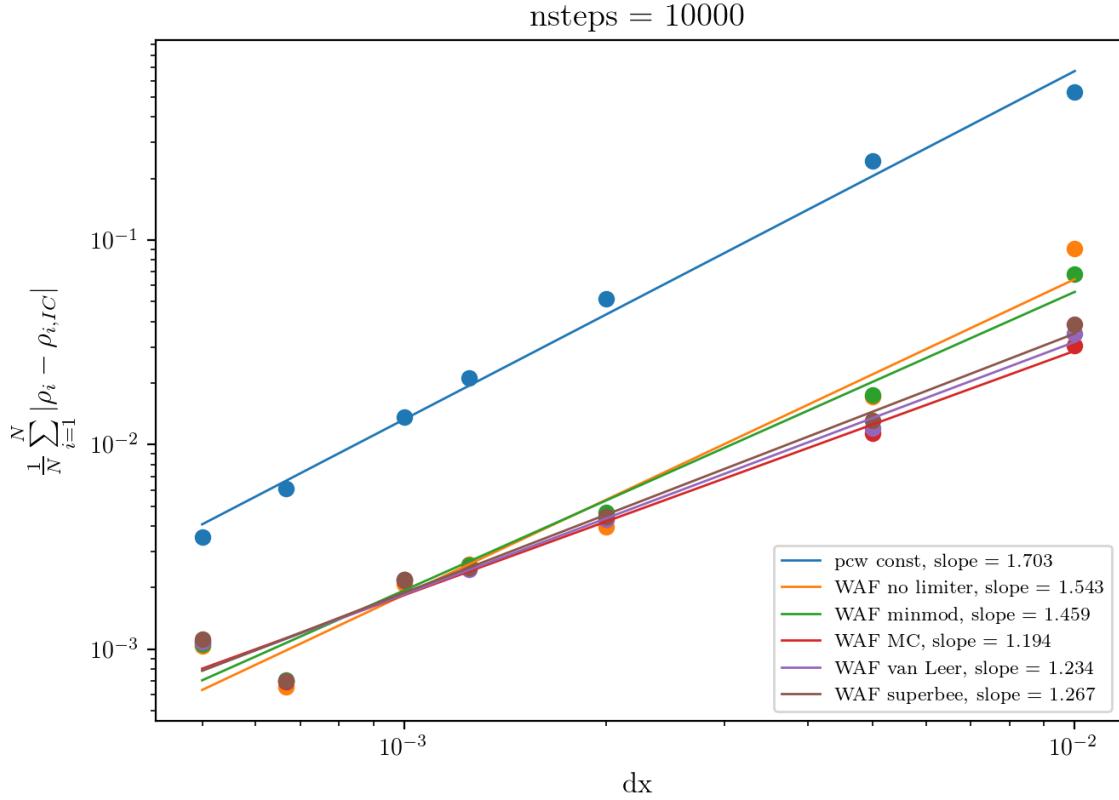
Points are measurements, the straight lines just connect the dots. For an accurate comparison, the simulations are stopped after the same number of steps, and  $\Delta x$  is kept constant. Initial conditions was a Gaussian profile. Here for piecewise linear advection.

$nx = 500, nsteps = 10000$

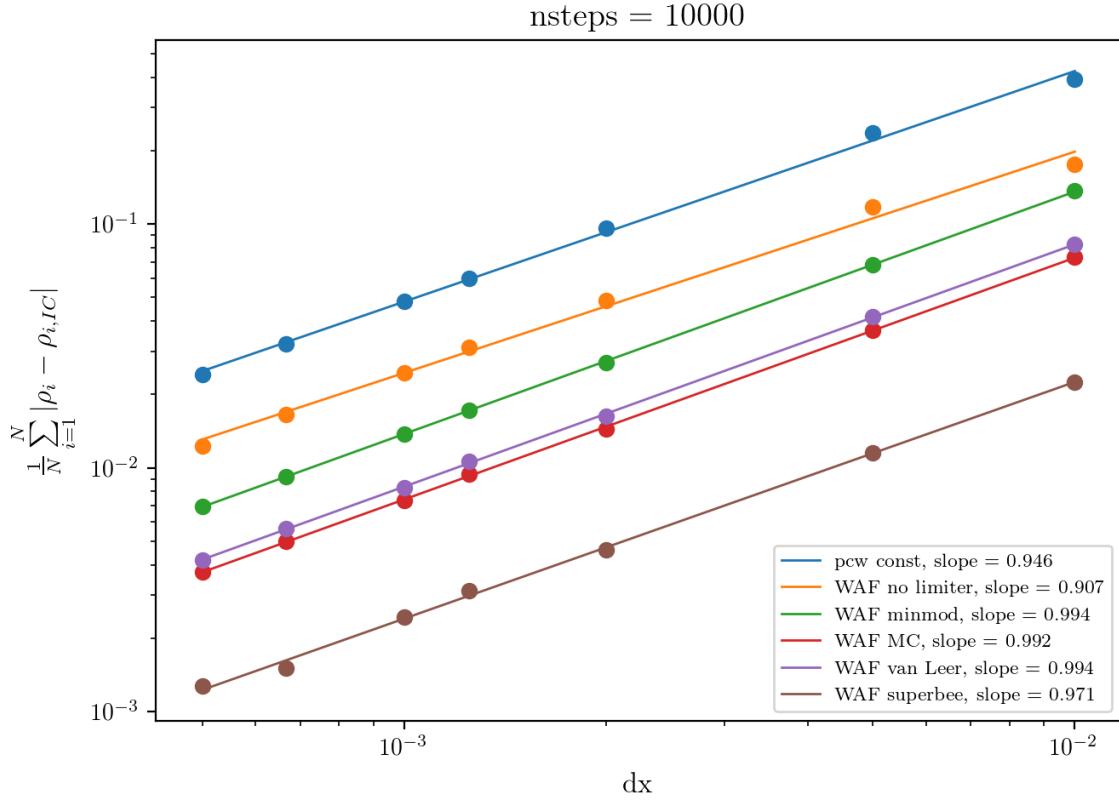


**Figure 24:** Convergence Study with respect to the Courant number  $C_{cfl}$  using the  $L_1$  norm.

Points are measurements, the straight lines just connect the dots. For an accurate comparison, the simulations are stopped after the same number of steps, and  $\Delta x$  is kept constant. Initial conditions was a step function. Here for piecewise linear advection.

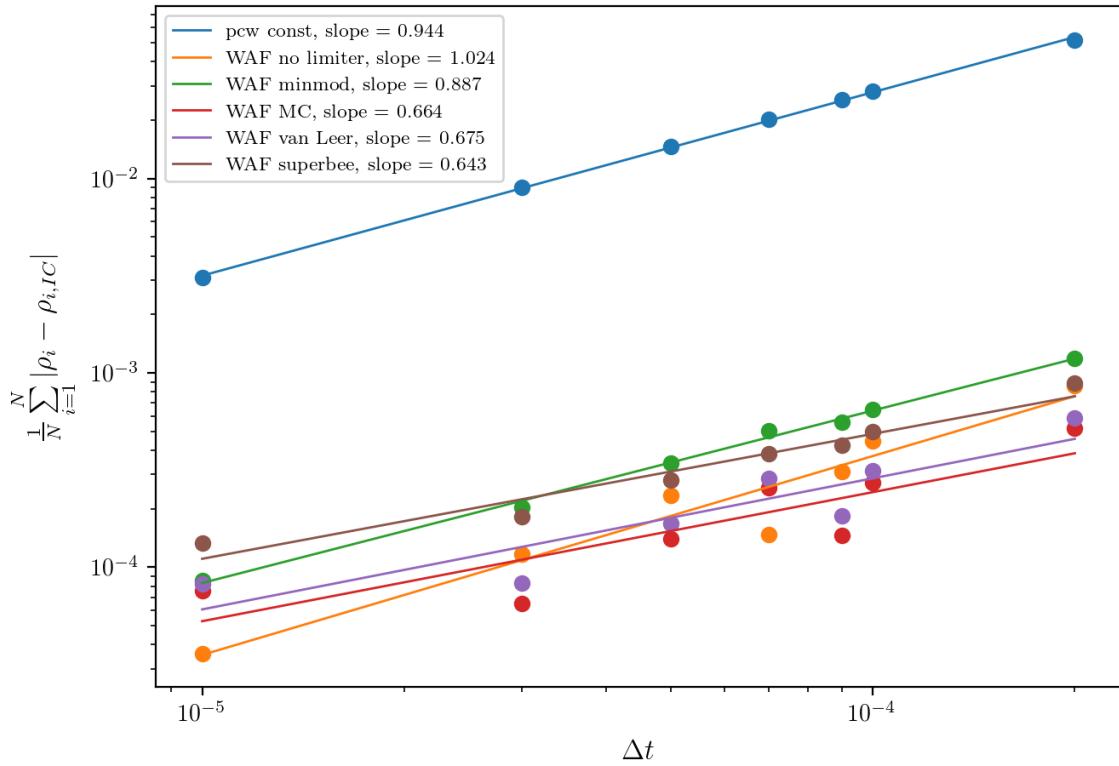


**Figure 25:** Convergence Study with respect to cell size  $\Delta x$  using the  $L1$  norm. Points are measurements, the straight lines are fitted curves, with their slope given in the legend. For an accurate comparison, the simulations are stopped after the same number of steps. This way, the  $C_{cfl}$  can also be kept constant throughout the different  $\Delta x$ . Initial conditions was a Gaussian profile. Here for WAF advection.



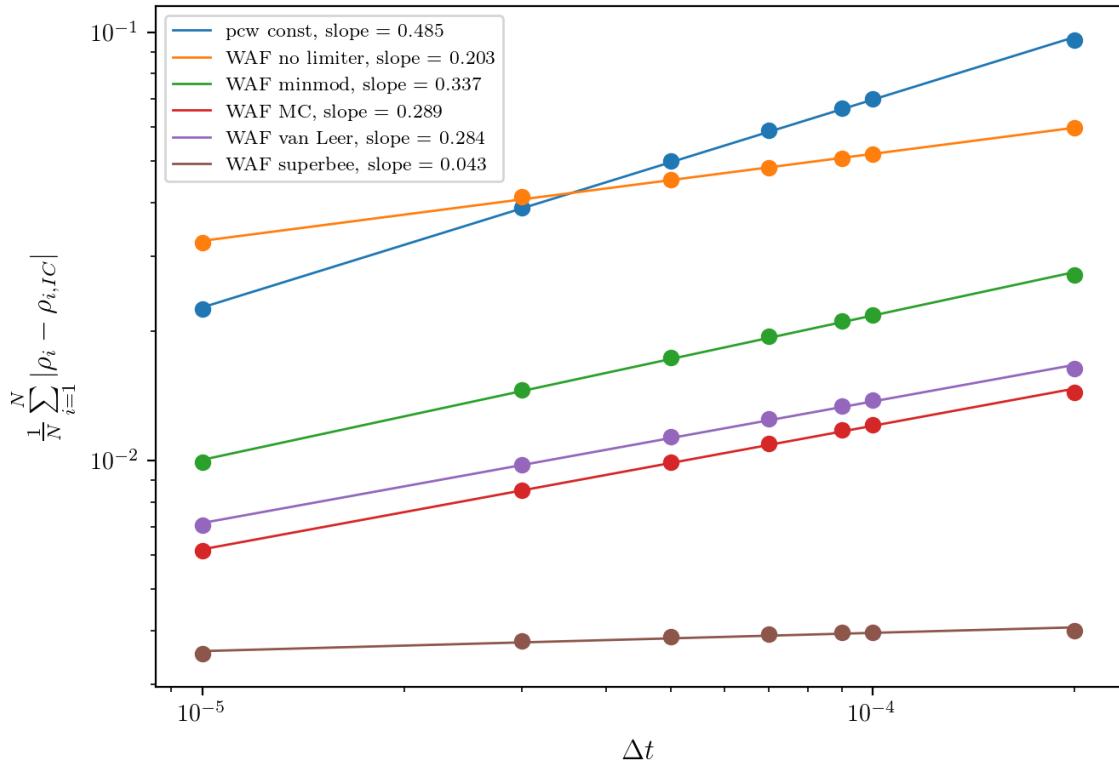
**Figure 26:** Convergence Study with respect to cell size  $\Delta x$  using the  $L1$  norm. Points are measurements, the straight lines are fitted curves, with their slope given in the legend. For an accurate comparison, the simulations are stopped after the same number of steps. This way, the  $C_{cfl}$  can also be kept constant throughout the different  $\Delta x$ . Initial conditions was a step function. Here for WAF advection.

$nx = 500, nstep = 10000$



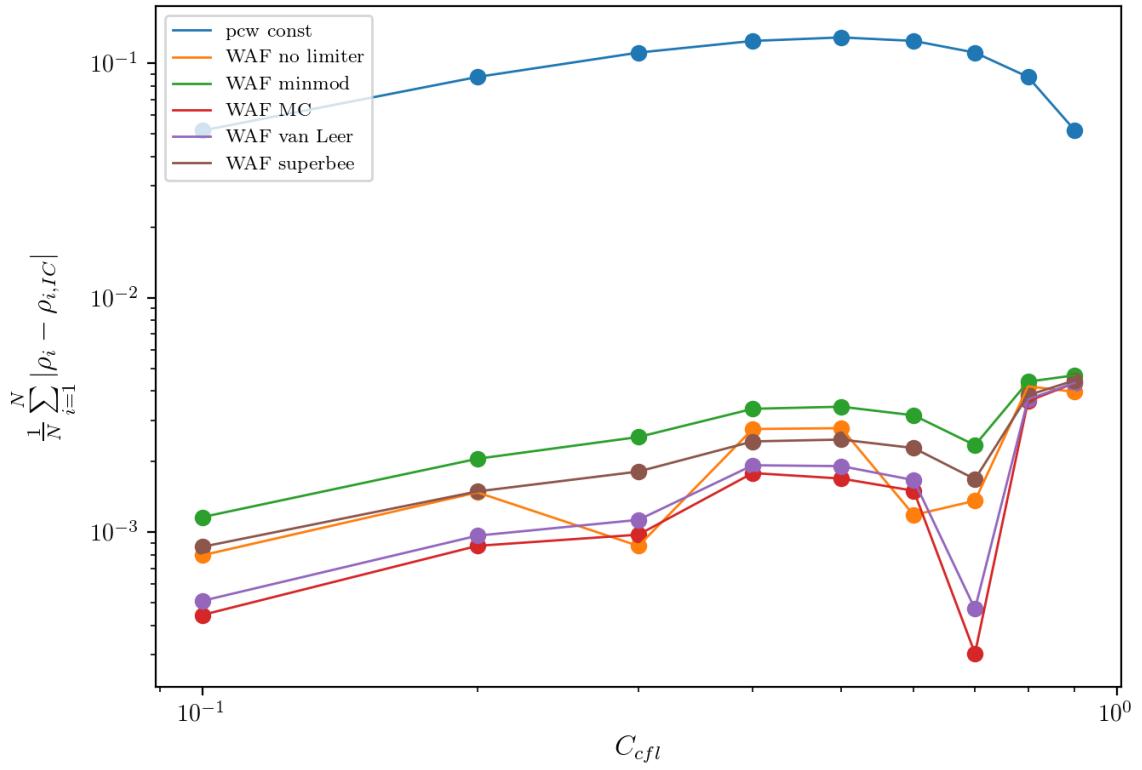
**Figure 27:** Convergence Study with respect to time step size  $\Delta t$  using the  $L1$  norm. Points are measurements, the straight lines are fitted curves, with their slope given in the legend. For an accurate comparison, the simulations are stopped after the same number of steps, and  $\Delta x$  is kept constant. Highest  $\Delta t$  corresponds to  $C_{cfl} = 0.001$ . Initial conditions was a Gaussian profile. Here for WAF advection.

$nx = 500, nstep = 10000$



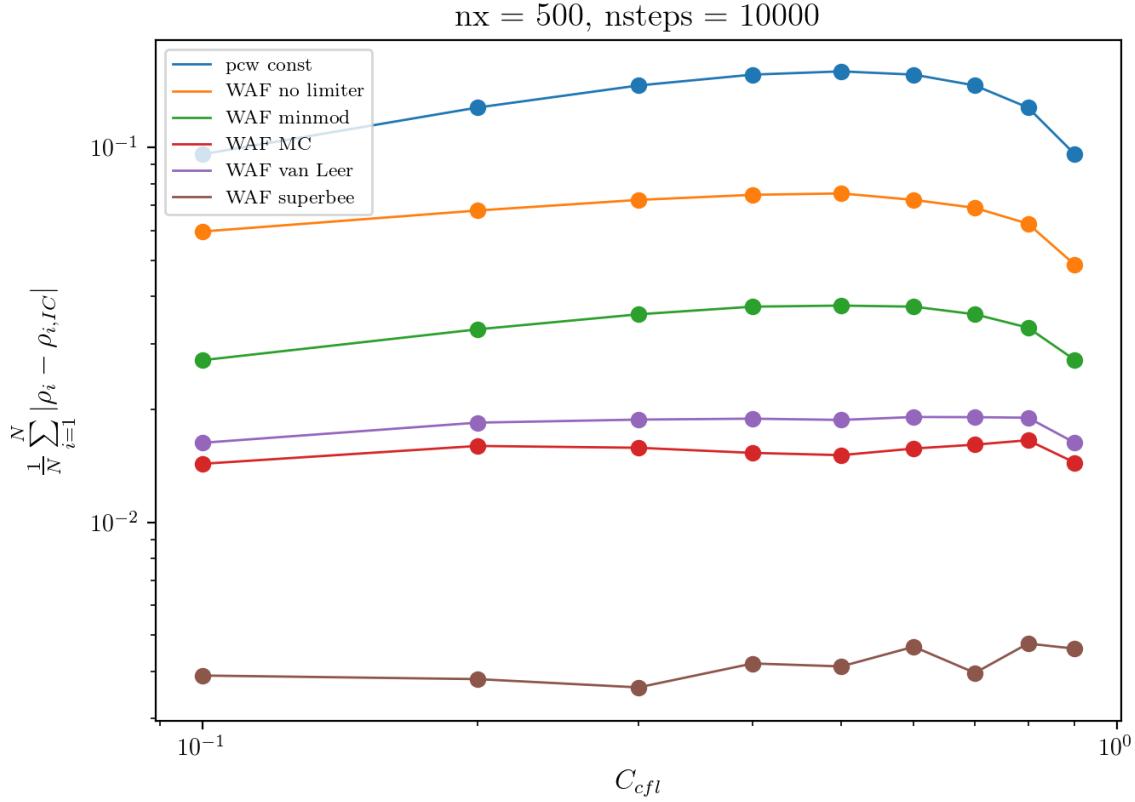
**Figure 28:** Convergence Study with respect to time step size  $\Delta t$  using the  $L1$  norm. Points are measurements, the straight lines are fitted curves, with their slope given in the legend. For an accurate comparison, the simulations are stopped after the same number of steps, and  $\Delta x$  is kept constant. Highest  $\Delta t$  corresponds to  $C_{cfl} = 0.001$ . Initial conditions was a step function. Here for WAF advection.

$nx = 500, nsteps = 10000$



**Figure 29:** Convergence Study with respect to the Courant number  $C_{cfl}$  using the  $L1$  norm.

Points are measurements, the straight lines just connect the dots. For an accurate comparison, the simulations are stopped after the same number of steps, and  $\Delta x$  is kept constant. Initial conditions was a Gaussian profile. Here for WAF advection.



**Figure 30:** Convergence Study with respect to the Courant number  $C_{cfl}$  using the  $L1$  norm.

Points are measurements, the straight lines just connect the dots. For an accurate comparison, the simulations are stopped after the same number of steps, and  $\Delta x$  is kept constant. Initial conditions was a step function. Here for WAF advection.

## 2.4 Conclusions

- Advection is diffusive (fig 1).
- It is diffusive even if  $C_{CFL} = 1!$  (fig 3). This is most probably because of round-off/float errors. For high  $t$ , the whole shape moves a bit to the right even.
- Using a lower CFL number leads to stronger diffusion. Compare figs. 1 and 2. Why?

We are solving the 1D advection equation with  $c$  being a constant velocity:

$$\frac{\partial u}{\partial t} + c \frac{\partial u}{\partial x} = 0 \quad (1)$$

Discretising this equation, we get (using an explicit time scheme and upwind differencing):

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} + \frac{u_i^n - u_{i-1}^n}{\Delta x} = 0 \quad (2)$$

This is however not an exact expression, but only an approximate one. If we use a Taylor expansion

$$u_i^{n+1} = u_i^n + \Delta t \frac{\partial u}{\partial t} + \frac{\Delta t^2}{2} \frac{\partial^2 u}{\partial t^2} + O(\Delta t^3) \quad (3)$$

$$u_{i-1}^n = u_i^n - \Delta x \frac{\partial u}{\partial x} + \frac{\Delta x^2}{2} \frac{\partial^2 u}{\partial x^2} + O(\Delta x^3) \quad (4)$$

and insert it into eq 2, we get (neglecting third order terms from now on)

$$\frac{\partial u}{\partial t} + c \frac{\partial u}{\partial x} + \frac{1}{2} \Delta t \frac{\partial^2 u}{\partial t^2} - c \frac{\Delta x}{2} \frac{\partial^2 u}{\partial x^2} = 0 \quad (5)$$

$$\Rightarrow \frac{\partial u}{\partial t} + c \frac{\partial u}{\partial x} = -\frac{1}{2} \Delta t \frac{\partial^2 u}{\partial t^2} + c \frac{\Delta x}{2} \frac{\partial^2 u}{\partial x^2} \quad (6)$$

$$= 0 + Err \quad (7)$$

$$Err = -\frac{1}{2} \Delta t \frac{\partial^2 u}{\partial t^2} + c \frac{\Delta x}{2} \frac{\partial^2 u}{\partial x^2} \quad (8)$$

which is the advection equation 1 plus some error term.

Now using eq. 1 we find:

$$\frac{\partial u}{\partial t} + c \frac{\partial u}{\partial x} = 0 \quad (9)$$

$$1) \quad \frac{\partial^2 u}{\partial t^2} + c \frac{\partial^2 u}{\partial t \partial x} = 0 \quad (10)$$

$$2) \quad \frac{\partial u}{\partial t} = -c \frac{\partial u}{\partial x} \quad (11)$$

$$\Rightarrow 3) \quad \frac{\partial^2 u}{\partial t^2} - c^2 \frac{\partial^2 u}{\partial x^2} = 0 \quad (12)$$

This gives us for the error term:

$$Err = -\frac{1}{2}\Delta t \frac{\partial^2 u}{\partial t^2} + c \frac{\Delta x}{2} \frac{\partial^2 u}{\partial x^2} \quad (13)$$

$$= -\frac{c^2 \Delta t}{2} \frac{\partial^2 u}{\partial x^2} + c \frac{\Delta x}{2} \frac{\partial^2 u}{\partial x^2} \quad (14)$$

$$= \frac{c \Delta x}{2} \left(1 - \frac{c \Delta t}{\Delta x}\right) \frac{\partial^2 u}{\partial x^2} \quad (15)$$

Inserting the CFL condition:

$$\frac{c \Delta t_{max}}{\Delta x} = C_{cfl} \leq 1 \quad (16)$$

we obtain:

$$Err = \frac{c \Delta x}{2} (1 - C_{cfl}) \frac{\partial^2 u}{\partial x^2} \quad (17)$$

The second derivative in space is characteristic for diffusion. But you can immediately see that the diffusion term (coefficient) depends on  $C_{cfl}$ , and indeed increases with decreasing  $C_{cfl}$ !

- 2D advection:

For advection only in x or y direction, respectively, the method reduces to the one dimensional case, and the results are identical when using Strang splitting or the naive case. Compare figs 5 vs 6, and 7 vs 8. But when we have both  $v_x$  and  $v_y$  be non-zero, the naive method (i.e. without Strang splitting, where we just extend the 1D method to 2D and don't think about it) starts getting "stripes" perpendicular to the diagonal along which it is being advected along (fig 9 vs 10). The reason is that a) the upwinding is not complete, i.e. the value  $u_{i-1,j-1}$  along the diagonal in the naive case never gets properly advected to  $u_{i,j}$ , and b) the method is diffusive, so the diffused material from  $u_{i-1,j}$  and  $u_{i,j-1}$  come together in that cell, messing everything up. Letting the code run for longer times actually leads to stripe-like instabilities. See also fig 16.

- The piecewise linear scheme can/will introduce oscillations around sharp edges (fig 11. The oscillations can go into the negative regime. Even though it's unphys-

ical, the total density content remains constant! This is because the scheme is fundamentally conservative.

- On the order of convergence:

- Dependence on  $\Delta x$ :

For the smooth Gaussian profile IC: The results are nicely as expected, fig. 19. The piecewise constant advection is of  $\mathcal{O}(\Delta x)$ , piecewise linear is of  $\mathcal{O}(\Delta x^2)$ . Remember that the order of convergence computation on paper gives you only the upper boundary of the error, so a faster convergence is possible, but shouldn't be greater than 1 order than predicted, which we have here.

For the step function in fig 20 we see as expected that the convergence rate drops to  $\mathcal{O}(\Delta x)$  because that's what slope limiters do. So then why is the unlimited linear method also dropping to first order? The computations we did on paper for the order estimate assume a smooth initial condition, which we don't have here. It turns out having a discontinuity drops your order of convergence.

- Dependence on  $\Delta t$ :

Fixing the time step  $\Delta t$  is essentially the same as fixing the Courant number  $C_{cfl}$  for an already fixed velocity  $u$  and grid spacing  $\Delta x$ . For the order of convergence to be measured w.r.t.  $\Delta t$ , we must start with a  $\Delta t$  that implies a very low  $C_{cfl}$ . The reason is that  $C_{cfl}$  determines the amplitude of the diffusivity, see eqn. 17 and fig. 23. So for a good comparison, we need to start with a low enough  $C_{cfl}$  such that the difference in diffusivity is negligible. Otherwise, you don't see the power law that emerges.

In the following analysis, let's focus on piecewise constant (first order) advection alone. I don't have the theory present for the piecewise linear scheme to back up my findings. What makes things more difficult is that for the piecewise linear advection,  $\Delta t$  also enters the computation of the fluxes between the cells, thus also affects the spatial component. It is not trivial to separate between the  $\Delta x$  and  $\Delta t$  dependence in these cases.

For the Gaussian profile, fig. 21, we get a nice power law with slope 1, as expected. (The piecewise linear methods get close to  $\mathcal{O}(\Delta t^{1/2})$ . Maybe because they start developing jump discontinuities? Compare fig 42.) For the step profile of the piecewise constant advection, fig. 22, we get  $\mathcal{O}(\Delta t^{1/2})$ . The reason behind it is that the convergence analysis in theory assumes a smooth initial condition, such that we can use derivatives and Taylor-expansions.

The step function is not smooth though. It can be shown that for a jump discontinuity, the error goes as

$$Err \propto \sqrt{t}$$

So if we have  $N$  steps of equal size  $\Delta t$ , if we keep  $N$  constant, we get

$$Err \propto \sqrt{\Delta t} \tag{18}$$

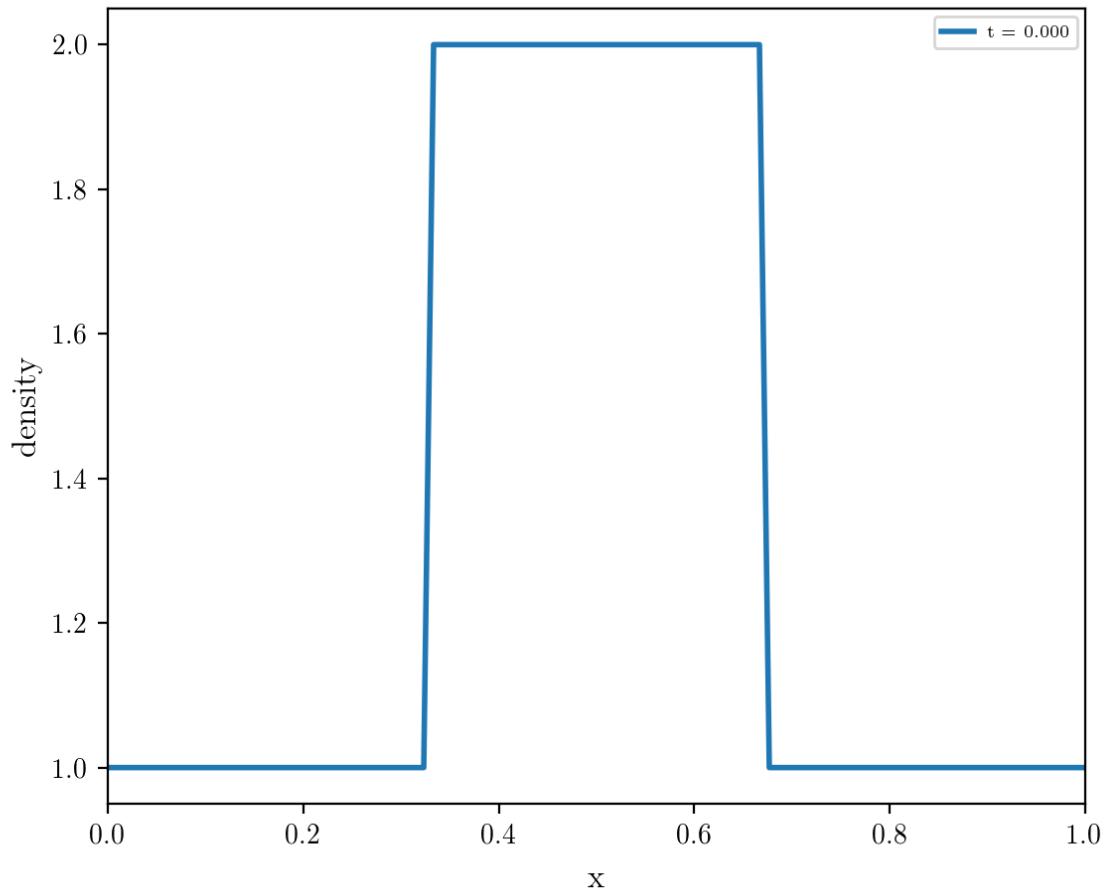
See LeVeque [2002], chapter 8.6 for details.

which is exactly what we see in fig. 22.

- Dependence on  $C_{CFL}$ :

Tweaking around  $C_{CFL}$  for a fixed  $nx$  and advection velocity  $u$  is essentially the same as tweaking  $\Delta t$ . For  $C_{cfl}$  comparable to 1 however we see the effects of the increased diffusivity, as described by equation 17. Indeed, measuring the convergence of  $0.1 \leq C_{cfl} \leq 0.9$  in figs. 23 and 24 shows that it doesn't behave like a power law at all.

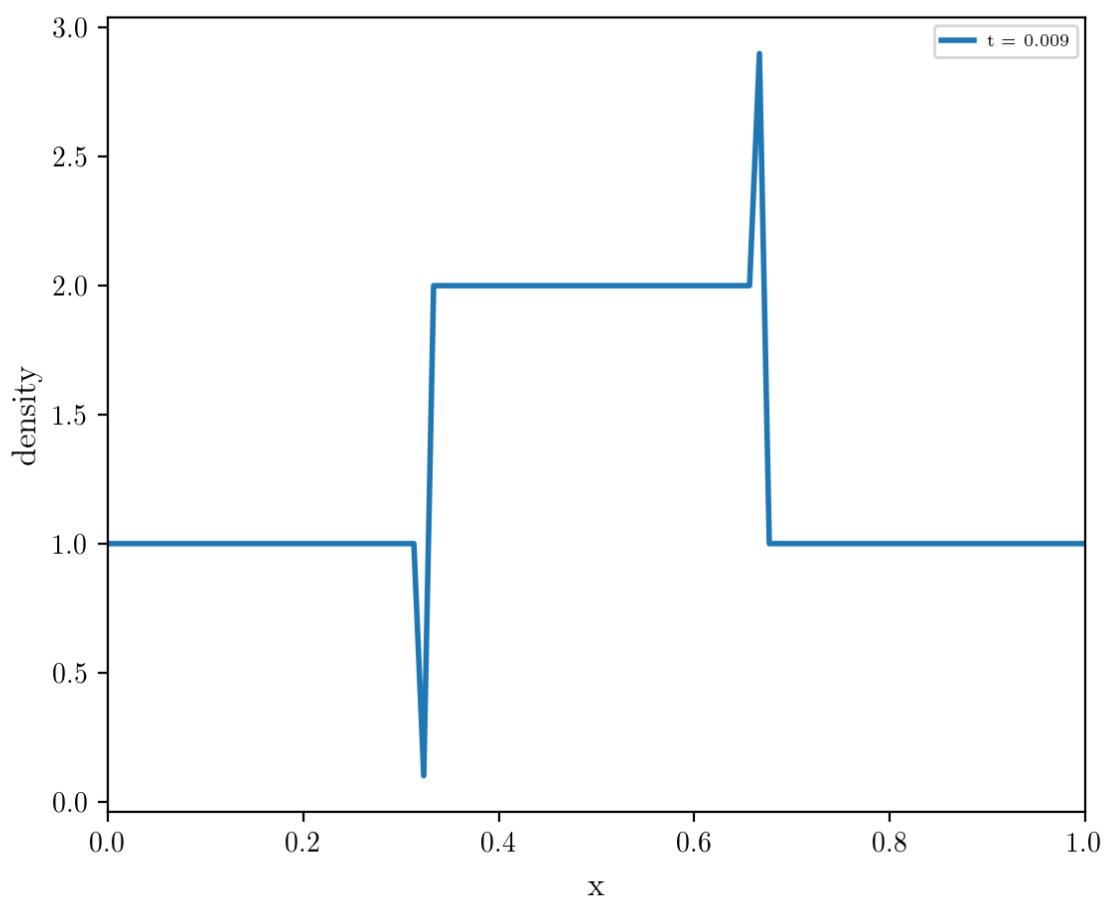
## 2.5 Future Debugging Hints



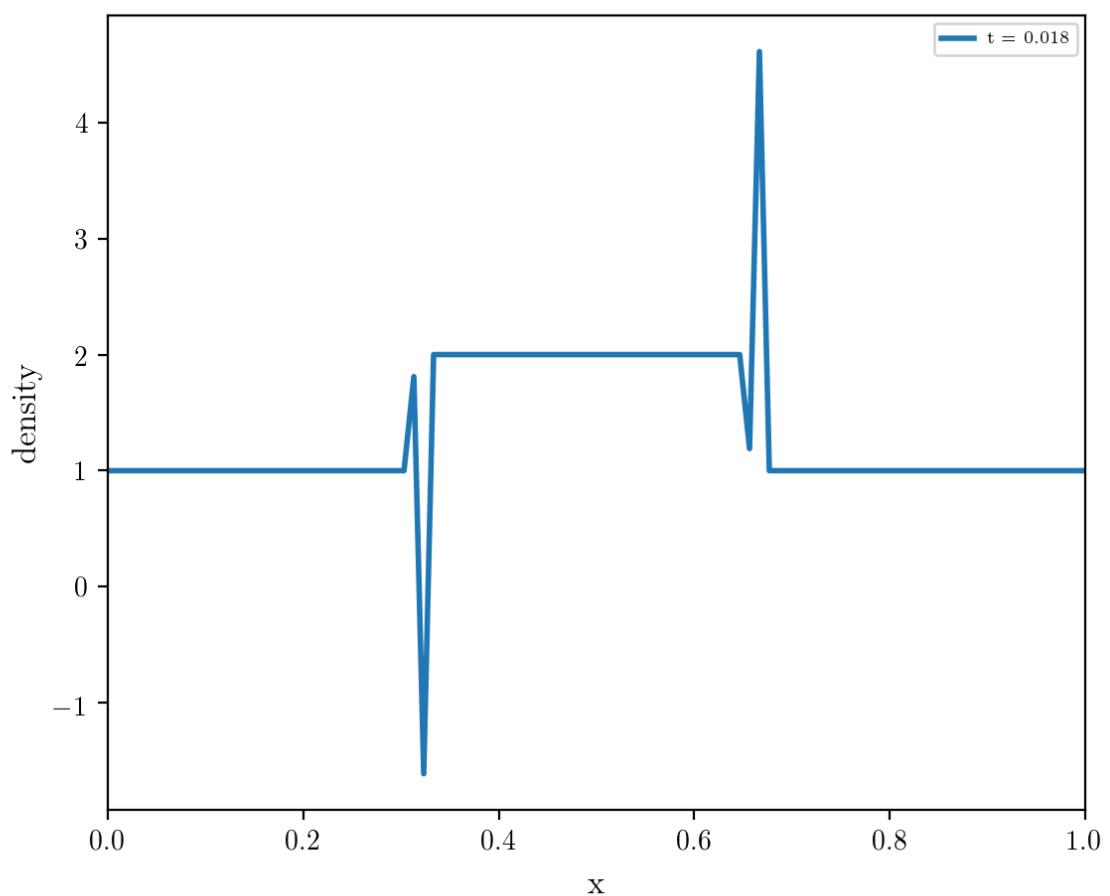
**Figure 31:** Initial conditions used to demonstrate debugging hints.  $u = 1$ .

### 2.5.1 When you're using downwind differencing

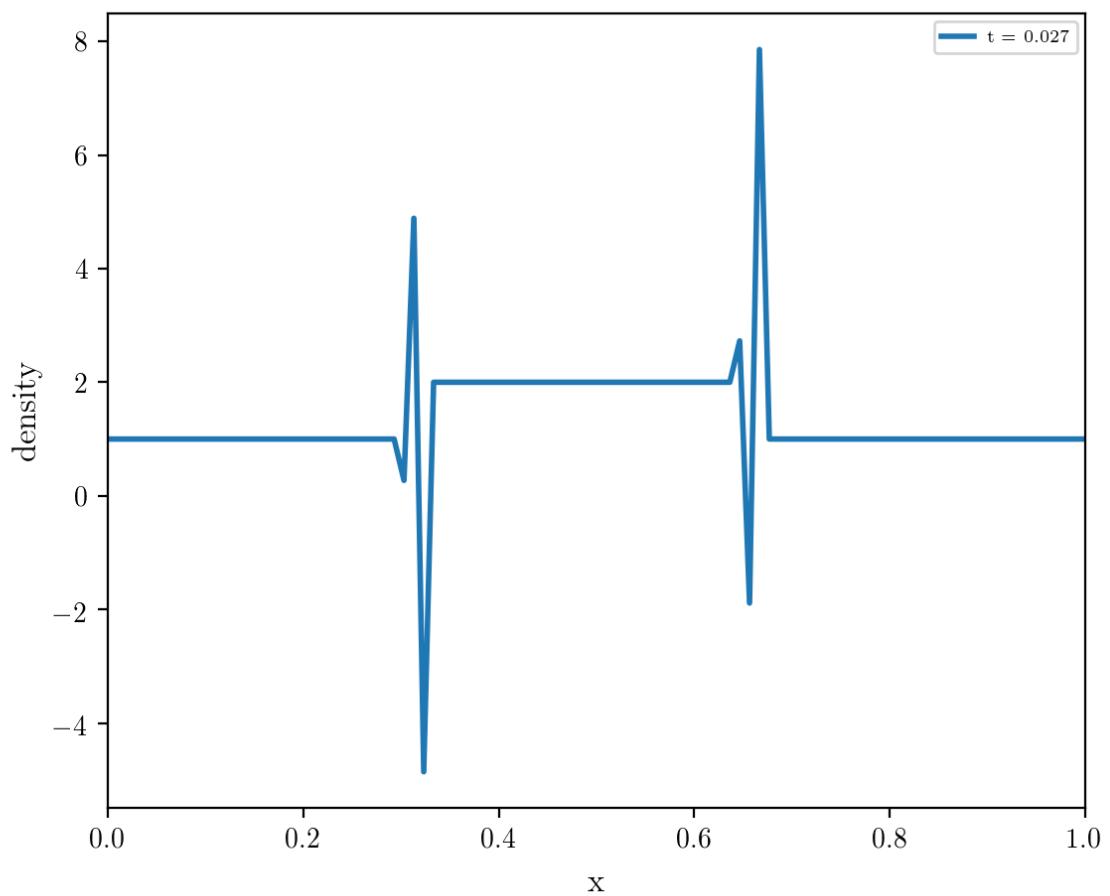
Downwind differencing is unphysical and violently unstable. Note how the wave goes in the wrong direction!



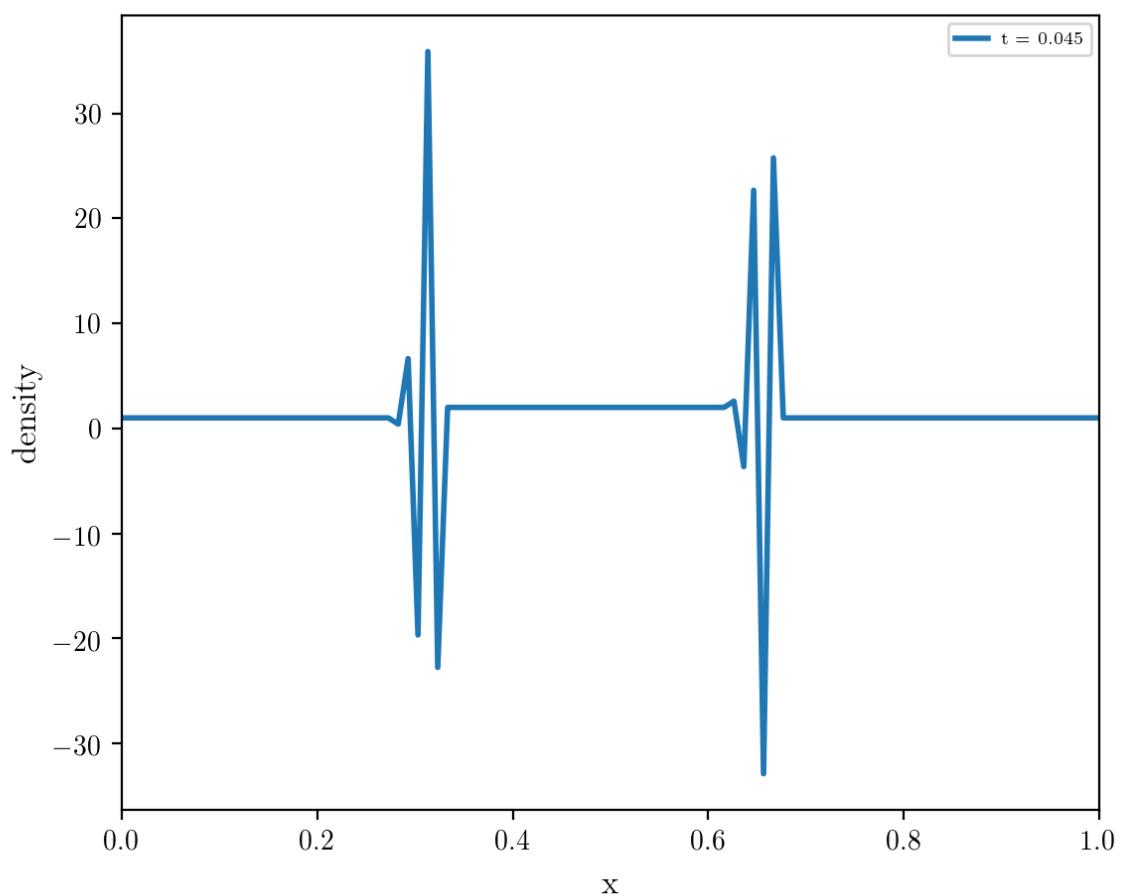
**Figure 32:** Downwind differencing after 1 step.



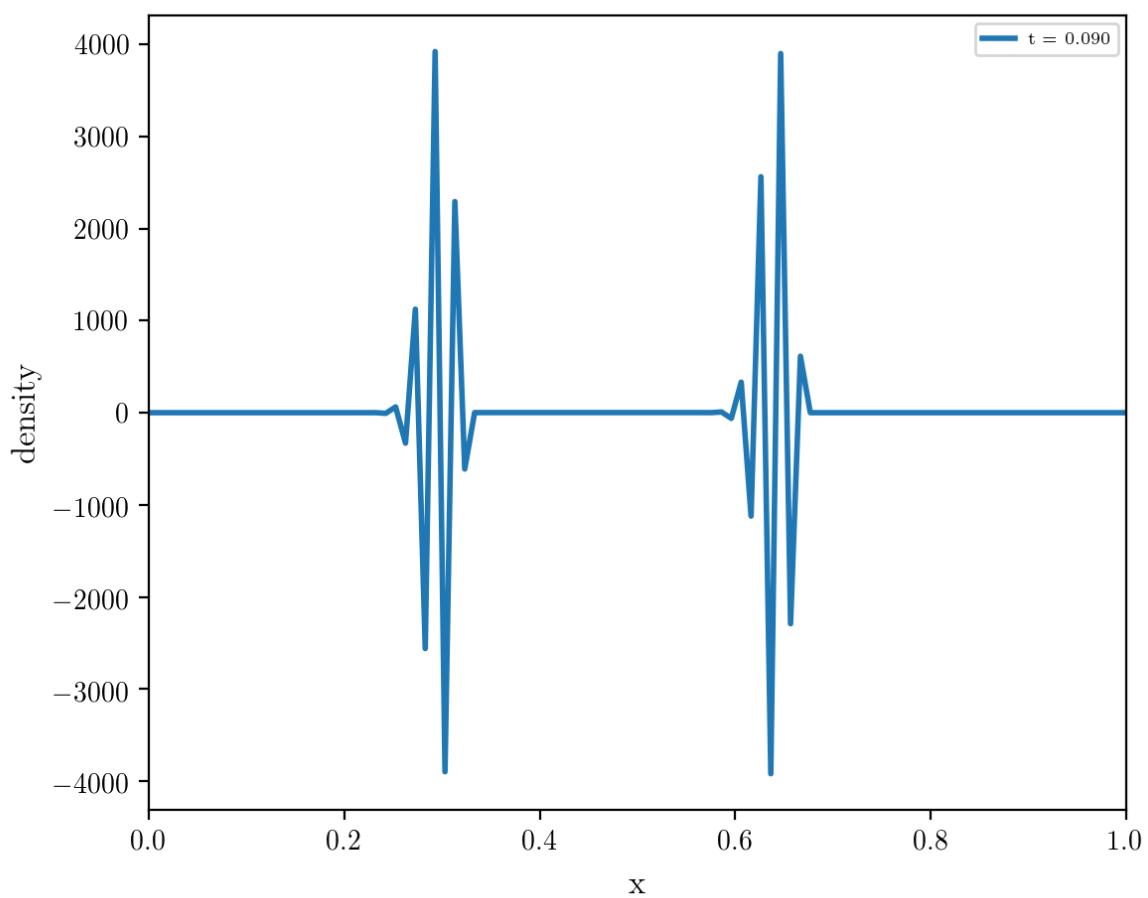
**Figure 33:** Downwind differencing after 2 steps.



**Figure 34:** Downwind differencing after 3 steps.

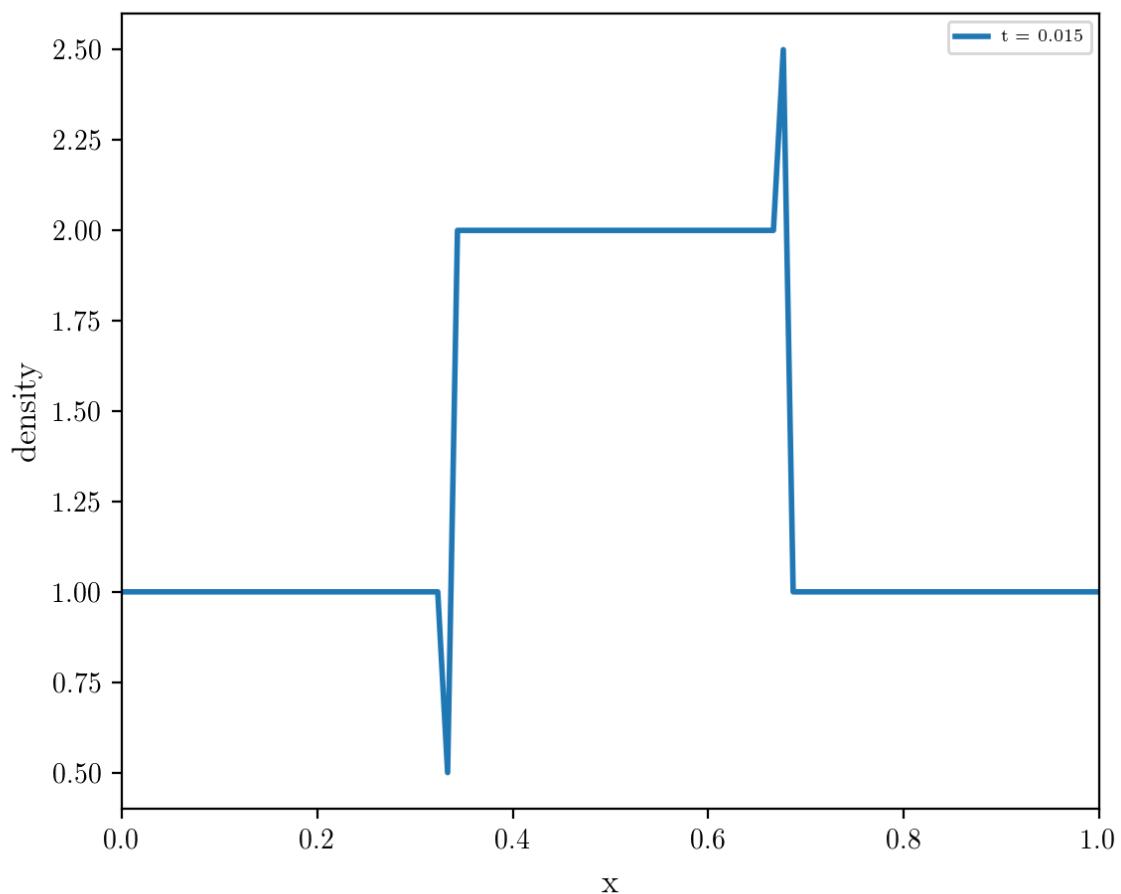


**Figure 35:** Downwind differencing after 5 steps.

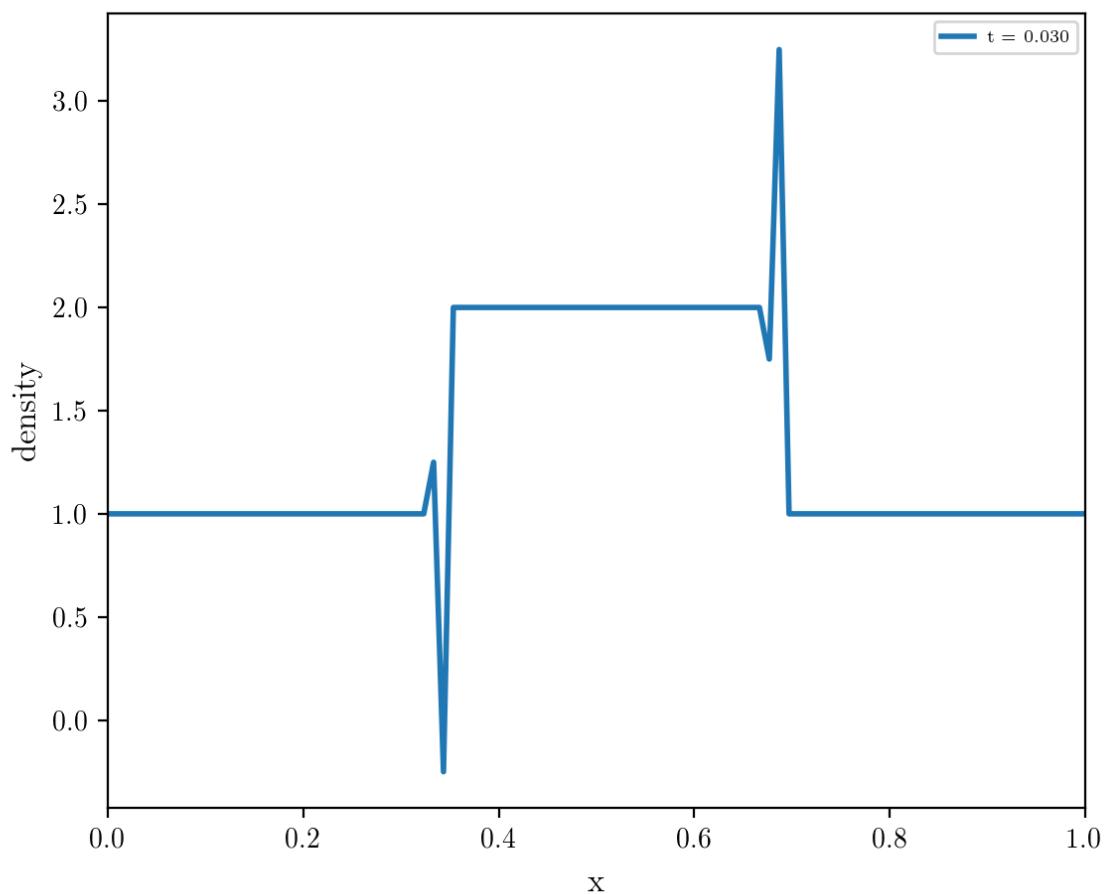


**Figure 36:** Downwind differencing after 10 steps.

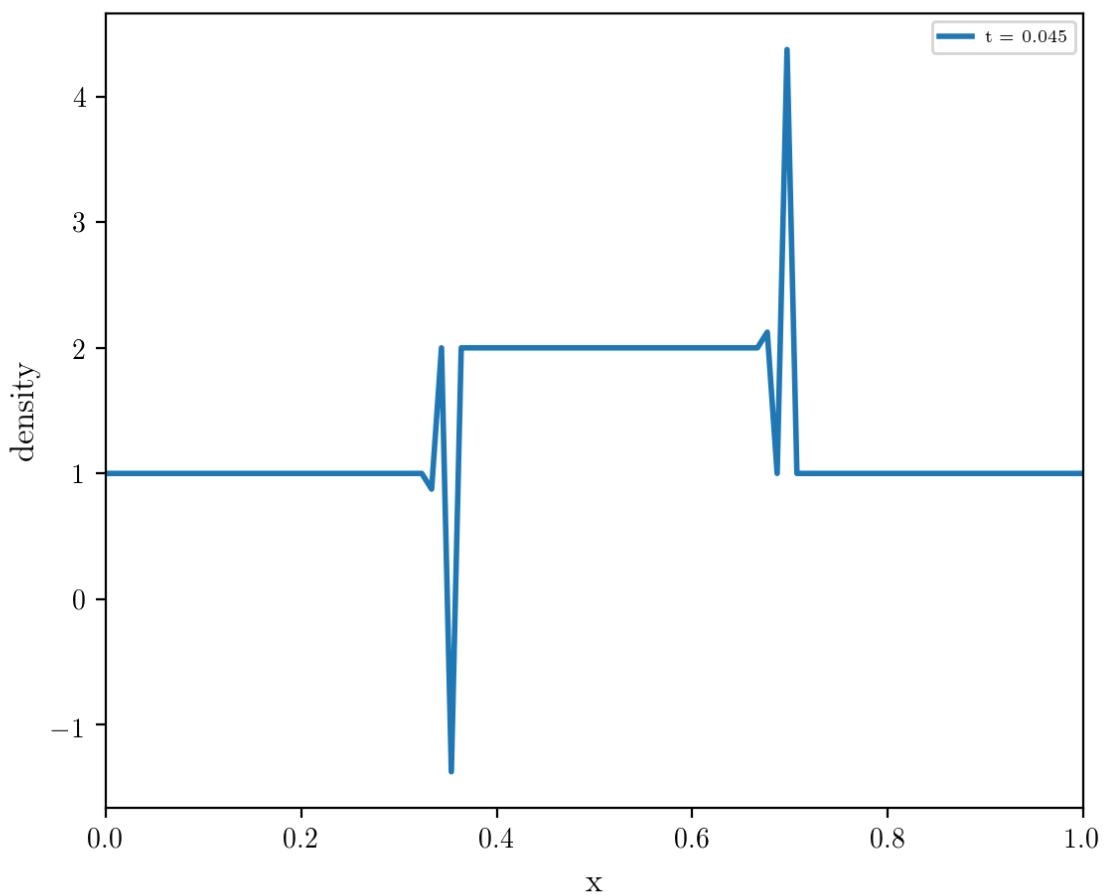
### 2.5.2 When the CFL condition is violated



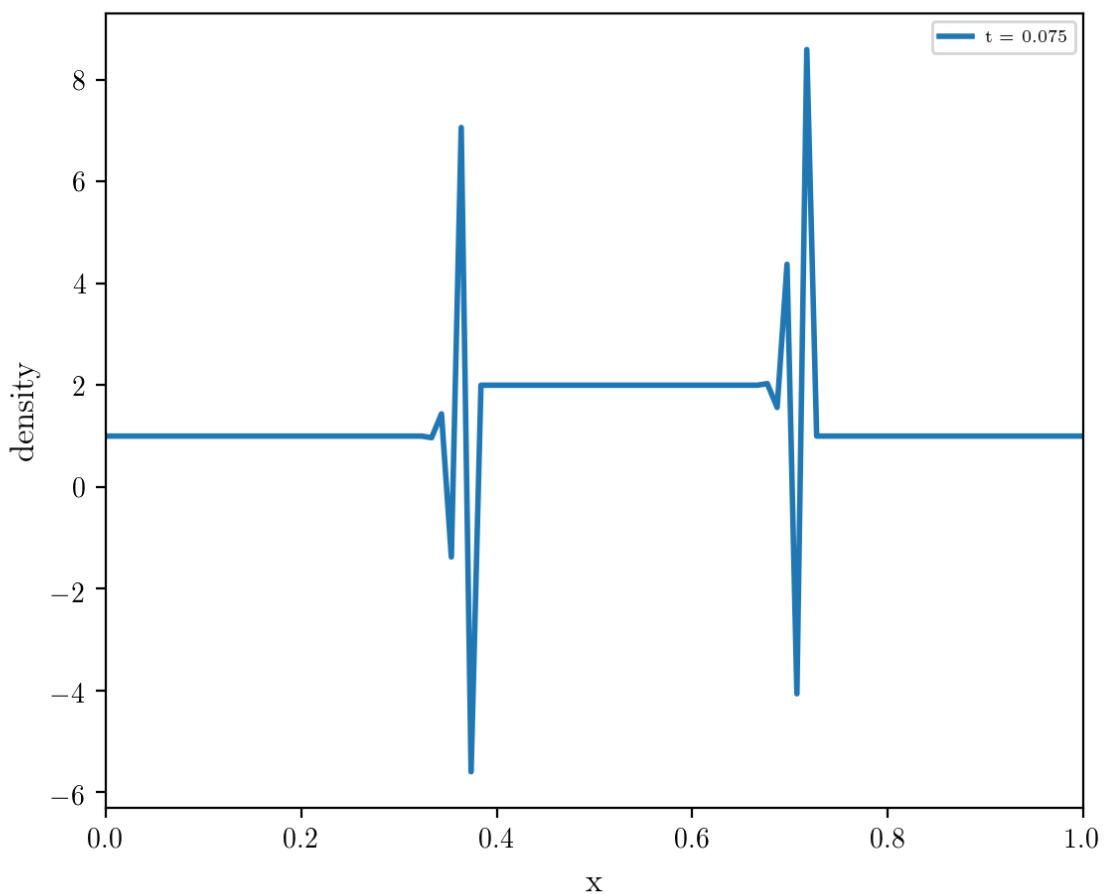
**Figure 37:** Violating the CFL condition (here  $C_{cfl} = 1.5$ ) after 1 step.



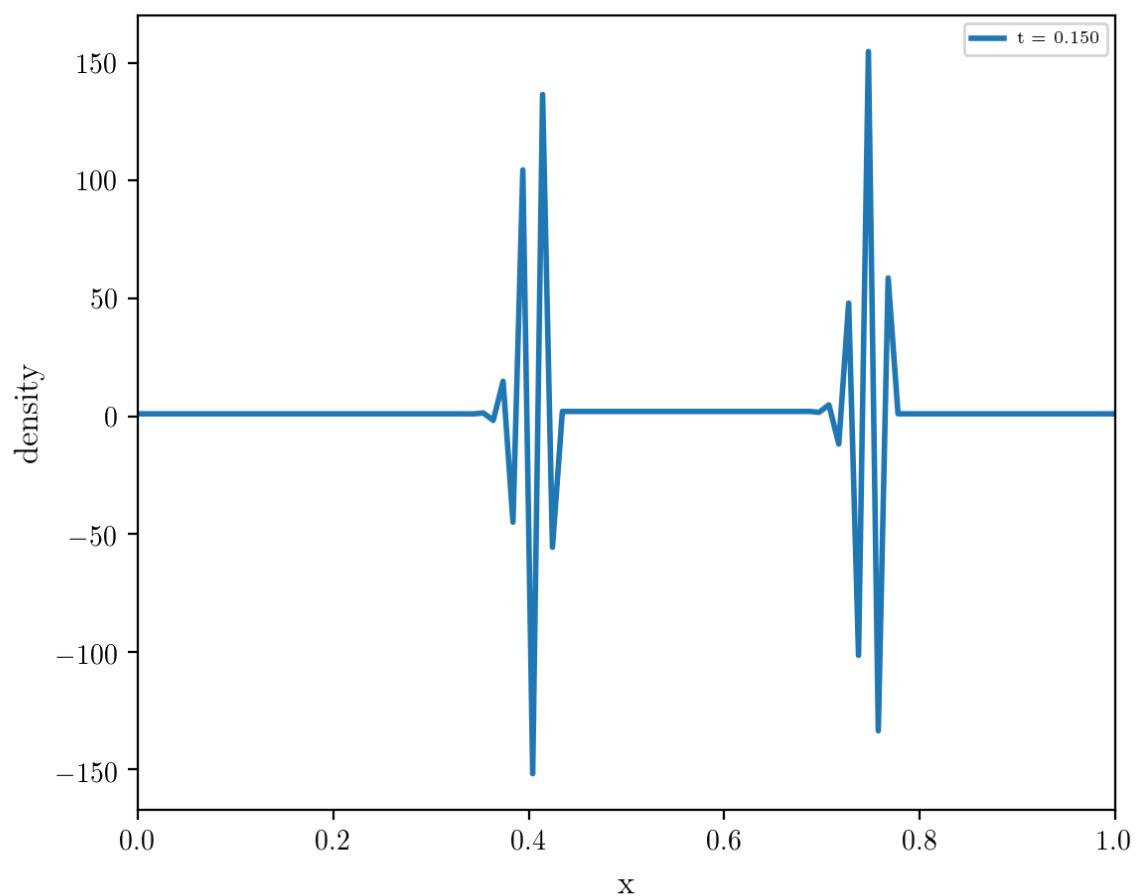
**Figure 38:** Violating the CFL condition (here  $C_{cfl} = 1.5$ ) after 2 steps.



**Figure 39:** Violating the CFL condition (here  $C_{cfl} = 1.5$ ) after 3 steps.



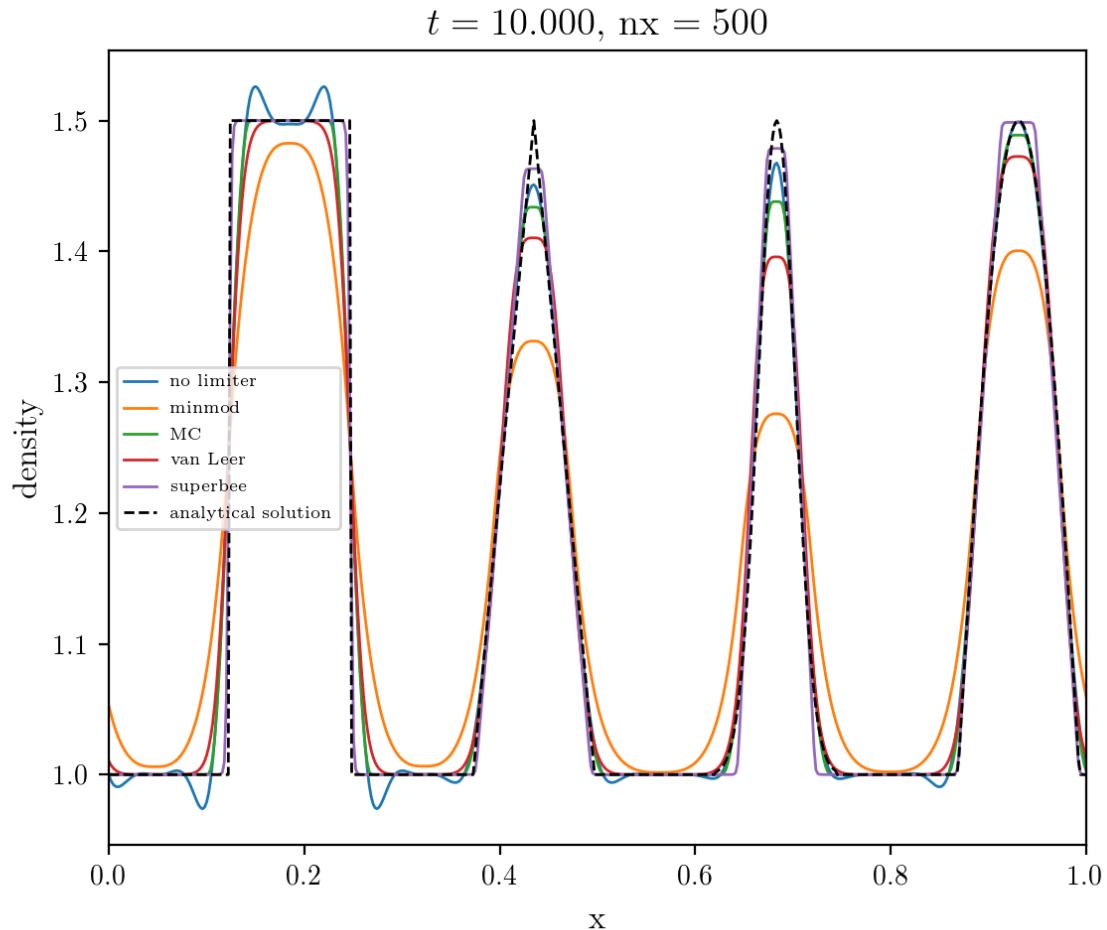
**Figure 40:** Violating the CFL condition (here  $C_{cfl} = 1.5$ ) after 5 steps.



**Figure 41:** Violating the CFL condition (here  $C_{cfl} = 1.5$ ) after 10 steps.

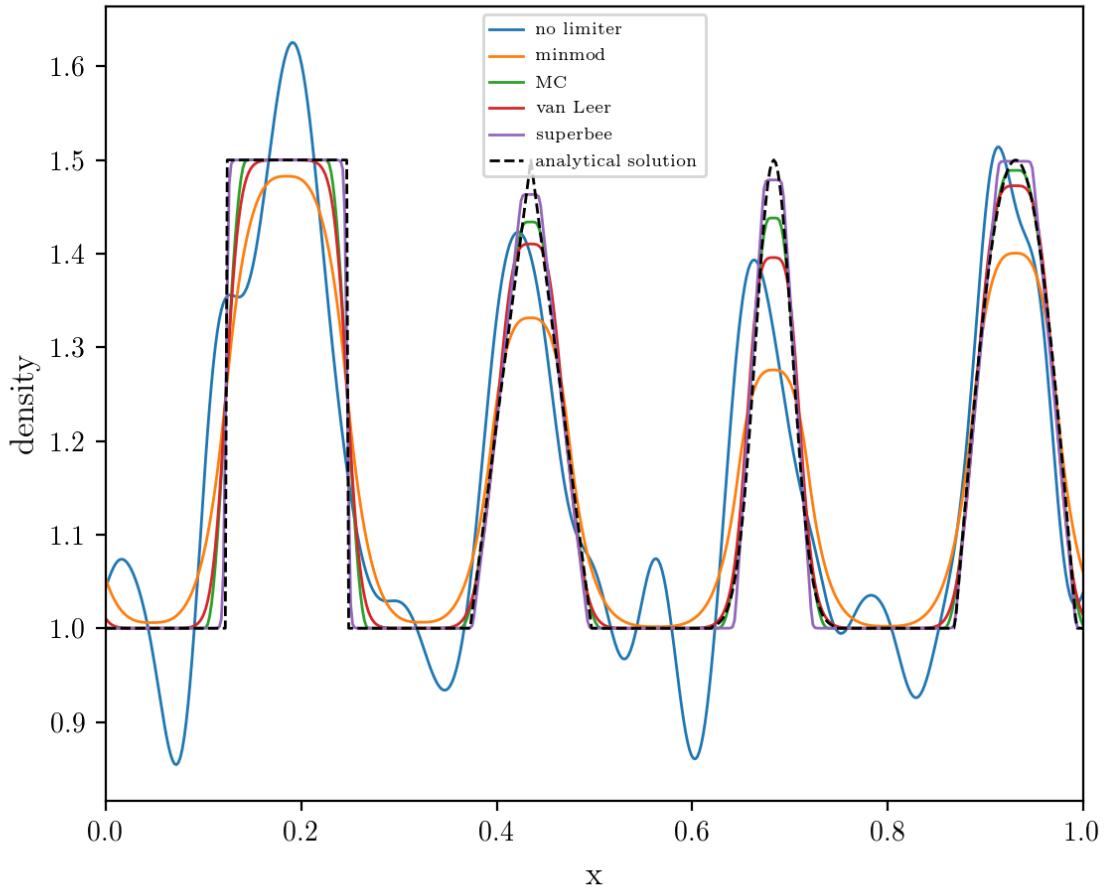
### 3 Slope and Flux Limiters

#### 3.0.1 Effects on linear advection



**Figure 42:** The effect of different slope limiters on linear advection, applied to piecewise linear advection

$t = 10.000, nx = 500$

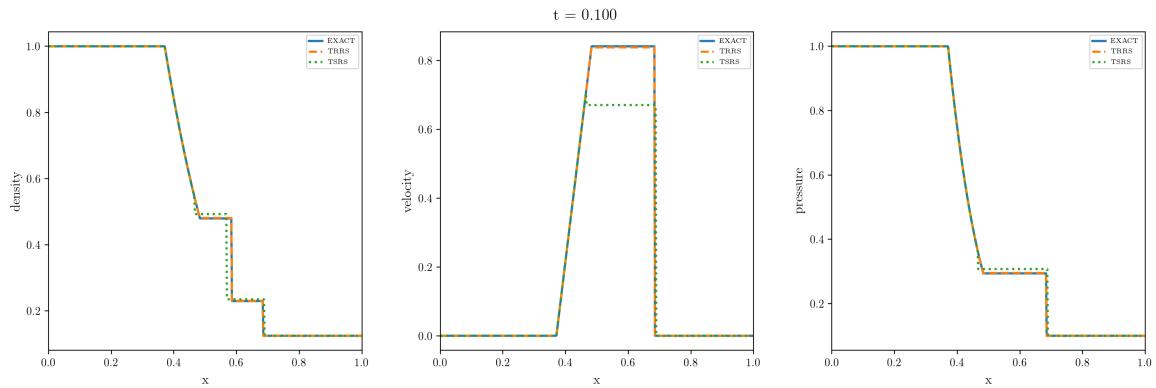


**Figure 43:** The effect of different slope limiters on linear advection, applied to WAF advection

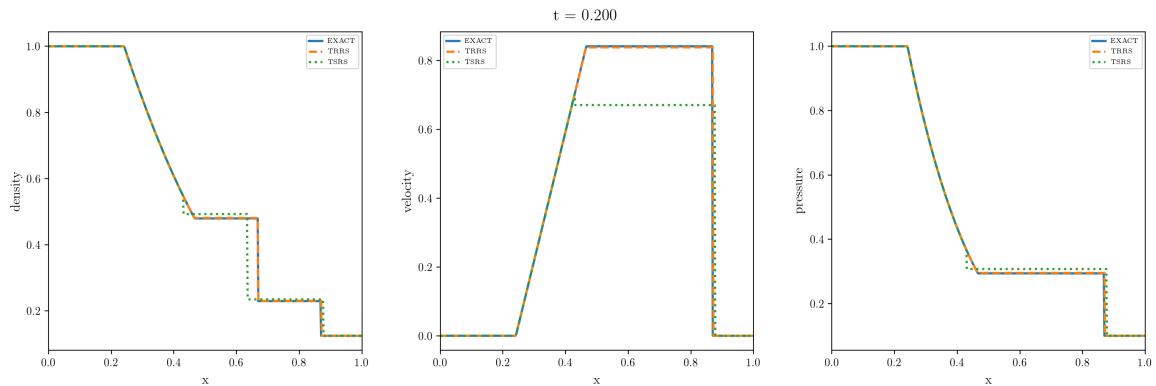
- All limiters except superbee still contain diffusion. You can't get rid of it entirely, but we got rid of the oscillations.
- The minmod resembles the solution of the piecewise constant advection, but pay attention that this is at much later times!
- Some limiters flatten continuous maxima. Van leer, then MC, then superbee in order of ascending “flattening”
- It's as if superbee tries to produce jump discontinuities
- For order of convergence study, see figs. 19 - 30, and discussion in section 2.4.

## 4 Riemann Solvers

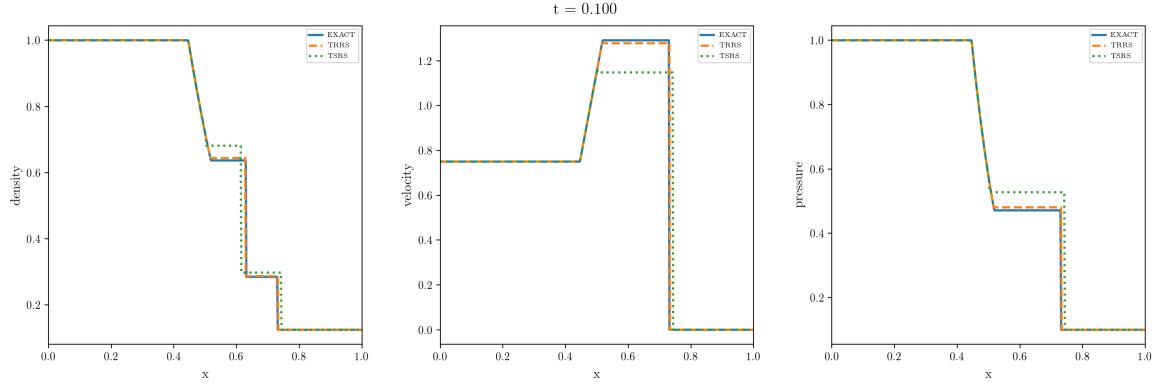
### 4.1 Approximate Riemann Solutions



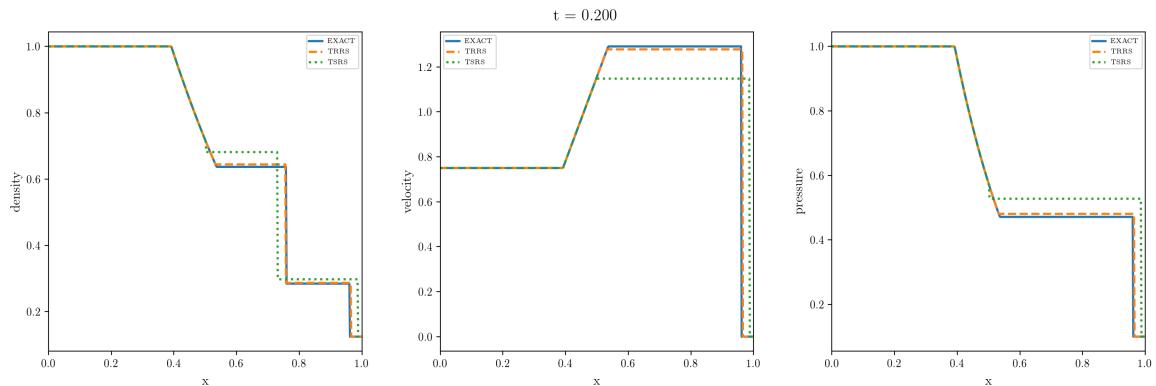
**Figure 44:** The Sod test solved using the exact and approximate Riemann solvers



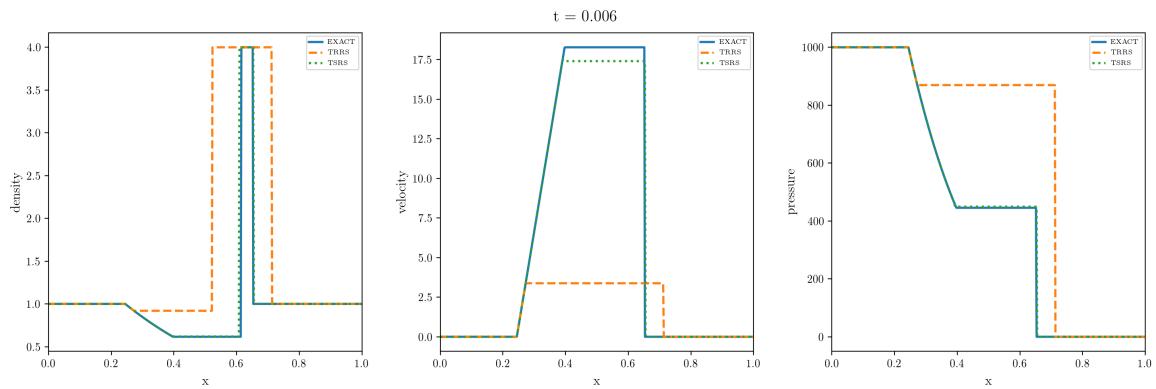
**Figure 45:** The Sod test solved using the exact and approximate Riemann solvers, at a later time



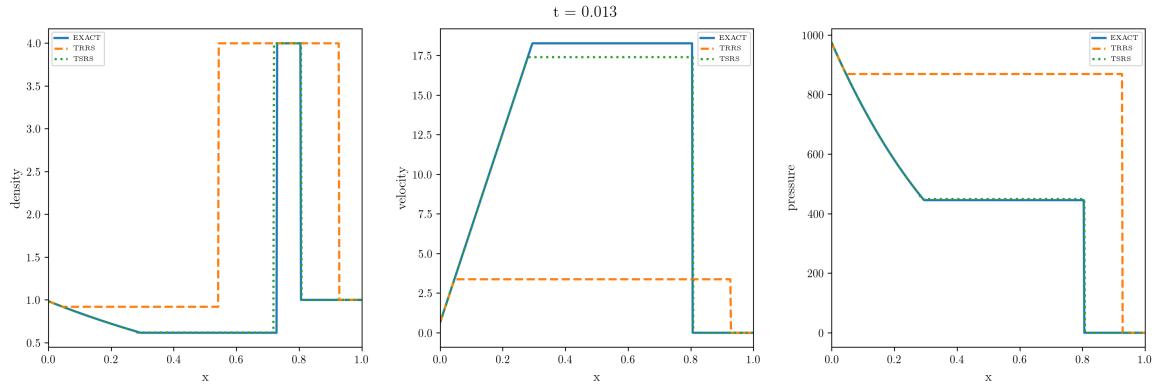
**Figure 46:** The modified Sod test solved using the exact and approximate Riemann solvers



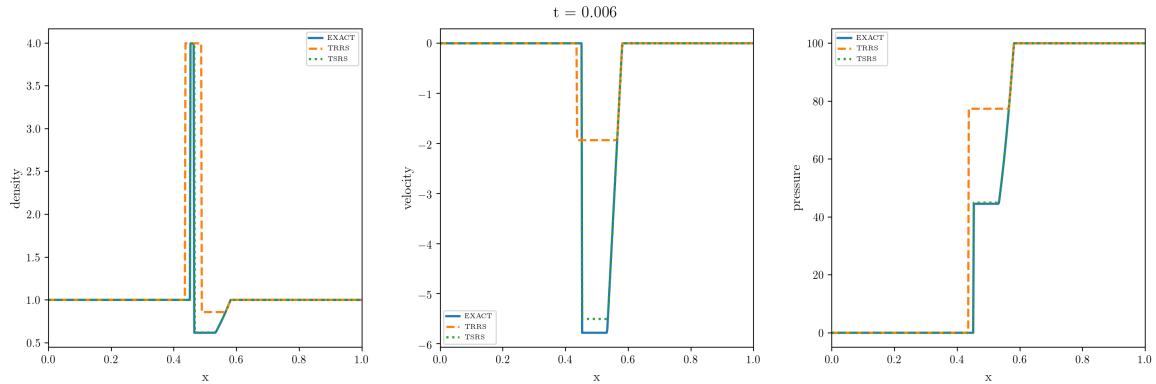
**Figure 47:** The modified Sod test solved using the exact and approximate Riemann solvers, at a later time



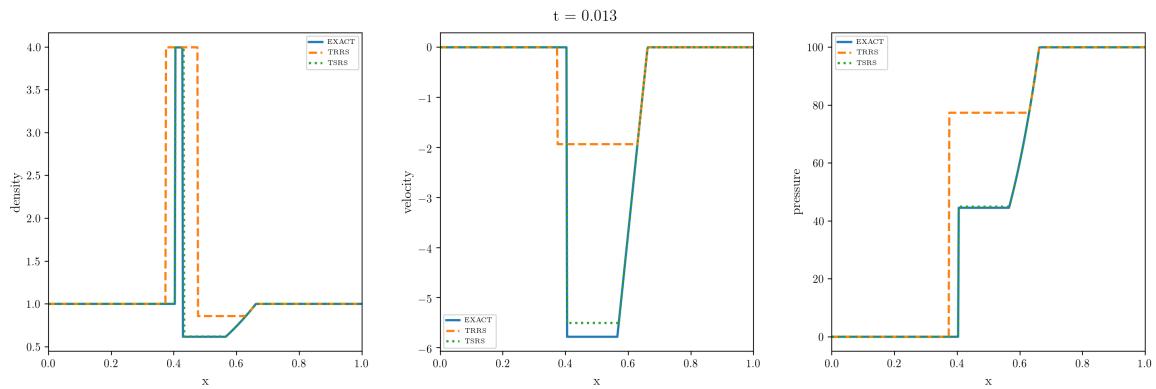
**Figure 48:** The left blast wave solved using the exact and approximate Riemann solvers



**Figure 49:** The left blast wave solved using the exact and approximate Riemann solvers, at a later time



**Figure 50:** The left blast wave solved using the exact and approximate Riemann solvers



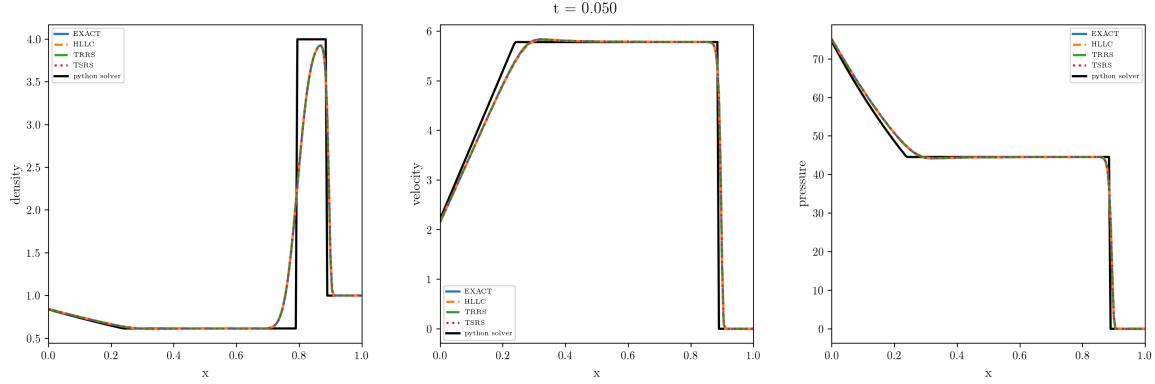
**Figure 51:** The left blast wave solved using the exact and approximate Riemann solvers, at a later time

## 4.2 Conclusions

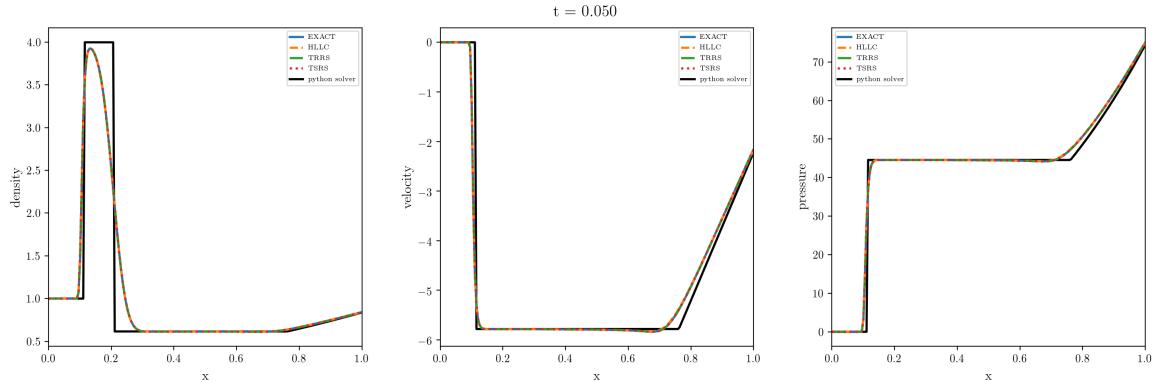
- The TSRS solver may have trouble with rarefactions, e.g. figs 44 - 47. Which is to be expected, since it assumes that we have two shocks present. It deals much better with shocks, e.g. figs 48 - 51.
- The TRRS solver may have trouble with shocks, e.g. figs 48 - 51. Which is to be expected, since it assumes that we have two rarefactions present. It deals much better with rarefactions, e.g. figs 44 - 47.
- It looks like the results get worse over time. Compare figs 44 vs 45, 46 vs 47, etc. But recall that for the Riemann solver, we only solve the solution once, and then sample the solution for a given  $x$  and  $t$ . So once the four regions are determined initially, all the solver does is “smear them out” while sampling at a later time  $t$ .

## 5 Godunov's Method

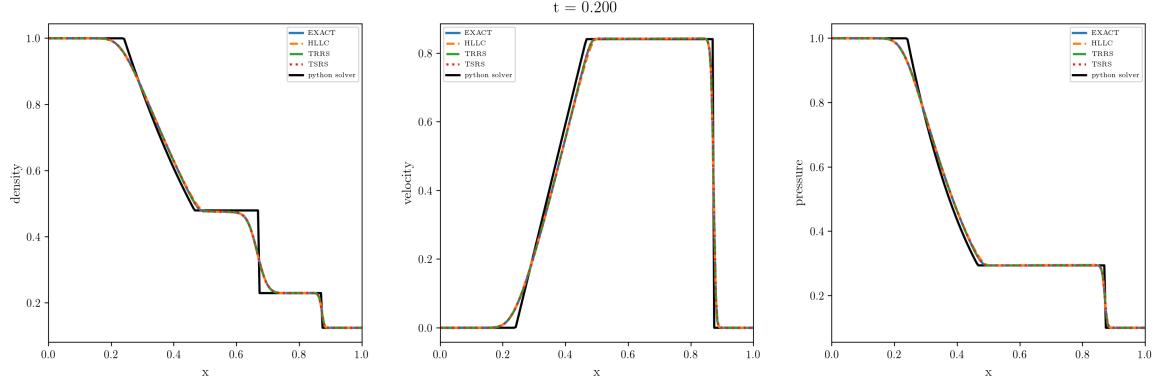
### 5.1 Solving Riemann Problems using Godunov's Method



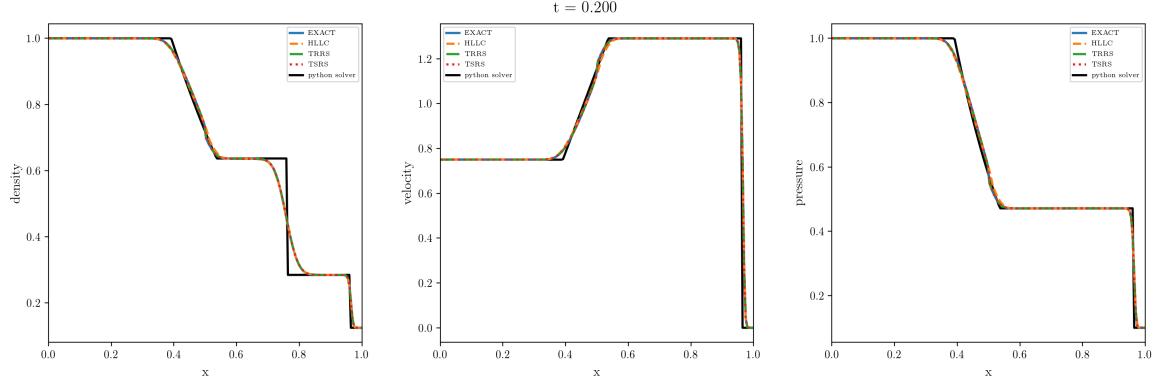
**Figure 52:** Solution to the left blast wave Riemann problem using Godunov's method and various Riemann solvers. The black line is the exact solution.



**Figure 53:** Solution to the right blast wave Riemann problem using Godunov's method and various Riemann solvers. The black line is the exact solution.



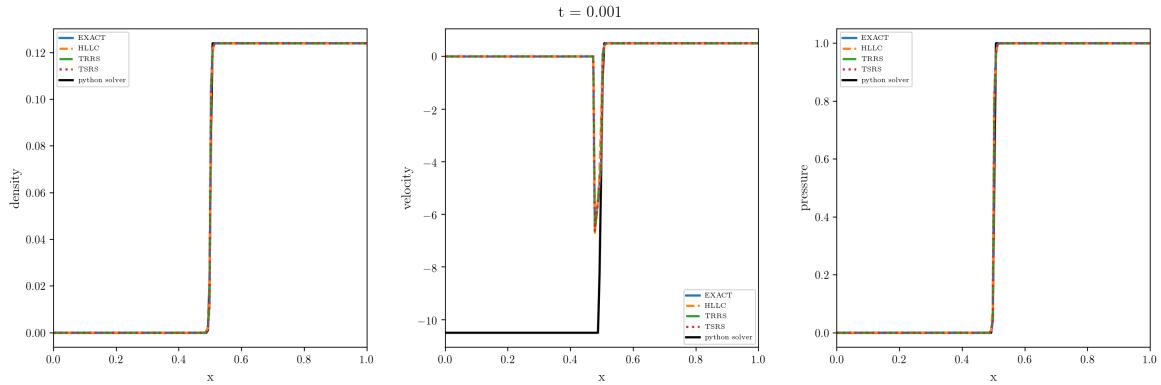
**Figure 54:** Solution to the sod test Riemann problem using Godunov's method and various Riemann solvers. The black line is the exact solution.



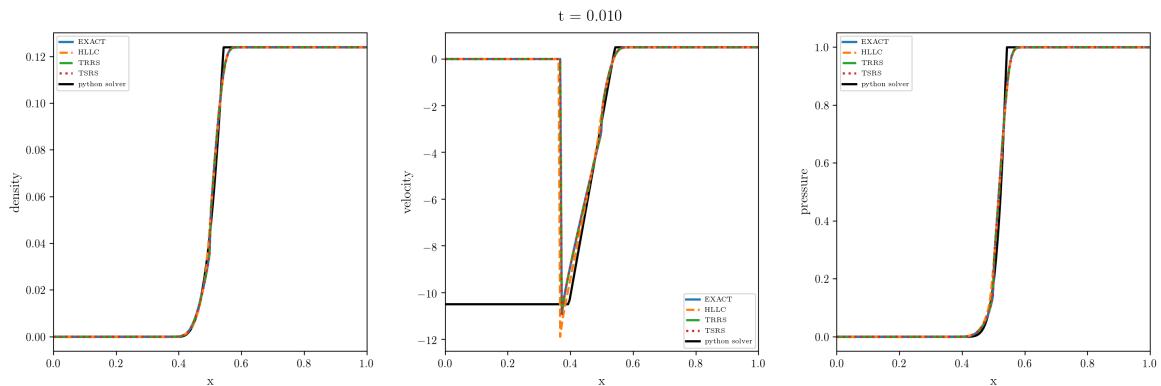
**Figure 55:** Solution to the modified sod test Riemann problem using Godunov's method and various Riemann solvers. The black line is the exact solution.

## 5.2 Solving Vacuum Riemann Problems using Godunov's Method

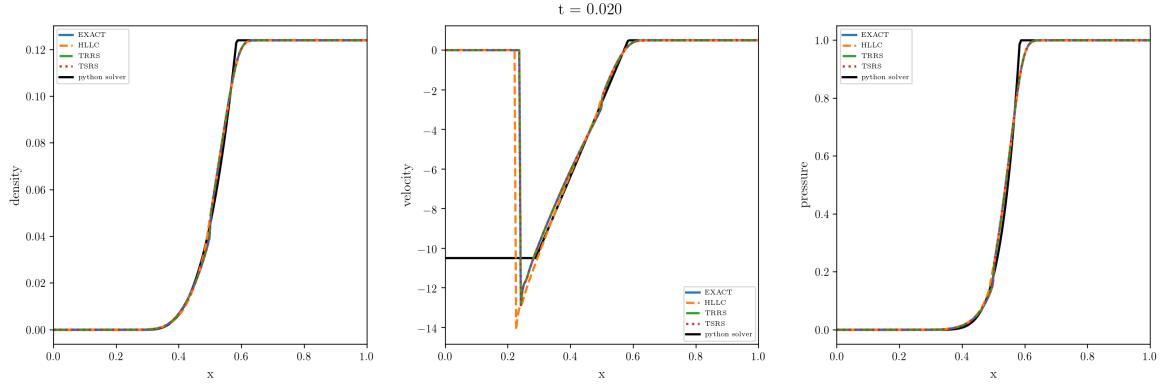
### 5.2.1 Left Vacuum State



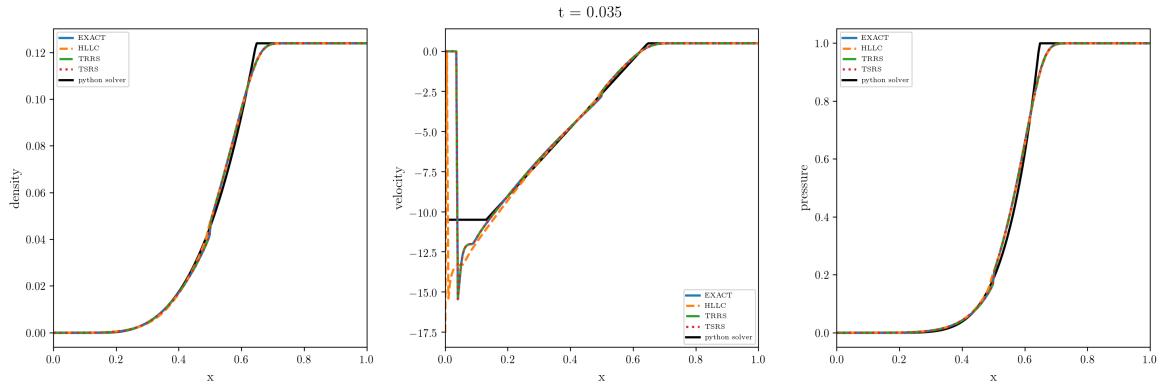
**Figure 56:** Left Vacuum Riemann problem at  $t = 0.001$  for different Riemann solvers using Godunov's method



**Figure 57:** Left Vacuum Riemann problem at  $t = 0.010$  for different Riemann solvers using Godunov's method

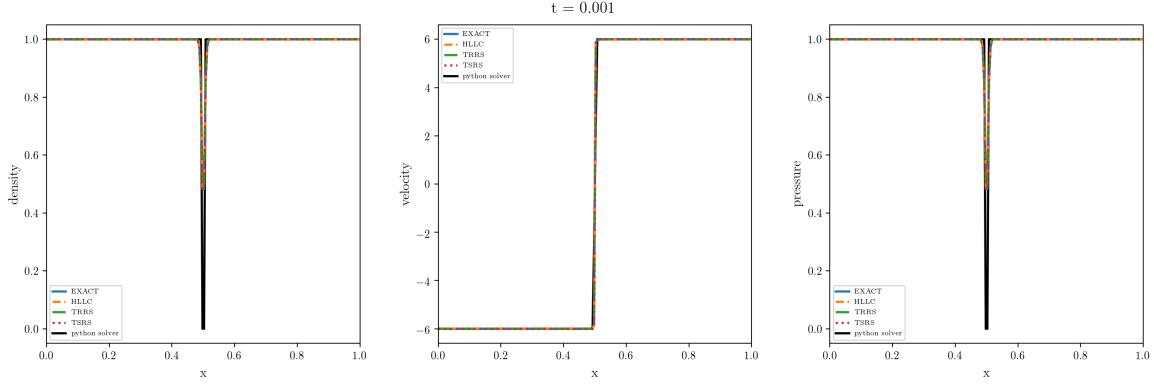


**Figure 58:** Left Vacuum Riemann problem at  $t = 0.02$  for different Riemann solvers using Godunov's method

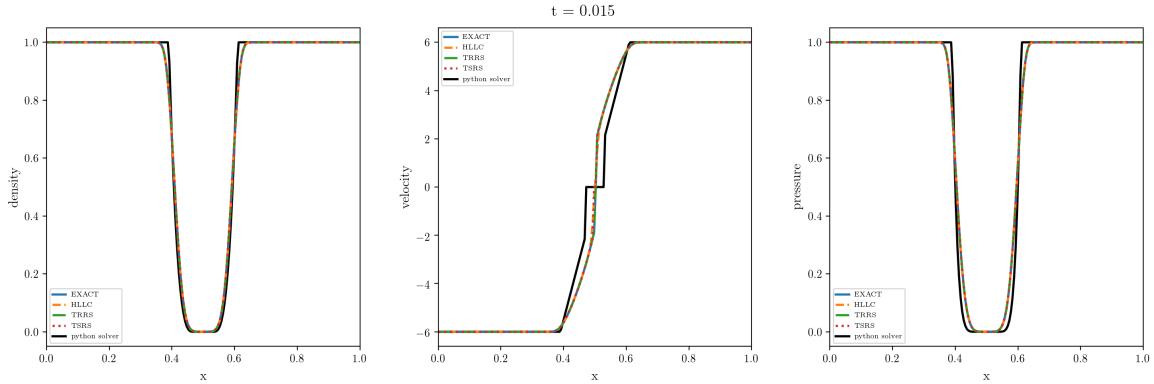


**Figure 59:** Left Vacuum Riemann problem at  $t = 0.035$  for different Riemann solvers using Godunov's method

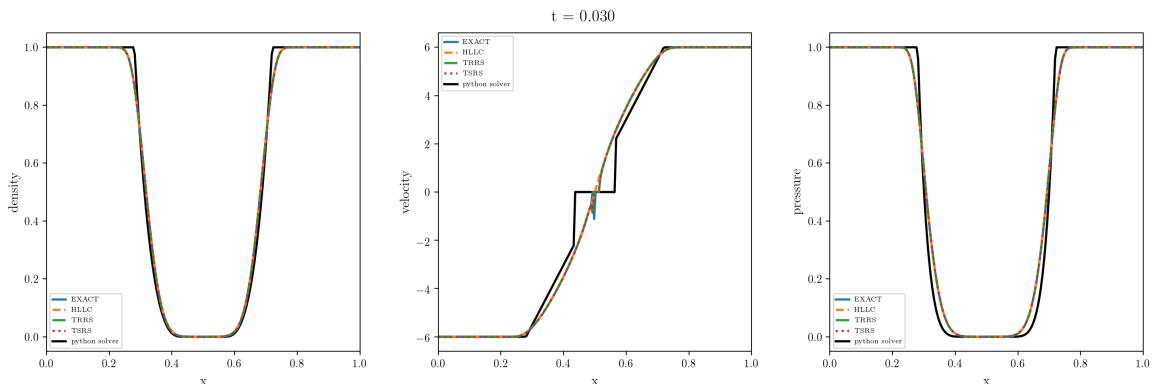
### 5.2.2 Vacuum Generating ICs



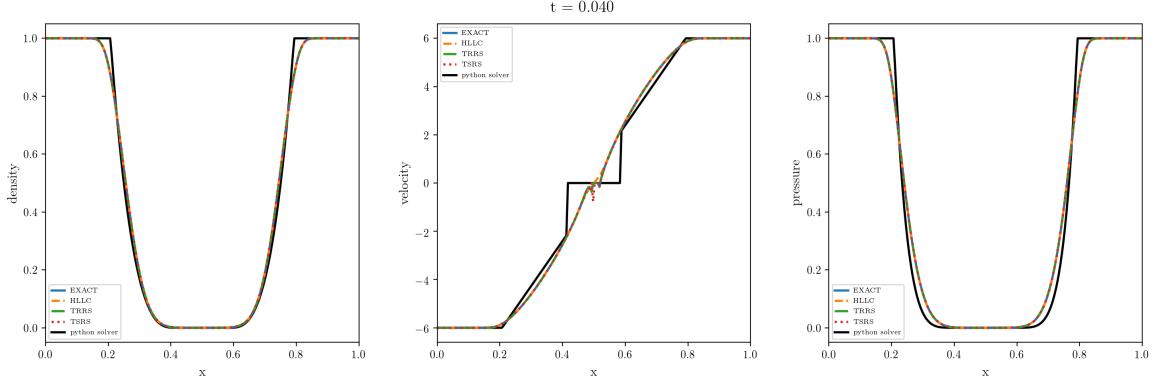
**Figure 60:** Vacuum Generating Riemann problem at  $t = 0.001$  for different Riemann solvers using Godunov's method



**Figure 61:** Vacuum Generating Riemann problem at  $t = 0.015$  for different Riemann solvers using Godunov's method

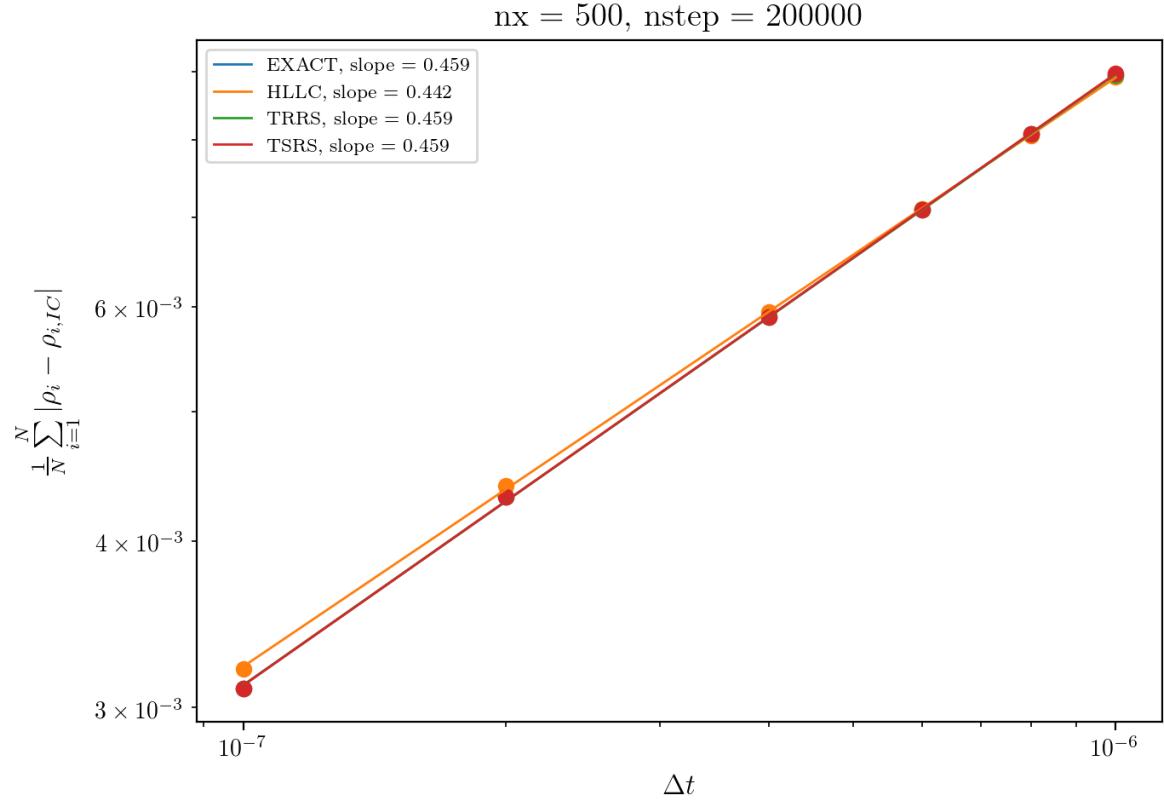


**Figure 62:** Vacuum Generating Riemann problem at  $t = 0.030$  for different Riemann solvers using Godunov's method

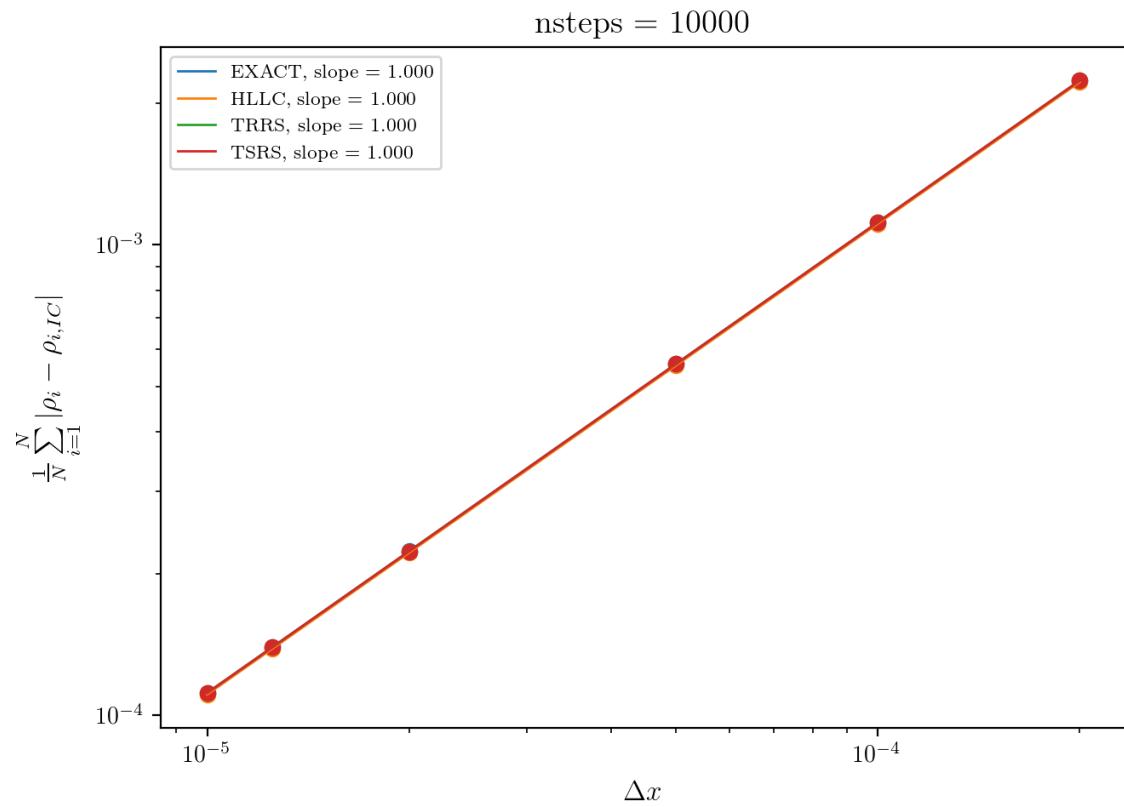


**Figure 63:** Vacuum Generating Riemann problem at  $t = 0.040$  for different Riemann solvers using Godunov's method

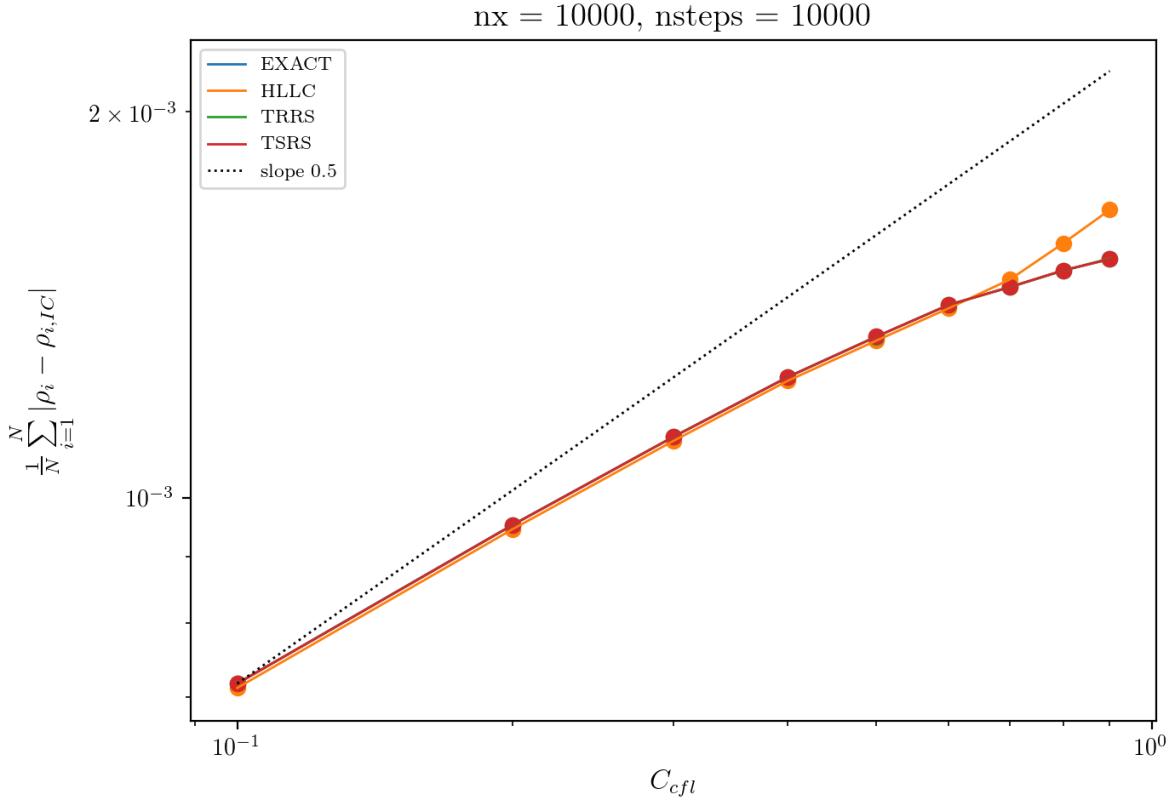
### 5.3 Order of Convergence Study



**Figure 64:** Testing the method's convergence with respect to the time step  $\Delta t$  on Sod test initial conditions. For a fair comparison, both the cell width  $\Delta x$  and the number of steps taken are fixed. The points are measurements, the lines are a linear fit, with the slope of the line given in the legend for each Riemann solver used in the legend.



**Figure 65:** Testing the method's convergence with respect to the cell width  $\Delta x$  on Sod test initial conditions. For a fair comparison, the Courant number  $C_{CFL}$  and the total number of steps are fixed. The points are measurements, the lines are a linear fit, with the slope of the line given in the legend for each Riemann solver used in the legend.



**Figure 66:** Testing the method's convergence with respect to the Courant number  $C_{CFL}$  on Sod test initial conditions. The points are measurements, the lines are just connecting them. The slope of 1/2 is plotted for comparison, and to demonstrate the deviation from it.

## 5.4 Conclusions

- Similar to the piecewise constant advection, the method is diffusive around sharp jump discontinuities. See figs. 52 - 55.
- Godunov's method has some trouble dealing with vacuum. The issue is on one hand that the Riemann solver would characterize the vacuum state with some escape velocity, while if we compute the primitive values of a cell state in vacuum using the conserved variables, we have no way of predicting this escape velocity inside the cell; The cell might be in the interior of the vacuum state, so looking at neighbouring states and solving the Riemann problem there doesn't help. We'd have to keep looking to the left or to the right to find the appropriate neighbour state, and this would only be a solution for Riemann problems, not if we want to solve arbitrary problems.

On the other hand, the code was reprogrammed to return zero velocity when the Riemann solvers are used in Godunov's scheme. This leads to very sharp jump discontinuities, see for example figs 56 - 59.

Similarly, some instabilities/inaccuracies/discontinuities develop in the vacuum-region in the vacuum generating case, figs. 60 - 63.

- **Order of Convergence**

- Looking at the time step dependence (fig. 64), we always get slopes around  $\sim 0.5$ . Considering that a Sod test contains multiple jump discontinuities, this is absolutely as expected if we follow the same argumentation as for linear advection. See eq. 18 and derivation leading up to it for comparison. All in all, it remains true that jump discontinuities reduce the order of convergence w.r.t. the time step.
- For the cell width dependence (fig. 65), we get a remarkable slope of 1.000 for all solvers. Even more remarkable, all the  $L1$  norms are identical. No approximate solver introduces more or less errors in this test case.
- For the  $C_{cfl}$  dependence, we see that it deviates more stronger for higher  $C_{CFL}$  from the 0.5 power law that we get for small  $\Delta t$ . However, it is significantly better than what we get for advection (fig. 24).

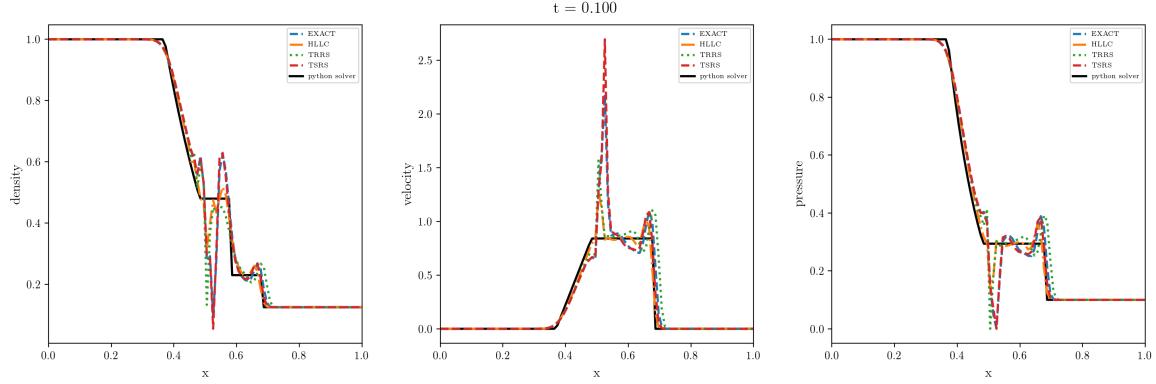
Why?

Well, we don't necessarily have constant time steps any more, nor do we have constant velocities. It is conceivable that the errors "correct themselves" by demanding/allowing a smaller/larger timestep.

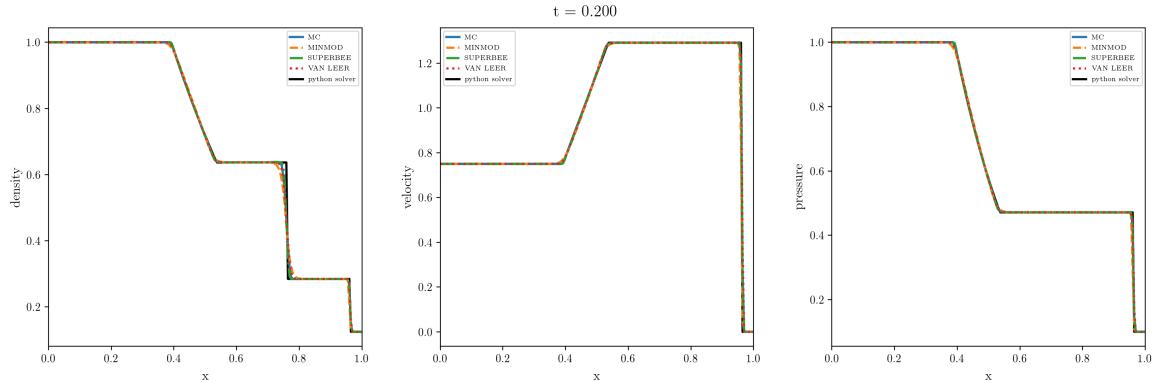
What also might decrease the accuracy for high  $C_{cfl}$  is that I don't properly compute the emerging wave speeds, but estimate them; So there is a possibility that for high  $C_{cfl}$ , things are just computed wrongly, i.e. the chosen time step is too large to be stable or accurate.

## 6 WAF Method

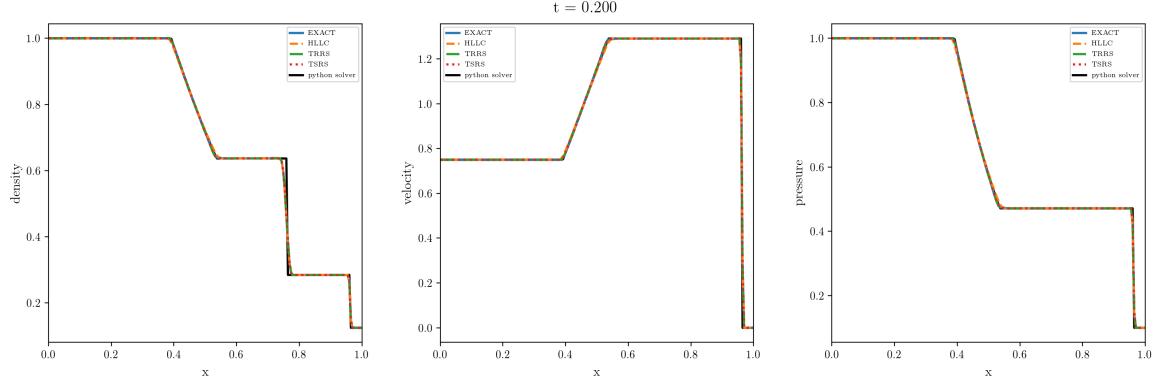
### 6.1 Solving Riemann Problems using the WAF Method



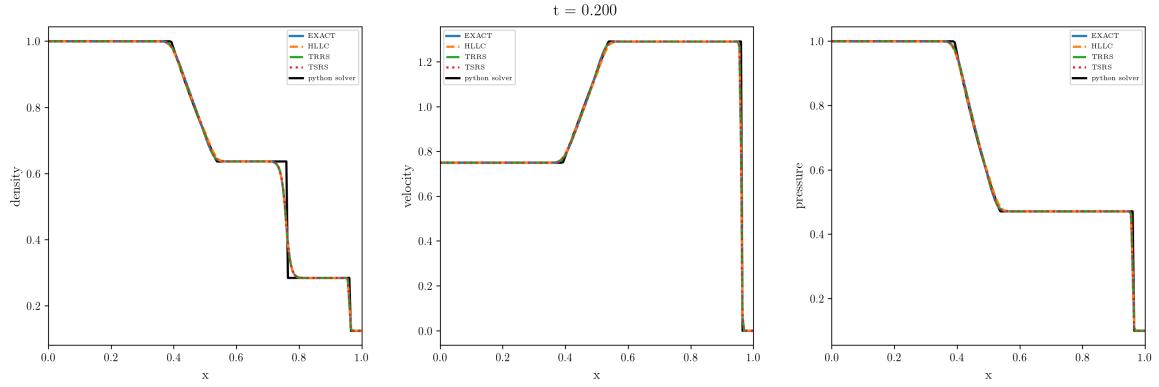
**Figure 67:** Solution to the sod shock Riemann problem using the WAF method without flux limiters. The black line is the exact solution.



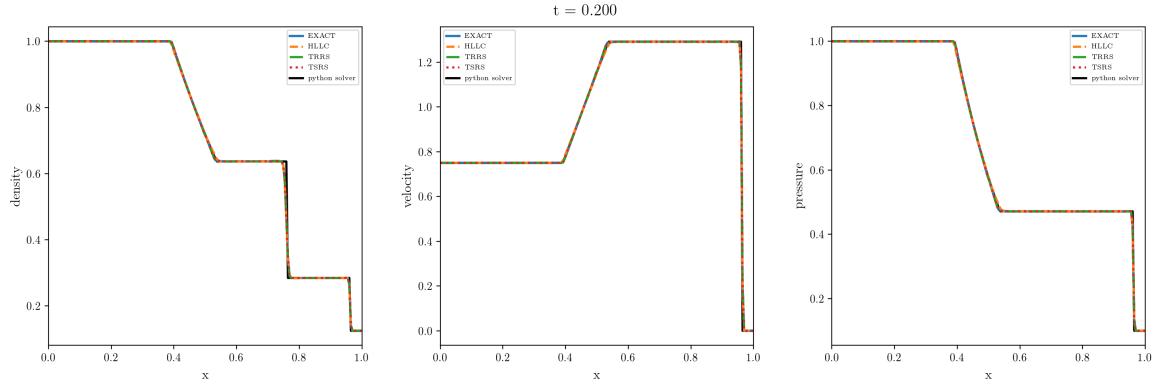
**Figure 68:** Solution to the modified sod test Riemann problem using the WAF method, an exact Riemann solver and various limiters. The black line is the exact solution.



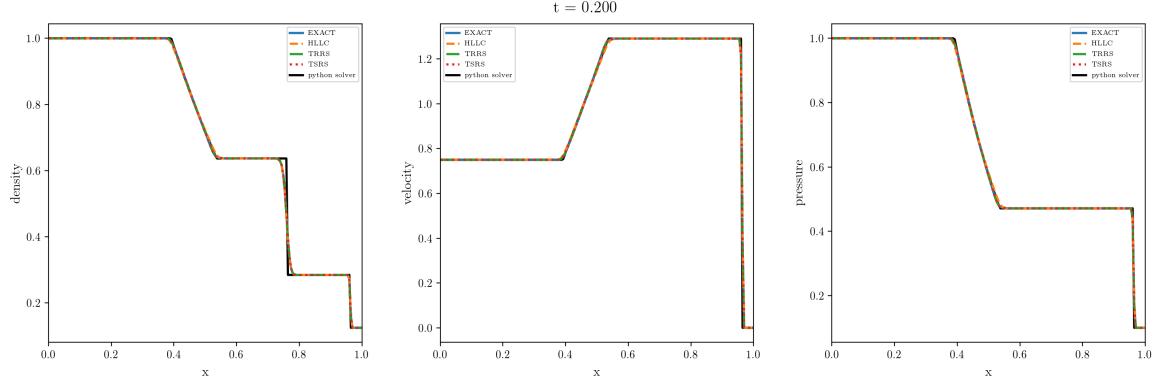
**Figure 69:** Solution to the modified sod test Riemann problem using the WAF method, an MC limiter, and various Riemann solvers. The black line is the exact solution.



**Figure 70:** Solution to the modified sod test Riemann problem using the WAF method, a minmod limiter, and various Riemann solvers. The black line is the exact solution.



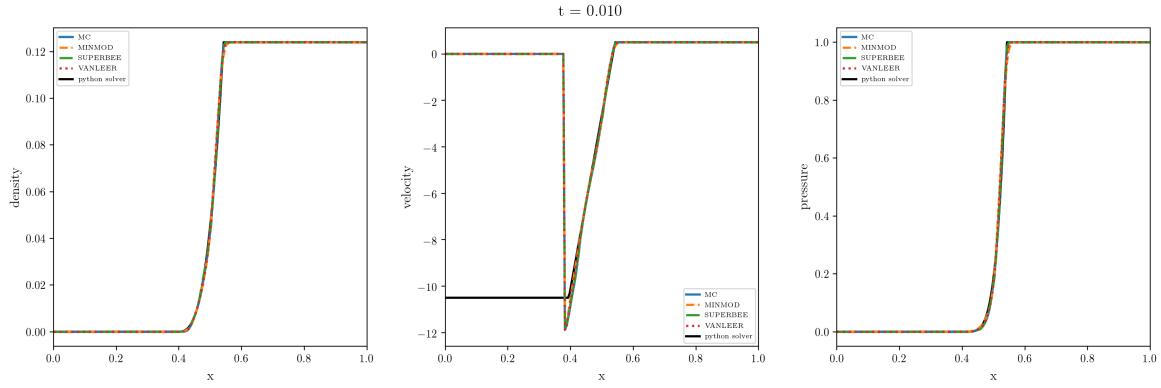
**Figure 71:** Solution to the modified sod test Riemann problem using the WAF method, a superbee limiter, and various Riemann solvers. The black line is the exact solution.



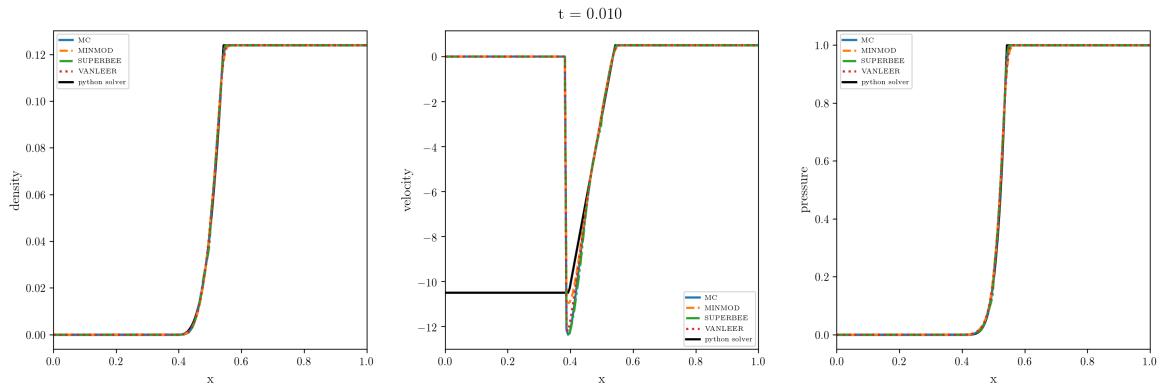
**Figure 72:** Solution to the modified sod test Riemann problem using the WAF method, a van Leer limiter, and various Riemann solvers. The black line is the exact solution.

## 6.2 Solving Vacuum Riemann Problems using WAF Method

### 6.2.1 Left Vacuum State

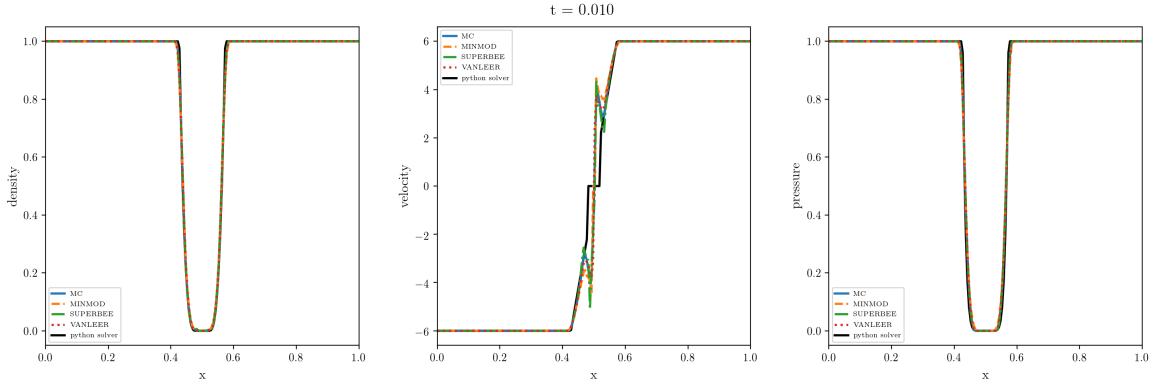


**Figure 73:** Left Vacuum Riemann problem at for Exact Riemann solver using WAF method and various limiters

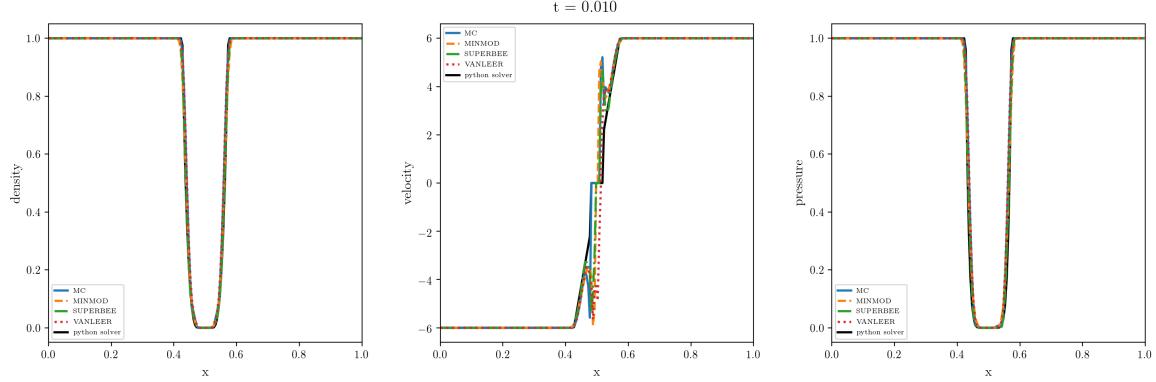


**Figure 74:** Left Vacuum Riemann problem at for HLLC Riemann solver using WAF method and various limiters

### 6.2.2 Vacuum Generating ICs

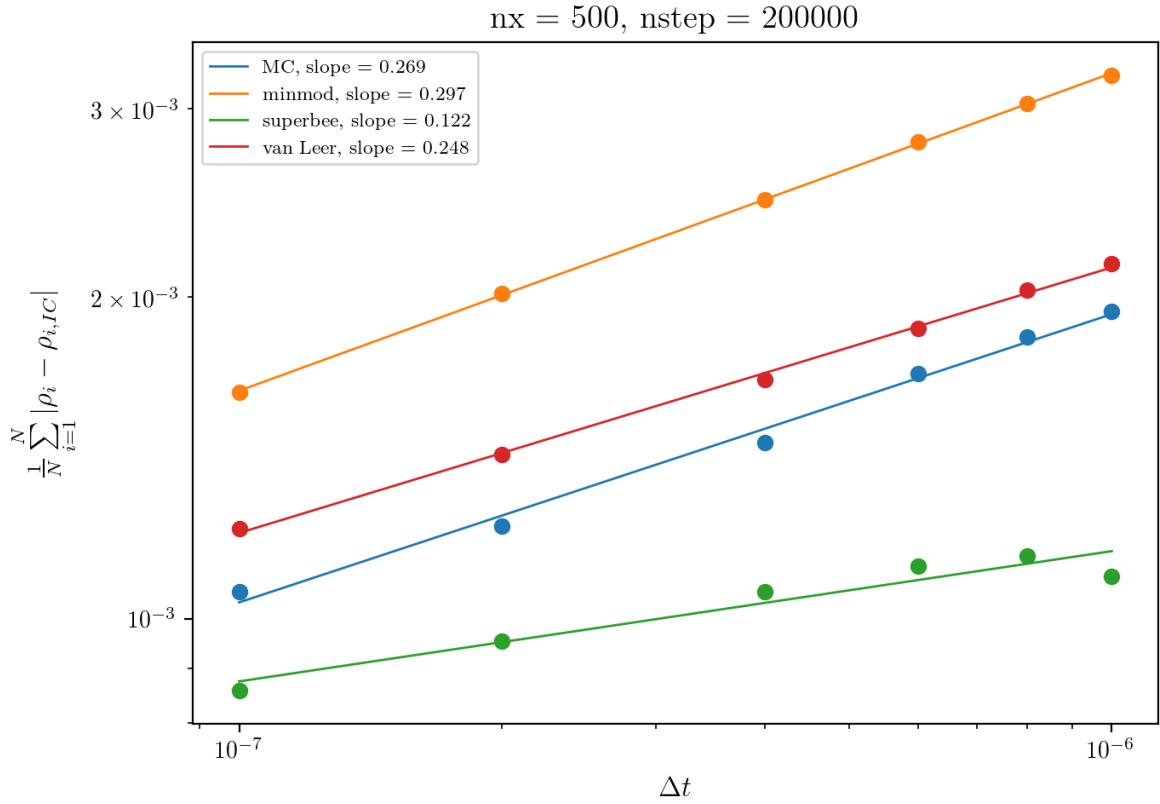


**Figure 75:** Vacuum Generating Riemann problem with Exact Riemann solver using WAF method

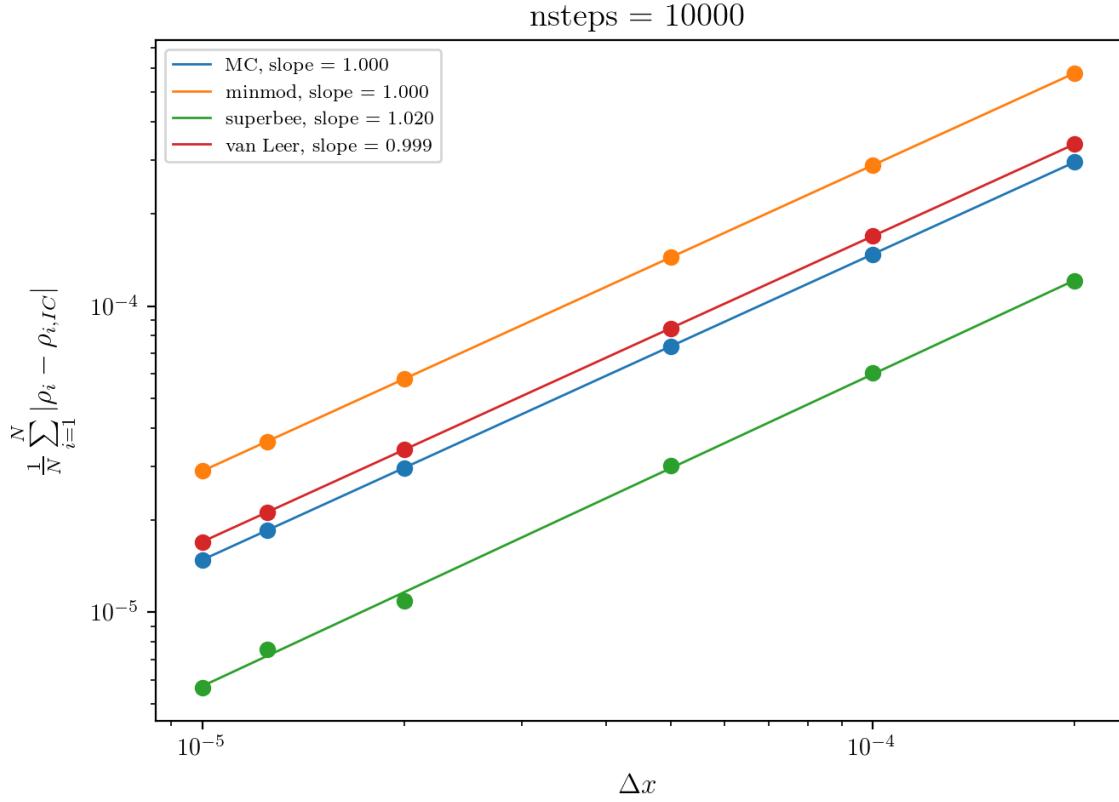


**Figure 76:** Vacuum Generating Riemann problem with HLLC Riemann solver using WAF method

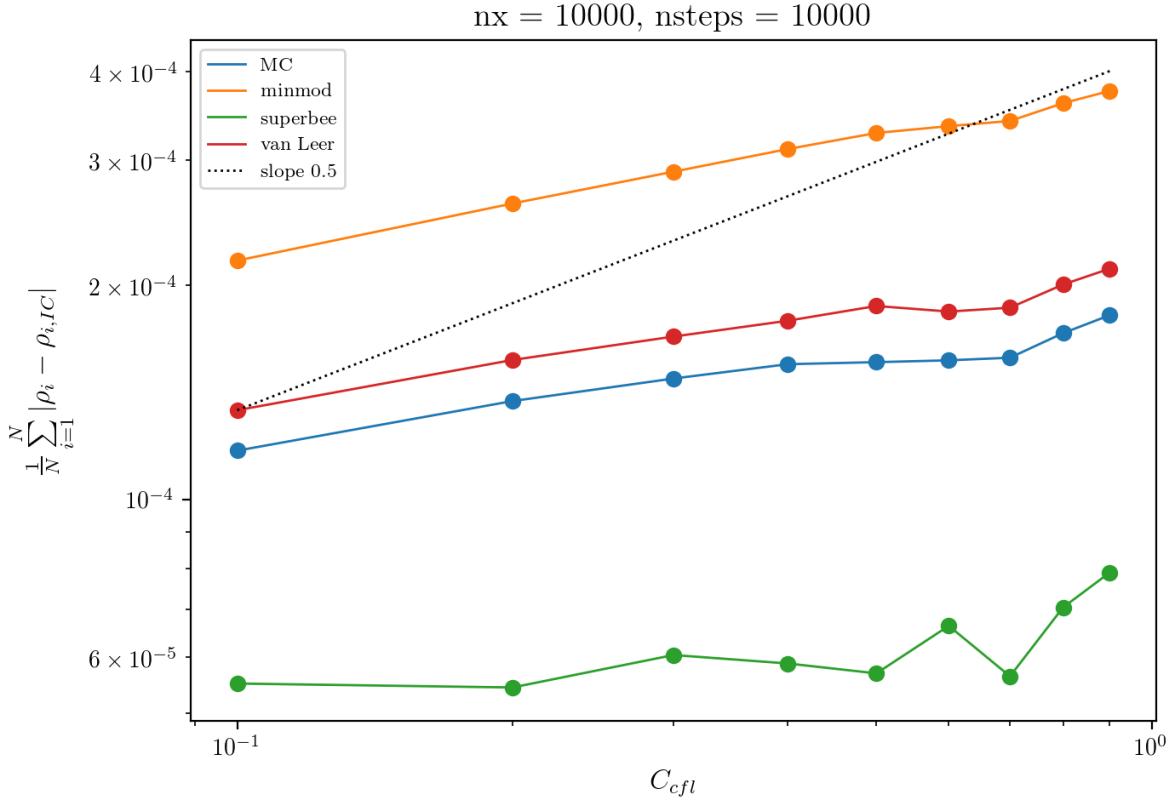
### 6.3 Order of Convergence Study



**Figure 77:** Testing the method's convergence with respect to the time step  $\Delta t$  on Sod test initial conditions. For a fair comparison, both the cell width  $\Delta x$  and the number of steps taken are fixed. The points are measurements, the lines are a linear fit, with the slope of the line given in the legend for each Riemann solver used in the legend.



**Figure 78:** Testing the method's convergence with respect to the cell width  $\Delta x$  on Sod test initial conditions. For a fair comparison, the Courant number  $C_{CFL}$  and the total number of steps are fixed. The points are measurements, the lines are a linear fit, with the slope of the line given in the legend for each limiter used in the legend. The exact Riemann solver has been used.



**Figure 79:** Testing the method's convergence with respect to the Courant number  $C_{CFL}$  on Sod test initial conditions. The points are measurements, the lines are just connecting them. The slope of 1/2 is plotted for comparison, and to demonstrate the deviation from it. The exact Riemann solver has been used.

## 6.4 Conclusions

- The WAF method without limiters, fig. 67, gives terrible results. In fact, in a lot of cases the oscillations would grow too strong and the code would crash.
- Similar to the piecewise linear advection, the drops around jump discontinuities are now much sharper, i.e. less diffusive, when limiters are applied. Compare fig. 68 to fig. 55.
- The choice of the Riemann solver has no noticeable effect on the solution. See fig. 68.
- The effect of the limiter is comparable to how they behave on linear advection. See fig. 69 - 72.

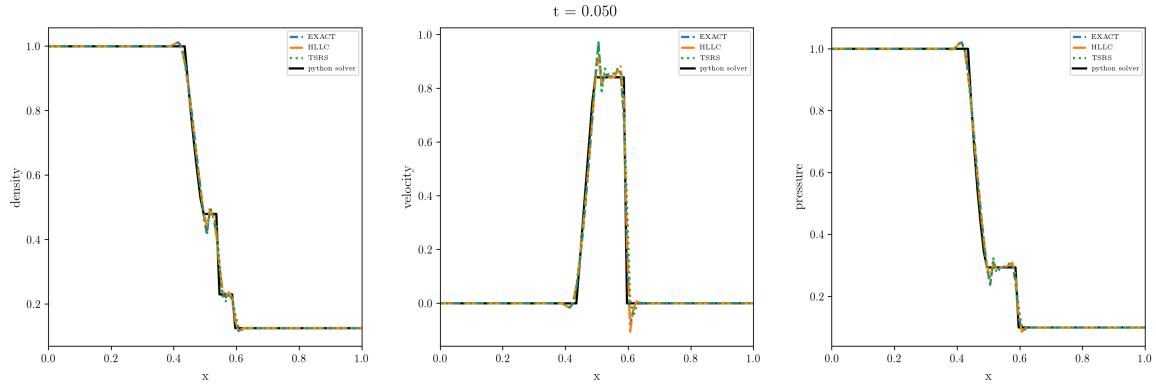
- The trouble with vacuum that Godunov’s method already had persists. I couldn’t find literature on how to deal with vacuum using the WAF method, so I made up my own. For small time steps, it “works”, as in it doesn’t crash, and gives better results than the Godunov method if a left or a right state is vacuum. But for vacuum generating conditions, the velocity in the middle develops massive spikes earlier than Godunov’s method; Compare figs. 75 and 76 to 61.

- **Order of Convergence**

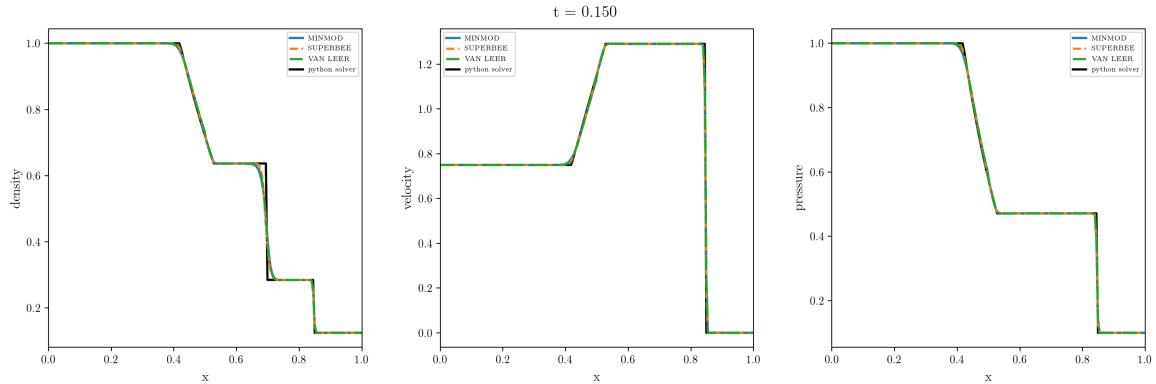
- Looking at the time step dependence (fig. 77), it’s absolutely terrible. I’m not sure why, but it is comparable to the results of WAF advection, fig. 28.
- For the cell width dependence (fig. 78), we get a remarkable slope of  $\approx 1$  for all solvers.
- For the  $C_{cfl}$  dependence, it again doesn’t follow the expected slope of the 0.5 power law that we get for small  $\Delta t$ . Similarly to the WAF advection (fig. 30) however, it gets close to a power law (straight line in log space) already at “high”  $C_{CFL} \approx 0.7$ , which is telling us that the diffusion which is  $\propto (1 - C_{CFL})$  for first order methods is strongly reduced.

## 7 MUSCL-Hancock Method

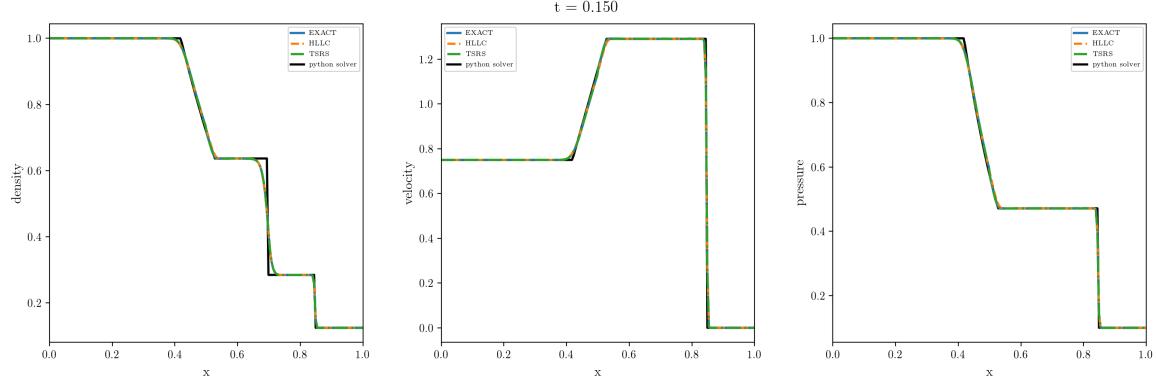
### 7.1 Solving Riemann Problems using the MUSCL-Hancock Method



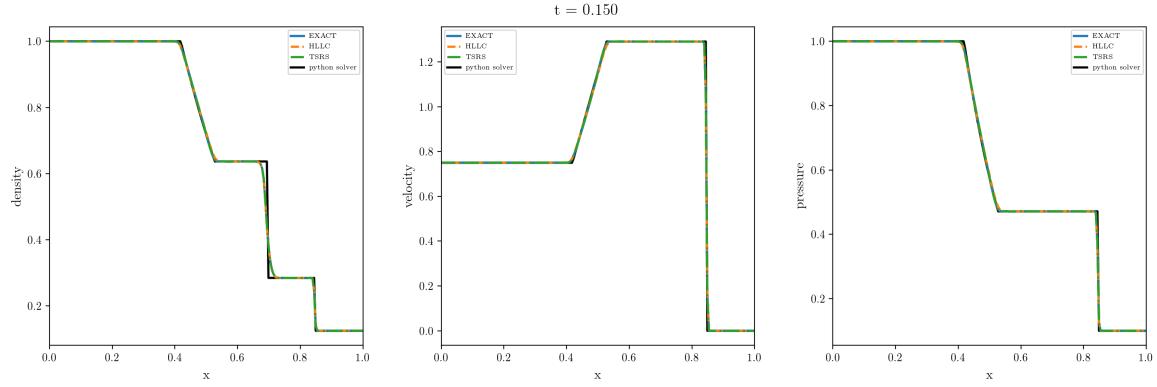
**Figure 80:** Solution to the sod shock Riemann problem using the MUSCL method without flux limiters. The black line is the exact solution.



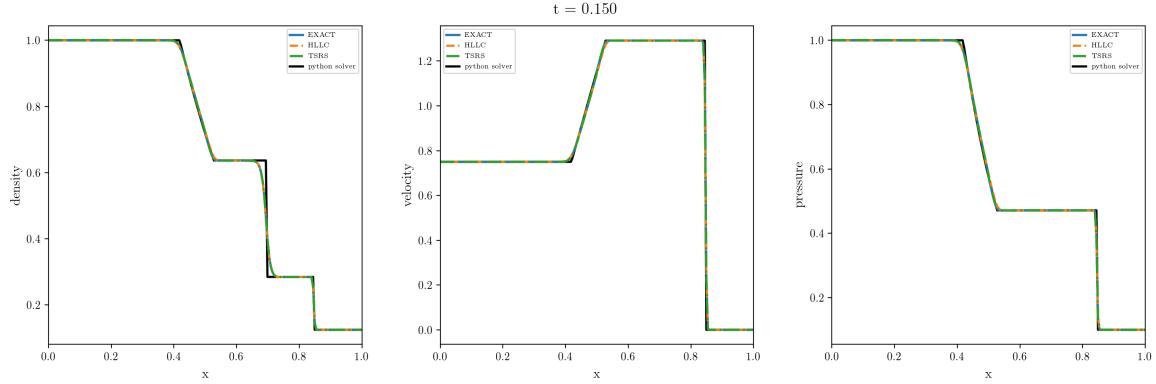
**Figure 81:** Solution to the modified sod test Riemann problem using the MUSCL-Hancock method, an exact Riemann solver and various limiters. The black line is the exact solution.



**Figure 82:** Solution to the modified sod test Riemann problem using the MUSCL-Hancock method, a minmod limiter, and various Riemann solvers. The black line is the exact solution.

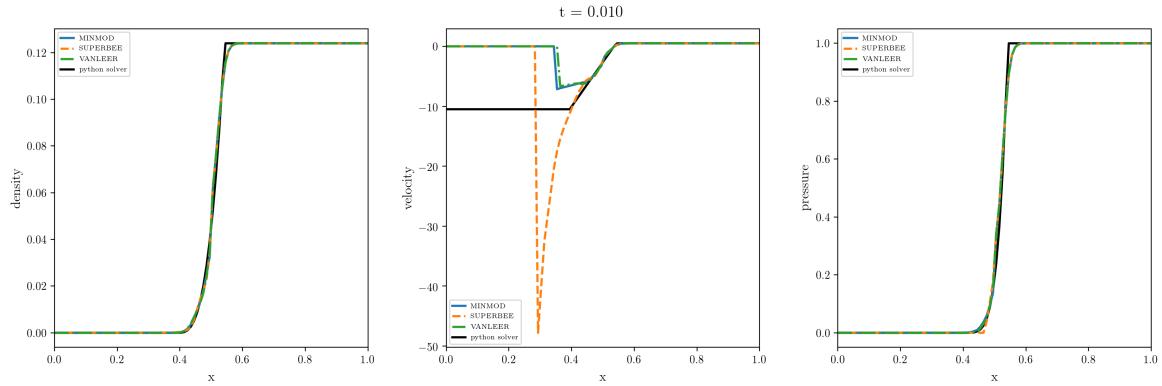


**Figure 83:** Solution to the modified sod test Riemann problem using the MUSCL-Hancock method, a superbee limiter, and various Riemann solvers. The black line is the exact solution.

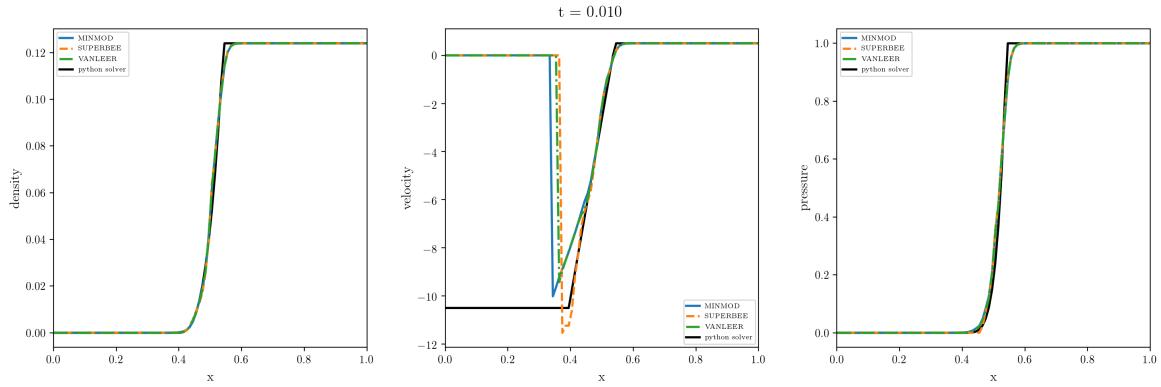


**Figure 84:** Solution to the modified sod test Riemann problem using the MUSCL-Hancock method, a van Leer limiter, and various Riemann solvers. The black line is the exact solution.

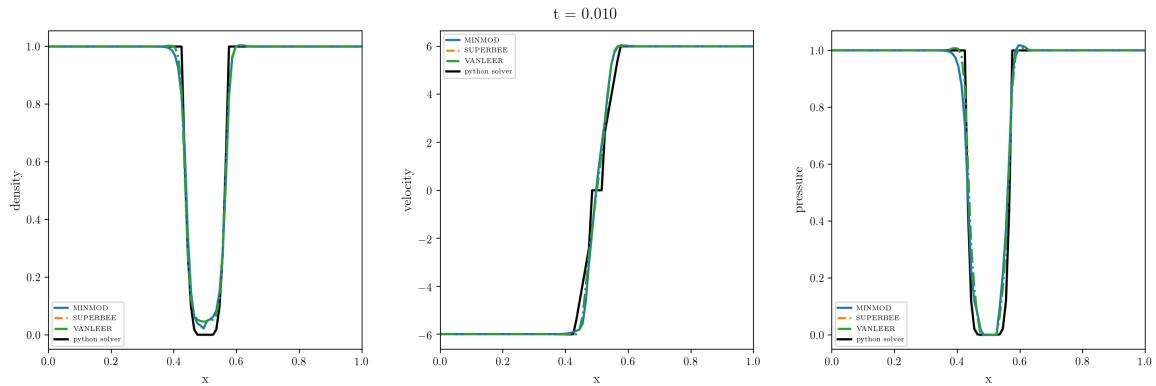
## 7.2 Solving Vacuum Riemann Problems using MUSCL-Hancock Method



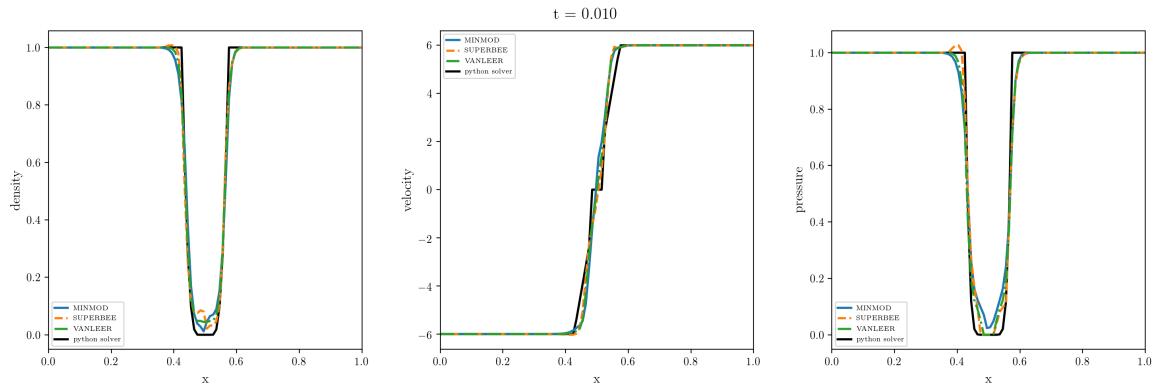
**Figure 85:** Left Vacuum Riemann problem for Exact Riemann solver using MUSCL-Hancock method and various limiters



**Figure 86:** Left Vacuum Riemann problem for HLLC Riemann solver using MUSCL-Hancock method and various limiters

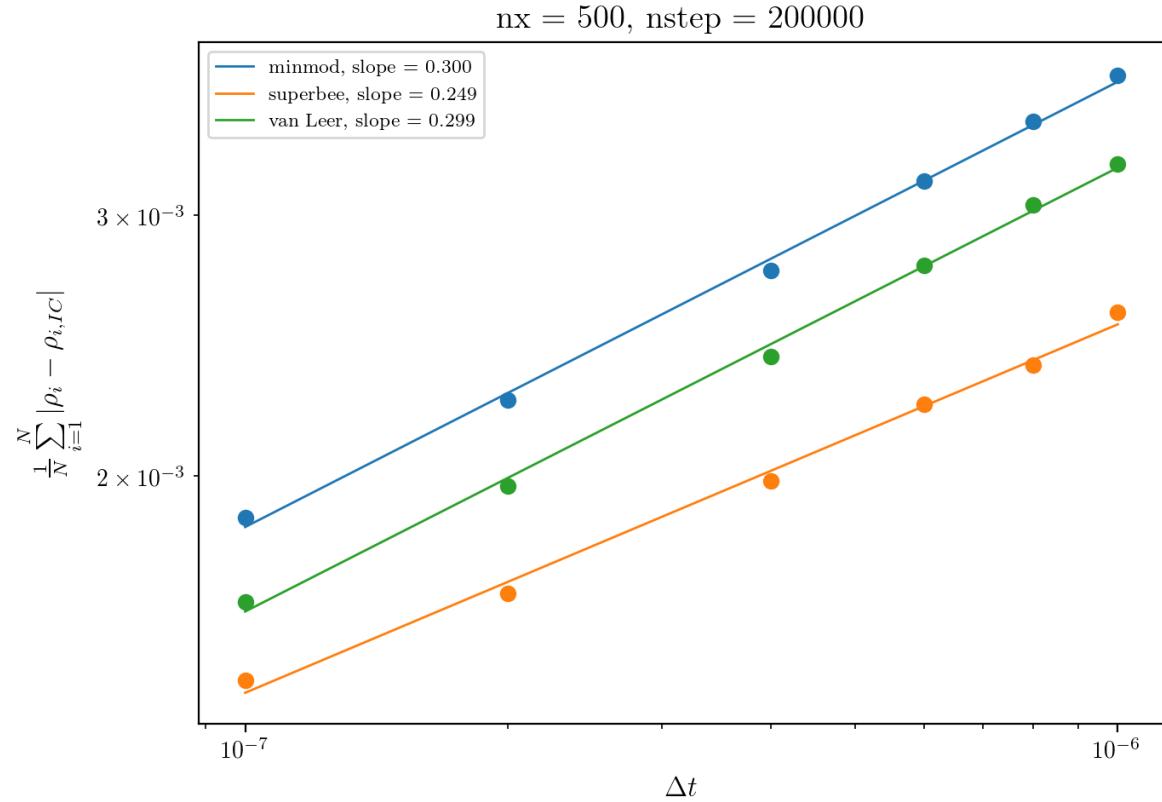


**Figure 87:** Vacuum Generating Riemann problem with Exact Riemann solver using MUSCL-Hancock method

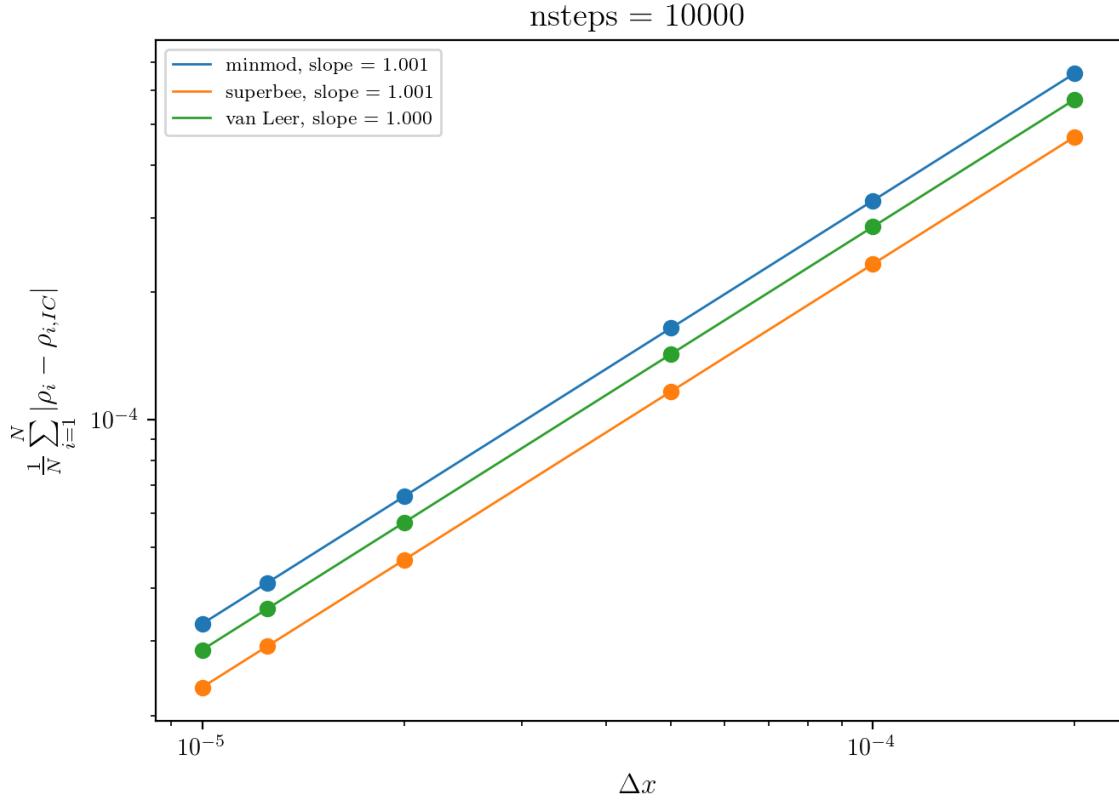


**Figure 88:** Vacuum Generating Riemann problem with HLLC Riemann solver using MUSCL-Hancock method

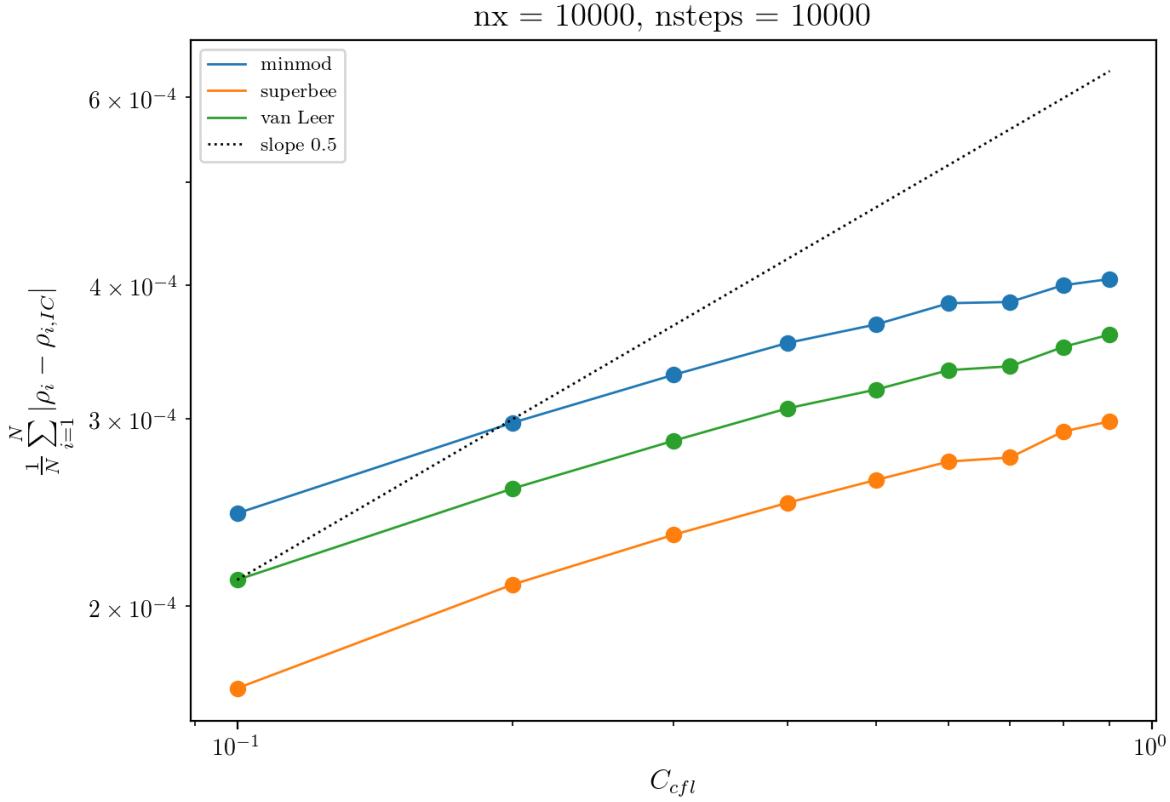
### 7.3 Order of Convergence Study



**Figure 89:** Testing the method's convergence with respect to the time step  $\Delta t$  on Sod test initial conditions. For a fair comparison, both the cell width  $\Delta x$  and the number of steps taken are fixed. The points are measurements, the lines are a linear fit, with the slope of the line given in the legend for each Riemann solver used in the legend.



**Figure 90:** Testing the method's convergence with respect to the cell width  $\Delta x$  on Sod test initial conditions. For a fair comparison, the Courant number  $C_{CFL}$  and the total number of steps are fixed. The points are measurements, the lines are a linear fit, with the slope of the line given in the legend for each limiter used in the legend. The exact Riemann solver has been used.



**Figure 91:** Testing the method's convergence with respect to the Courant number  $C_{CFL}$  on Sod test initial conditions. The points are measurements, the lines are just connecting them. The slope of 1/2 is plotted for comparison, and to demonstrate the deviation from it. The exact Riemann solver has been used.

## 7.4 Conclusions

- The MUSCL method without limiters, fig. 80, gives terrible results. In fact, in a lot of cases the oscillations would grow too strong and the code would crash.
- Similar to the piecewise linear advection, the drops around jump discontinuities are now much sharper, i.e. less diffusive, when limiters are applied. Compare fig. 81 to fig. 55.
- The choice of the Riemann solver has no noticeable effect on the solution. See fig. 81.
- The effect of the limiter is comparable to how they behave on linear advection. See fig. 82 - 84.

- If vacuum already exists, the method handles it reasonably well (fig 85, 86) compared with the other two finite volume methods. If there is vacuum generating conditions (fig 87, 88), things go down the drain. I had to use  $C_{cfl} = 0.1$  to get results like in those figure. Clearly the method is not TVD in that case, because new peaks arise. But I don't understand exactly why.

- **Order of Convergence**

- The time step dependence (fig. 89) is comparable to the results of piecewise linear advection, fig. 22.
- For the cell width dependence (fig. 90), we get a remarkable slope of  $\approx 1$  for all solvers.
- For the  $C_{cfl}$  dependence, it is essentially a power law (straight line in log space) for all  $C_{CFL}$ , which is telling us that the diffusion which is  $\propto (1 - C_{CFL})$  for first order methods is strongly reduced.

## References

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