

# Advection Project

---

Mladen Ivkovic

January 2018

# Introduction

The ideal gas equations form a set of hyperbolic differential equations of the form

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

In this project, consider the 1D advection of the mass density  $\rho$  with a global constant velocity  $u$ :

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

The analytical solution is given by any function  $q(x)$  with  $\rho(x, t) = q(x - ut)$ , which is  $q(x)$  again translated by  $ut$  along the  $x$ -direction.

## Solution Scheme: Finite Volume Method

Divide space into  $N$  cells of same size.

Make use of the divergence theorem: Convert volume integrals of finite volumes (cells) of divergence terms to surface integrals of fluxes through surfaces.

Advantage for conservation laws: Finite volume method is conservative, because  $F_{i+\frac{1}{2}} = F_{(i+1)-\frac{1}{2}}$

# Solution Scheme: Problems

## Direction of information

In this problem, all characteristics propagate downstream:

Information strictly travels in the flow direction.

$\Rightarrow \rho_i$  mustn't depend on downstream value. Otherwise, you'll be using "information that isn't there yet".

$\Rightarrow$  The discretisation will depend on the sign of the velocity  $u$ .

## CFL Condition

For explicit integration methods, we need to limit the time step such that information can travel at most by 1 cell during each time step to avoid instabilities.

In 1D:  $\Delta t \leq \Delta x/u$  ; In ND:  $\Delta t \leq [\sum_i^N u_i / \Delta x_i]^{-1}$

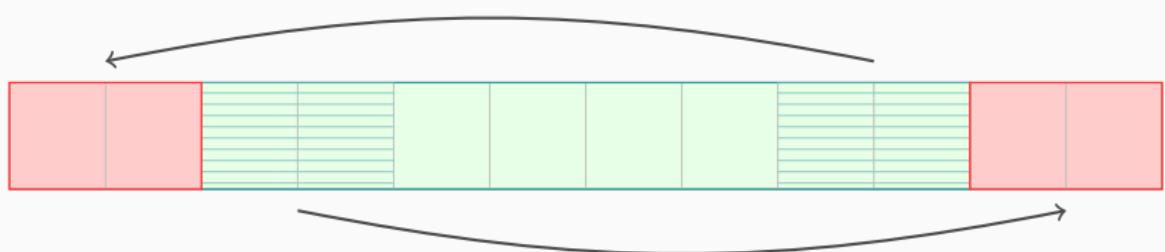
## Numerical Diffusion

The solution is not advected perfectly (except for  $\Delta t = \frac{\Delta x}{u}$ ) but smoothed out. The numerical algorithm has a diffusion term as a by-product. For a first order algorithm, we're effectively solving

$$\frac{\partial \rho}{\partial t} + u \frac{\partial \rho}{\partial x} = \frac{\Delta xu}{2} \frac{\partial^2 \rho}{\partial x^2}$$

# Periodic Boundary Conditions

Simulate a periodic boundary by introducing ghost cells:



# Piecewise Constant Method: Donor-Cell Advection

Assume cell state within cell is constant.

For  $u = 1$ :

$$\rho_i^{n+1} = \rho_i^n + \frac{\Delta t}{\Delta x} (f_{i-1/2}^{n+1/2} - f_{i+1/2}^{n+1/2})$$

$$f_{i \pm 1/2}^{n+1} = u_{i \pm 1/2} \quad \rho_{i-1/2 \pm 1/2}$$

First order accurate in time and space.

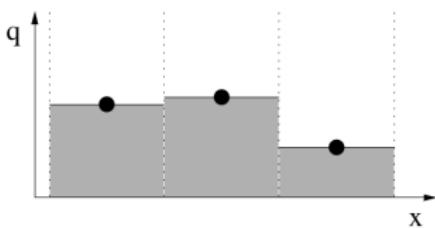
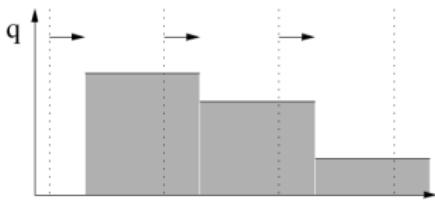
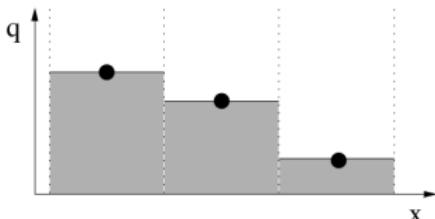
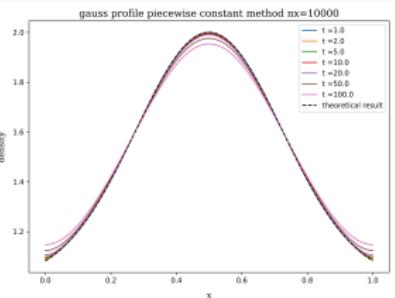
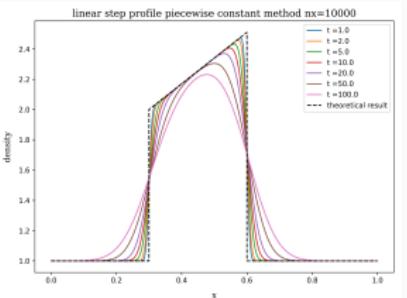
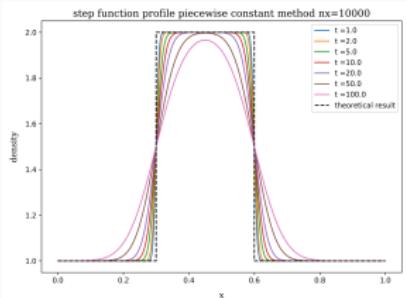
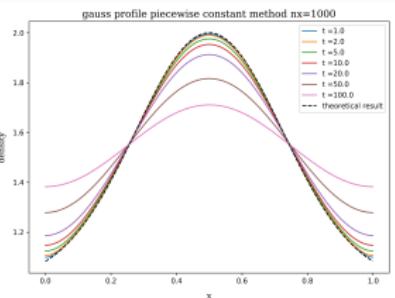
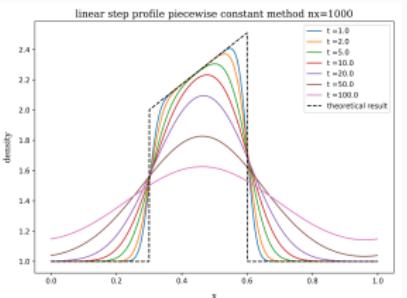
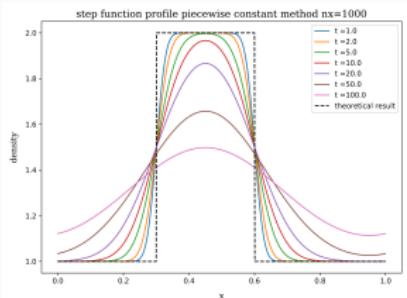
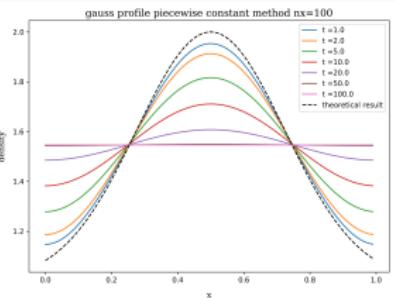
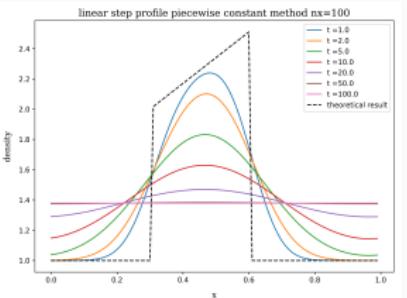
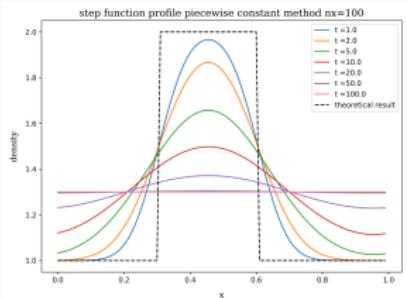


Image adapted from "Lecture Numerical Fluid Dynamics", Lecture given by C.P. Dullemond and H.H. Wang at Heidelberg University, 2009



# Piecewise Linear Method

Assume that the state in the cell is piecewise linear with some slope  $s$ . This gives a second-order accurate method.

For  $x_{i-1/2} < x_i < x_{i+1/2}$ :  $\rho(x, t = t_n) = \rho_i^n + s_i^n(x - x_i)$

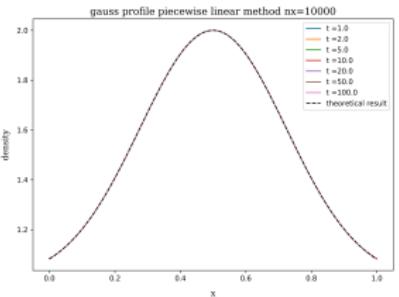
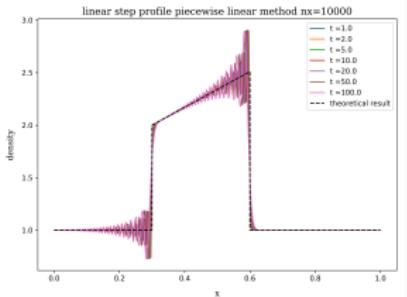
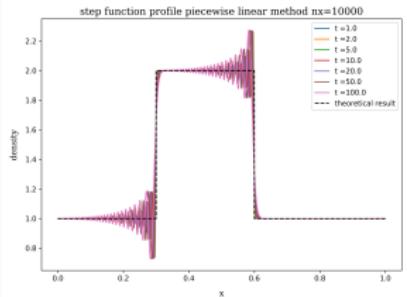
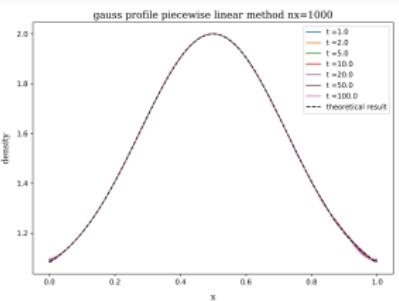
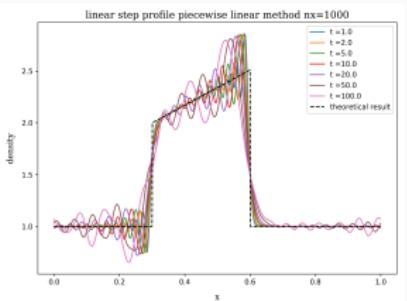
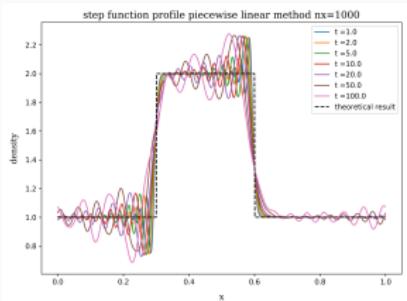
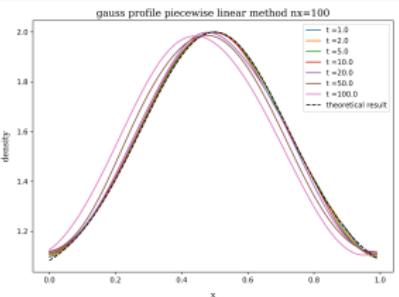
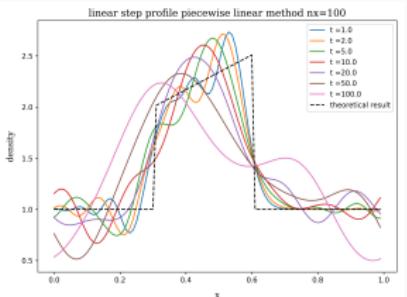
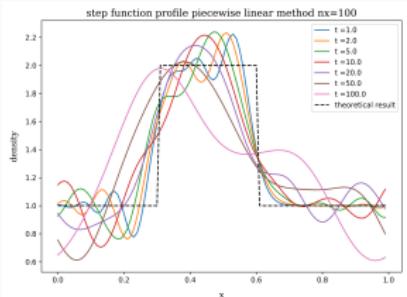
The flux over the interface is then

$$\begin{aligned} f_{i-1/2}(t) &= u\rho(x = x_{i-1/2}, t) \\ &= u\rho_{i-1} + us_i^n(\Delta x/2 - u(t - t_n)) \end{aligned}$$

Finally averaging the fluxes over a time step gives:

$$\rho_i^{n+1} = \rho_i^n - \frac{u\Delta t}{\Delta x}(\rho_i^n - \rho_{i-1}^n) - \frac{u\Delta t}{\Delta x} \frac{1}{2}(s_i^n - s_{i-1}^n)(\Delta x - u\Delta t)$$

Choice of slope:  $s_i^n = \frac{\rho_{i+1}^n - \rho_i^n}{\Delta x}$  ( Lax-Wendroff method )



## New Problem: Oscillations

The piecewise linear elements can have overshoots, leading to the oscillations seen in the previous plots.

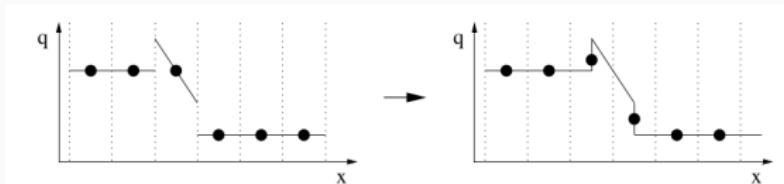


Image adapted from "Lecture Numerical Fluid Dynamics", Lecture given by C.P. Dullemond and H.H. Wang at Heidelberg University, 2009

**Godunov's theorem:** *any linear algorithm for solving partial differential equations, with the property of not producing new extrema, can be at most first order.*

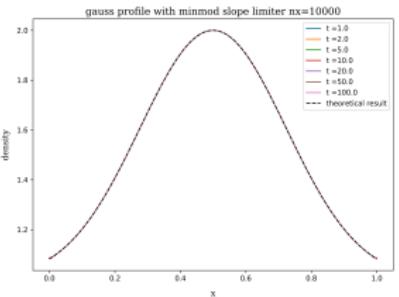
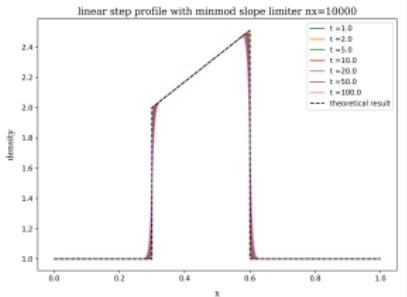
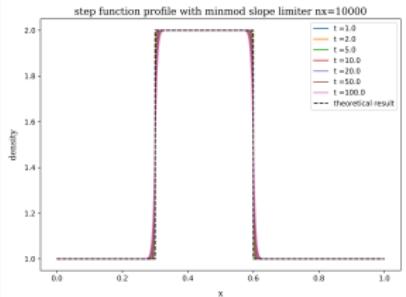
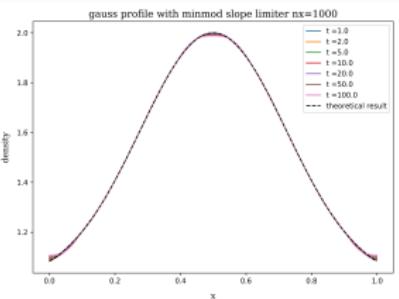
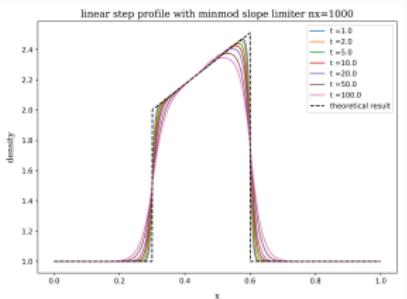
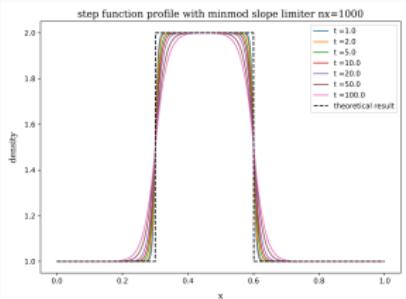
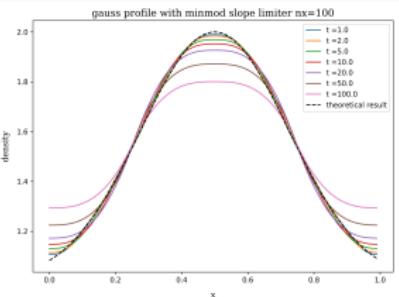
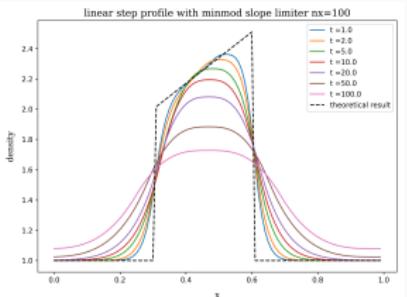
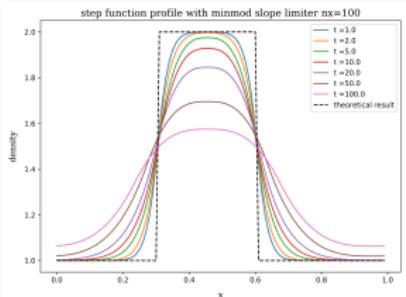
⇒ use non-linear conditions (slope limiters) to modify the slope  $s_i^n$  to prevent overshoots. Requirement: total variation diminishing:  $TV(\rho^{n+1}) \leq TV(\rho^n) \equiv \sum |\rho_i - \rho_{i-1}|$ . Such a scheme will not develop oscillations near a jump, because a jump is a monotonically in/decreasing function and a TVD scheme will not increase the  $TV$ .

# Minmod Slope Limiter

$$s_i^n = \text{minmod} \left( \frac{\rho_i^n - \rho_{i-1}^n}{\Delta x}, \frac{\rho_{i+1}^n - \rho_i^n}{\Delta x} \right)$$

with

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



# Van Leer Slope Limiter

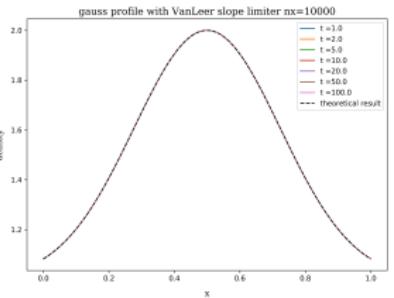
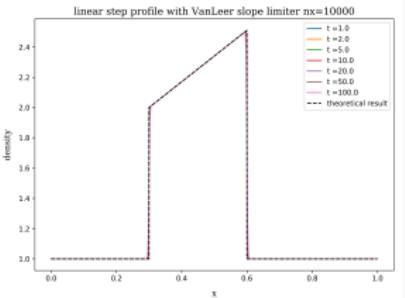
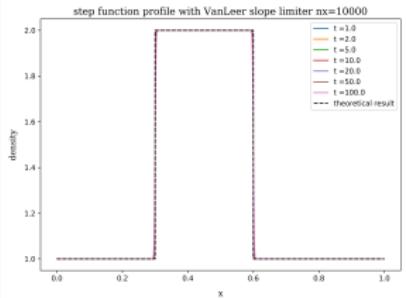
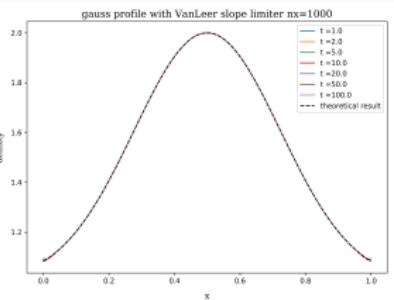
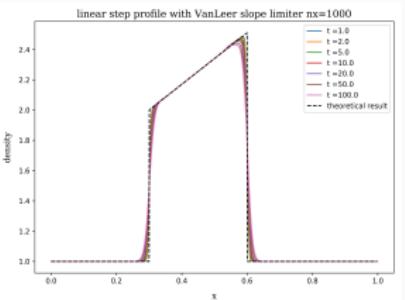
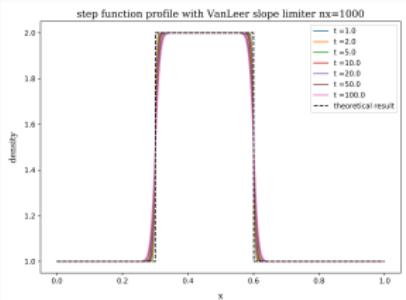
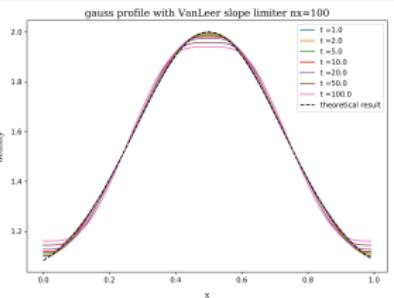
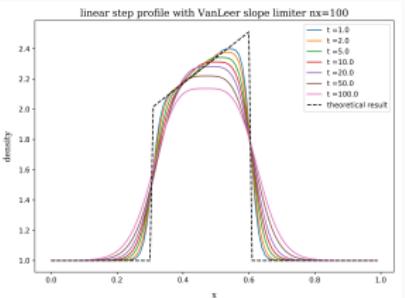
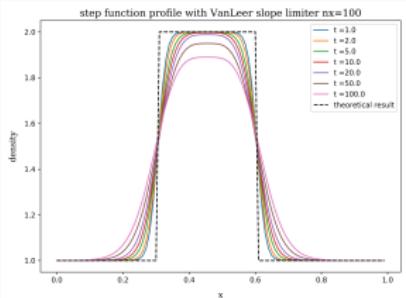
Rewrite flux (assuming  $u = 1$ ) as

$$f_{i-1/2}^{n+1/2} = u\rho_{i-1} + \frac{1}{2}u \left(1 - \frac{u\Delta t}{\Delta x}\right) \phi(r_{i-1/2}^n)(q_i^n - q_{i-1}^n)$$
$$r_{i-1/2}^n = \frac{q_{i-1}^n - q_{i-2}^n}{q_i^n - q_{i-1}^n}$$

Here,  $\phi$  is the slope/flux limiter.

The Van Leer flux limiter is defined as

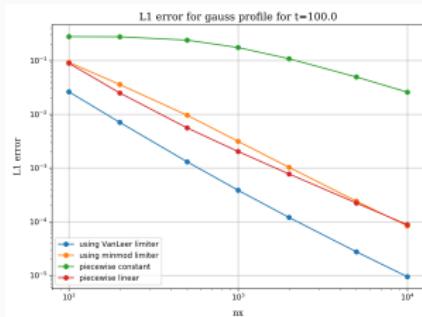
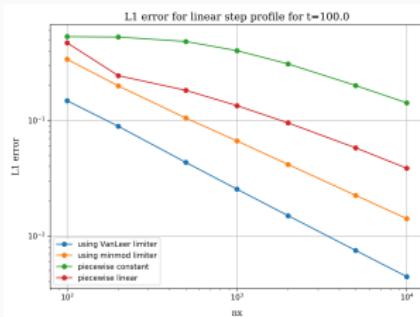
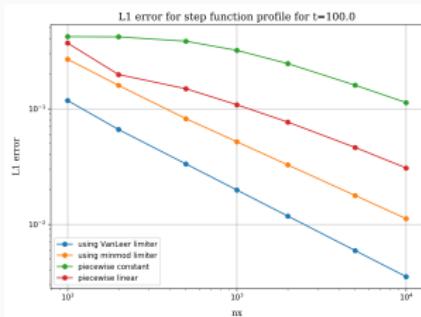
$$\phi(r) = \frac{r - |r|}{1 - |r|}$$



# Precision of the Algorithms

Quantify error through  $L1$  error norm:  $L1 = \frac{1}{N} \sum_i |\rho_i - \tilde{\rho}(x_i)|$ , where  $\tilde{\rho}(x_i)$  is the analytical solution.

For  $t = 100$ :



## 2D Advection

The 2D advection equation is given by

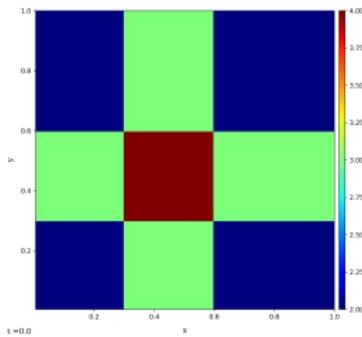
$$\frac{\partial \rho}{\partial t} + u \frac{\partial \rho}{\partial x} + v \frac{\partial \rho}{\partial y} = 0$$

Solution using the same algorithms for each dimension, adding fluxes simultaneously.

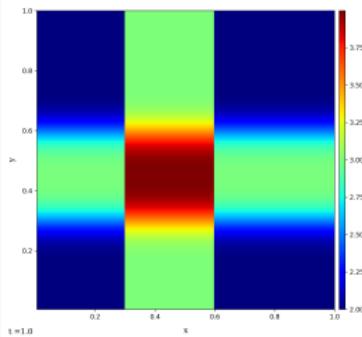
Results are shown for a step function density field with  $nx = ny = 200$ .

# $u = 0, v = 1$ Piecewise Constant Method

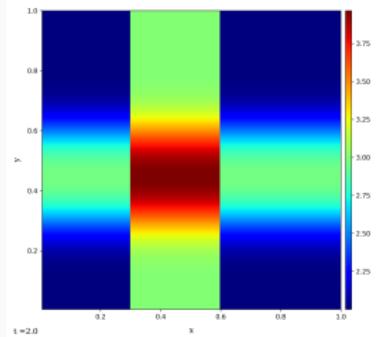
step function profile piecewise constant method nx=ny=200



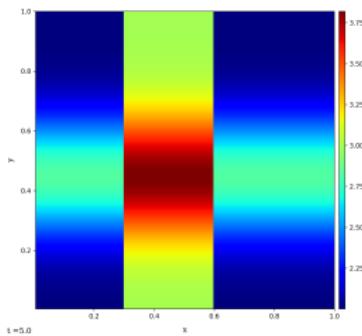
step function profile piecewise constant method nx=ny=200



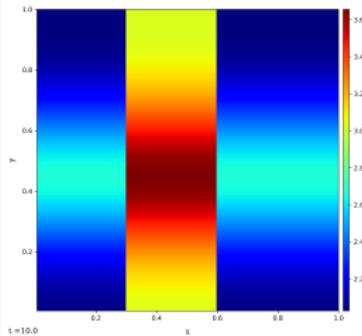
step function profile piecewise constant method nx=ny=200



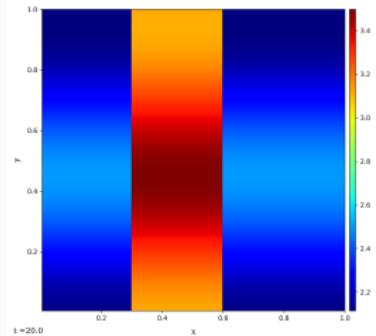
step function profile piecewise constant method nx=ny=200



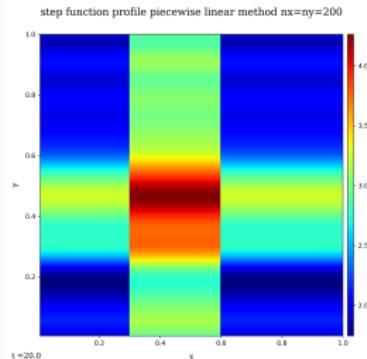
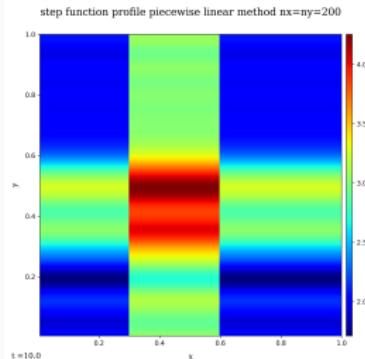
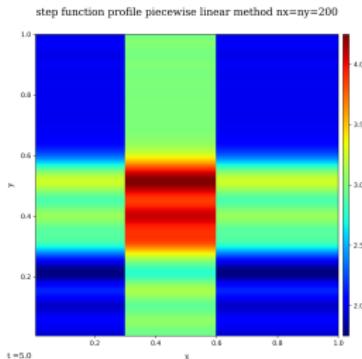
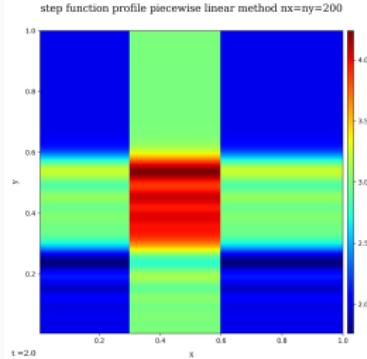
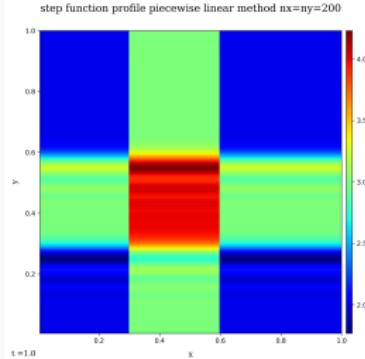
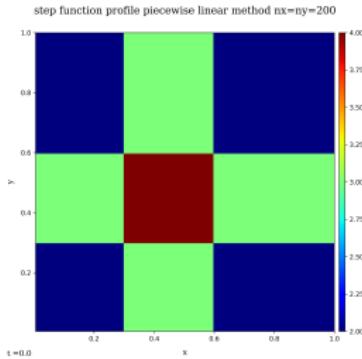
step function profile piecewise constant method nx=ny=200



step function profile piecewise constant method nx=ny=200

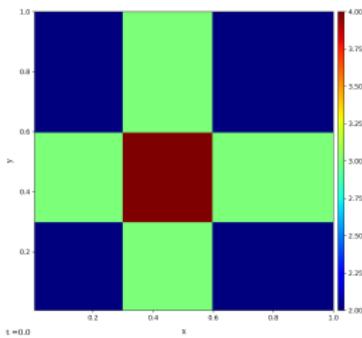


# $u = 0, v = 1$ Piecewise Linear Method

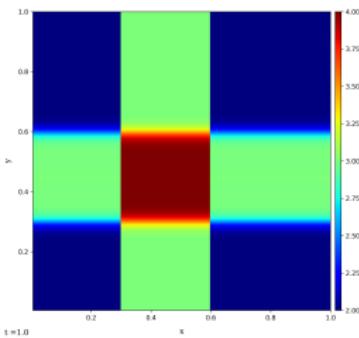


# $u = 0, v = 1$ Minmod Slope Limiter

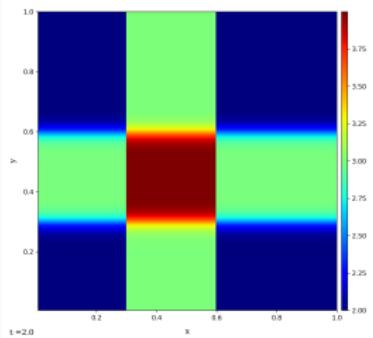
step function profile with minmod slope limiter nx=ny=200



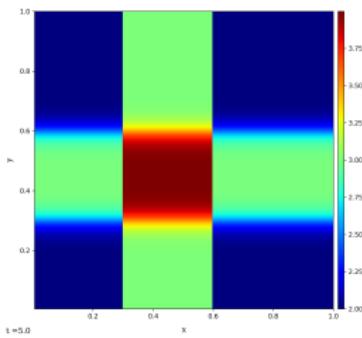
step function profile with minmod slope limiter nx=ny=200



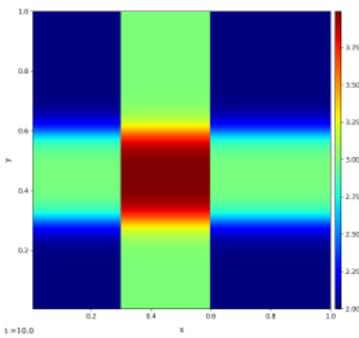
step function profile with minmod slope limiter nx=ny=200



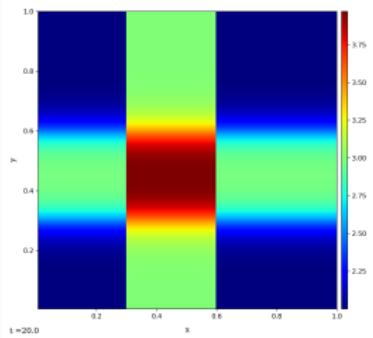
step function profile with minmod slope limiter nx=ny=200



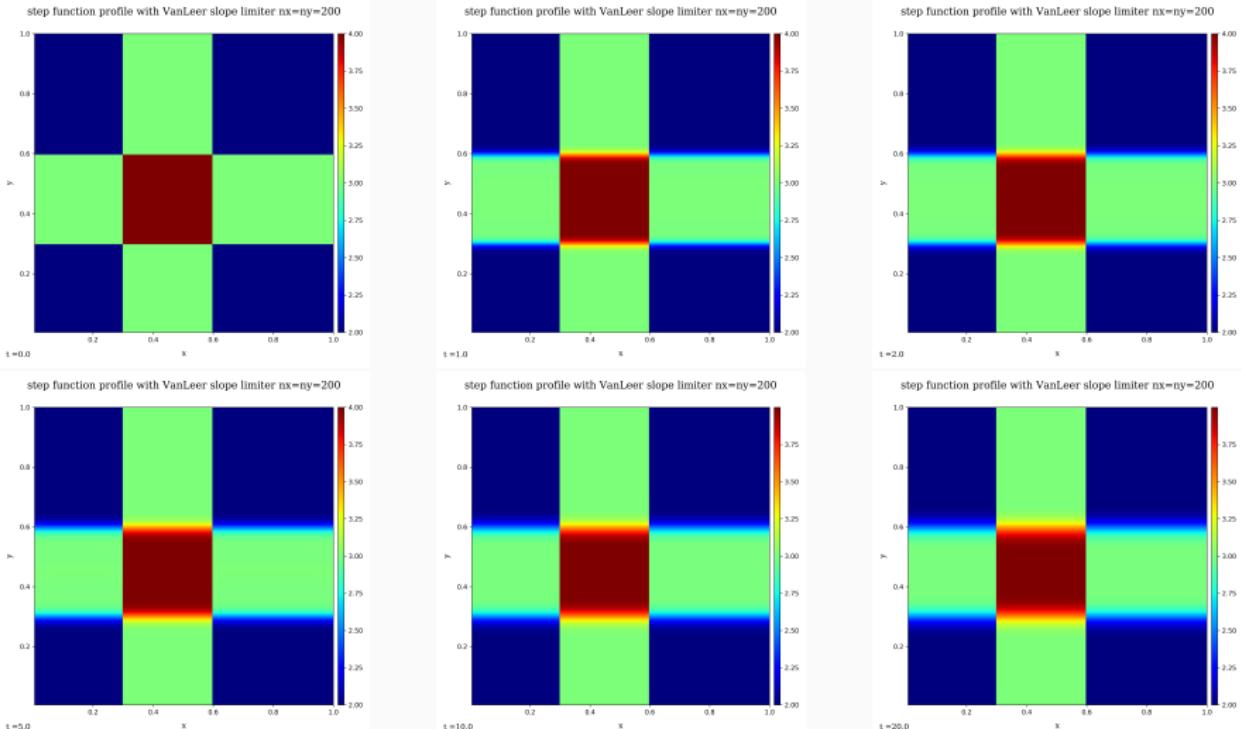
step function profile with minmod slope limiter nx=ny=200



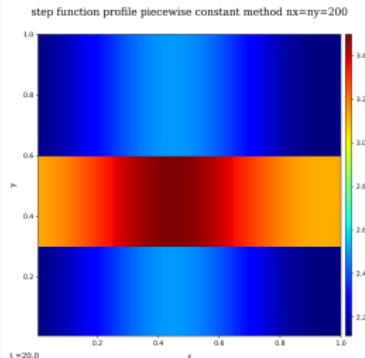
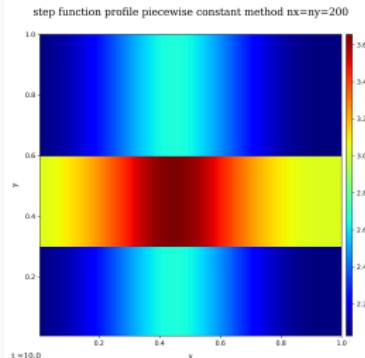
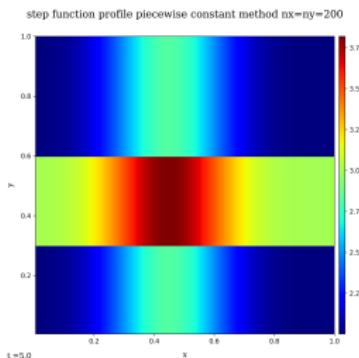
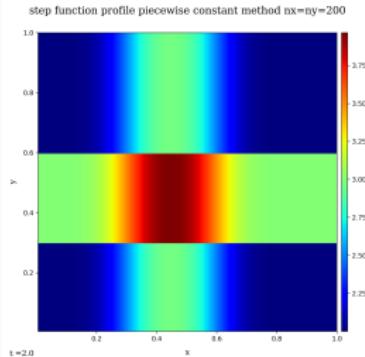
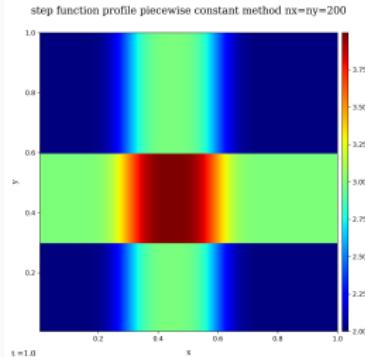
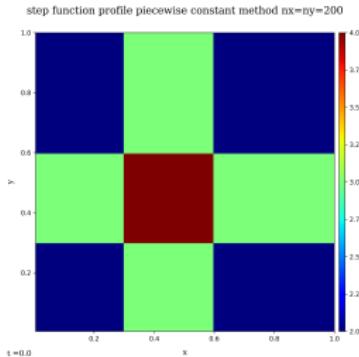
step function profile with minmod slope limiter nx=ny=200



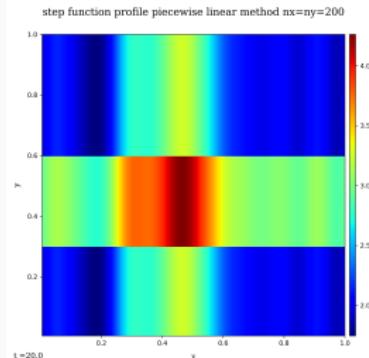
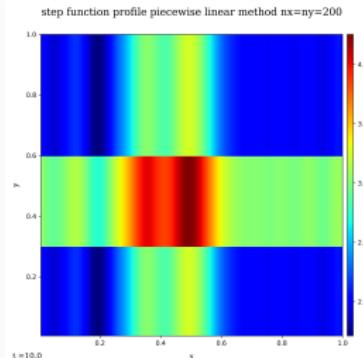
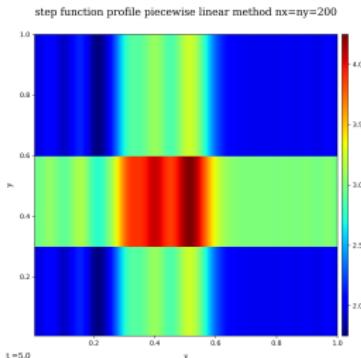
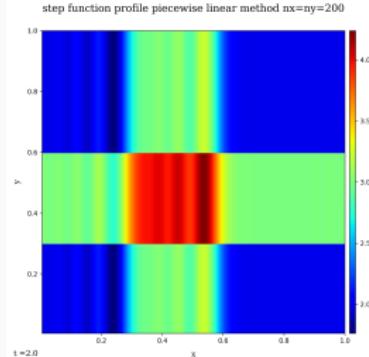
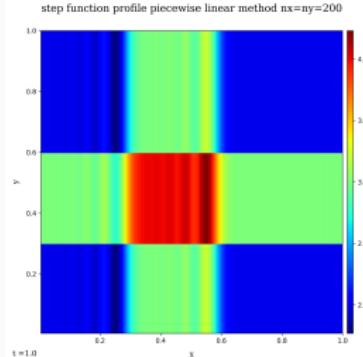
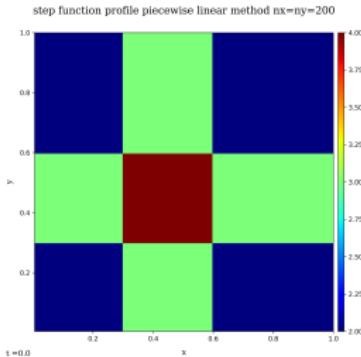
# $u = 0, v = 1$ Van Leer Slope Limiter



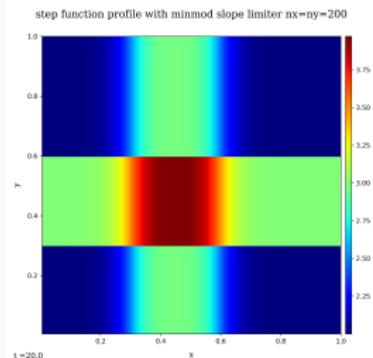
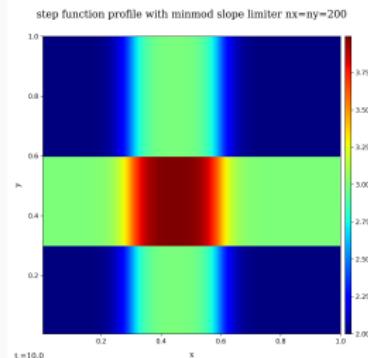
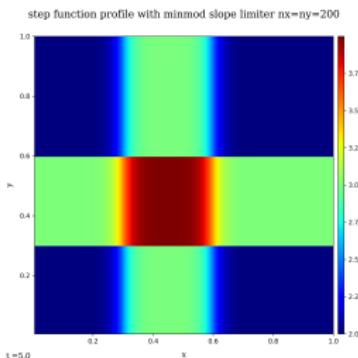
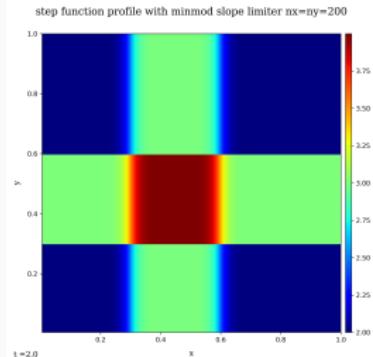
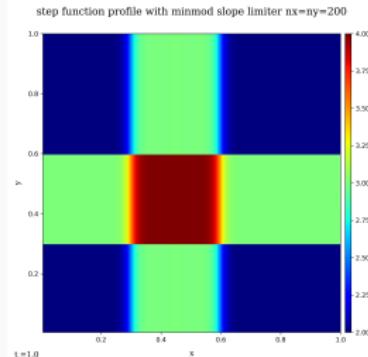
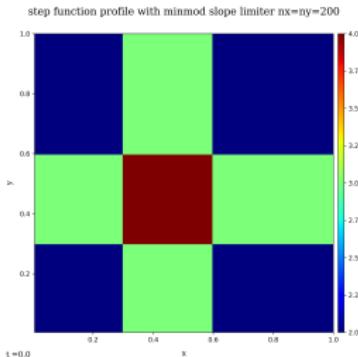
# $u = 1, v = 0$ Piecewise Constant Method



# $u = 1, v = 0$ Piecewise Linear Method

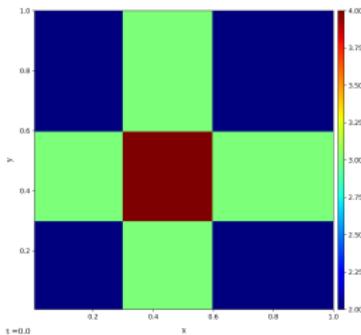


# $u = 1, v = 0$ Minmod Slope Limiter

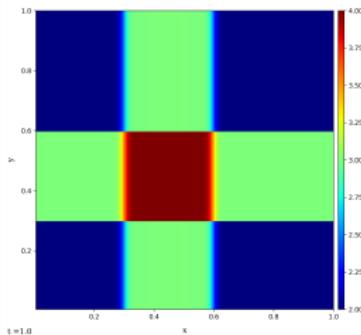


# $u = 1, v = 0$ Van Leer Slope Limiter

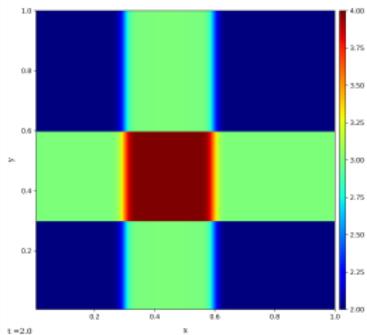
step function profile with VanLeer slope limiter nx=ny=200



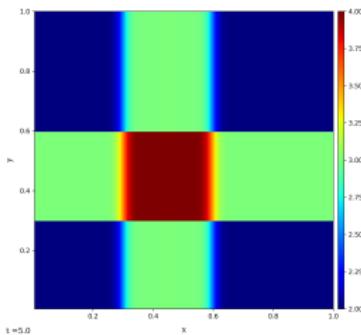
step function profile with VanLeer slope limiter nx=ny=200



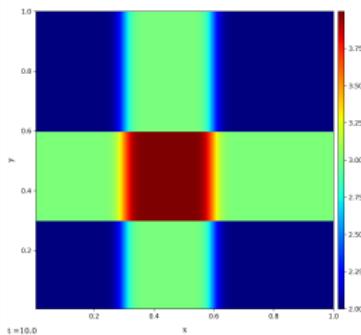
step function profile with VanLeer slope limiter nx=ny=200



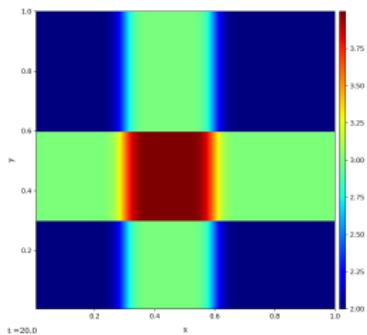
step function profile with VanLeer slope limiter nx=ny=200



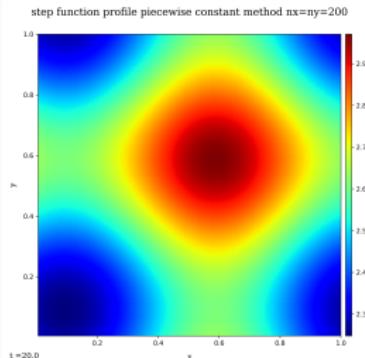
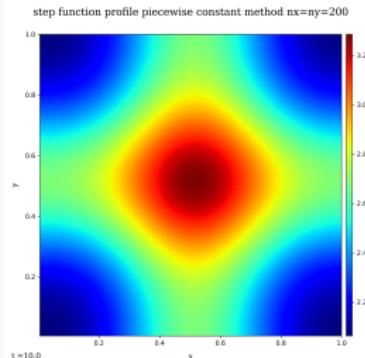
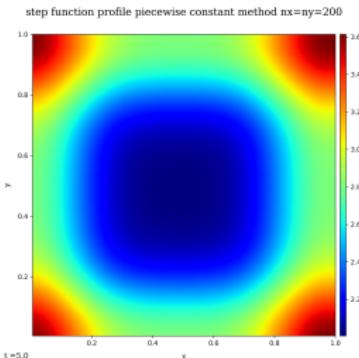
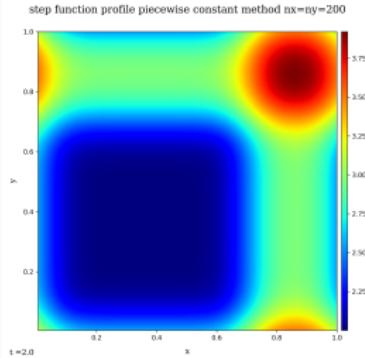
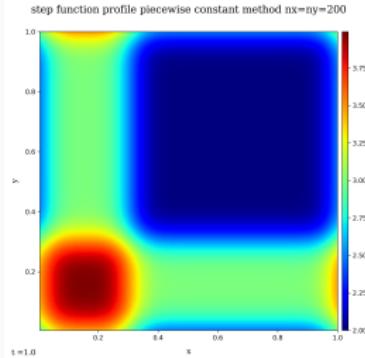
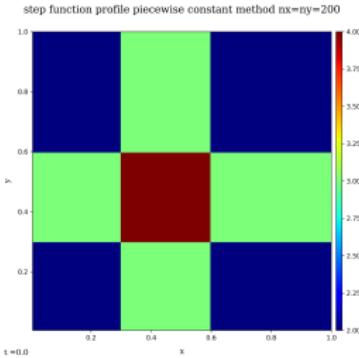
step function profile with VanLeer slope limiter nx=ny=200



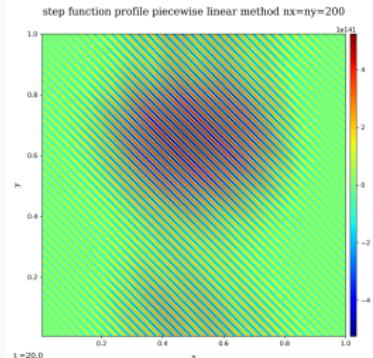
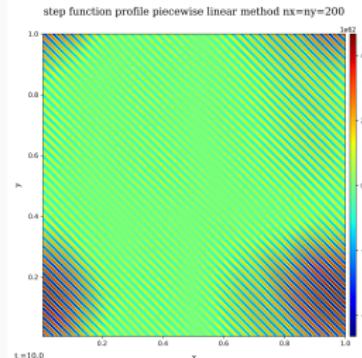
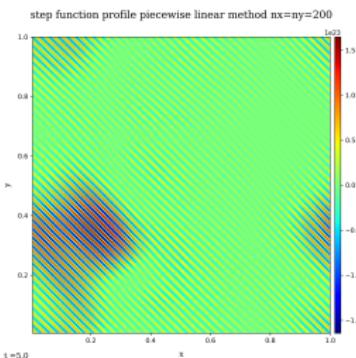
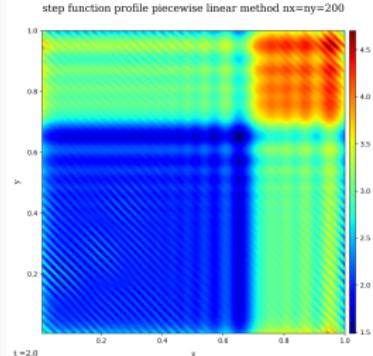
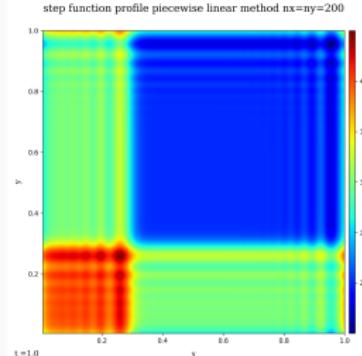
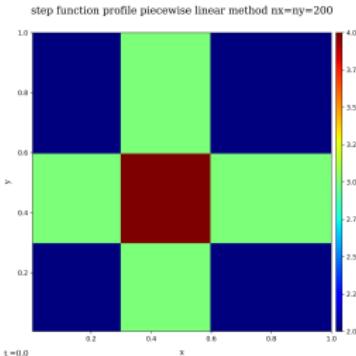
step function profile with VanLeer slope limiter nx=ny=200



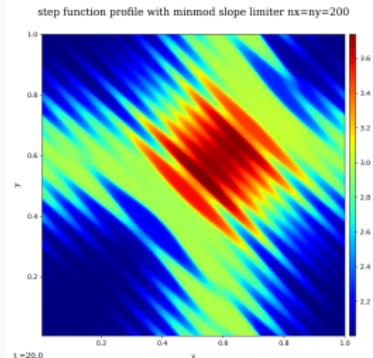
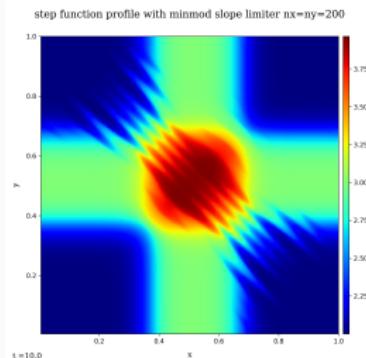
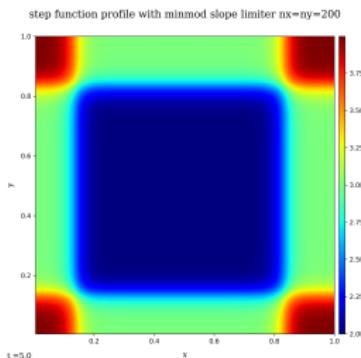
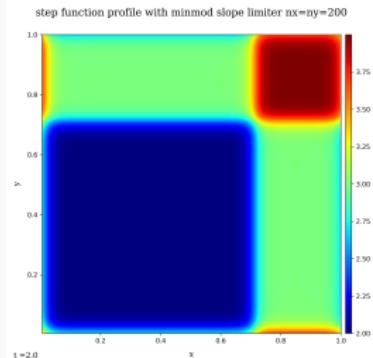
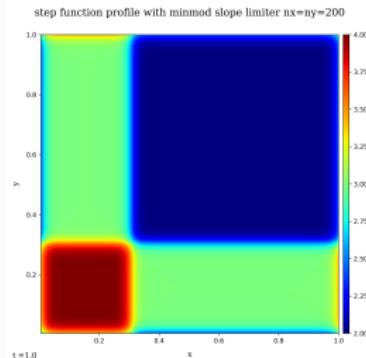
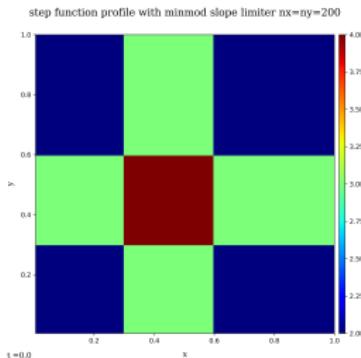
# $u = v = \sqrt{2}/2$ Piecewise Constant Method



# $u = v = \sqrt{2}/2$ Piecewise Linear Method

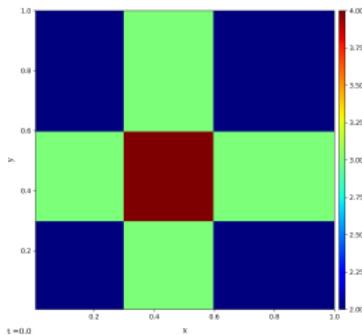


# $u = v = \sqrt{2}/2$ Minmod Slope Limiter

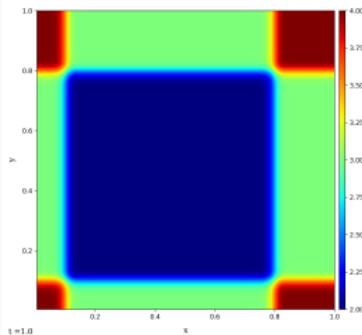


# $u = v = \sqrt{2}/2$ Van Leer Slope Limiter

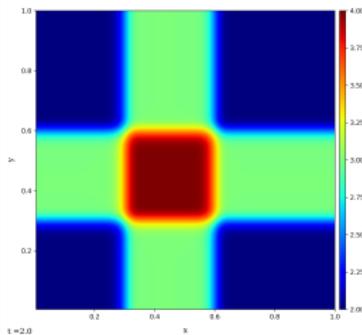
step function profile with VanLeer slope limiter nx=ny=200



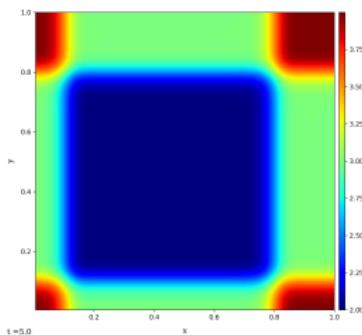
step function profile with VanLeer slope limiter nx=ny=200



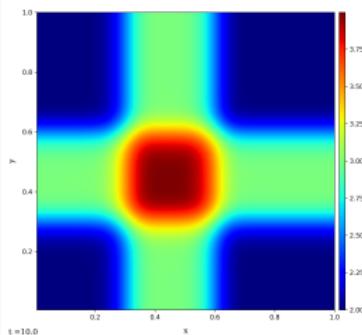
step function profile with VanLeer slope limiter nx=ny=200



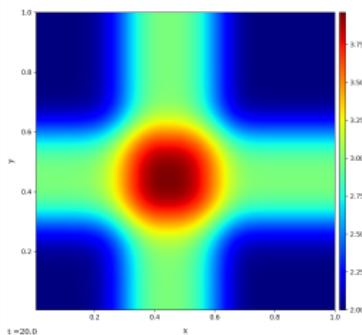
step function profile with VanLeer slope limiter nx=ny=200



step function profile with VanLeer slope limiter nx=ny=200



step function profile with VanLeer slope limiter nx=ny=200



Program, plotting scripts and this presentation available on  
[https://bitbucket.org/mivkov/computational\\_astrophysics](https://bitbucket.org/mivkov/computational_astrophysics)