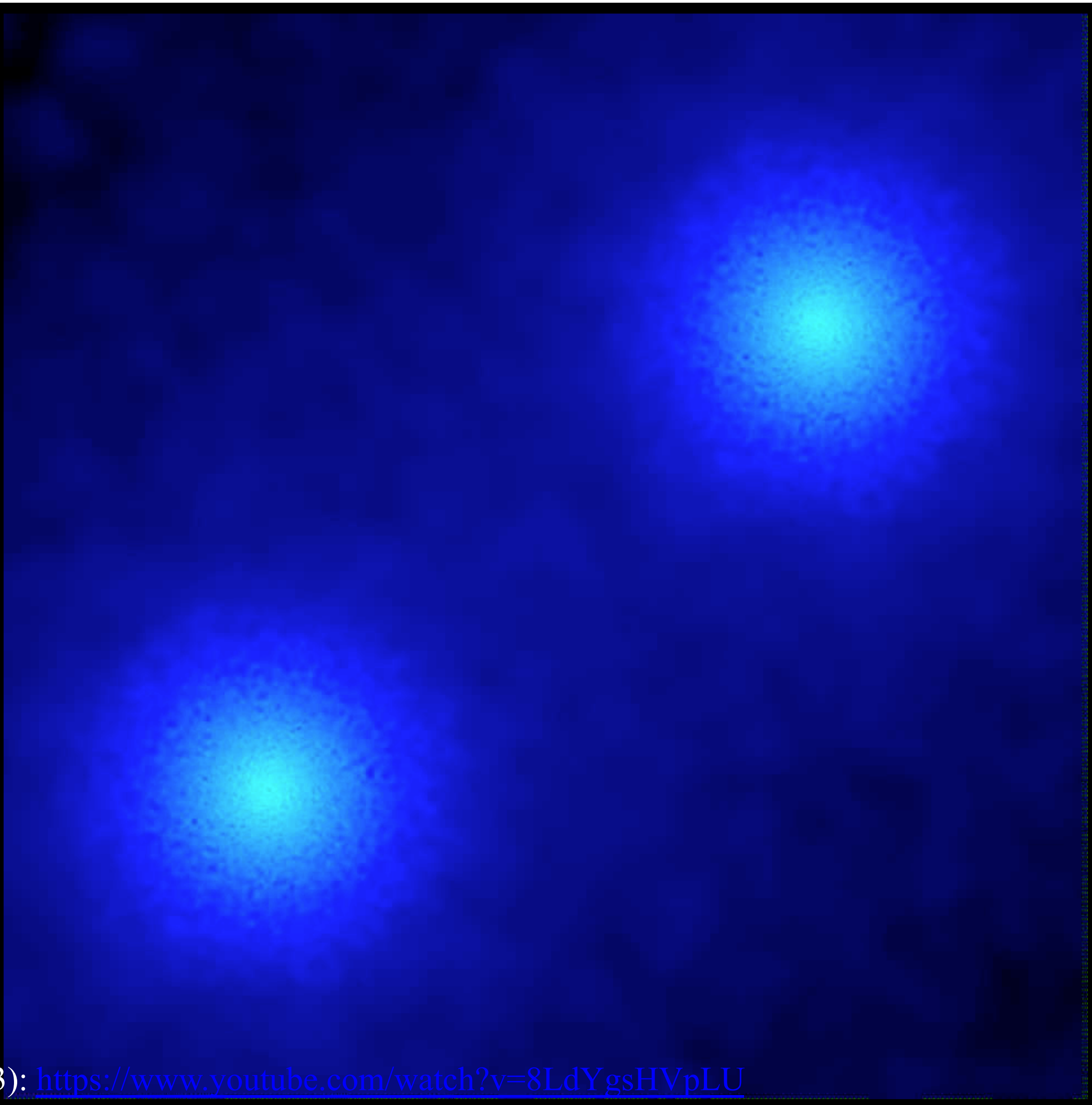
Numerical Hydrodynamics II

James Wurster

Fluids: PH4031

16 February 2021

Example: Astrophysics: Galaxy merger



Wurster & Thacker (2013): https://www.youtube.com/watch?v=8LdYgsHVpLU

Complex Calculation: Required Components

Re-cap

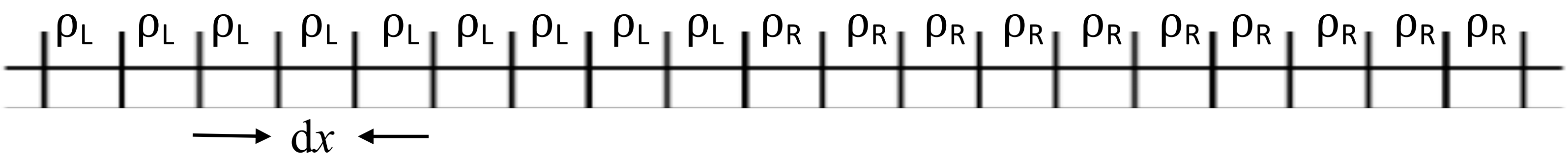
- To solve any system numerically, we require

A method to divide the region (e.g. grids)
A method to describe the evolution of the region
(i.e. the set of fluid dynamics equations)
A method to describe the edge of the region (i.e. boundary conditions)
The initial properties of the system (i.e. initial conditions)

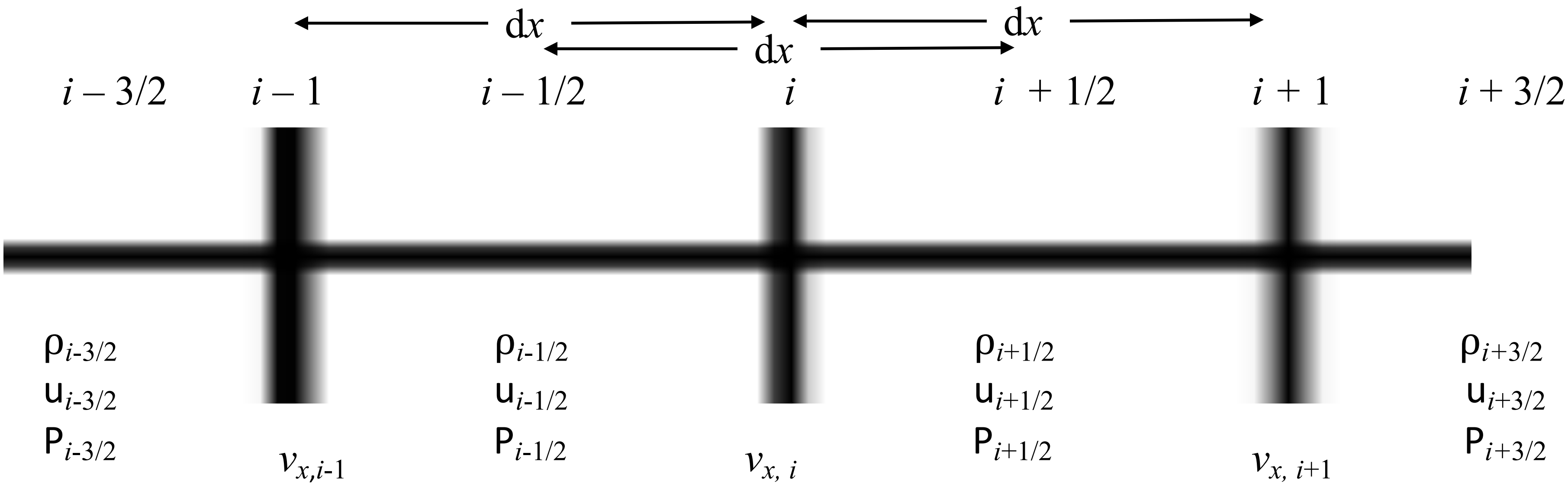
Defining your problem:
Defining quantities

Re-cap

- Eulerian grid: grid of constant spacing



- A few cells:



- Scalars are calculated at *cell-centre*
- Vectors are calculated at *cell-interface*

Fluid equations: Continuum vs 1D-Numerical

Re-cap

- Continuum equations

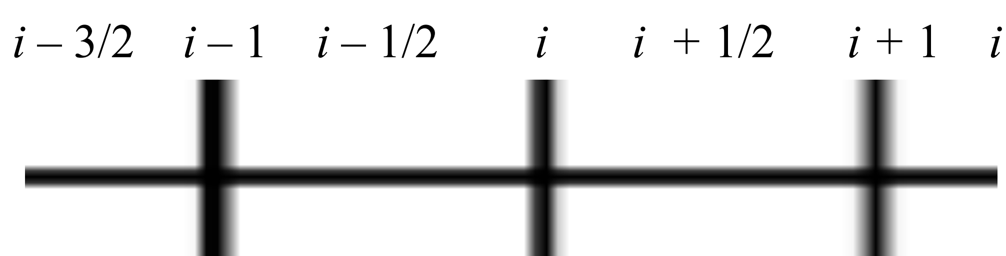
$$\text{Continuity equation:} \quad \frac{\mathrm{D}\rho}{\mathrm{D}t} = -\rho\nabla\cdot\boldsymbol{v}$$

$$\text{Equation of motion:} \quad \frac{\mathrm{D}\boldsymbol{v}}{\mathrm{D}t} = -\frac{1}{\rho}\nabla P$$

$$\text{Energy equation:} \quad \frac{\mathrm{D}u}{\mathrm{D}t} = -\frac{P}{\rho}\nabla\cdot\boldsymbol{v}$$

$$\text{Equation of state:} \quad P = (\gamma-1)\,\rho u$$

- There exists various schemes (e.g. donner-cell) that stabilise the code



Fluid equations: Continuum vs 1D-Numerical

Re-cap

- Discrete equations in Eulerian form:

$$
\begin{aligned}
v_{x,i}^{n+\frac{1}{2}} &= v_{x,i}^{n-\frac{1}{2}} - dt\left(\frac{2}{\rho_{i+\frac{1}{2}}^{n} + \rho_{i-\frac{1}{2}}^{n}}\frac{P_{i+\frac{1}{2}}^{n} - P_{i-\frac{1}{2}}^{n}}{dx} + v_{x,i}^{n-\frac{1}{2}} f(v)\right)\\
\rho_{i+\frac{1}{2}}^{n+1} &= \rho_{i+\frac{1}{2}}^{n} - dt\left(\rho_{i+\frac{1}{2}}^{n}\frac{v_{x,i+1}^{n+\frac{1}{2}} - v_{x,i}^{n+\frac{1}{2}}}{dx} + \frac{v_{x,i+1}^{n+\frac{1}{2}} + v_{x,i}^{n+\frac{1}{2}}}{2} f(\rho)\right)\\
u_{i+\frac{1}{2}}^{n+1} &= u_{i+\frac{1}{2}}^{n} - dt\left(\frac{P_{i+\frac{1}{2}}^{n}}{\rho_{i+\frac{1}{2}}^{n}}\frac{v_{x,i+1}^{n+\frac{1}{2}} - v_{x,i}^{n+\frac{1}{2}}}{dx} + \frac{v_{x,i+1}^{n+\frac{1}{2}} + v_{x,i}^{n+\frac{1}{2}}}{2} f(u)\right)\\
P_{i+\frac{1}{2}}^{n+1} &= (\gamma - 1)\,\rho_{i+\frac{1}{2}}^{n+1} u_{i+\frac{1}{2}}^{n+1}
\end{aligned}
$$

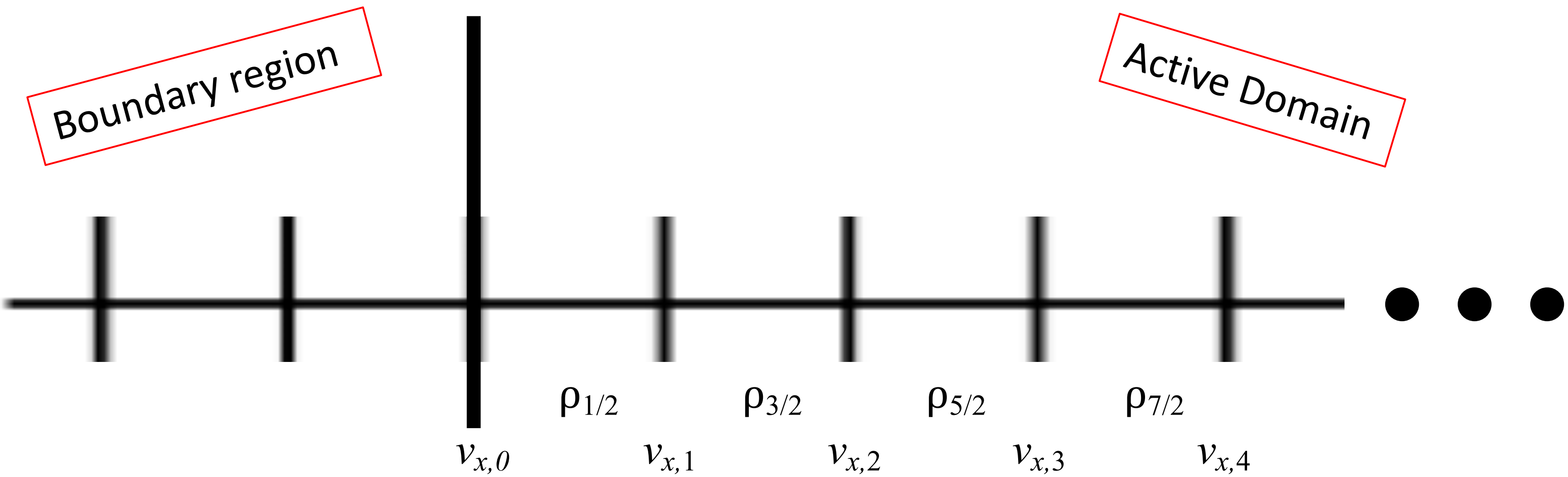
- *f(a)* represents the Lagrangian part of the derivative, and can be first, second, third, ..., order

$i-3/2$ $\quad i-1$ $\quad i-1/2$ $\quad i$ $\quad i+1/2$ $\quad i+1$ $\quad i+3/2$

Boundaries

Re-cap

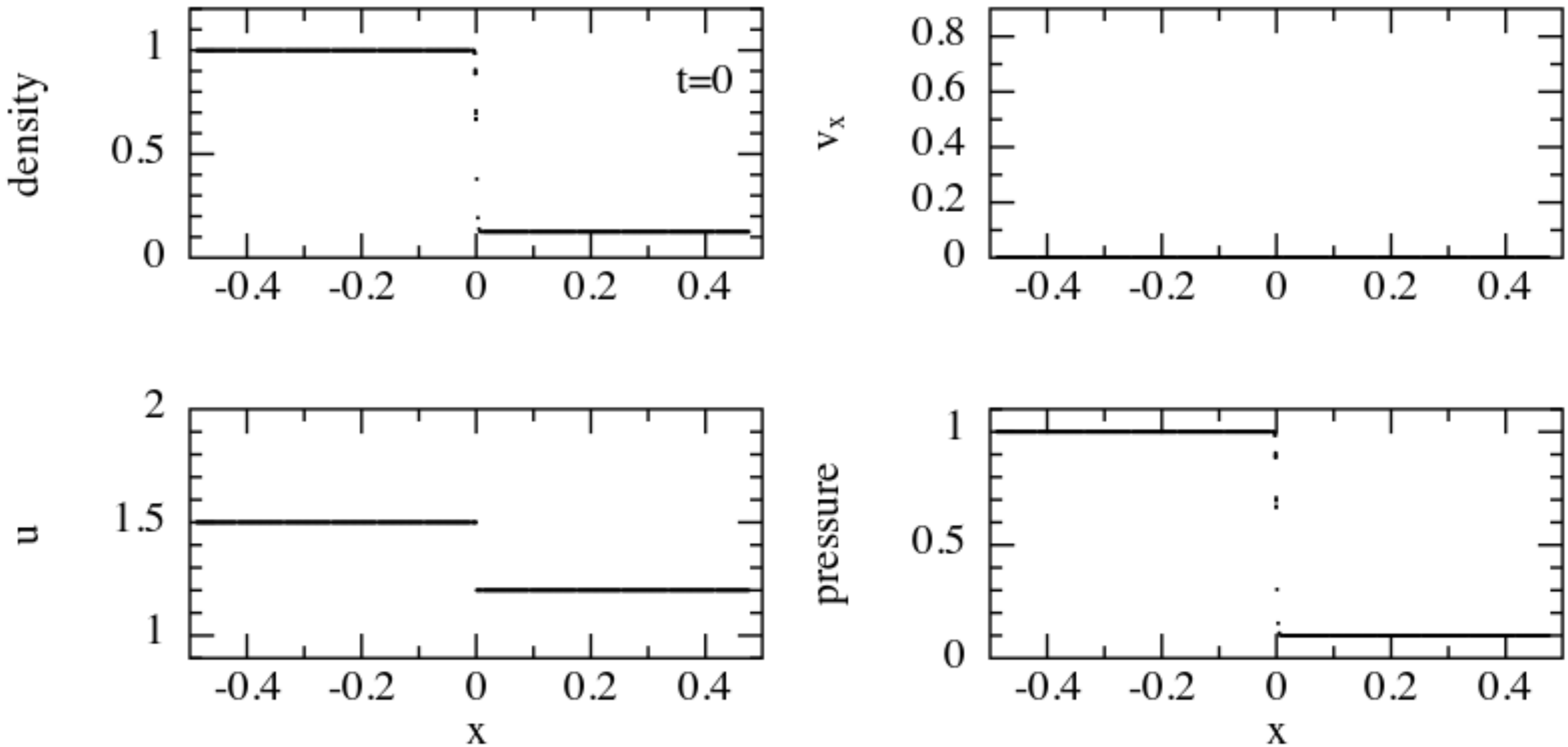
- We almost have enough information to run a simple simulation, but what happens at the edge of the simulation?
- Similar to solving differential equations, boundary conditions are required (e.g.)
  - Fixed / Inflow
  - Outflow
  - Reflective
  - Periodic



Initial conditions: Sod Shock

Re-cap

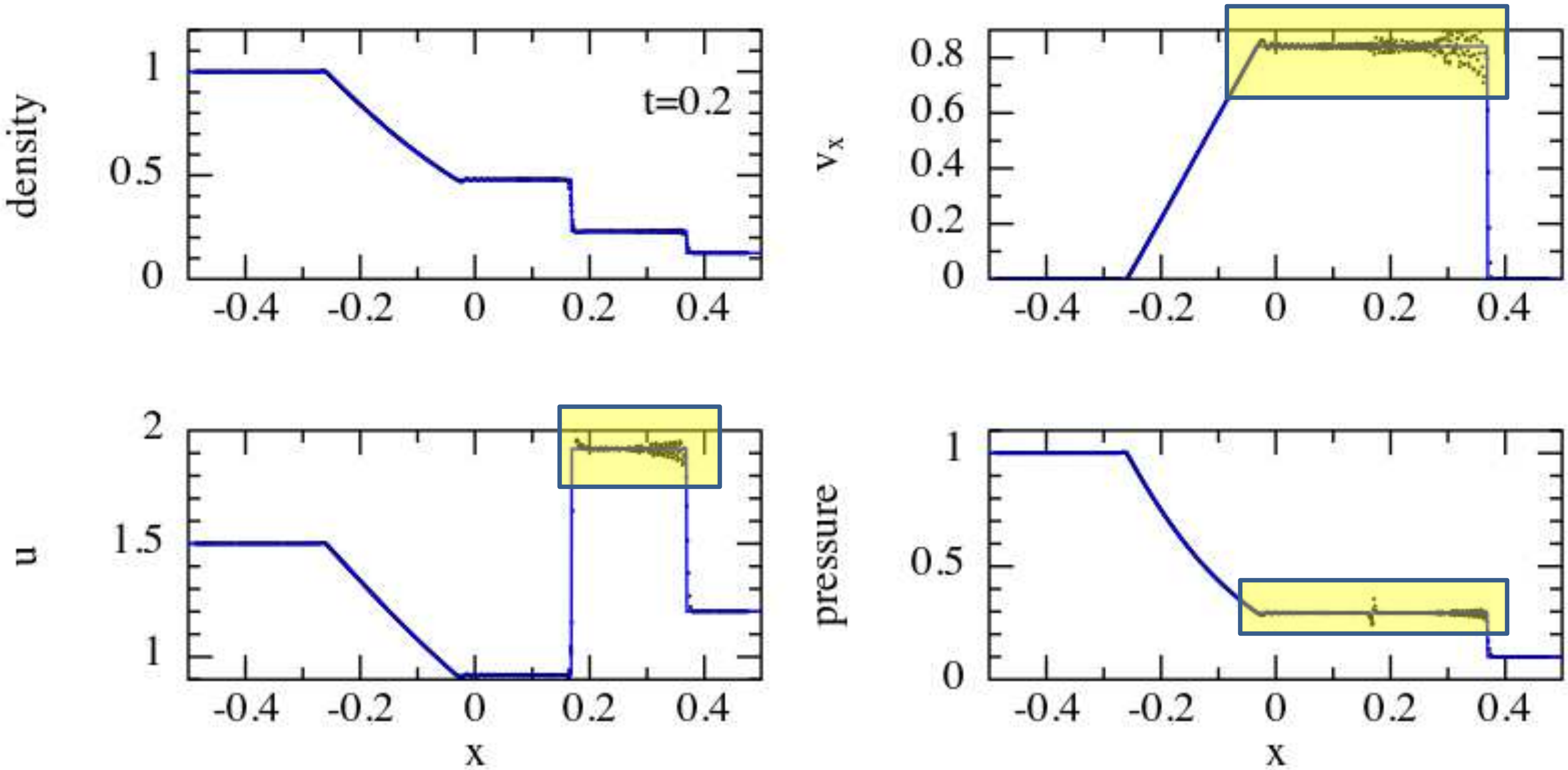
- Initial conditions for the Sod Shock
- Boundary Conditions: fixed (since we stop the problem before the shock hits the walls)



Sod Shock Evolution

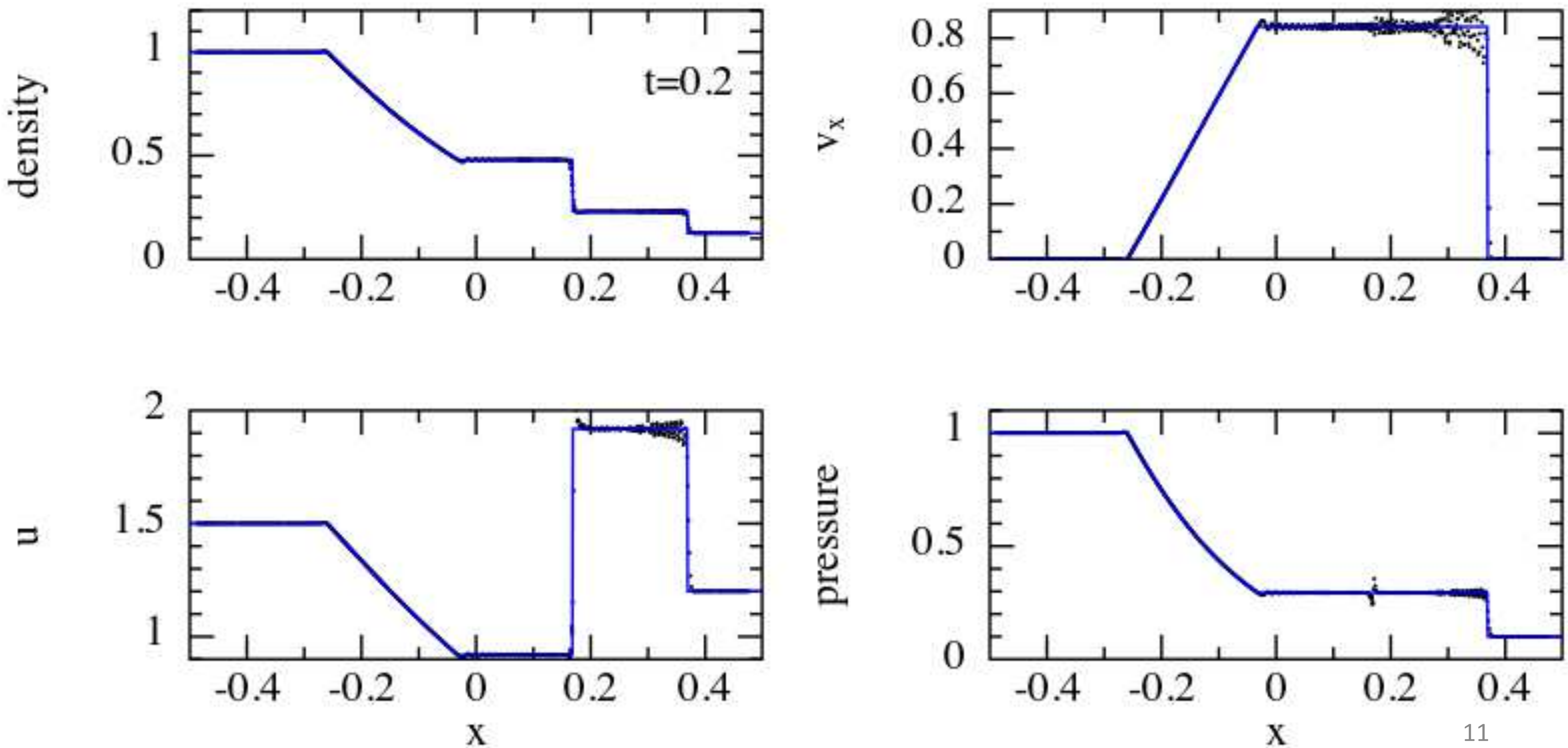
Re-cap

- Ringing and instabilities occur at the shock wave dampen as the shock propagates



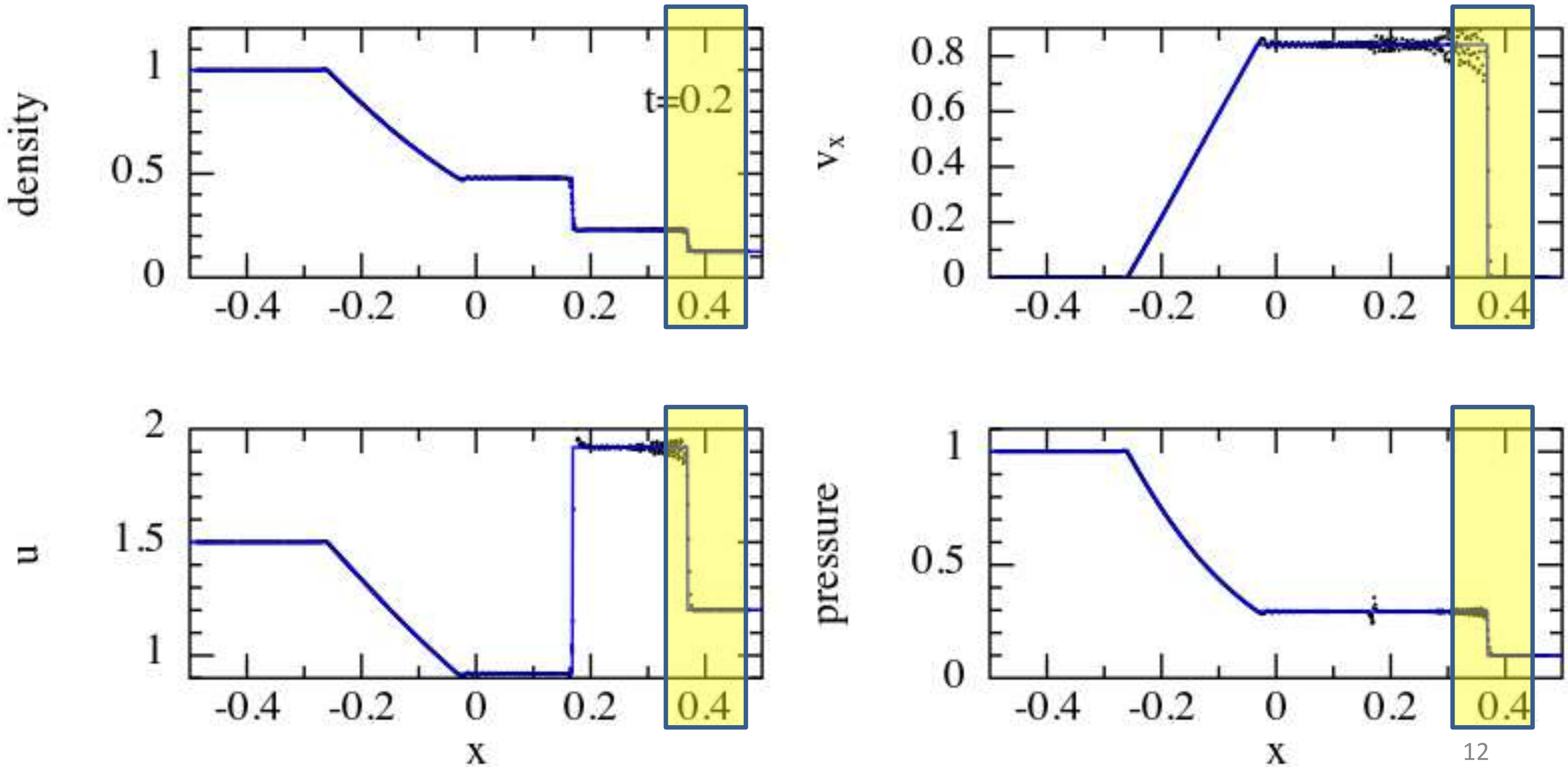
Sod Shock: Structure of shock

- Physical acceptable solution contains three distinct waves:
  - shock wave
  - contact discontinuity
  - rarefaction wave



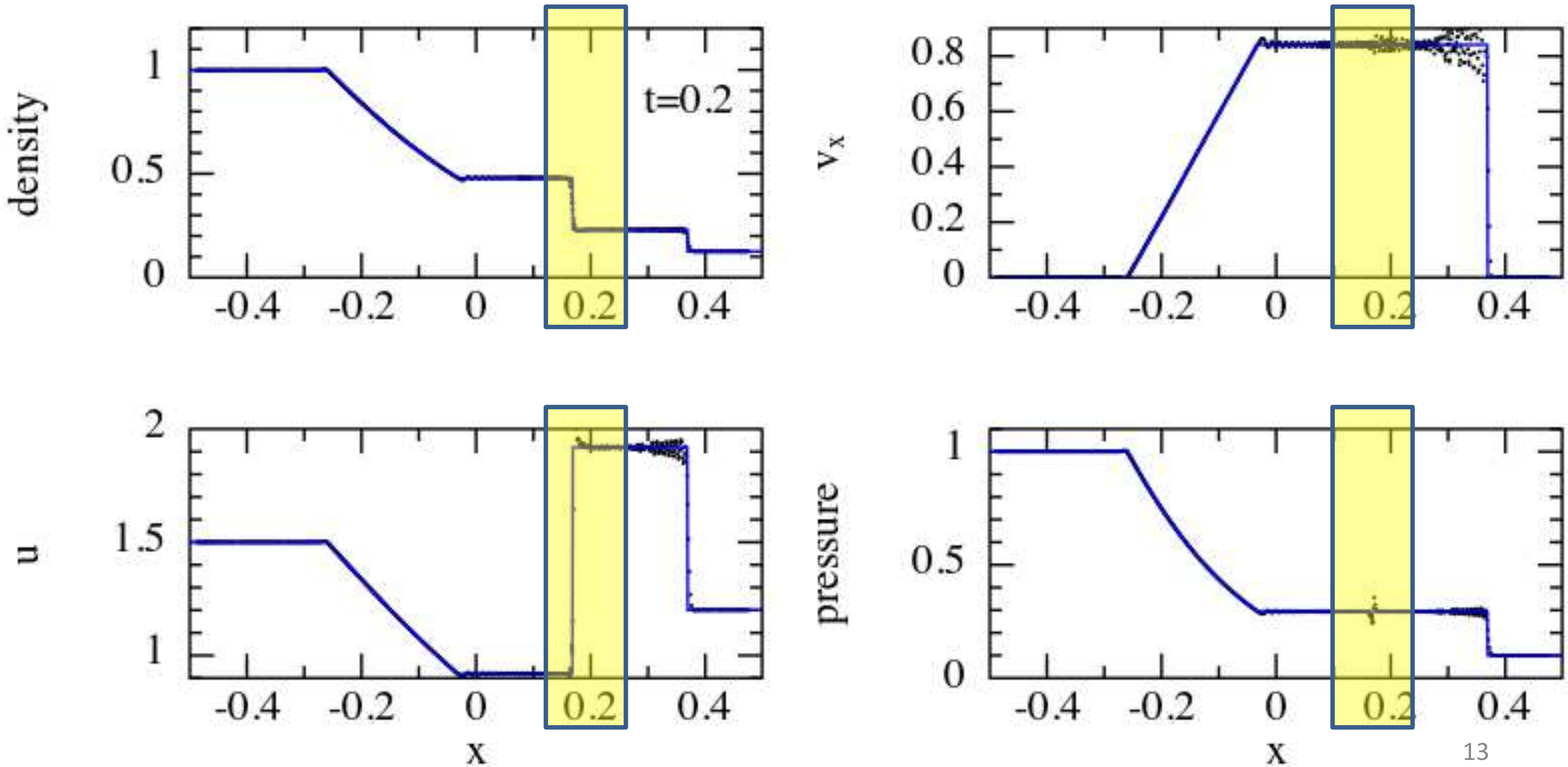
Sod Shock: Structure of shock

- Shock Wave
  - Strong discontinuity in density, pressure and fluid velocity
  - Supersonic movement caused by a strong pressure or velocity gradient



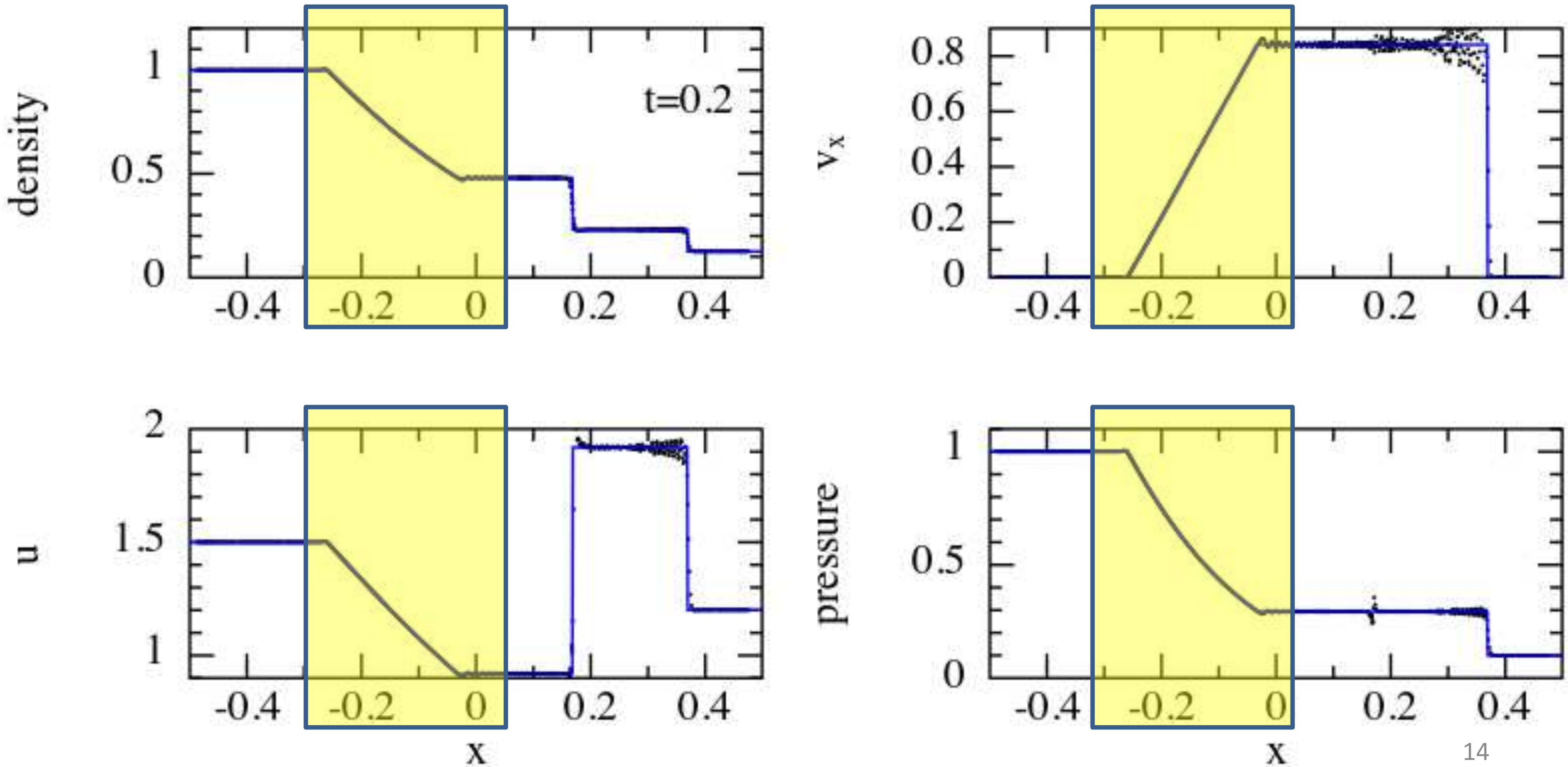
Sod Shock: Structure of shock

- Contact discontinuity
  - Discontinuity in density; pressure and fluid velocity are constant
  - Moving with the local fluid velocity



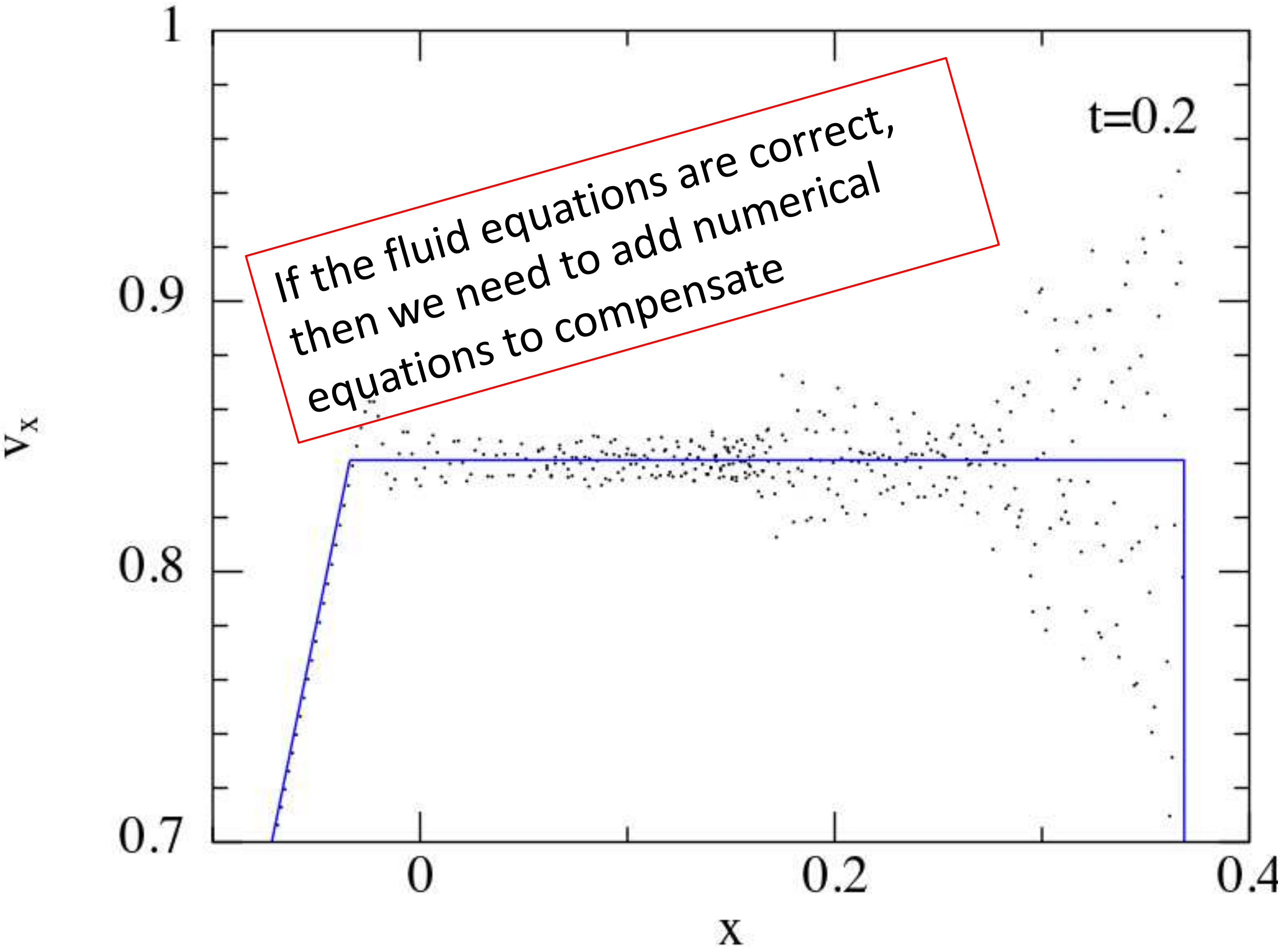
Sod Shock: Structure of shock

- Rarefaction wave
  - Continuous change in density, pressure and fluid velocity
  - Moving with the sound speed relative with respect to the local fluid velocity



Sod Shock Evolution

- Ringing and instabilities occur at the **shock wave** & dampen as the shock propagates
  - Numerical methods often have difficulty resolving sharp discontinuities
  - The algorithms typically overpredict in one cell, then underpredict in the next



Sod Shock:
Artificial terms: Artificial viscosity

- Numerical algorithms are required for stability
- The form and parameterisation of these requires careful consideration to suppress numerical instabilities but not physical instabilities
- Modify the velocity by adding in an artificial pressure term:

$$v_{x,i}^{n+\frac{1}{2}} = v_{x,i}^{n-\frac{1}{2}} - dt\left[\frac{2}{\rho_{i+\frac{1}{2}}^{n} + \rho_{i-\frac{1}{2}}^{n}} \frac{\left(P_{i+\frac{1}{2}}^{n} + q_{i+\frac{1}{2}}^{n}\right) - \left(P_{i-\frac{1}{2}}^{n} + q_{i-\frac{1}{2}}^{n}\right)}{dx} + v_{x,i}^{n-\frac{1}{2}} f(v)\right]$$

where

$$q_{i+\frac{1}{2}}^{n} = \begin{cases} \alpha^2 \rho_{i+\frac{1}{2}}^{n} \Delta v_i^2 & \text{if } \Delta v_i < 0 \\ 0 & \text{else} \end{cases}$$

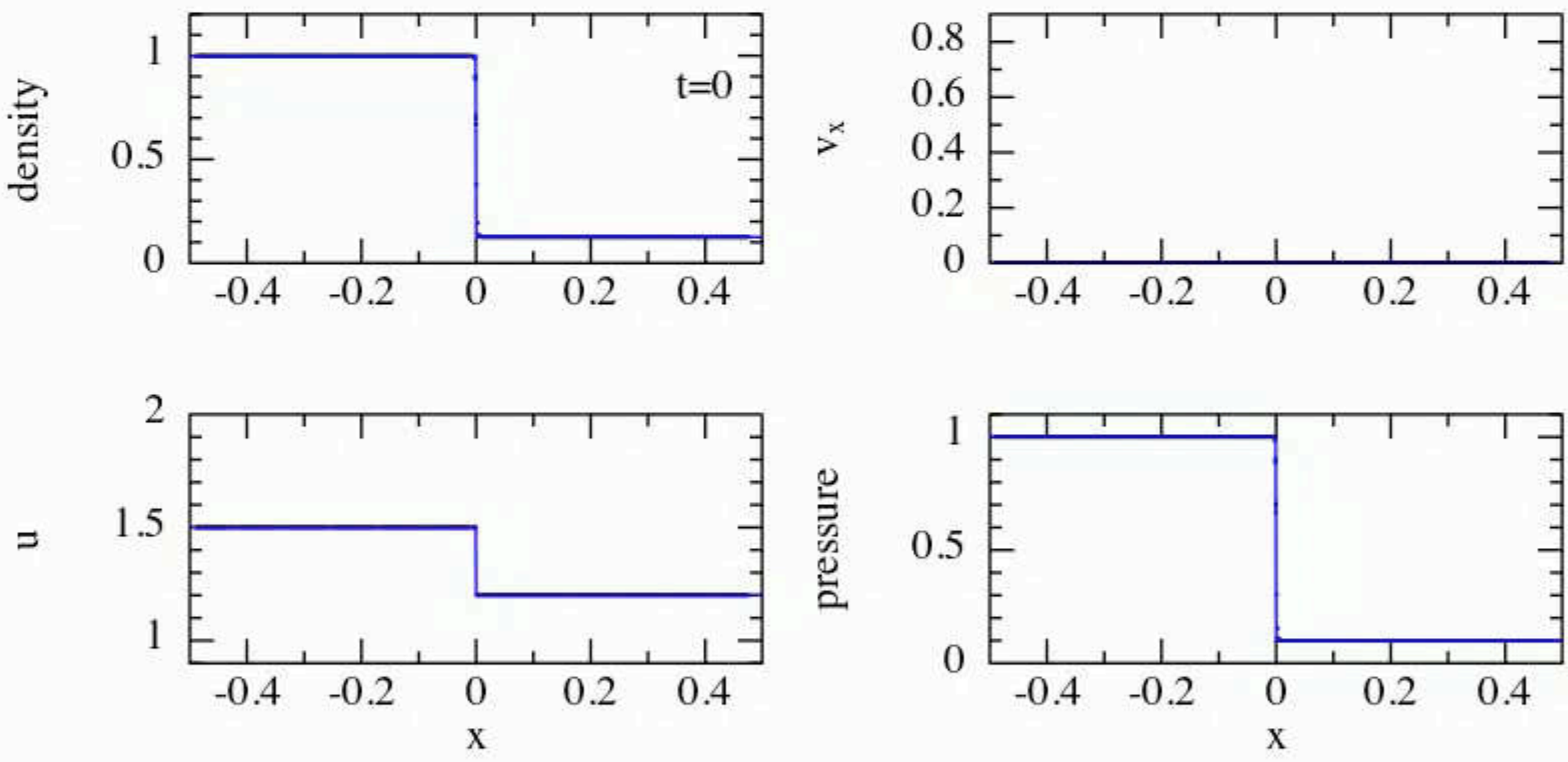
and

$$\Delta v_i = \left(v_{i+\frac{1}{2}}^{n-\frac{1}{2}} - v_{i-\frac{1}{2}}^{n-\frac{1}{2}}\right)$$

- Only add q for converging flows
- α is a value between 0 & 1, which can either be a fixed value, or dynamically calculated

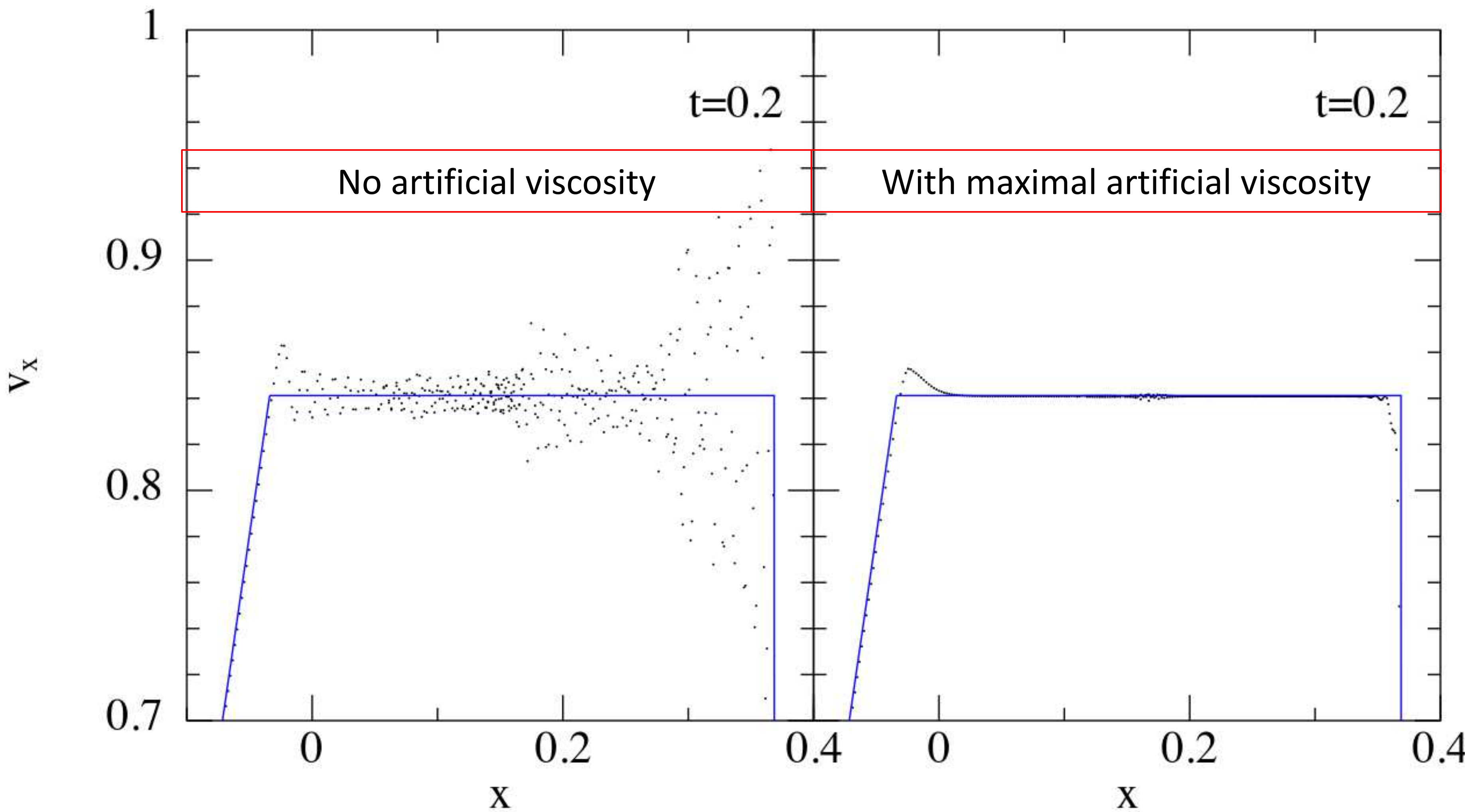
Sod Shock: Artificial terms: Artificial viscosity

- Numerical algorithms are required for stability
- The form and parameterisation of these requires careful consideration to suppress numerical instabilities but not physical instabilities



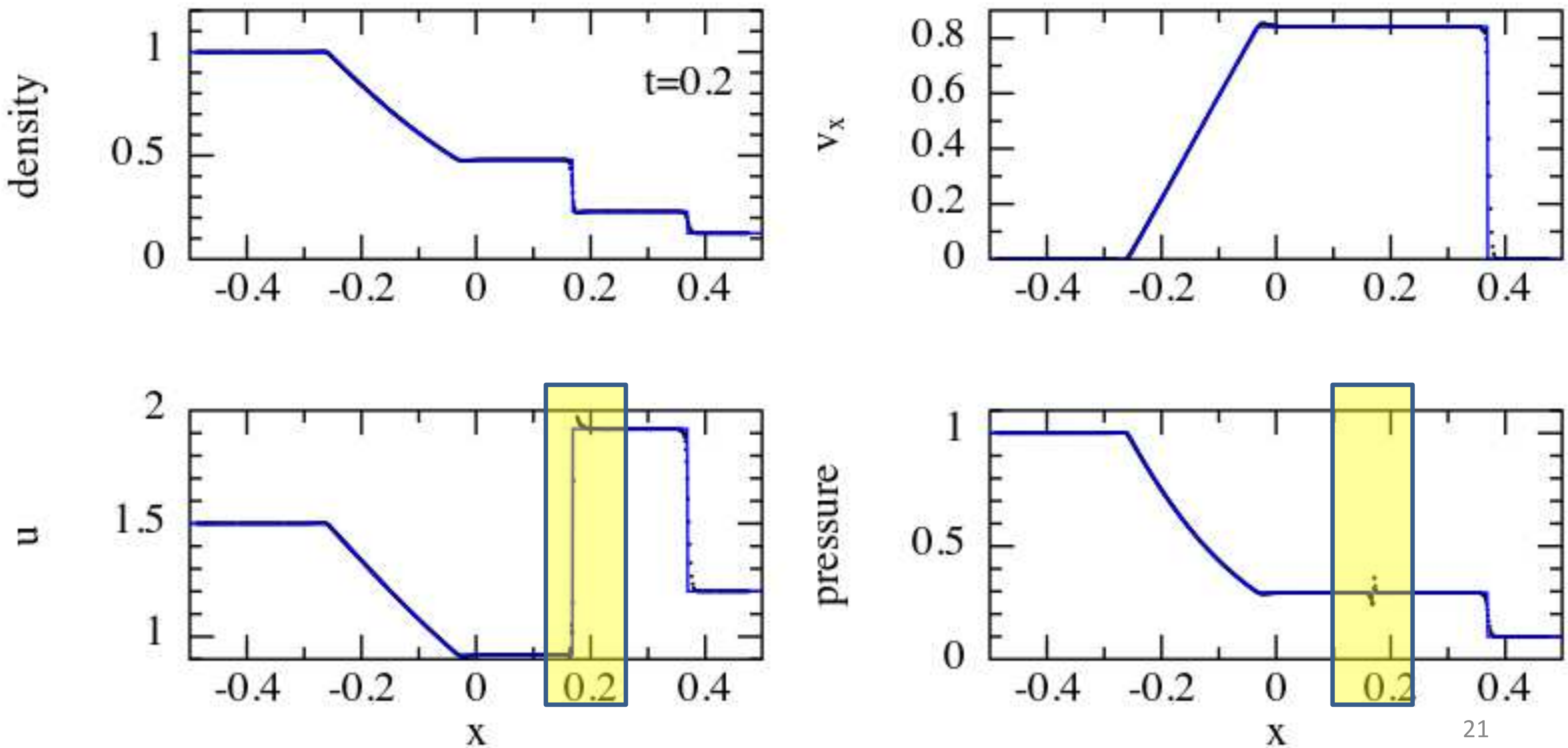
Sod Shock: Artificial terms: Artificial viscosity

- Artificial viscosity well suppresses the ringing, and the numerical results (dots) better match the analytical result (blue)



Sod Shock:
Artificial terms: Artificial viscosity

- Even with artificial viscosity, there is still some ‘blips’ in energy and pressure occurring at the **contact discontinuity**



Sod Shock:
Artificial terms: Artificial conductivity

- Numerical algorithms are required for stability
- The form and parameterisation of these requires careful consideration to suppress numerical instabilities but not physical instabilities
- Since there is no instabilities in velocity at the contact discontinuity, we need to add at artificial term to the internal energy: artificial conductivity:

$$\frac{\mathrm{d}u}{\mathrm{d}t} = -\frac{P}{\rho}\nabla\cdot\boldsymbol{v} + \left.\frac{\mathrm{d}u}{\mathrm{d}t}\right|_{\text{artificial}}$$

- There are various forms of artificial conductivity throughout the literature

Sod Shock: Artificial terms: Art. viscosity + art. conductivity

- Numerical algorithms are required for stability
- Alternatively, we can use

$$\frac{\mathrm{d}u}{\mathrm{d}t} = -\frac{P+q}{\rho}\nabla\cdot\boldsymbol{v}$$

- Given the similarities between the energy equation and equation of motion, we can instead use

$$P^{n+1}_{i+\frac{1}{2},j+\frac{1}{2}} = (\gamma-1)\,\rho^{n+1}_{i+\frac{1}{2},j+\frac{1}{2}}\,u^{n+1}_{i+\frac{1}{2},j+\frac{1}{2}} + q^{n}_{i+\frac{1}{2}}$$

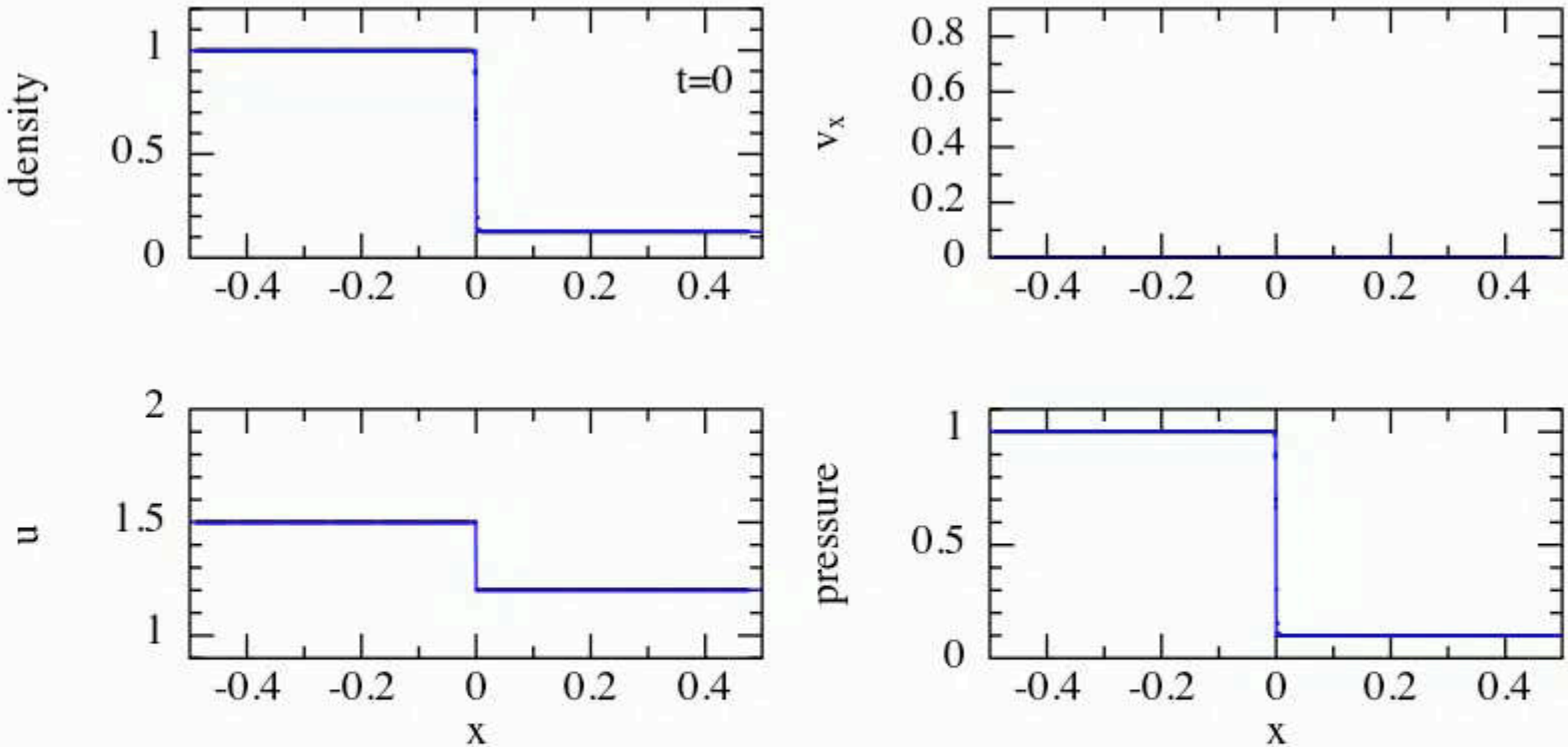
- with

$$v^{n+\frac{1}{2}}_{x,i} = v^{n-\frac{1}{2}}_{x,i} - dt\left[\frac{2}{\rho^n_{i+\frac{1}{2}}+\rho^n_{i-\frac{1}{2}}}\frac{P^n_{i+\frac{1}{2}}-P^n_{i-\frac{1}{2}}}{dx} + v^{n-\frac{1}{2}}_{x,i}\,f(v)\right]$$

$$u^{n+1}_{i+\frac{1}{2}} = u^n_{i+\frac{1}{2}} - dt\left(\frac{P^n_{i+\frac{1}{2}}}{\rho^n_{i+\frac{1}{2}}}\frac{v^{n+\frac{1}{2}}_{x,i+1}-v^{n+\frac{1}{2}}_{x,i}}{dx} + \frac{v^{n+\frac{1}{2}}_{x,i+1}+v^{n+\frac{1}{2}}_{x,i}}{2}f(u)\right)$$

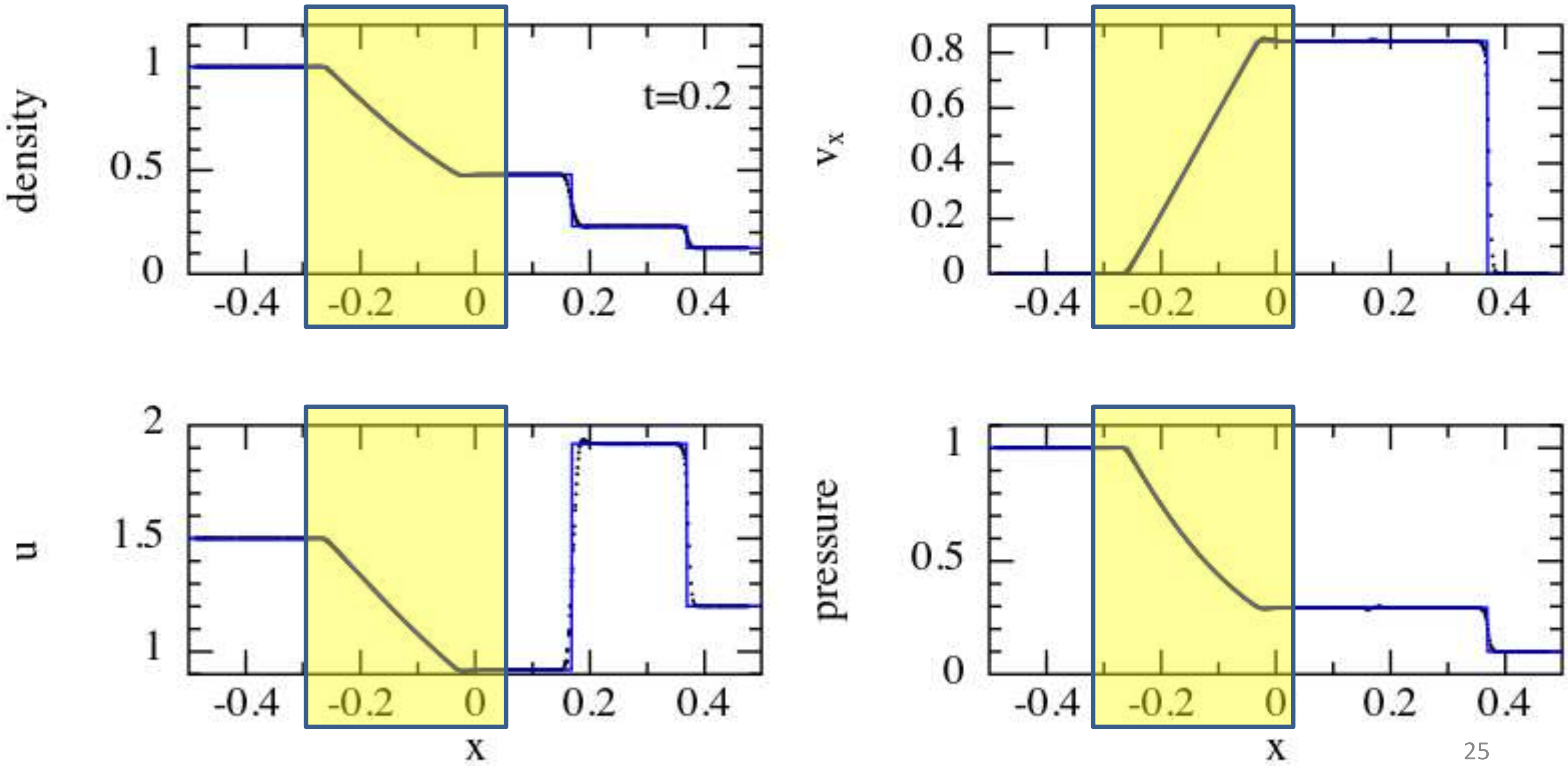
Sod Shock:
Artificial terms: Art. viscosity + art. conductivity

- Numerical algorithms are required for stability
- The form and parameterisation of these requires careful consideration to suppress numerical instabilities but not physical instabilities



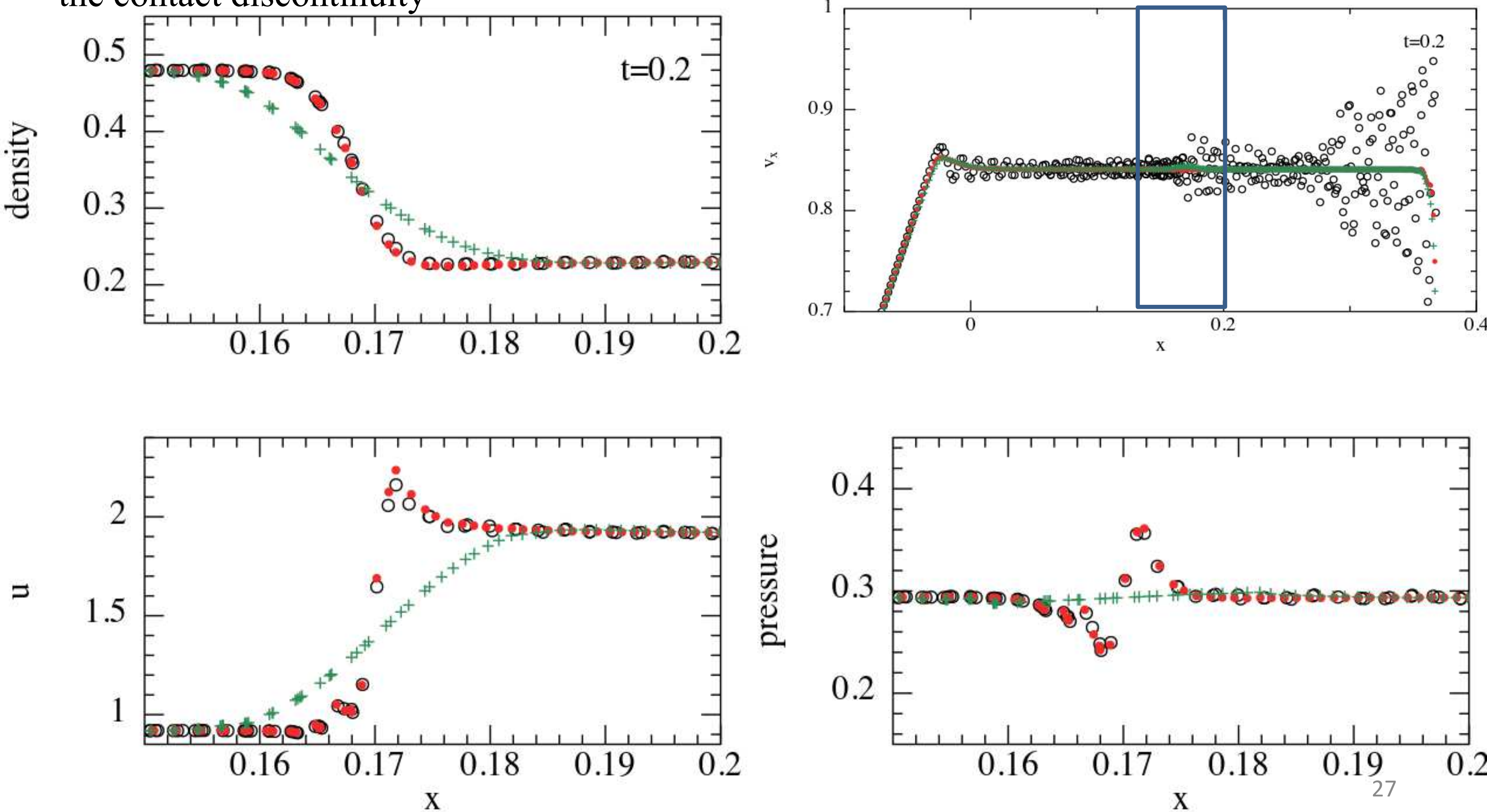
Sod Shock:
Artificial terms: Art. viscosity + art. conductivity

➢ The **rarefaction wave** is stable, thus we do not need to add in another numerical algorithms



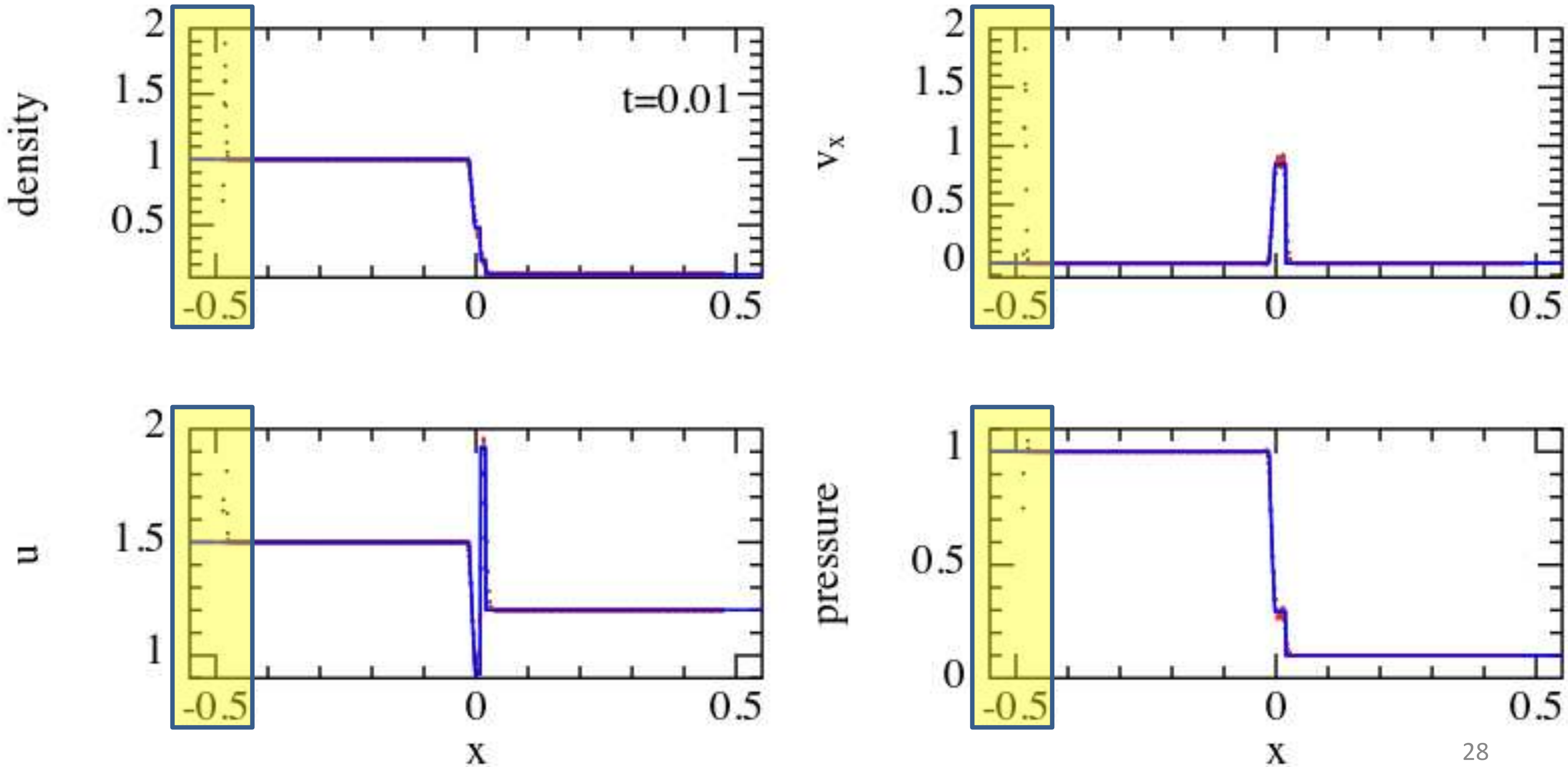
Sod Shock: Artificial terms: Art. viscosity + art. conductivity

- Comparing without artificial terms (black), with artificial viscosity (red) & both (green) at the contact discontinuity



Sod Shock: Boundaries: A Cautionary Tale

- Incorrect boundaries (either by choice or a bug) will lead to incorrect answers



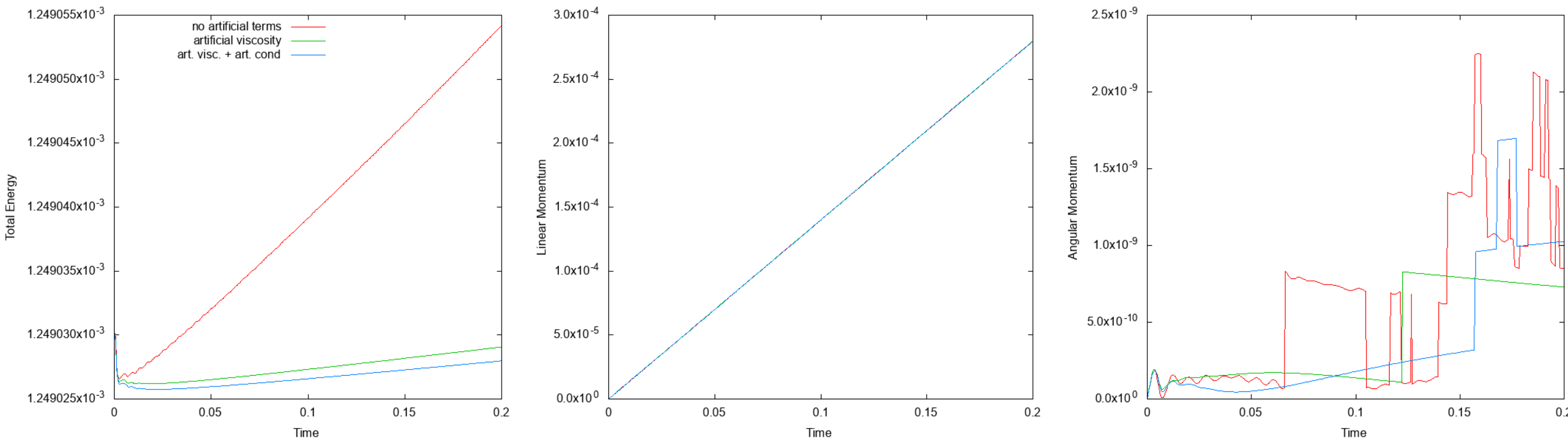
Conservation Laws

- Physics demands that certain quantities are conserved:
  - Mass
  - Energy
  - Linear momentum
  - Angular momentum

- Numerical experiments should also conserve these values
- Eulerian formalism (the equations in these slides) is not guaranteed to conserve mass
- SPH formalism (the graphs in these slides) is guaranteed to conserve mass by design

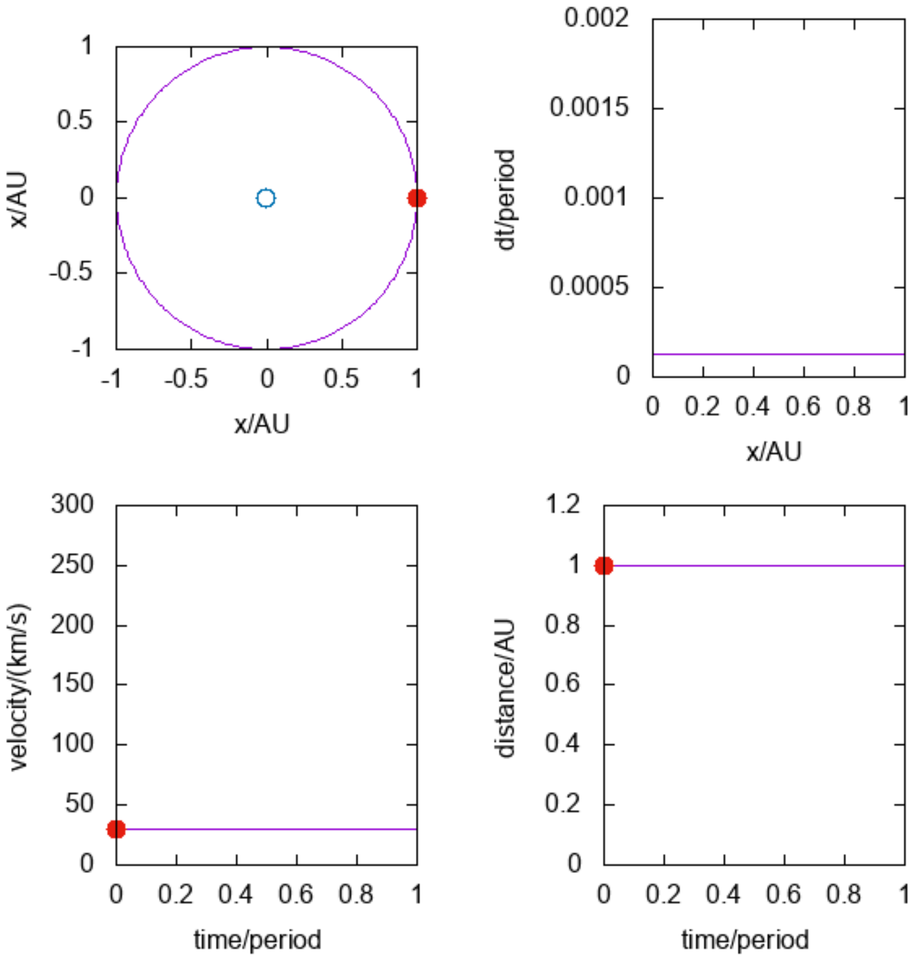
Conservation Laws

- Physics demands that certain quantities are conserved:
  - Mass
  - Energy
  - Linear momentum
  - Angular momentum
- Conserved quantities for the Sod Shock tube:



Timestepping

- As the simulation evolves, what timestep do we choose?
  - As long as possible, but short enough to resolve the physics
  - We want this to be chosen by the programme and not as an input parameter

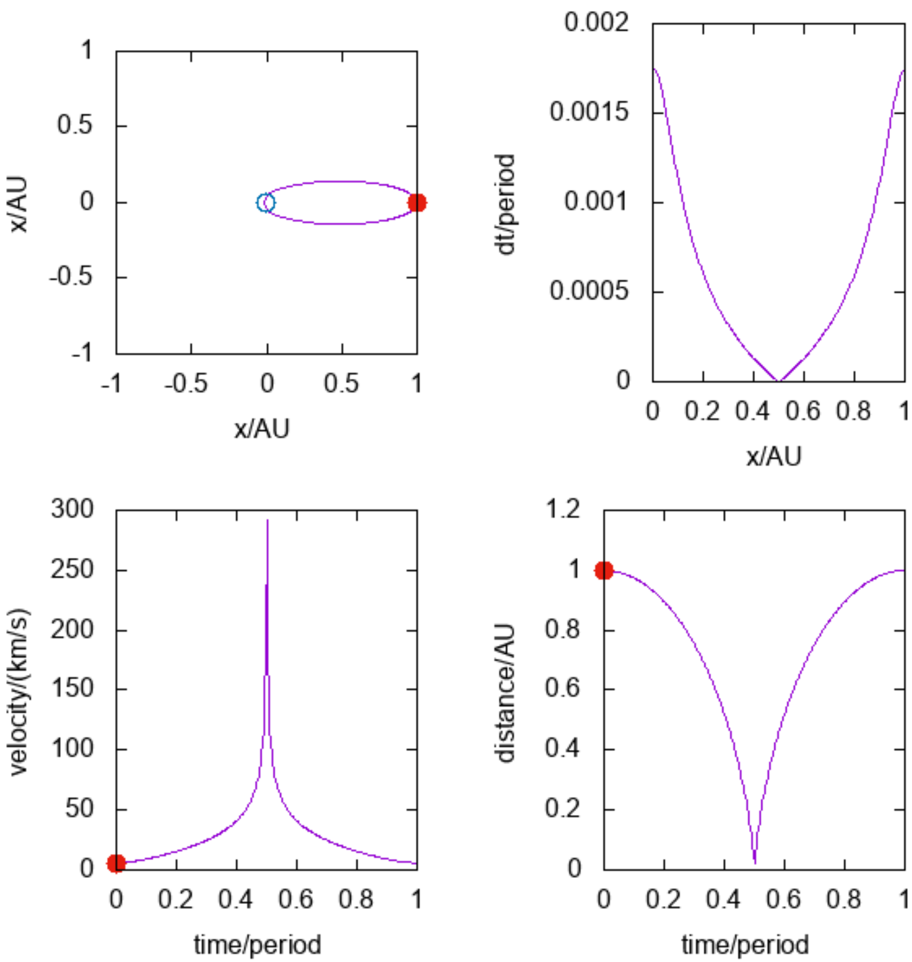


$$\mathrm{d}t = \frac{1}{N}\frac{2\pi\vec{r}}{\vec{v}} \quad \text{where } N \text{ is the number of steps per orbit}$$

$$= C\frac{\vec{r}}{\vec{v}} \quad \text{where } C \leq 1$$

Timestepping

- As the simulation evolves, what timestep do we choose?
  - As long as possible, but short enough to resolve the physics
  - We want this to be chosen by the programme and not as an input parameter



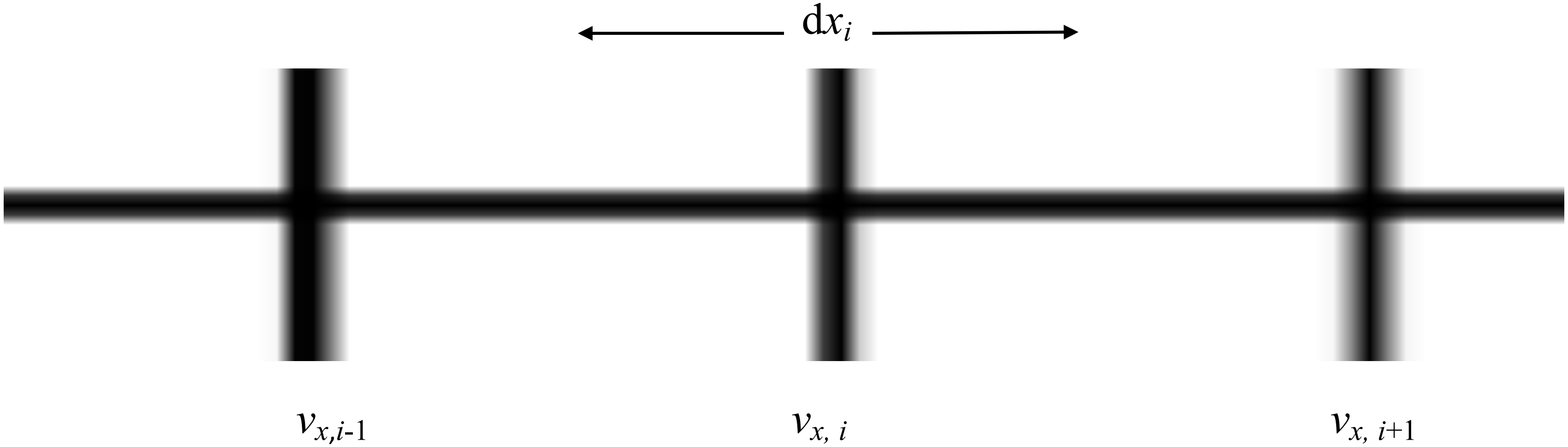
$$\mathrm{d}t = \frac{1}{N}\frac{2\pi\vec{r}}{\vec{v}} \quad \text{where } N \text{ is the number of steps per orbit}$$

$$= C\frac{\vec{r}}{\vec{v}} \quad \text{where } C \leq 1$$

Timestepping

- The previous timestep can be extended to numerical fluid dynamics
- Courant–Friedrichs–Lewy is a common limiting timestep:

$$dt = \min\left(C\frac{dx_i}{v_i}, C\frac{dx_i}{c_{\mathrm{s},i}}\right) \quad \text{where} \quad C \leq 1$$



- This is based upon “How long does it take information (e.g. a wave) to travel from one side of the cell to the other?”
- If a wave travels too fast, then it is unresolved and the simulation may break

Timestepping
A practical warning

- Although testable on well known problems, selecting a d*t* coefficient can be challenging in practical simulations
- A timestep governing non-ideal MHD is

$$dt = C\frac{dx_i^2}{\eta_i}$$

where

$C = 1/6$ (Bai 2014)

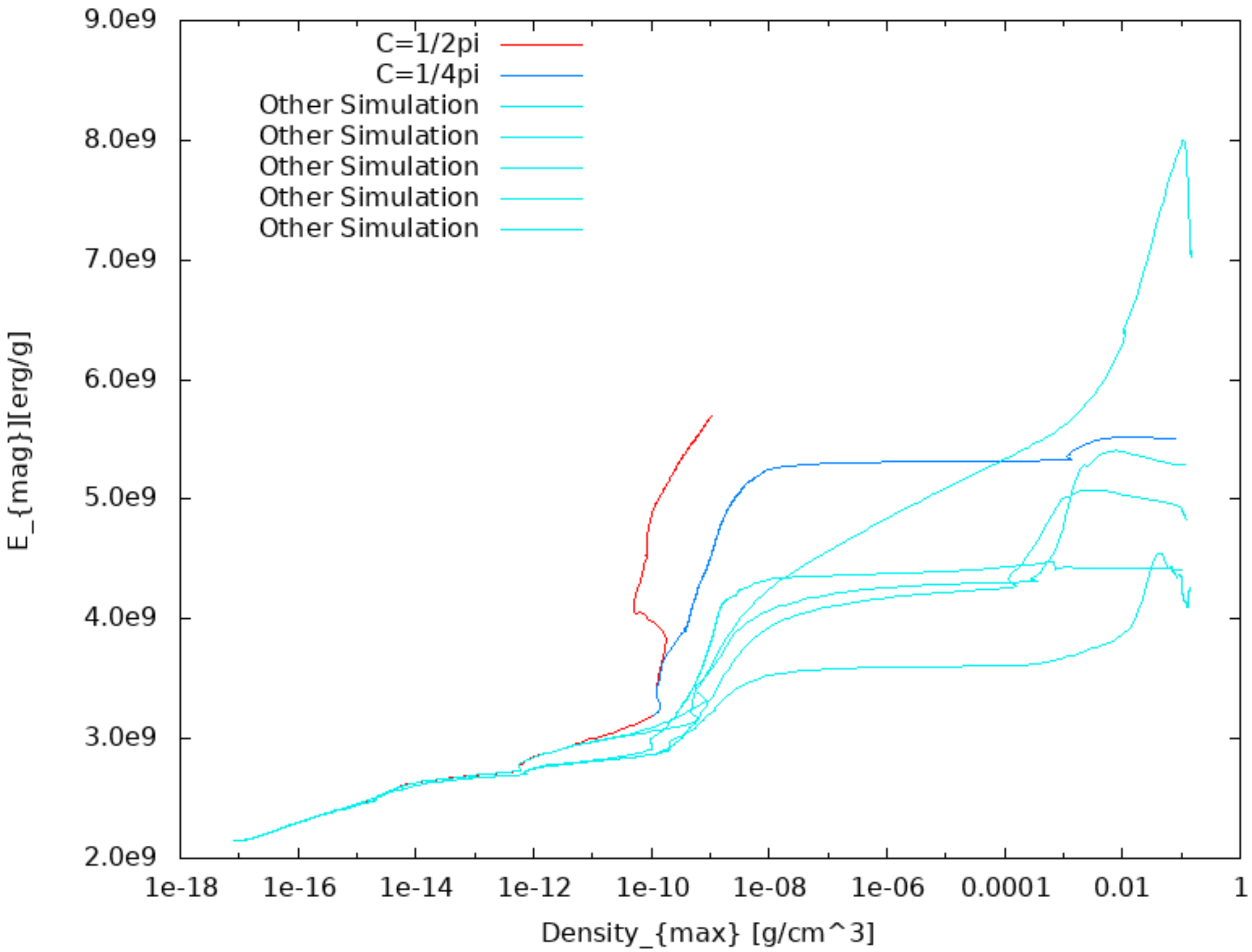
$C = 1/2\pi$ (Wurster+ 2016)

$C = 1/4\pi$ (Wurster, miscellaneous tests)

$C = 1/10\pi$ (Tsukamoto+2015)

Timestepping
A practical warning

- Although testable on well known problems, selecting a d*t* coefficient can be challenging
- Each line represent a different simulation. The red line does not match the rest of the trends
- The blue line is the same model, but with a smaller *C*



Resolution

- Computers have finite resources, and users have finite time/patience
- To determine the desired resolution, must carefully consider
  - What are you trying to model?
  - What are your computational resources?

- Our options are generally
  - Model a small region at high resolution
  - Model a large region at low resolution

Resolution: Geographic example

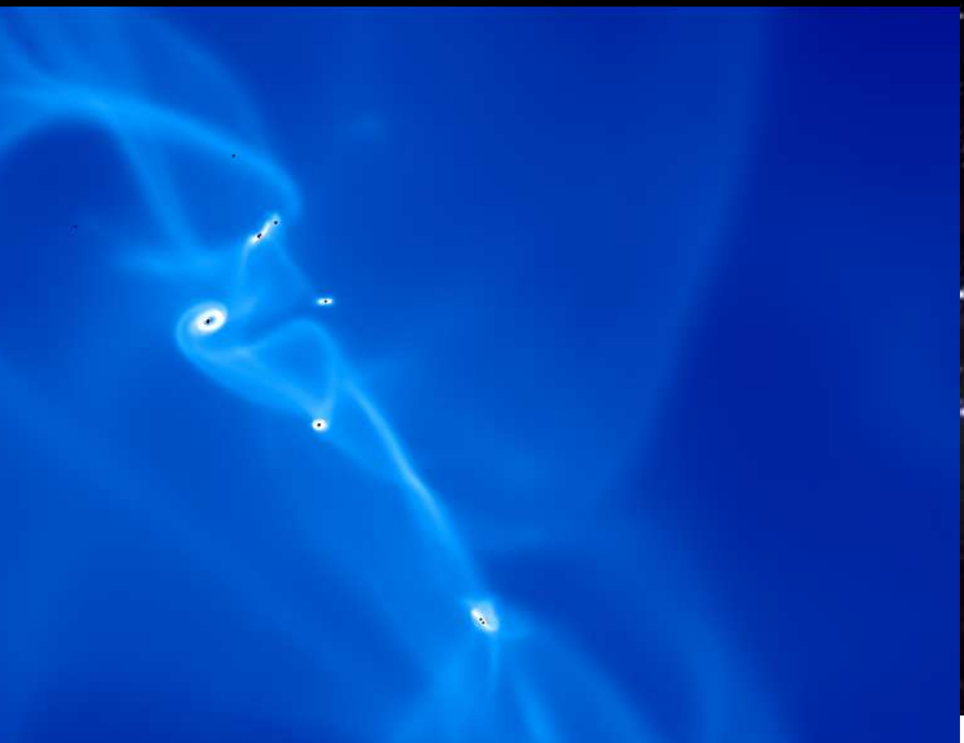
➢ Resolution example:

❖ small scales at high resolution (top left)

❖ large scales at low resolution (bottom right)

❖ in between (top right)

Resolution:
Example: Modelling a molecular cloud



- Want to resolve star formation
- Require 10^5 particles / M_{sun} or $10^{-5}M_{sun}$/particle
- Resources allow for $5x10^6$ particles
- Can reasonably model a cloud of $50M_{sun}$

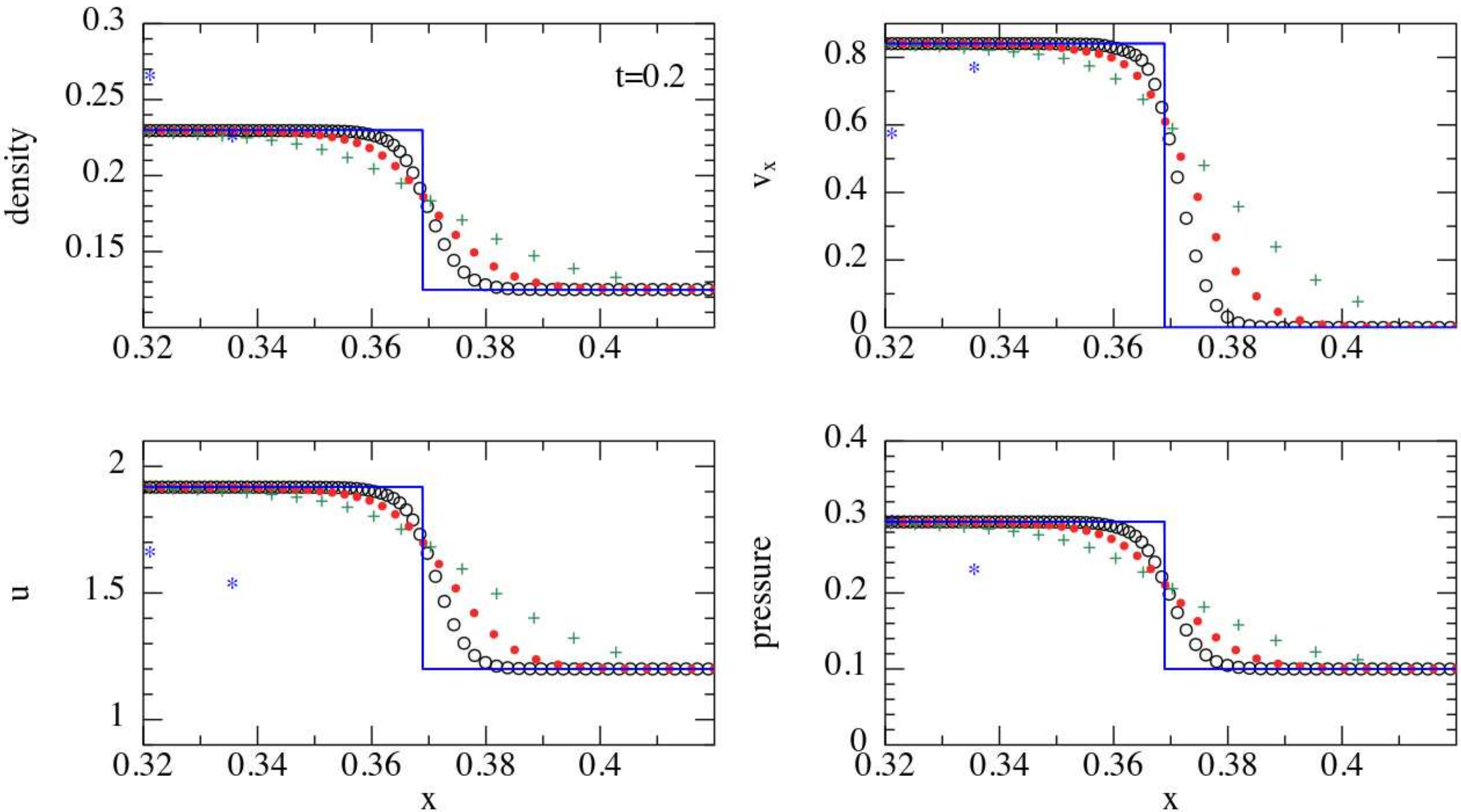
Wurster, Bate & Price (2019)

- Want to model an entire molecular cloud
- Cloud contains 10^5 M_{sun}
- Resources allow for $5x10^6$ particles
- Can reasonably resolve down to $0.02M_{sun}$

Northern Orion Molecular Cloud; researchgate.net

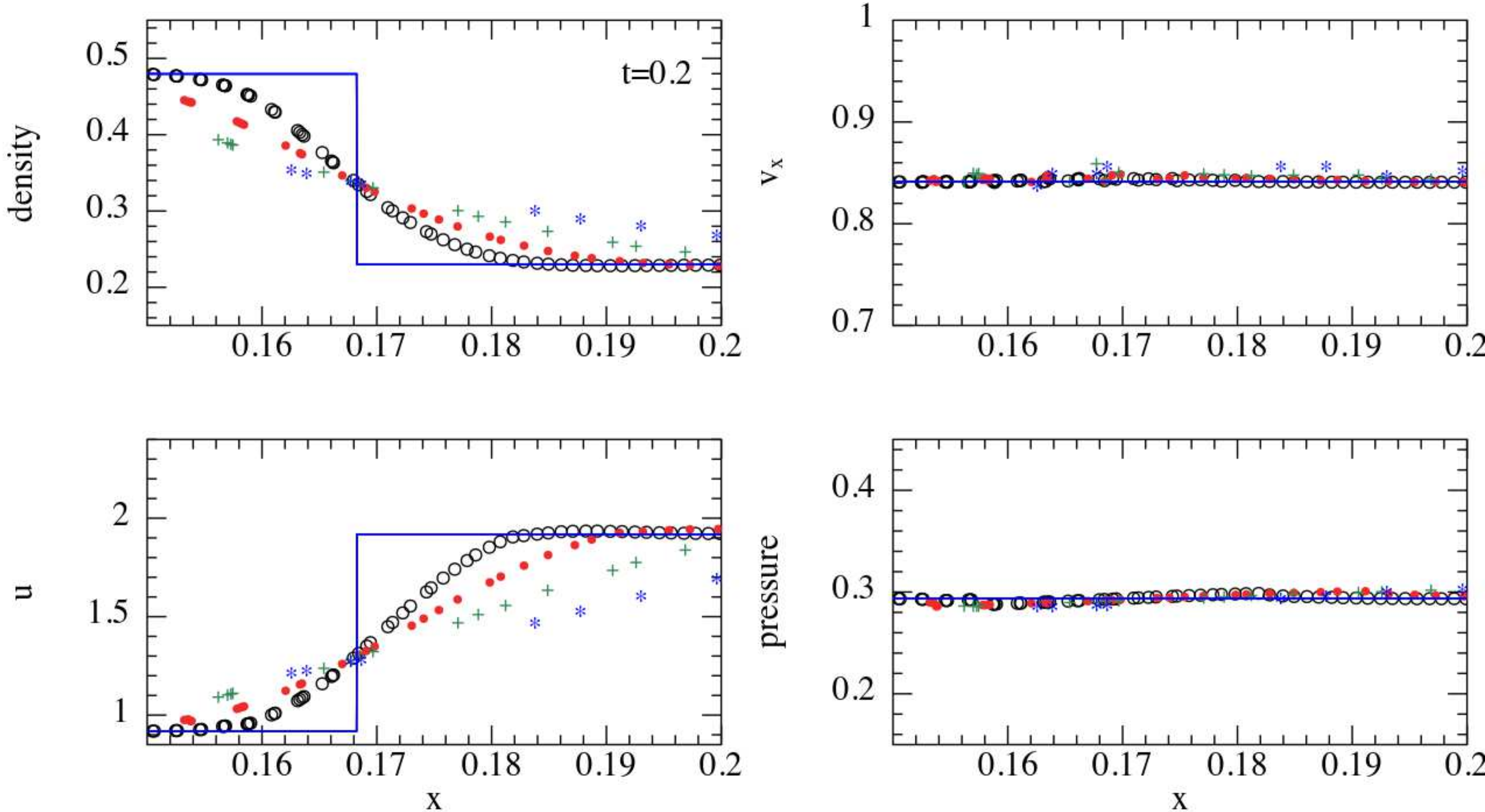
Resolution: Sod Shock

➢ Investigating the shock wave for four resolutions: $n_{x,\mathrm{left}}$ = 32,64,128 & 256



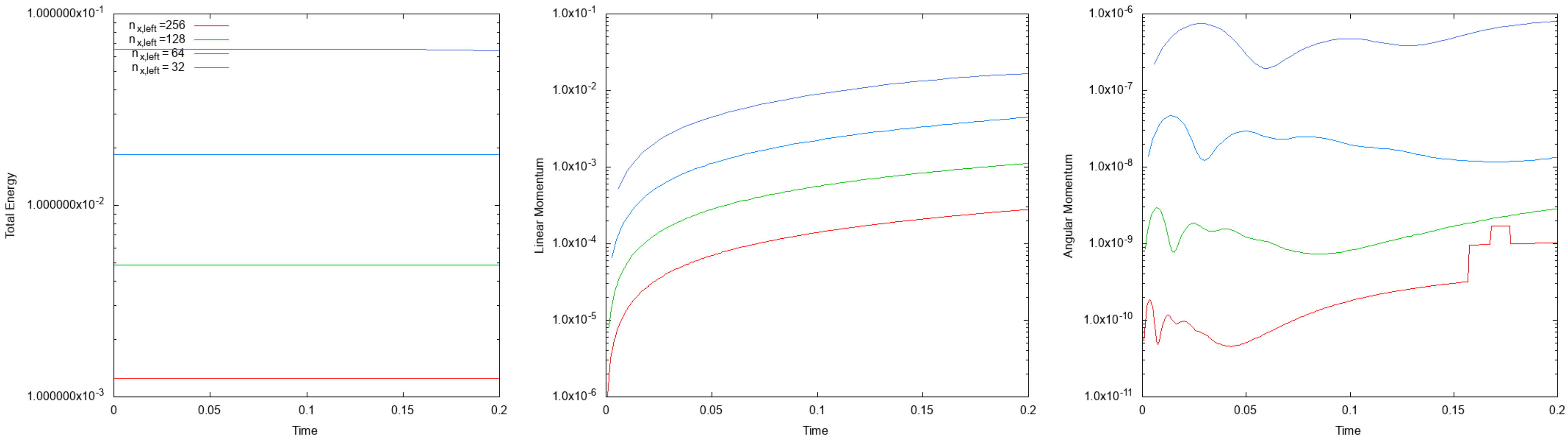
Resolution: Sod Shock

- Investigating the contact discontinuity for four resolutions: $n_{x,\mathrm{left}}$ = 32,64,128 & 256



Resolution: Sod Shock

- Testing four resolutions: $n_{x,\mathrm{left}}$ = 32,64,128 & 256
- Conserved quantities get better with increasing resolution

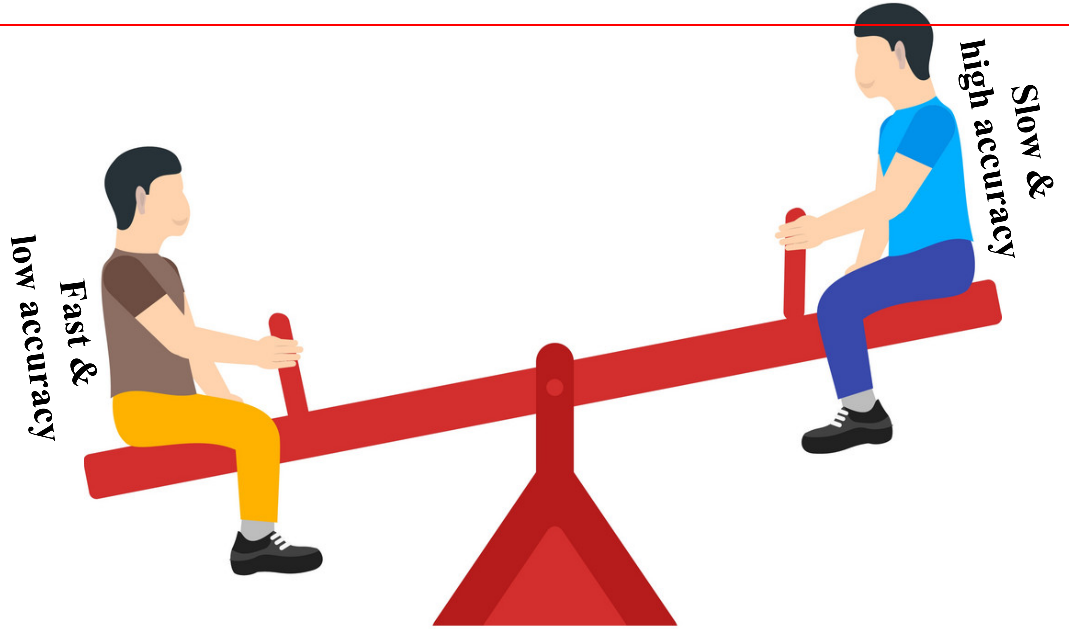


Resolution: Warning!

- Recall: decreasing *dx* by a factor 2
  - doubles the number of calculations per step
  - doubles the number of steps

$$dt = C\frac{dx_i}{v_i} \quad \text{where} \quad C \leq 1$$

In numerical studies, the user must always balance resolution with runtime!



Resolution: Sod Shock

- Testing four resolutions: $n_{x,\text{left}} = 32, 64, 128$ & 256
- Runtime is considerably longer for higher resolutions