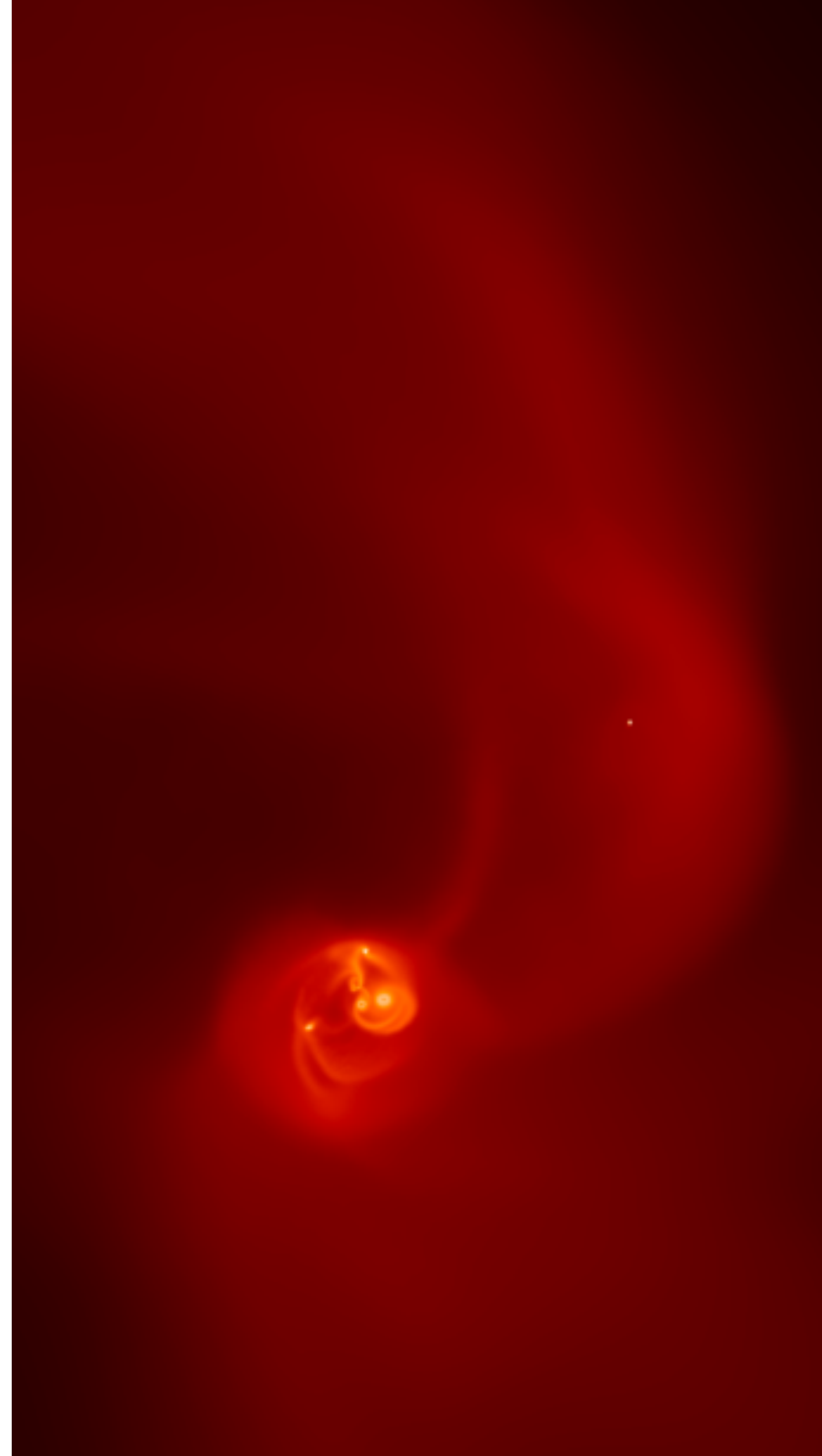
# Godunov methods in GANDALF

Stefan Heigl
David Hubber
Judith Ngoumou

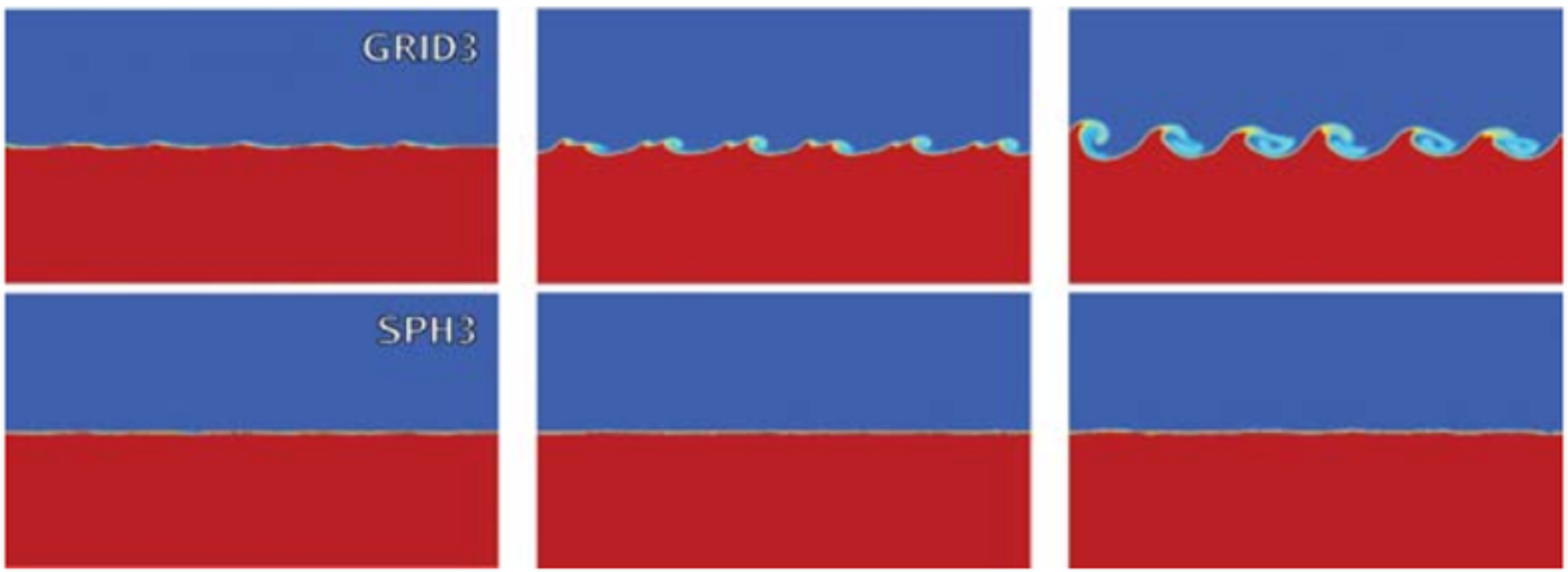
USM, LMU, München

28th October 2015



# Why not just stick with SPH?

- SPH is perfectly adequate in many scenarios but can fail, or at least prove sub-optimal in certain astrophysical contexts, e.g.
  - Gases of different temperatures/specific entropies are in contact or mixing together (e.g. a hot gas bubble pushing on a cold background medium)




Agertz et al. (2007)

  - High artificial viscosity causes unphysically high dissipation (e.g. evolution of a protoplanetary disc)

# Why not just stick with SPH?

- Other methods, in particular Godunov methods, have proven themselves to handle such hydrodynamical cases better than SPH
- However, Godunov methods have traditionally been used on static, Eulerian grid codes which introduce their own set of problems
  - Advection errors (i.e. numerical diffusion when the gas is travelling rapidly between grid cells)
  - Angular momentum conservation
- In the last few years, hybrid algorithms that attempt to retain as many of the advantages of both approaches have been developed
  - Moving-mesh Finite-Volume Hydrodynamics (cf. AREPO, Springel 2010)
  - Meshless Finite-Volume Hydrodynamics (cf. GIZMO, Lanson & Vila 2008)

# ‘Old-fashioned’ Finite Difference

- The **Finite-Difference** method is a discretization method, where a smooth function is discretized at regular points
- Differential equations are solved by **approximating derivatives by finite differences**
- e.g. using the Euler method:

$$\frac{\partial Q(x)}{\partial x} \approx \frac{Q(x+h) - Q(x)}{h}$$

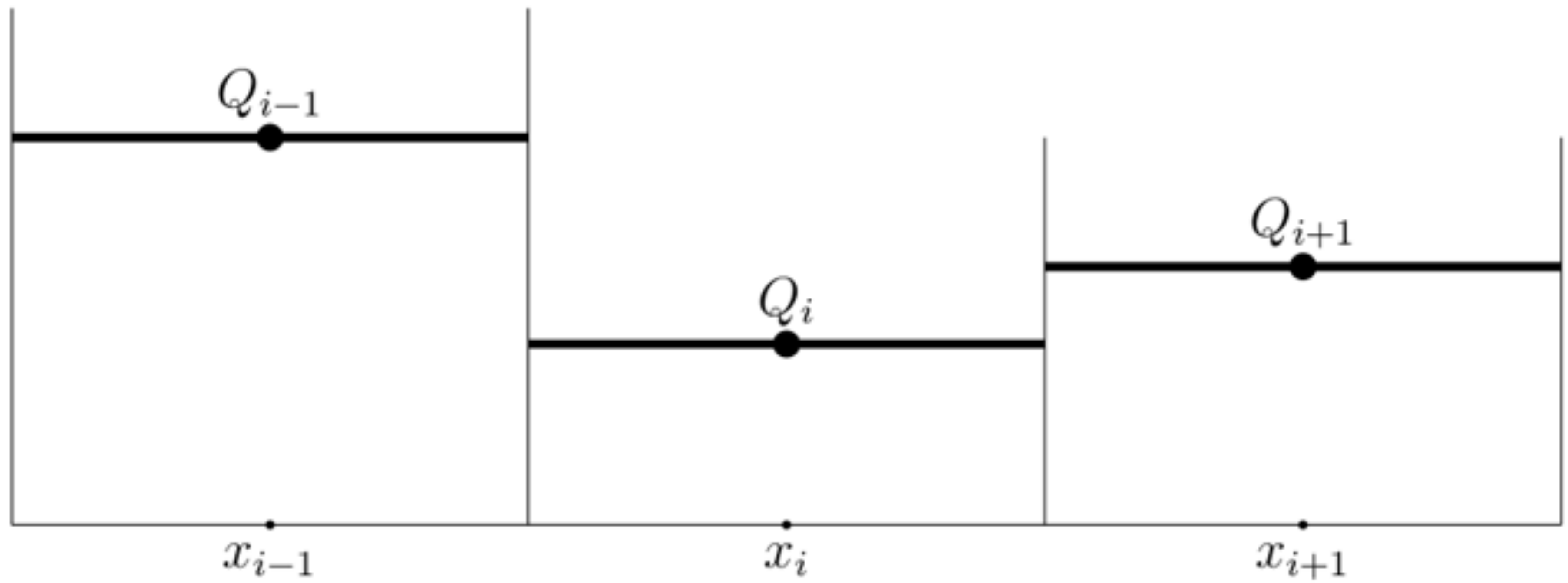
- This can be applied for gradients in space or time and for higher order derivatives
- e.g. the heat equation, using a second-order central difference for the space derivative

$$\frac{Q_i^{n+1} - Q_i^n}{h} = \alpha \frac{Q_{i+1}^n - 2Q_i^n + Q_{i-1}^n}{k^2}$$

$$\Rightarrow Q_i^{n+1} = Q_i^n + \alpha \frac{h}{k^2}(Q_{i+1}^n - 2Q_i^n + Q_{i-1}^n)$$

# Godunov methods : Finite Volume (FV) Hydrodynamics

- **Sergei Godunov (1959)** suggested a new approach to solving the Hydrodynamical equations which moved away from the traditional Finite-Difference scheme and towards a **Finite-Volume** approach.
- Instead of calculating effective forces from approximate gradients, the finite-volume approach calculates the **flux of the hydrodynamical quantities at the cell boundaries**.
- As the flux entering a given volume equals the flux leaving the adjacent volume, this method is **conservative.**
- The cell boundaries define a **Riemann problem**

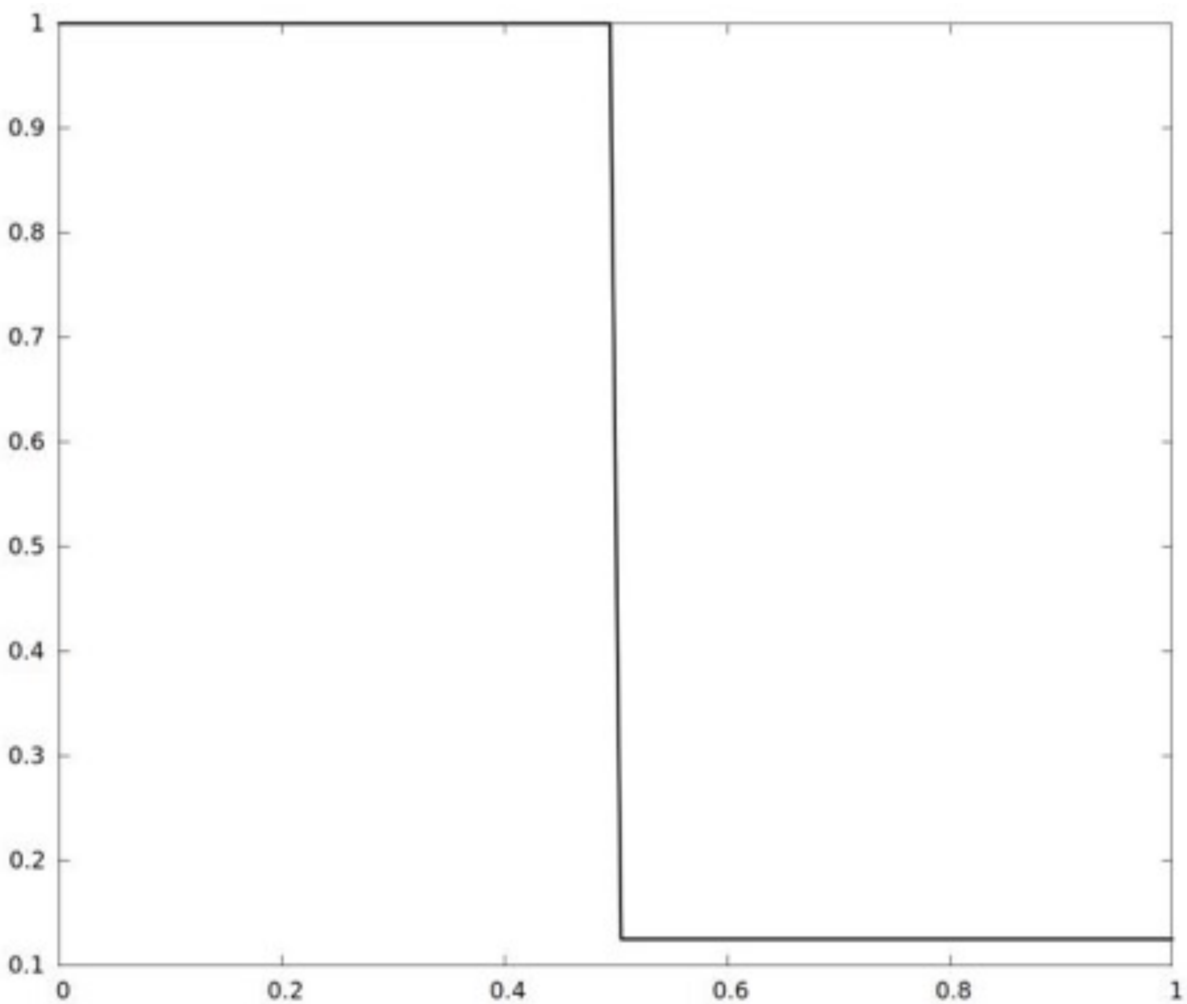


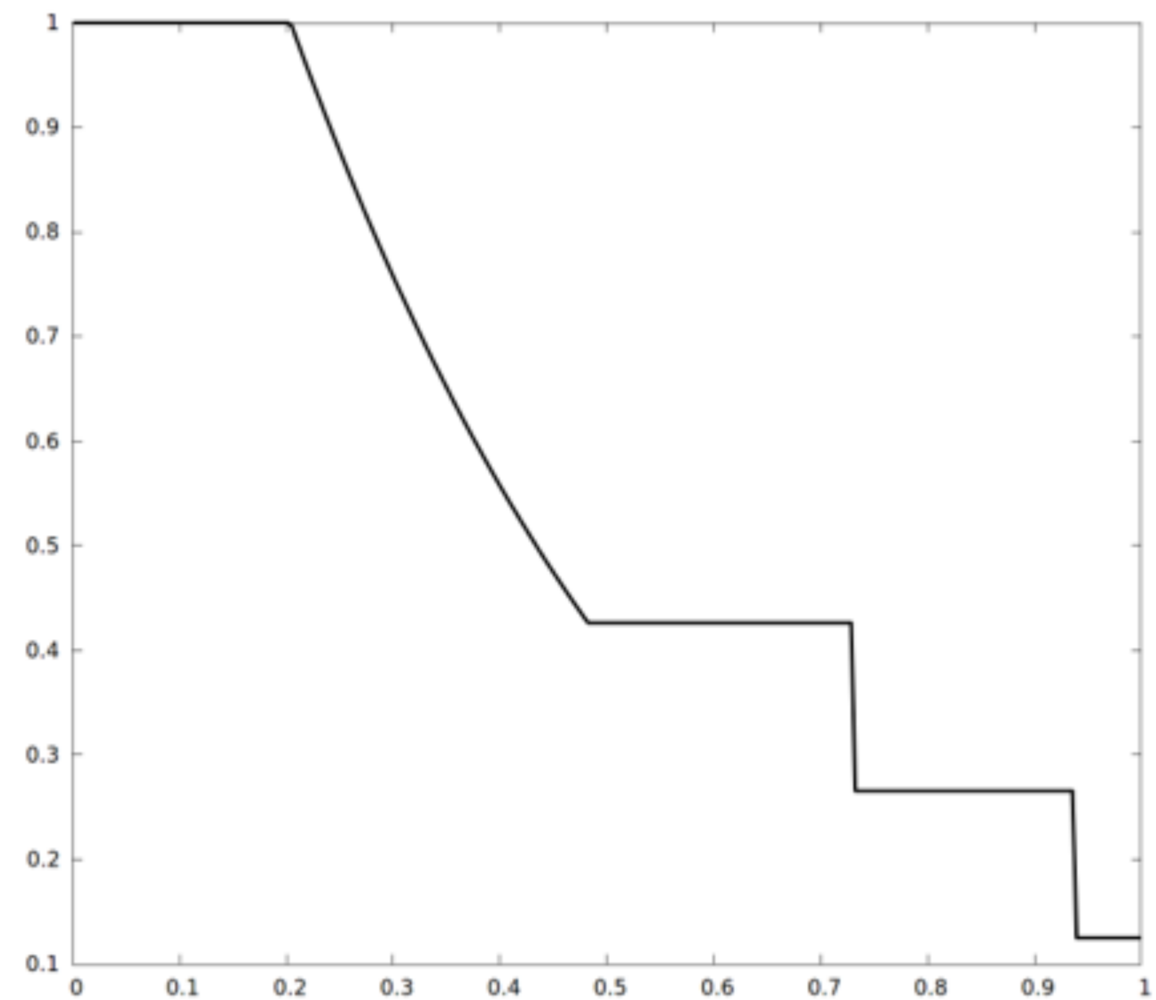
# What is a Riemann solver?

- A **Riemann solver** is an algorithm for computing the solution of a simple Riemann problem, for example
  - The state of the intermediate shock structure
  - The flux of mass, momentum and energy across the shock
- Why is using a Riemann solver important?
  - We could in theory compute the fluxes from the left and right states, but then we might fail at capturing shocks properly
  - A Riemann solver effectively allows us to capture shocks with the **minimum required dissipation** (BIG step up from schemes that use artificial viscosity, such as SPH)
- The Godunov method uses an exact/approximate Riemann solver locally

# Exact Riemann solvers

- The exact Riemann solver gives the **numerical exact solution** to a Riemann problem
- There is **no closed-form solution** to the Riemann problem, even not for ideal gases, not for the isothermal, nor the isentropic equations
- Thus, one has to use an initial pressure guess and **iterate** to find the solution up to any desired accuracy
- The flux is then calculated according to the wave pattern at the boundary



# Approximate Riemann solvers

- An approximate Riemann solver does what it says, it calculates an **approximate solution** to the Riemann problem
- It is much faster than the exact Riemann solver as it uses **no iteration**
- In most cases an approximate solution is perfectly adequate and can speed up the code considerably
- In the rare cases that it fails (e.g. near a strong shock), we can switch to the exact Riemann solver
- The most used approximate solvers are the Roe solver, the HLLE and the HLLC

# Integrating the Euler equations

- Our scheme now integrates the five Euler equations:

$$\frac{\partial}{\partial t}\rho + \nabla(\rho\mathbf{u}) = 0$$
$$\frac{\partial}{\partial t}(\rho\mathbf{u}) + \nabla(\rho\mathbf{u}\otimes\mathbf{u}) = -\nabla p$$
$$\frac{\partial}{\partial t}(\rho e_{tot}) + \nabla(\rho e_{tot}\mathbf{u}) = -\nabla(\mathbf{u}p)$$

- The new state is then: $\mathbf{Q}_i^{n+1} = \mathbf{Q}_i^n + \frac{\Delta t}{\Delta x}\left[\mathbf{F}_{i-1/2} - \mathbf{F}_{i+1/2}\right]$

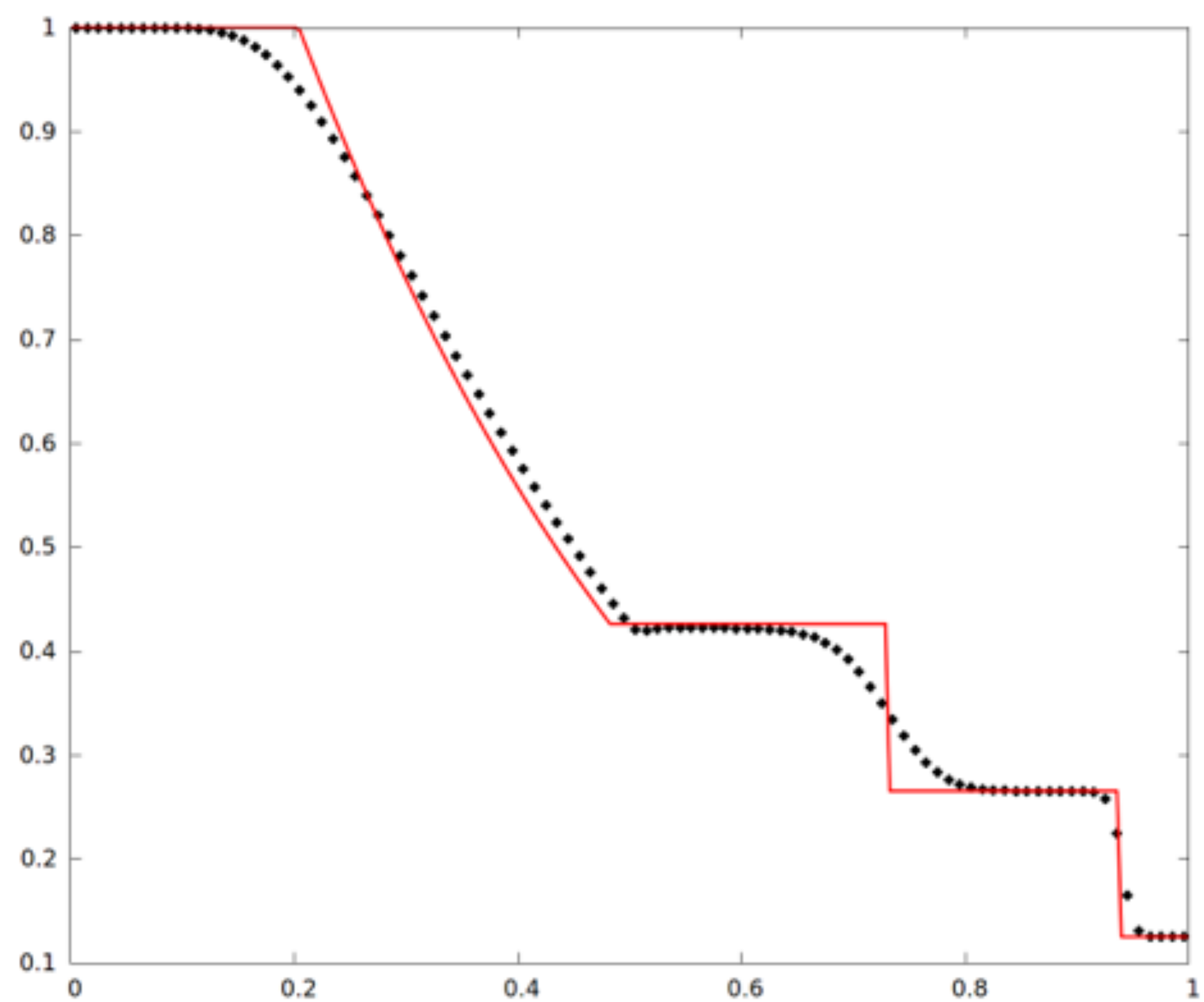
- The timestep size has to be confined in order to prevent wave interaction. This is usually done by the **CFL-condition**:

$$\Delta t = \frac{C_{cfl}\Delta x}{S_{max}} \qquad 0 < C_{cfl} \leq 1$$

- where the maximum wave speed is the maximum of the sum of the velocity and the sound speed in the domain

# So, that'll work right??? ….

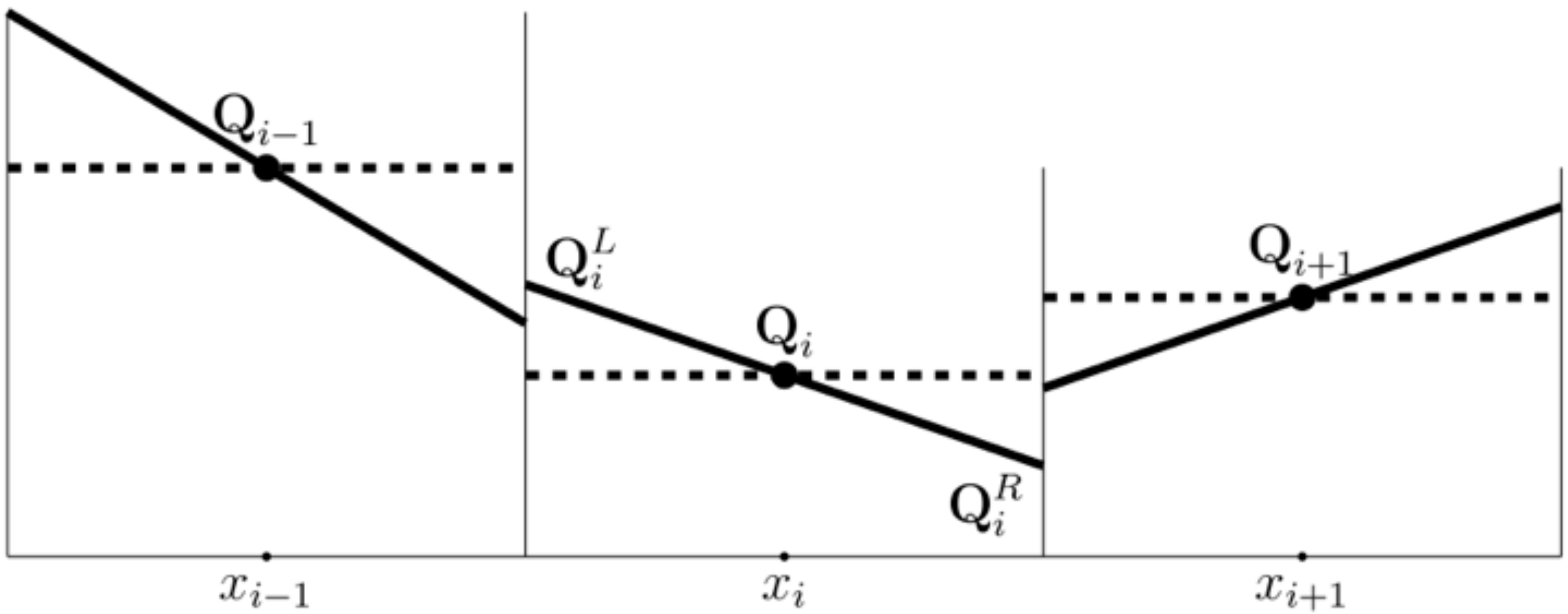
- Well, sort of …




- It's correct, but has been horribly 'smoothed-out' over the discontinuities
- This is caused by **using the average cell values as the boundary conditions** for the Riemann solver
- Effectively leads to a spatially **1st-order scheme**

# Moving to 2nd order
# MUSCL-Hancock scheme

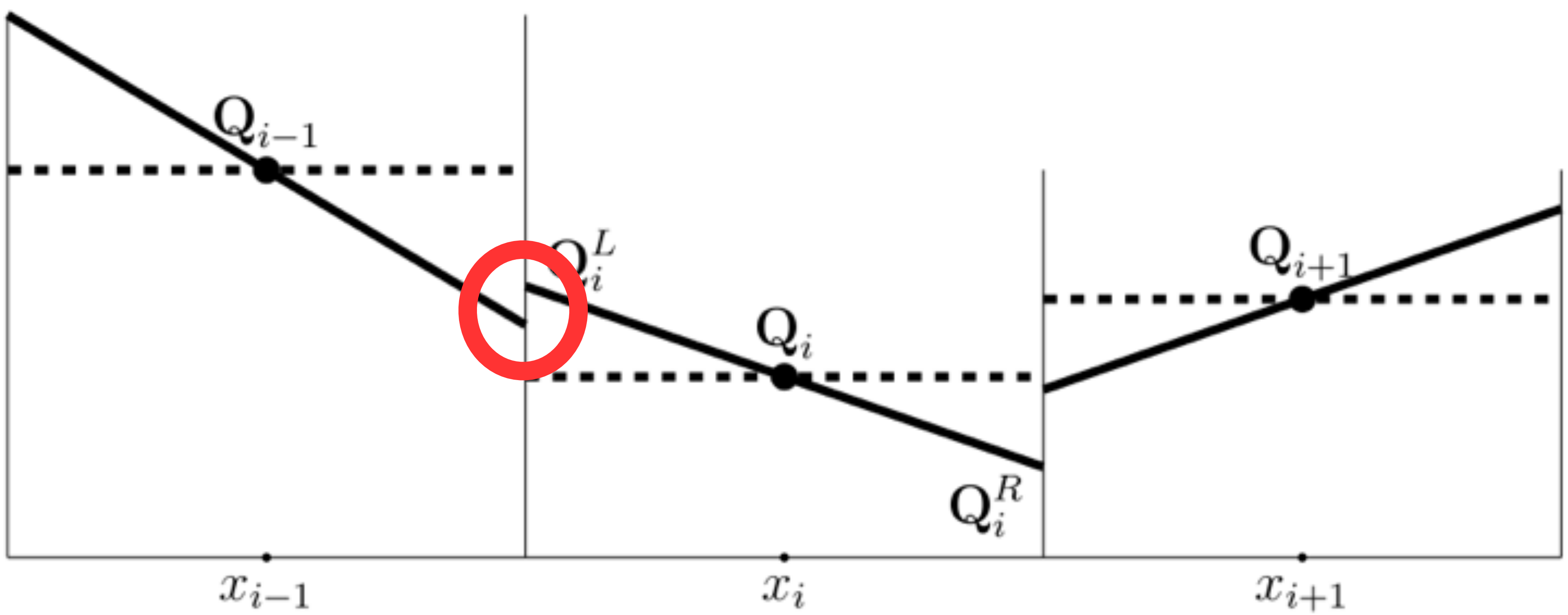
- In the 1970s, Van Leer developed the *Monotonic Upstream-Centered Scheme for Conservation Laws,* aka **MUSCL**.
- The main ingredient in MUSCL is that the gradient is calculated and used to **extrapolate the cell properties to the cell boundaries**
- The extrapolated values are evolved half a timestep to make the scheme stable




- The time evolved extrapolated values are then used to solve the Riemann problem

# Ooops, something's gone wrong!

- If the code does not break (which it usually does), Godunov's theorem states, that using a second order scheme will produce **oscillations at large gradients**
- This is due to the fact that the slopes can develop **overshoots** if the cell gradient is steeper than the overall gradient

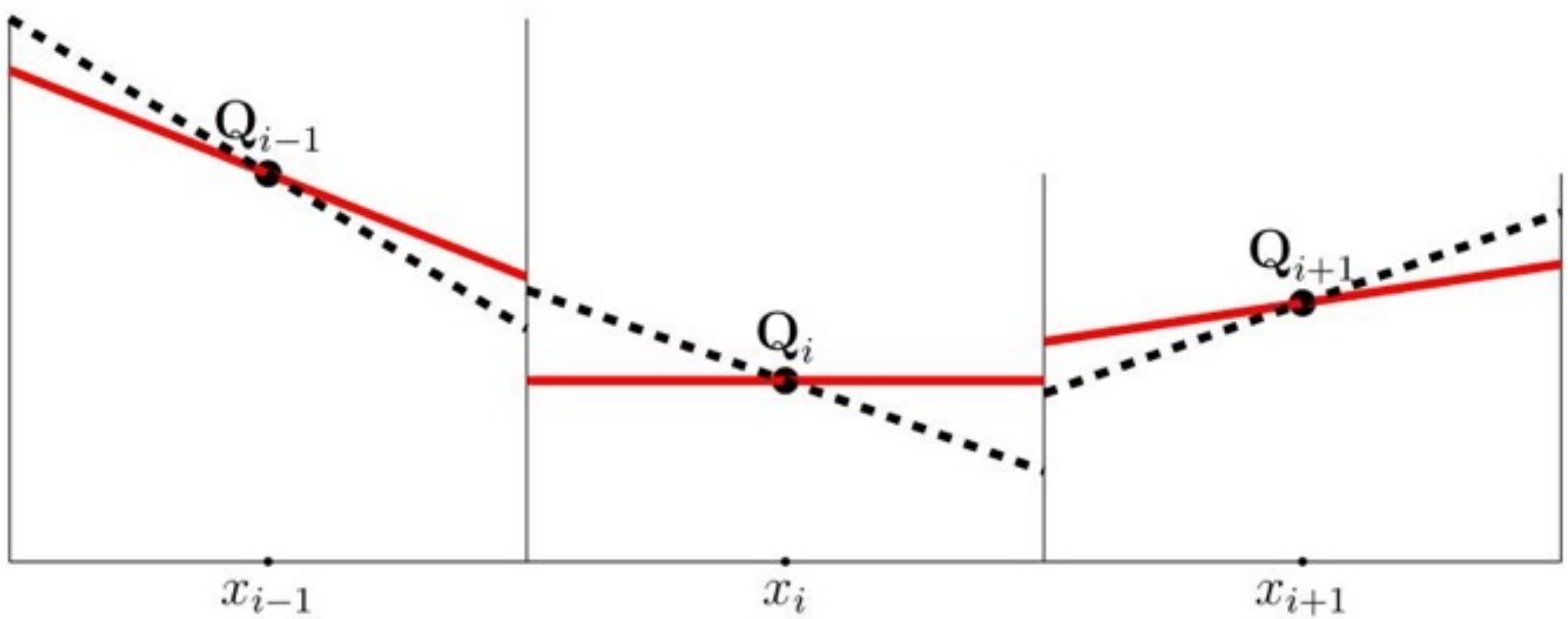



- One can prevent these with the application of **slope limiters**

# Slope limiters
# A world of pain

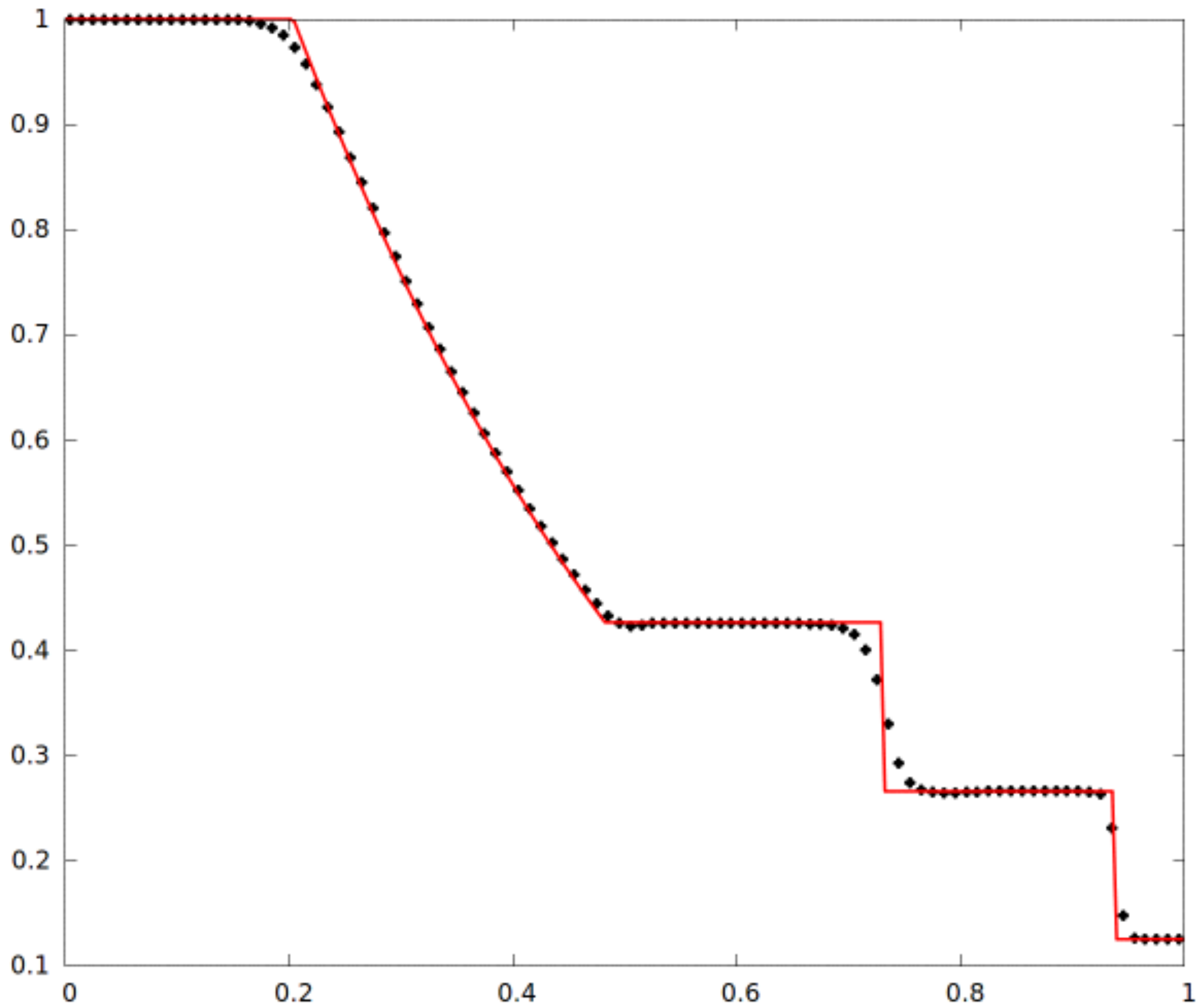
- A **slope limiter** is an algorithm for preventing the 'wiggling' in the shock solution
- Effectively the slope limiter forces the hydrodynamics to be solved in 1st-order near any discontinuity, such as a shock
- Everywhere else where the flow is smooth, the hydrodynamics is solved in 2nd-order
- Let's consider this super simple slope limiter, called a **minmod**

$$\Delta \mathbf{Q}_L = \mathbf{Q}_i - \mathbf{Q}_{i-1}$$
$$\Delta \mathbf{Q}_R = \mathbf{Q}_{i+1} - \mathbf{Q}_i$$




$$\Delta \mathbf{Q}_i = 0 \qquad for\ \Delta \mathbf{Q}_L \Delta \mathbf{Q}_R < 0$$
$$\Delta \mathbf{Q}_i = \min(\Delta \mathbf{Q}_L, \Delta \mathbf{Q}_R) \qquad else$$

# Finally, success

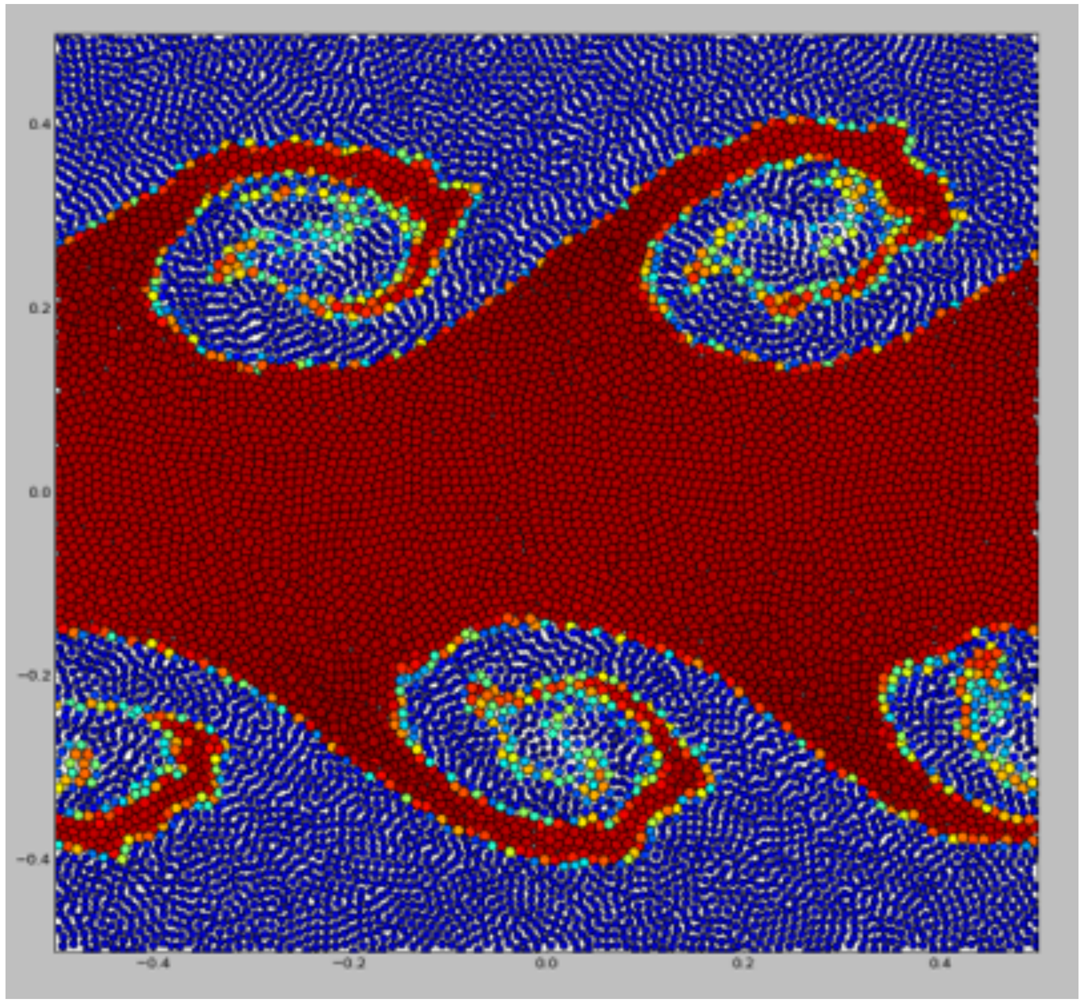


# Moving mesh FV schemes

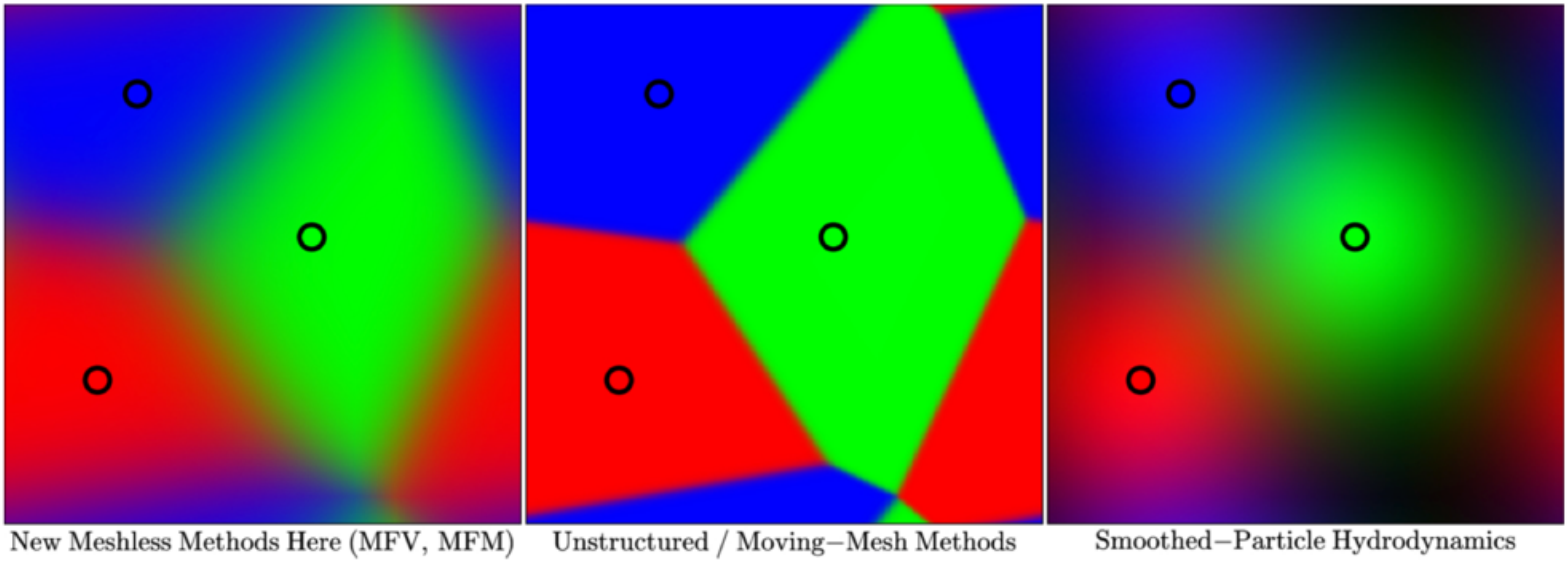
- One can use the finite volume method on a moving grid, defined by particles similar to SPH
- In order to define boundaries, one has to perform a **Delauney triangulation**
- Most famous variant is the **AREPO** (Springel 2010) code

**COMING SOON!!!!**

Moving mesh in **GANDALF:**



# Meshless FV schemes




Hopkins (2015)

# Meshless FV schemes

Lanson & Vila (2008a,b):

- proposed an alternative hybrid algorithm, between traditional FV and SPH methods
- consistent and conservative mesh-free finite-volume method

Gaburov & Nitadori (2011):

- Astrophysical Weighted Particle Magnetohydrodynamics

Hopkins (2014, 2015):

- GIZMO Mesh-free hydrodynamic simulation Methods

# Meshless FV schemes

## Advantages (Hopkins, 2015)

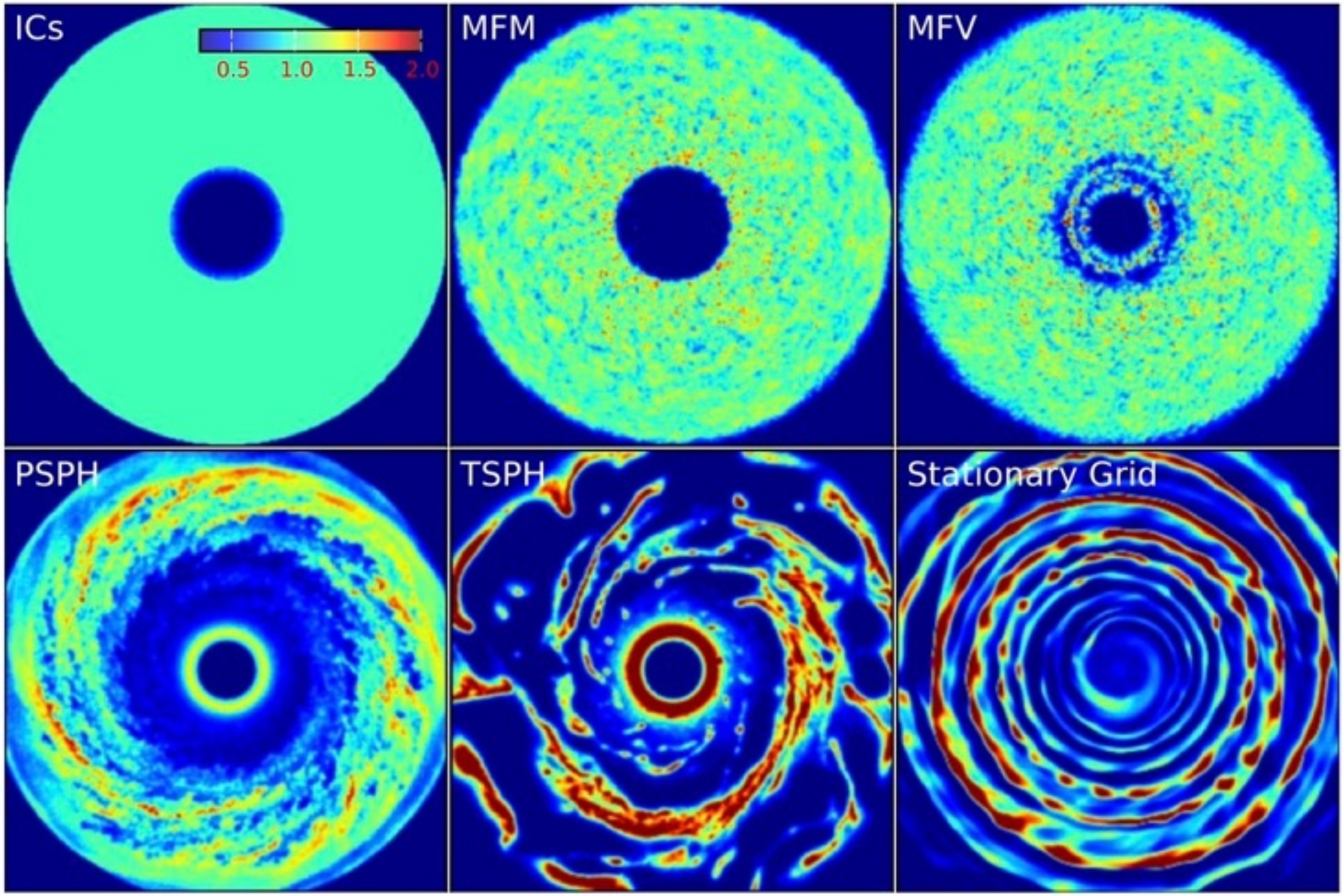
**compared to SPH**

- proper convergence
- good capturing of fluid-mixing instabilities
- no artificial viscosity
- more accurate sub-sonic flow evolution (reduced noise)
- better shock-capturing

**compared to fixed grid methods**

- automatic adaptivity
- reduced advection errors and numerical diffusion
- velocity-independence of numerical errors
- accurate coupling to N-body gravity solvers
- good angular momentum conservation
- elimination of grid alignment effects

# Meshless FV :
# Kelvin-Helmholtz instability




Hopkins (2015)

# Meshless FV schemes

Hopkins (2015)

**Table 1.** Summary of Some Popular Numerical Hydrodynamics Methods

| Method Name | Consistency /Order | Conservative? (Mass/Energy /Momentum) | Conserves Angular Momentum | Numerical Dissipation | Long-Time Integration Stability? | Number of Neighbors | Known Difficulties |
|---|---|---|---|---|---|---|---|
| **Smoothed-Particle Hydro. (SPH)** | | | | | | | |
| "Traditional" SPH (GADGET, GASOLINE, TSPH) | 0 | ✓ | up to AV | artificial viscosity (AV) | ✓ | $\sim 32$ | fluid mixing, noise, E0 errors |
| "Modern" SPH (P-SPH, SPHS, PHANTOM, SPHGal) | 0 | ✓ | up to AV | AV+conduction +switches | ✓ | $\sim 128-442$ | excess diffusion, E0 errors |
| "Corrected" SPH (rpSPH, Integral-SPH, Morris96 SPH, Moving-Least-Squares SPH) | 0-1 | × | × | artificial viscosity | × | $\sim 32$ | errors grow non-linearly, "self-acceleration" |
| "Godunov" SPH (GSPH, GSPH-I02, Cha03 SPH) | 0 | ✓ | up to gradient errors | Riemann solver | ✓ | $\sim 300$ | instability, expense, E0 errors remain |
| **Finite-Difference Methods** | | | | | | | |
| Gridded/Lattice Finite Difference (ZEUS [some versions], Pencil code) Lagrangian Finite Difference (PHURBAS, FPM) | 2-3 | × | × | artificial viscosity | × | $\sim 8-128$<br>$\sim 60$ | instability, lack of conservation, advection errors |
| **Finite-Volume Godunov Methods** | | | | | | | |
| Static Grids (ATHENA, PLUTO) | 2-3 | ✓ | × | Riemann solver | ✓ | $\sim 8$ (geometric) $\sim 8-125$ (stencil) | over-mixing, ang. mom., velocity-dependent errors (VDE) |
| Adaptive-Mesh Refinement (AMR) (ENZO, RAMSES, FLASH) | 2-3 (1) | ✓ | × | Riemann solver | ✓ | $\sim 8-48$<br>$\sim 24-216$ | over-mixing, ang. mom., VDE, refinement criteria |
| Moving-Mesh Methods (AREPO, TESS, FVMHD3D) | 2 | ✓ | × | Riemann solver | ✓ | $\sim 13-30$ | mesh deformation, ang. mom. (?), "remeshing" |
| **New Methods In This Paper** | | | | | | | |
| Meshless Finite-Mass & Meshless Finite-Volume (MFM, MFV) | 2 | ✓ | up to gradient errors | Riemann solver | ✓ | $\sim 32$ | remeshing noise ? (TBD) |

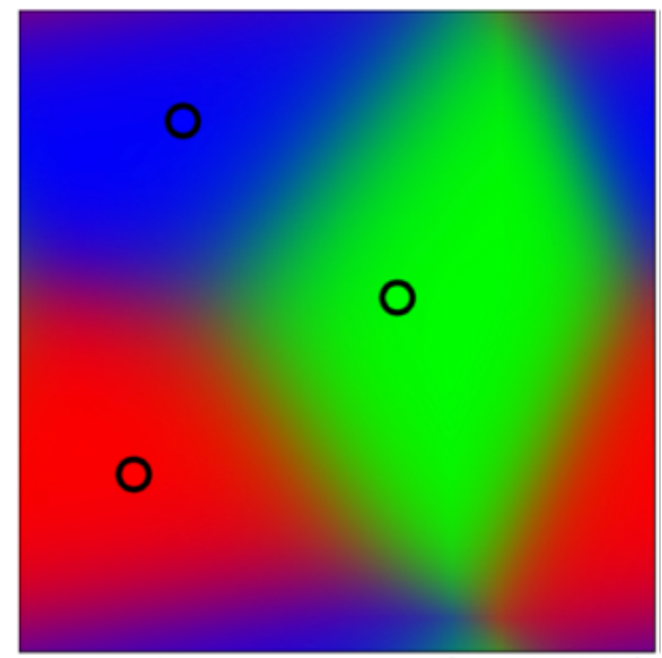
# Meshless FV in GANDALF

Volume partition according to the nearest particles (SPH Kernel)

$$V_i = \int \psi_i(\mathbf{x})\mathbf{d}^\nu\mathbf{x}$$

$$\psi_i = \frac{1}{\omega(\mathbf{x})} W(|\mathbf{x} - \mathbf{x_j}|, \mathbf{h}(\mathbf{x}))$$

$$\omega(\mathbf{x}) = \sum_{\mathbf{j}} \mathbf{W}(|\mathbf{x} - \mathbf{x_j}|, \mathbf{h}(\mathbf{x}))$$



New Meshless Methods Here (MFV, MFM)

Hopkins (2015)

# Meshless FV in GANDALF

## The meshless equations of motion

Discretization of an integral solution to a scalar conservation law on a number of particles at positions $(\mathbf{x}_i)$

$$\frac{d}{dt}(V_i U_i) + \sum_j \left[ V_i F_i \psi_j(\mathbf{x}_i) - V_j F_j \psi_i(\mathbf{x}_j) \right] = V_i S_i$$

Conservation of mass, momentum and energy

$$\frac{d}{dt}(V_i U_i) + \sum_j \mathbf{F}_{ij} \cdot \mathbf{A}_{ij} = V_i S_i$$

Godunov-like finite-volume equations

# Meshless FV in GANDALF

## Solving the mesh-less equations of motion

- MUSCL-Hancock scheme
- 2nd order accurate gradient estimation used to extrapolate the cell properties to the “cell boundaries”
- Slope-limited, linear reconstruction of face-centered quantities from each particle
- extrapolated values are evolved over half a timestep
- time-averaged fluxes for the timestep are derived from the solution to the Riemann problem in the rest frame of the effective face between two particles

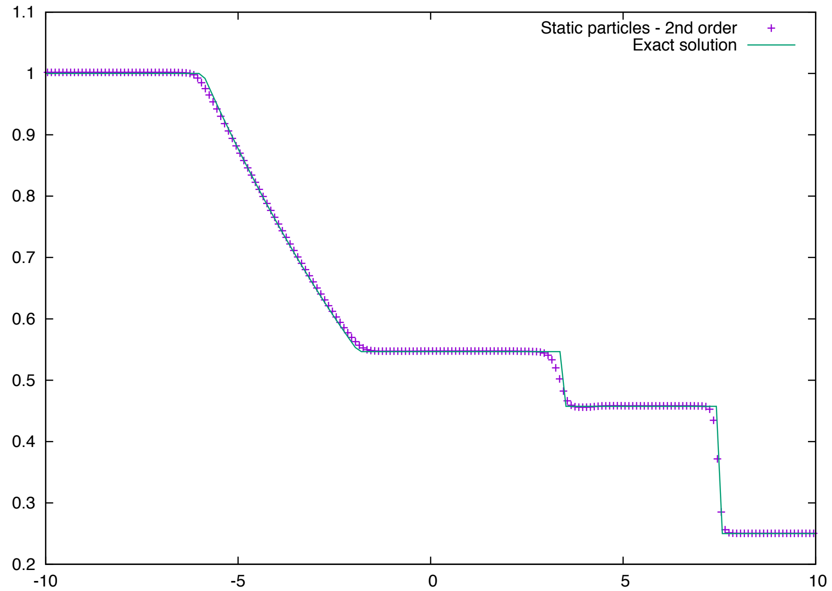
# Shocks (work in progress)

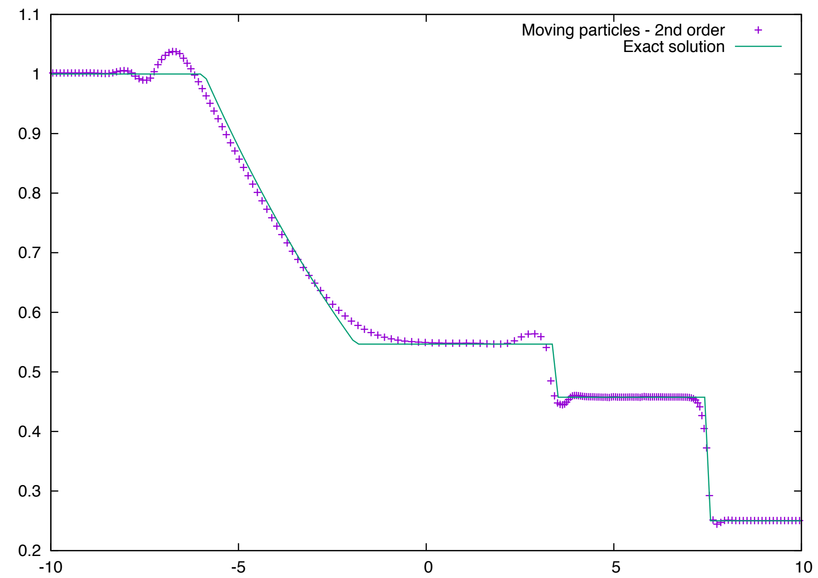
Static particles

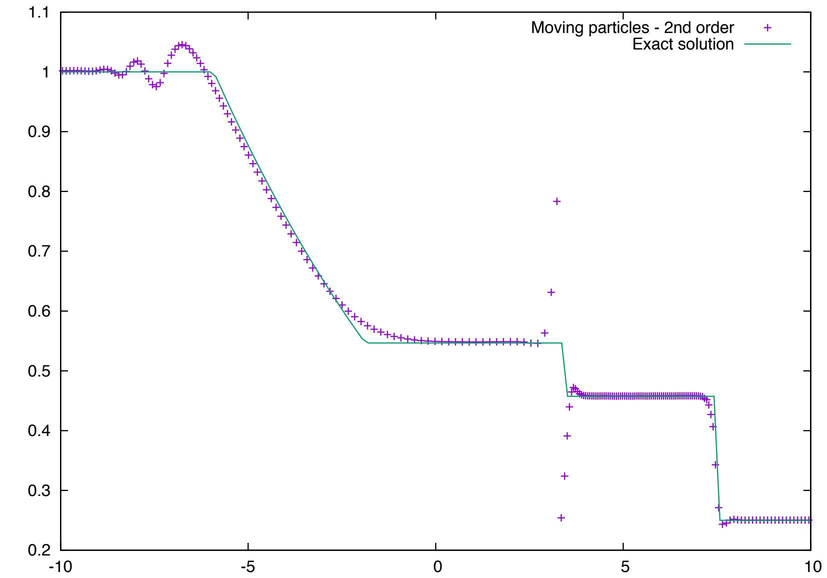
Moving particles (Meshless Finite-Volume)

Moving particles (Meshless Finite-Mass)

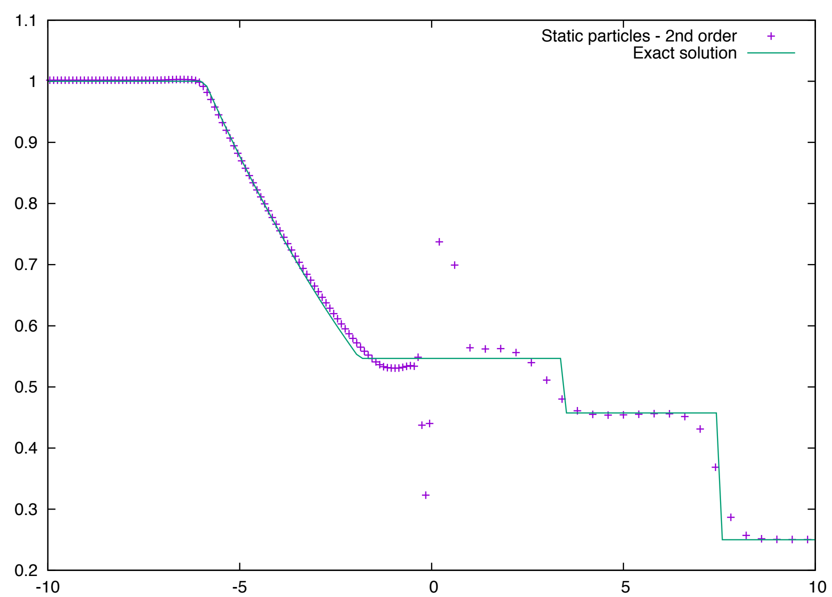
Equal separation

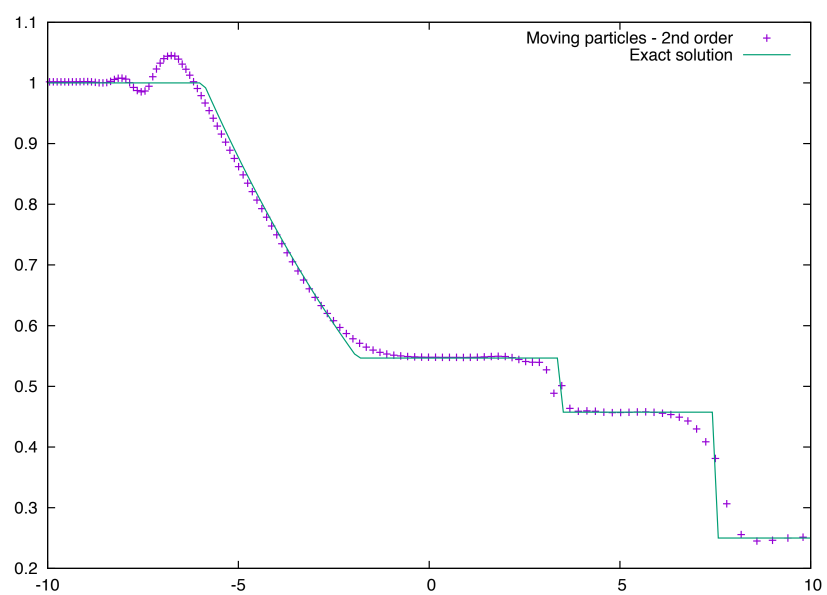


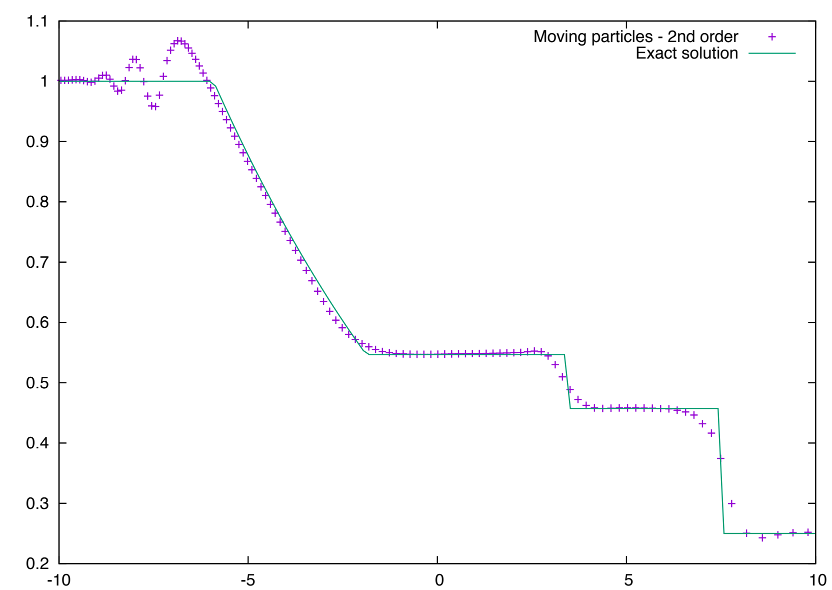



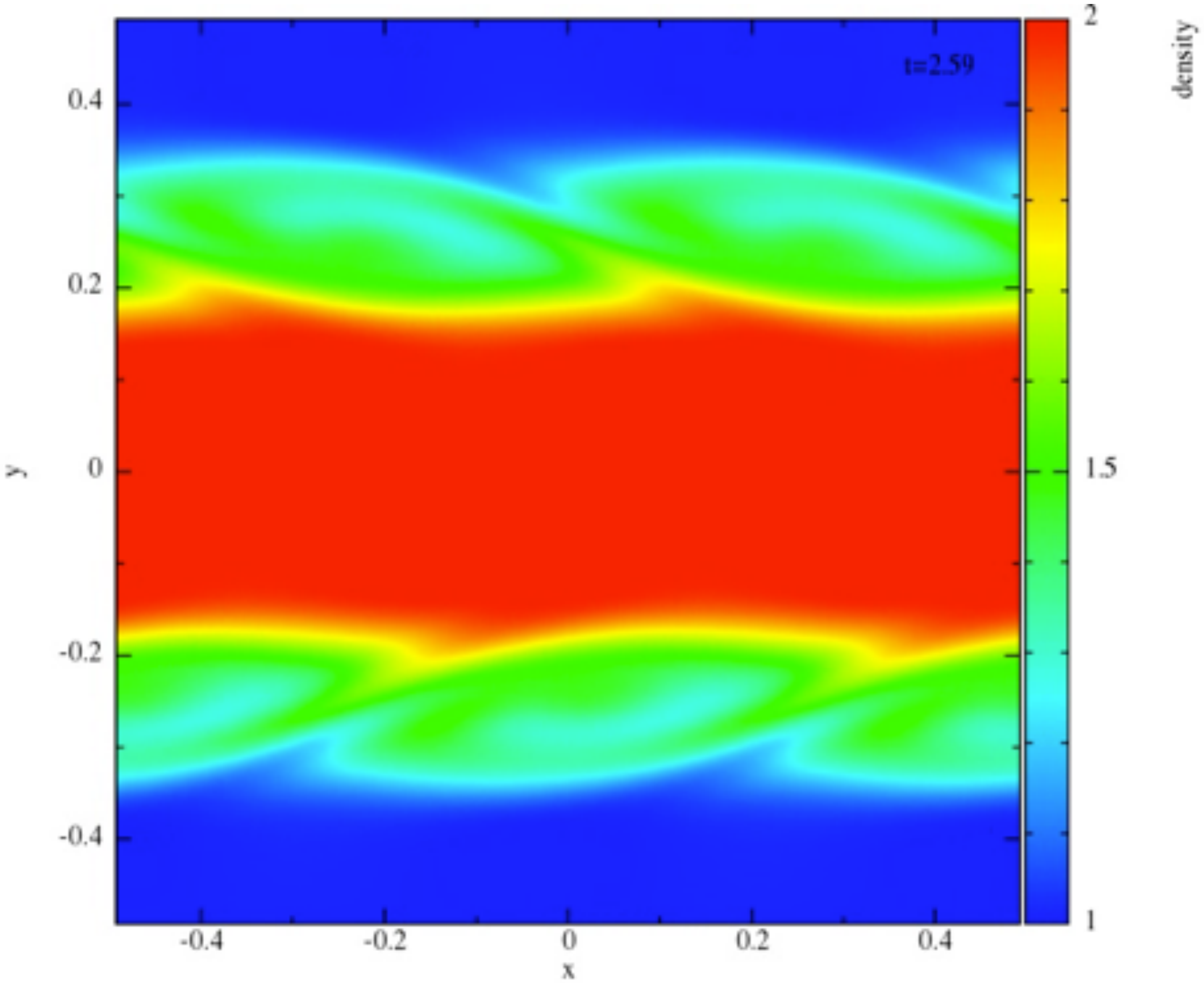

Equal mass

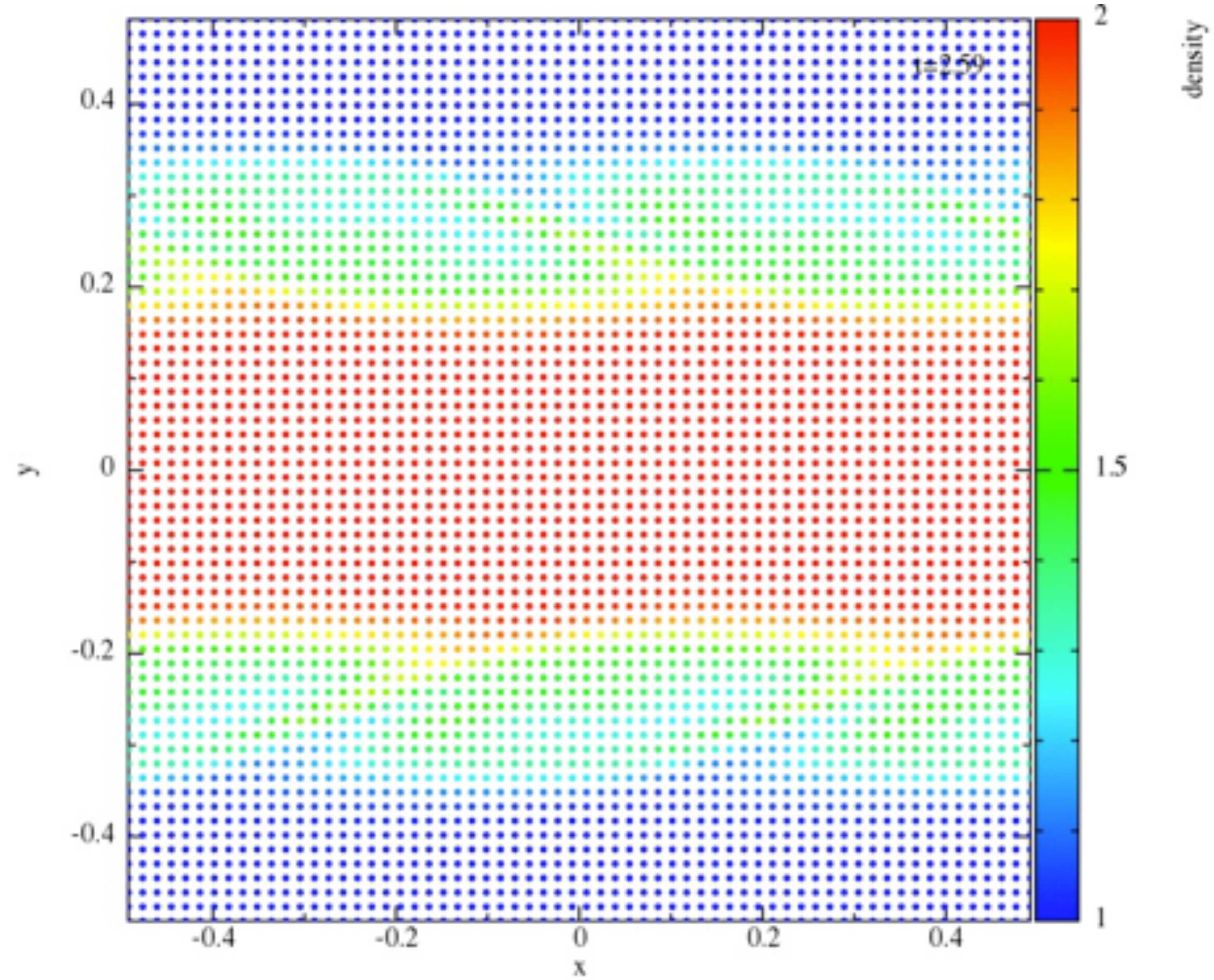


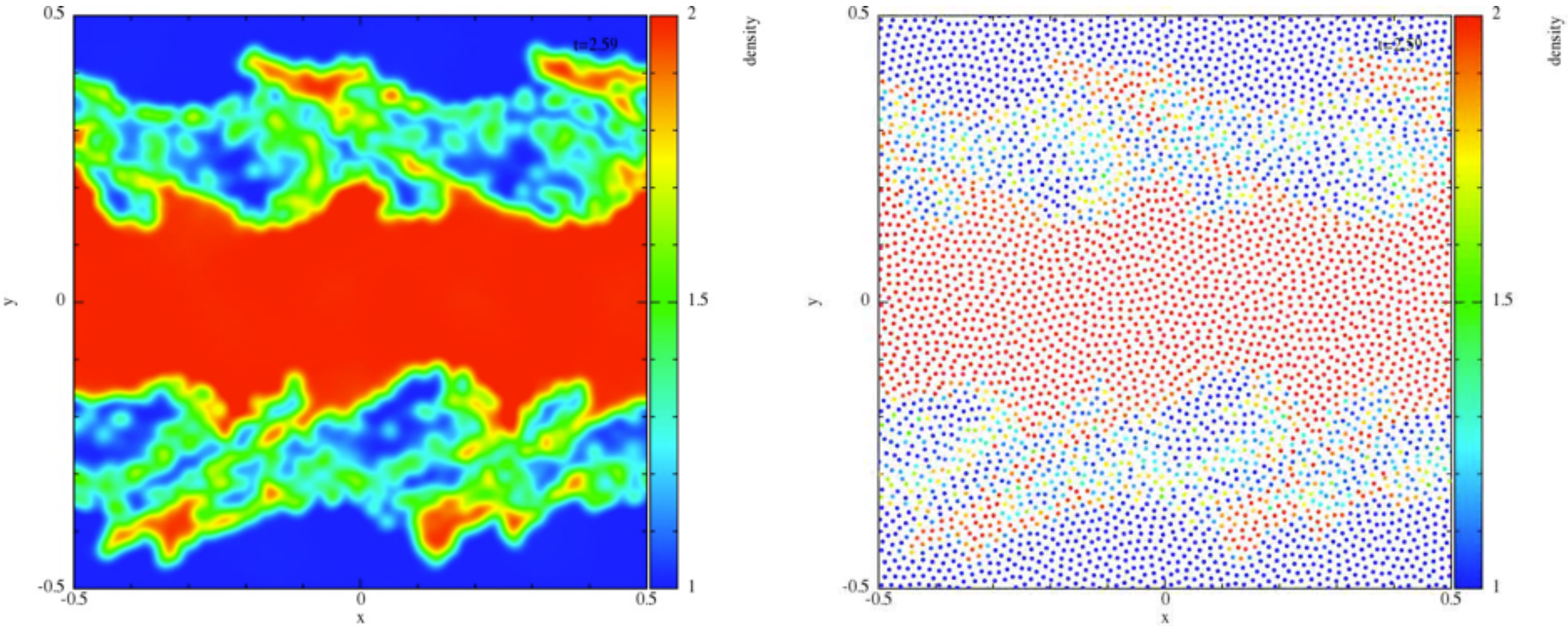



# Meshless FV : (work in progress)
# Kelvin-Helmholtz instability : Static particles



# Meshless FV : (work in progress)
# Kelvin-Helmholtz instability : Moving particles



# Meshless FV in GANDALF

## To be tested/implemented…

- Multiple timesteps
- Various equations of state (Isothermal, barotropic…)
- Sinks and N-body
- Thermodynamics: heating / cooling

## Caveats

- Slope limiters
- Advection errors (reduced but not eliminated)
- Not thoroughly tested with gravity and N-body…

# Godunov methods in GANDALF

## Running a meshless simulation in GANDALF

### Parameters

- `sim` : Simulation type

| | |
|---|---|
| sph | = SPH (+ N-body) algorithm (default : ‘grad-h’ SPH) |
| gradhsph | = ‘grad-h’ SPH simulation (+ N-body) |
| sm2012sph | = Saitoh & Makino (2012) SPH (+ N-body) |
| meshlessfv | = Meshless Finite-Volume algorithm (default : ‘mfvmuscl’) |
| mfvmuscl | = Meshless FV MUSCL integration simulation |
| mfvrk | = Meshless FV Runge-Kutta integration |
| nbody | = N-body only simulation |

**Meshless finite-volume parameters**

- `riemann_solver` : Riemann solver in FV scheme

| | |
|---|---|
| exact | = Exact Riemann solver (e.g. Toro 1999) |
| hllc | = HLLC approximate Riemann solver |

- `slope_limiter` : Slope limiter for TVD condition

| | |
|---|---|
| null | = No limiting |
| zeroslope | = Set all slopes to zero (effectively 1st order Godunov) |
| balsara2004 | = Balsara (2004) slope-limiter |
| springel2009 | = Original AREPO (Springel 2009) slope limiter |
| tess2011 | = TESS slope limtier |
| gizmo | = Original GIZMO paper (Hopkins 2015) slope limiter |
| minmod | = simplified implementation of minmod slope limiter |

- `zero_mass_flux` : Use Meshless-Finite Mass scheme to prevent mass-flux between particles? (1 or 0)
- `static_particles` : Use static particles (Eulerian approach)? (1 or 0)