Numerical Galaxy Formation and Cosmology

Lecture3: Numerical hydrodynamics on a mesh

Ewald Puchwein & Benjamin Moster

Why hydrodynamics?

- Everything we see is gas or made from gas

image credit: NASA

- Need to follow the hydrodynamics:
  - To form galaxies and stars
  - To study the interstellar, intergalactic and intracluster medium


bullet cluster
image credit: NASA

bullet cluster

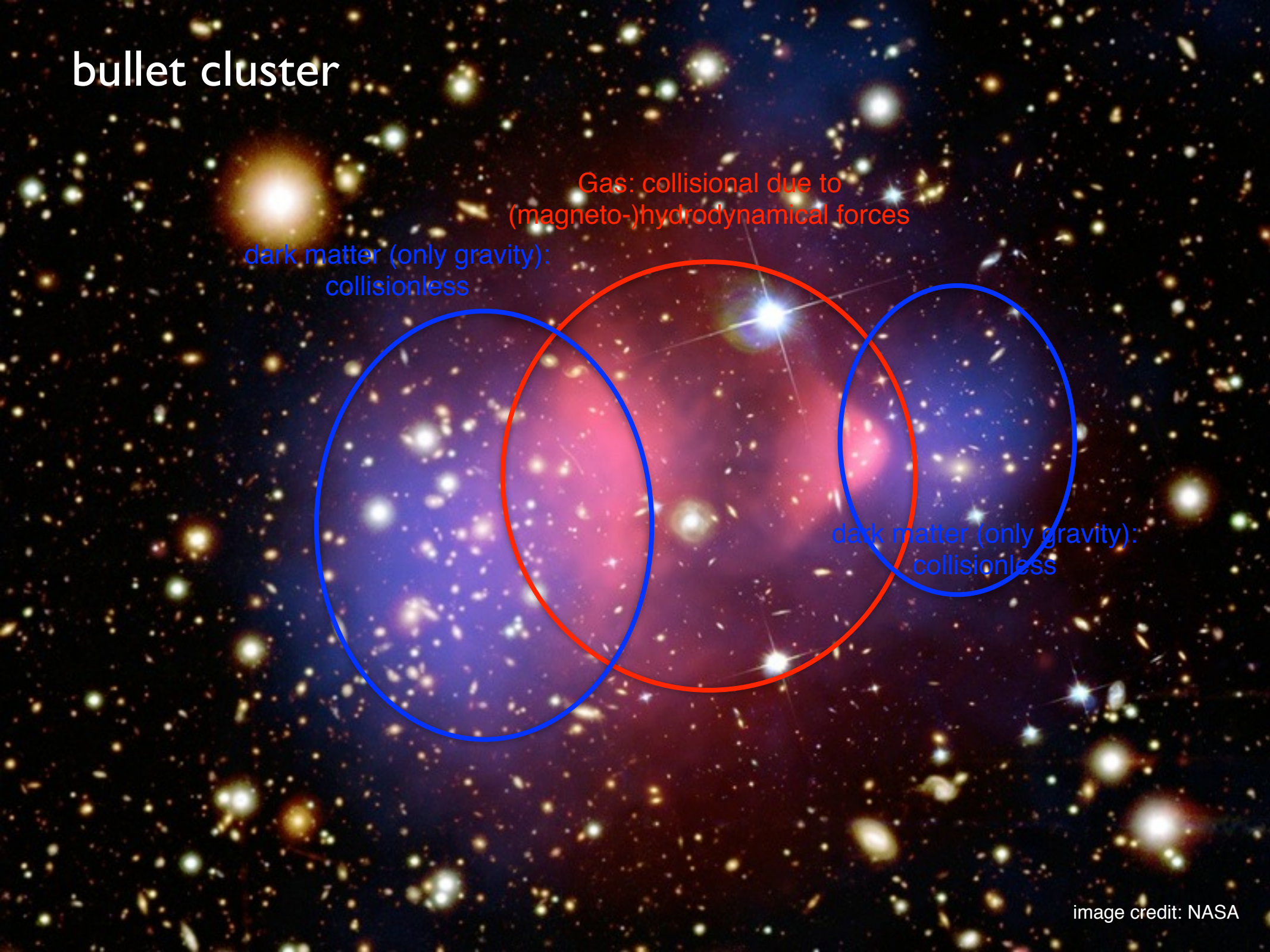
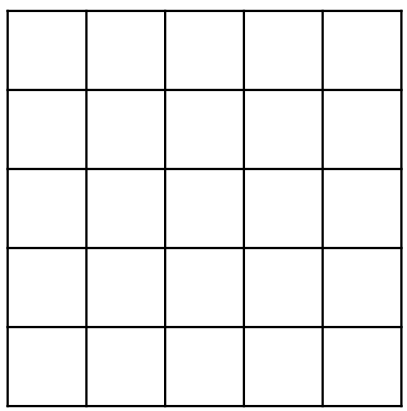


image credit: NASA

Eulerian vs Lagrangian methods

Eulerian methods

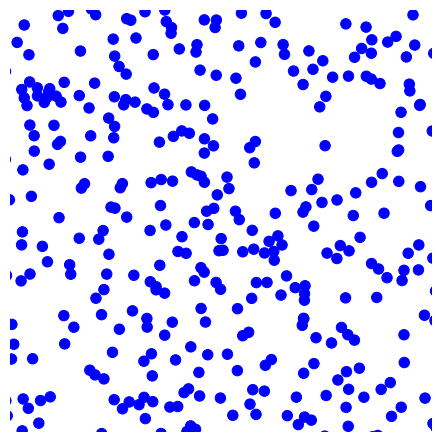
discretize space
(finite-volume scheme)



use a grid fixed in space

Lagrangian methods

discretize mass

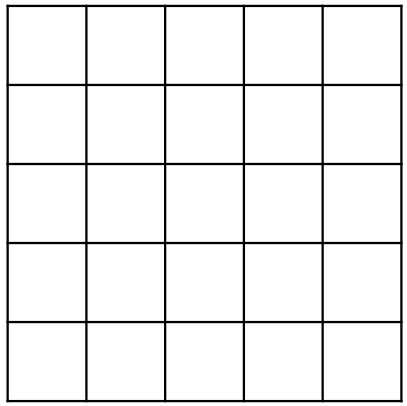


use particles for the gas (like in n-body) which move with the flow

Eulerian vs Lagrangian methods

Eulerian methods

discretize space
(finite-volume scheme)



use a grid fixed in space

moving-mesh

discretize space
(finite-volume scheme)

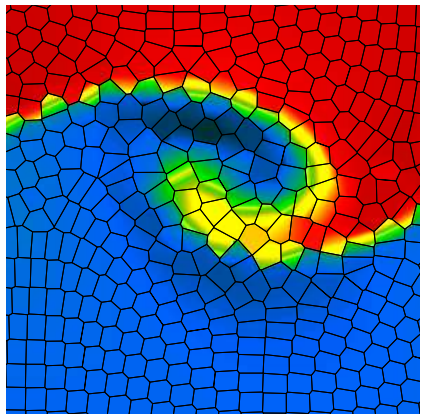
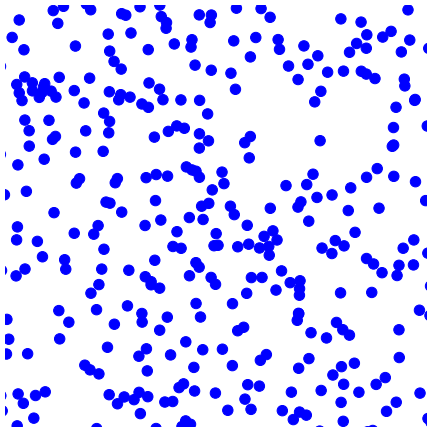


image credit: V. Springel

uses an unstructured
mesh moving with the flow

Lagrangian methods

discretize mass



use particles for the gas (like
in n-body) which move with
the flow

The Euler equations

- The equations of hydrodynamics can be written in terms of conserved quantities

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0$$ mass conservation

$$\frac{\partial (\rho \mathbf{u})}{\partial t} + \nabla \cdot (\rho \mathbf{u}\mathbf{u}) + \nabla P = \mathbf{0}$$ momentum conservation

$$\frac{\partial \rho e_{\rm tot}}{\partial t} + \nabla \cdot ((\rho e_{\rm tot} + P)\mathbf{u}) = 0$$ energy conservation

with $e_{\rm tot} = e + \frac{u^2}{2}$ and e = internal energy per unit mass

$$P = (\gamma - 1)\rho e$$ equation of state

Advection on a grid

- Let's consider a simpler problem first

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0$$

  and assuming u = const.

- tesselate space into cells for numerical treatment
- need to advect the mass such that total mass is conserved
  - ➡ calculate mass fluxes at cell interfaces
  - ➡ remove mass from the cell on one side of the interface and add it to the cell on the other side (this ensures mass conservation)

Advection on a grid

- update conserved quantities

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

- simplest method: donor-cell algorithm

$$f_{i-1/2} = q_{i-1} u \quad \text{for u > 0}$$

- or in general

$$f_{i-1/2} = \frac{1}{2} u_{i-1/2} \left[(1 + \theta_{i-1/2}) q_{i-1} + (1 - \theta_{i-1/2}) q_i \right]$$

where $\theta_{i-1/2} = \mathrm{sgn}(u_{i-1/2})$

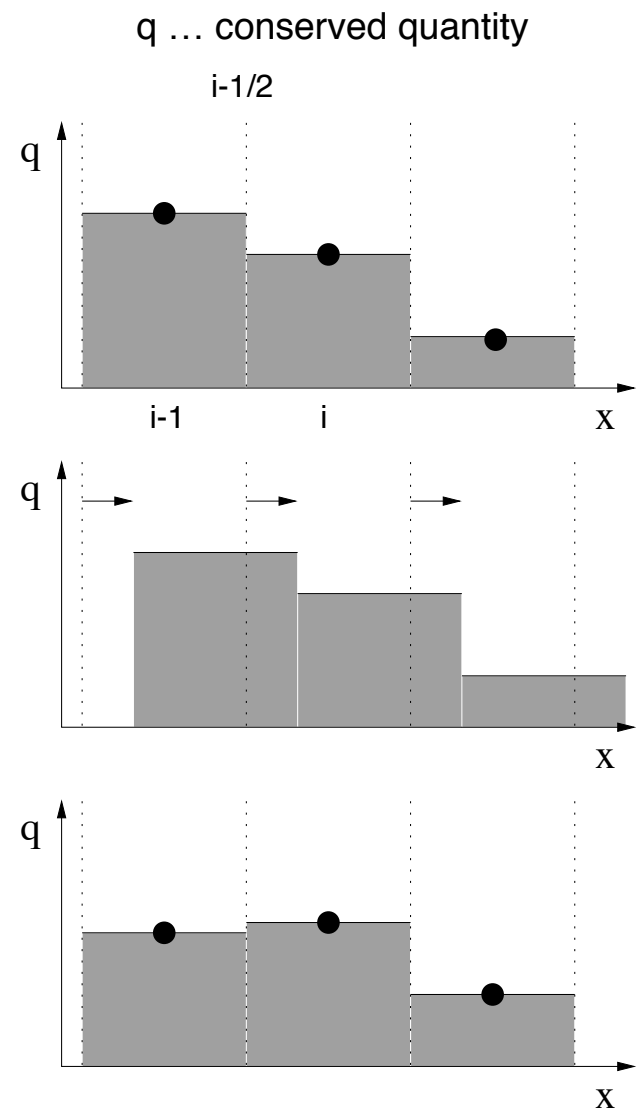


image credit: C.P. Dullemond

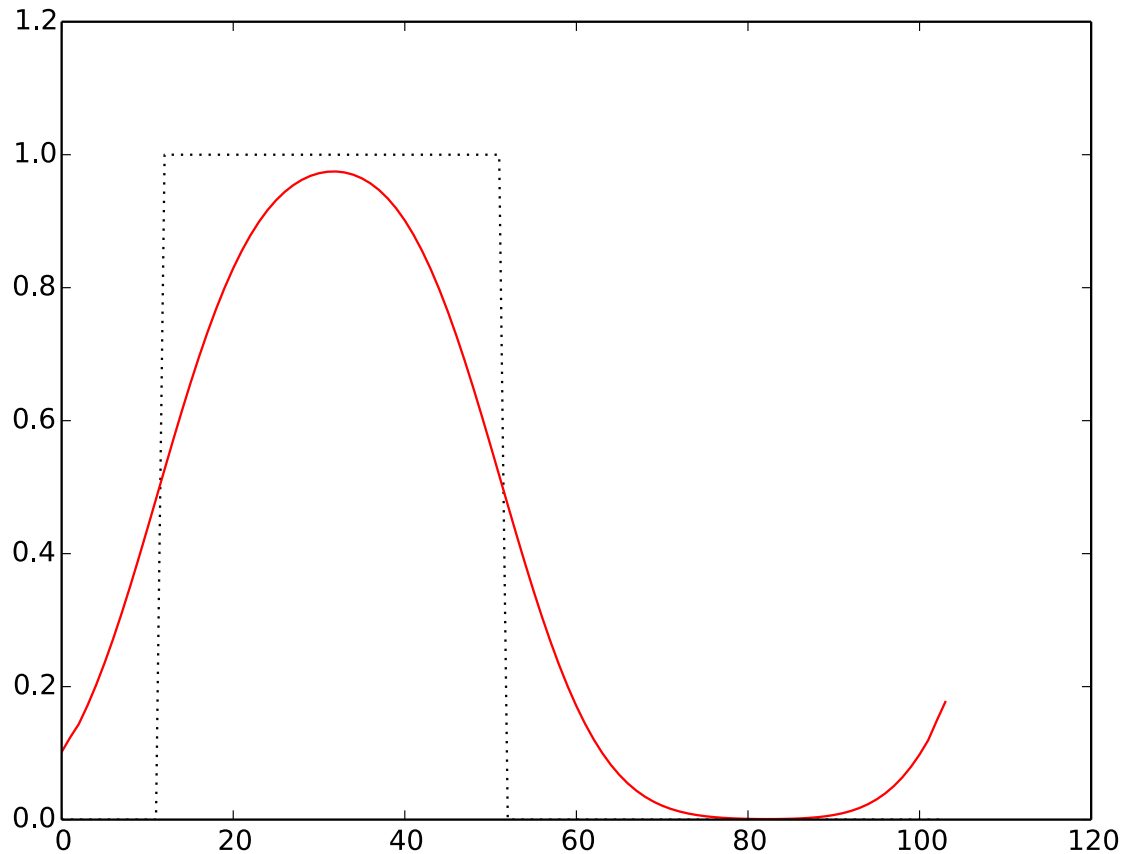
Advection on a grid

- simplest method: donor-cell algorithm
- advection of top hat:

Advection on a grid

- simplest method: donor-cell algorithm
- advection of top hat:

-> scheme is very diffusive



Advection on a grid

- next higher order: piecewise linear within cell

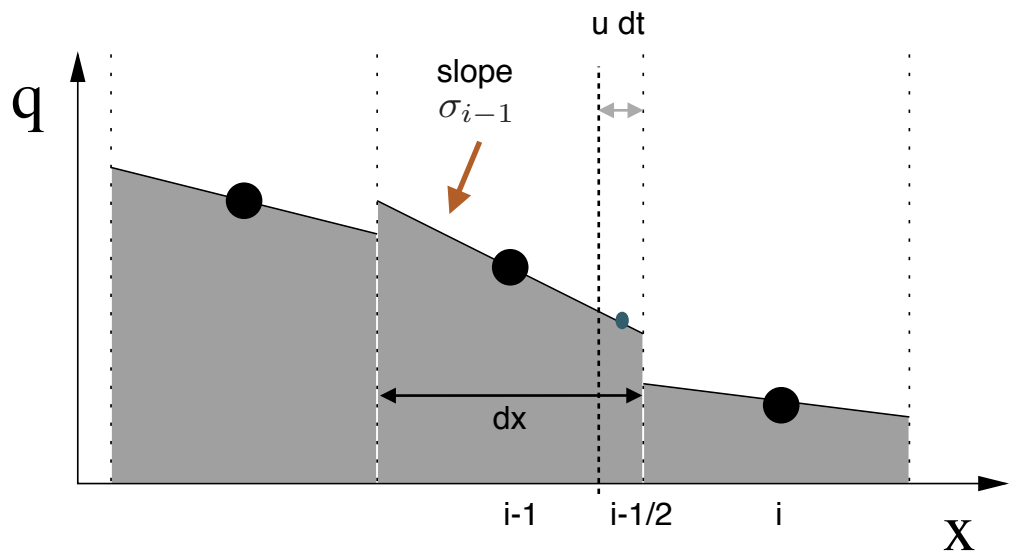


image credit: C.P. Dullemond

average density at interface over time step Δt

$$q_{\mathrm{average},i-1/2} = q_{i-1} + \sigma_{i-1}\left(\frac{\Delta x}{2} - \frac{u\Delta t}{2}\right)$$

flux -> $f_{i-1/2} = q_{\mathrm{average},i-1/2}u$

Advection on a grid

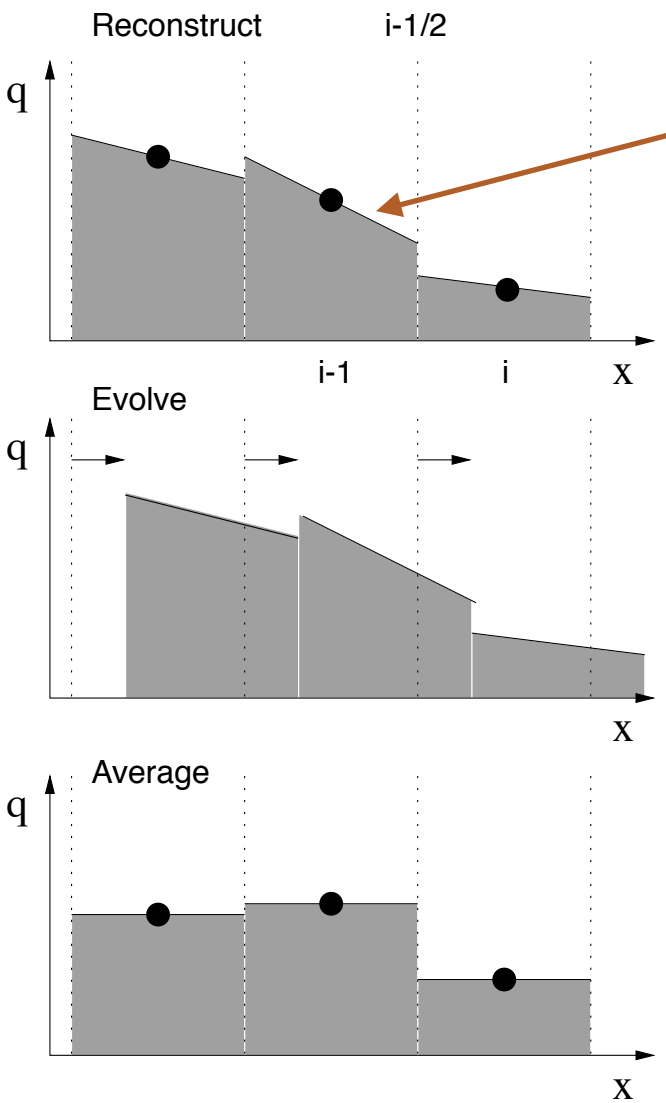


image credit: C.P. Dullemond

What should we use for the slope (for u>0)?

Lax-Wendroff (downwind)

$$\sigma_{i-1} = \frac{q_i - q_{i-1}}{\Delta x}$$

Advection on a grid

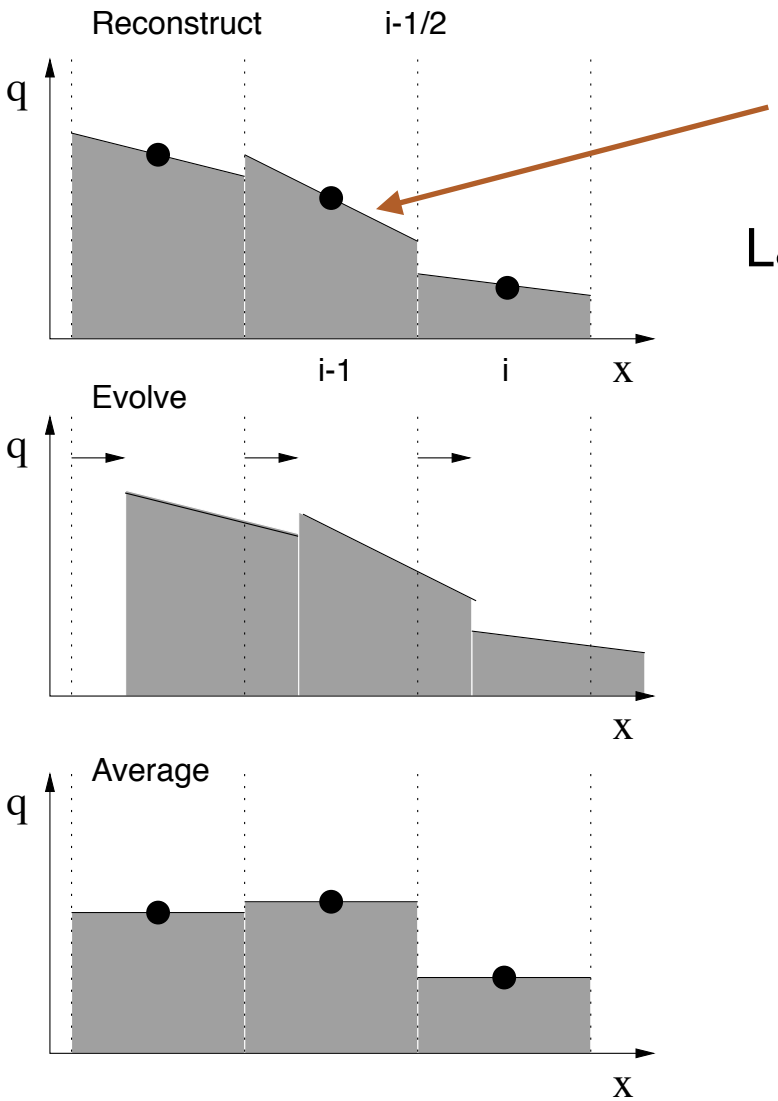
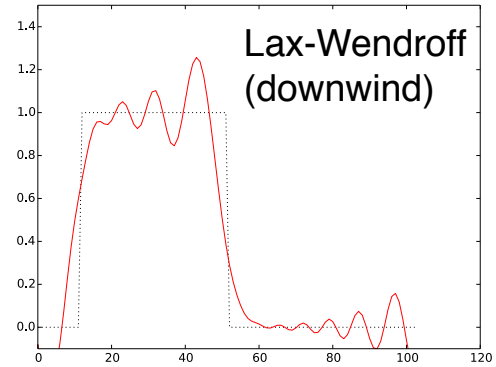


image credit: C.P. Dullemond

What should we use for the slope (for u>0)?

Lax-Wendroff (downwind)

$$\sigma_{i-1} = \frac{q_i - q_{i-1}}{\Delta x}$$



Advection on a grid

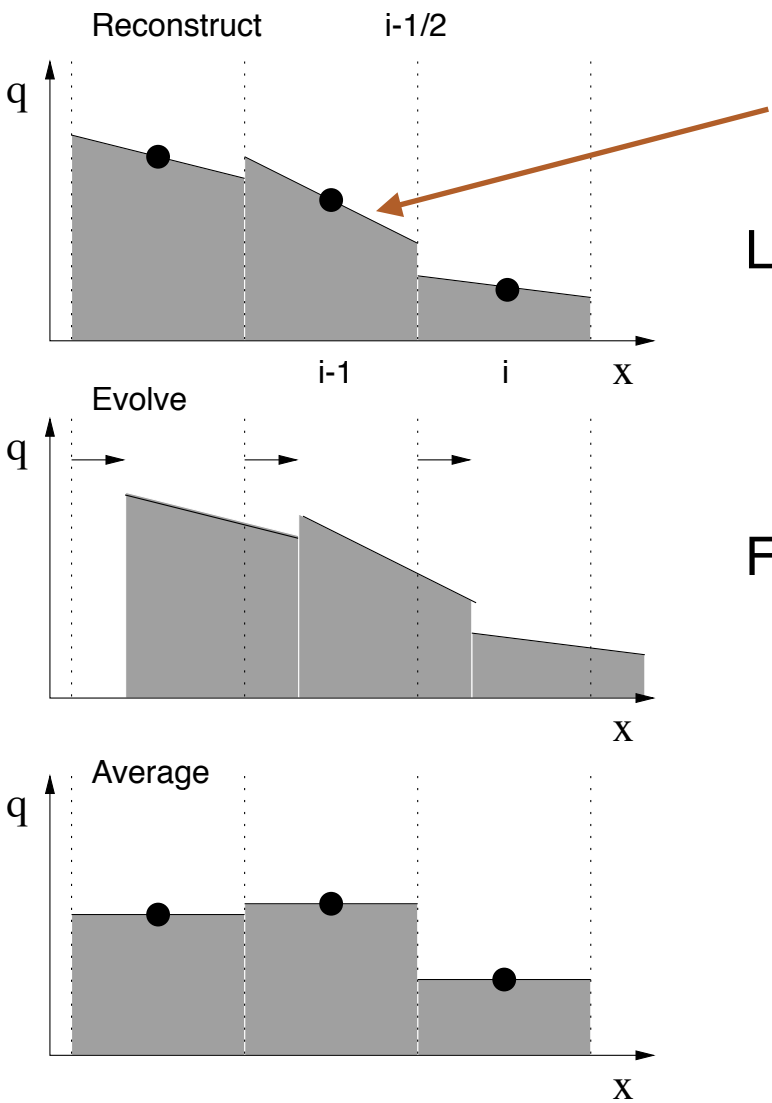


image credit: C.P. Dullemond

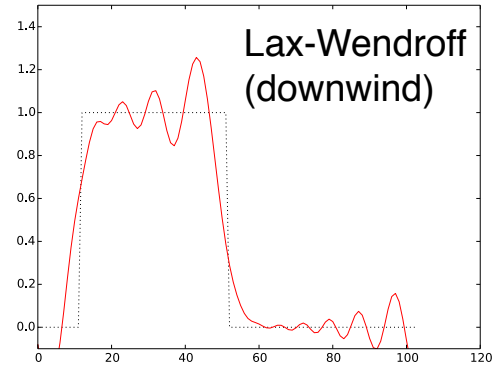
What should we use for the slope (for u>0)?

Lax-Wendroff (downwind)

$$\sigma_{i-1} = \frac{q_i - q_{i-1}}{\Delta x}$$

Fromm (cell-centered)

$$\sigma_{i-1} = \frac{q_i - q_{i-2}}{2\Delta x}$$



Advection on a grid

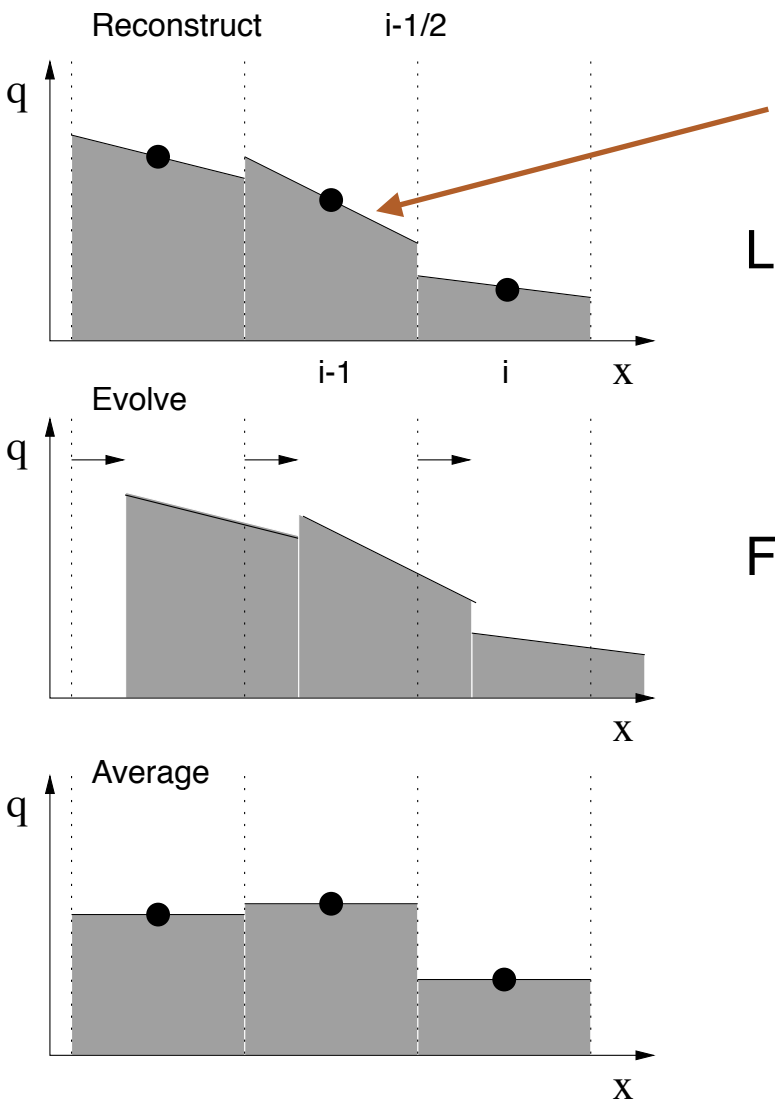


image credit: C.P. Dullemond

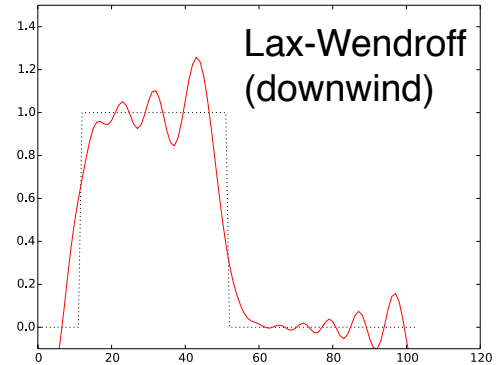
What should we use for the slope (for u>0)?

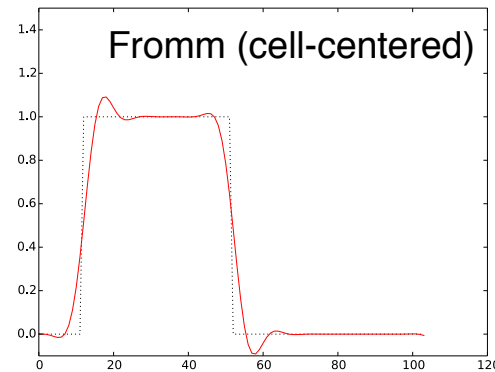
Lax-Wendroff (downwind)

$$\sigma_{i-1} = \frac{q_i - q_{i-1}}{\Delta x}$$

Fromm (cell-centered)

$$\sigma_{i-1} = \frac{q_i - q_{i-2}}{2\Delta x}$$





Advection on a grid

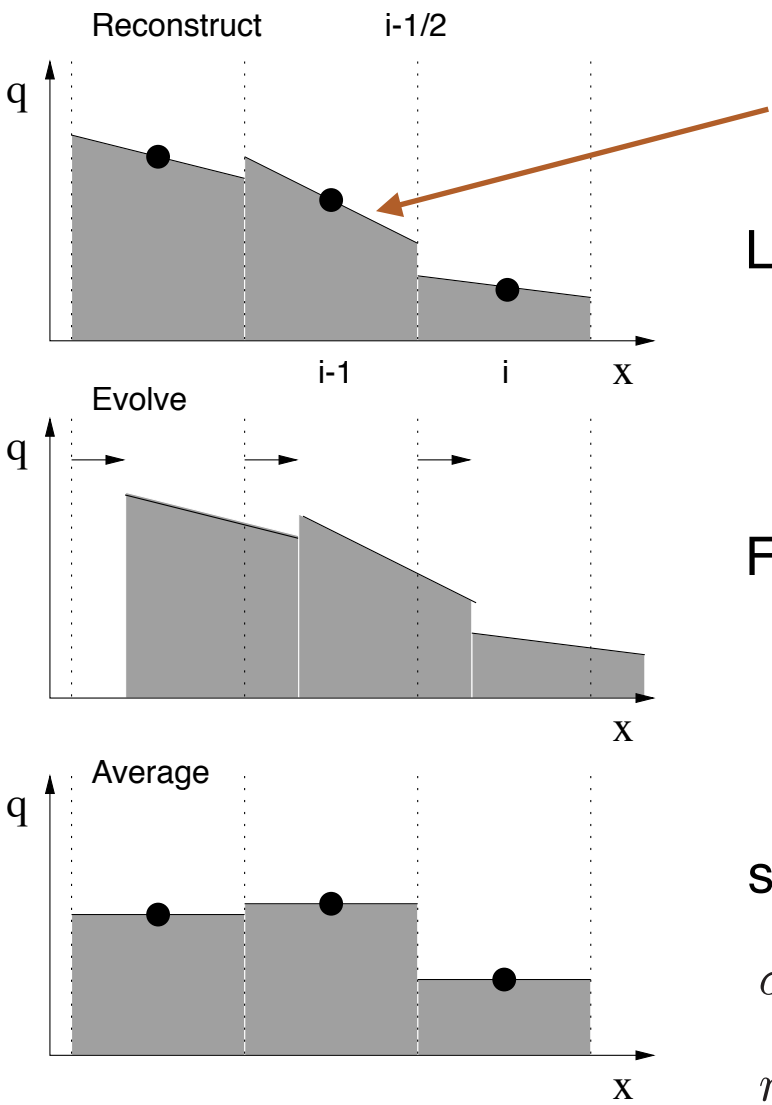


image credit: C.P. Dullemond

What should we use for the slope (for u>0)?

Lax-Wendroff (downwind)

$$\sigma_{i-1} = \frac{q_i - q_{i-1}}{\Delta x}$$

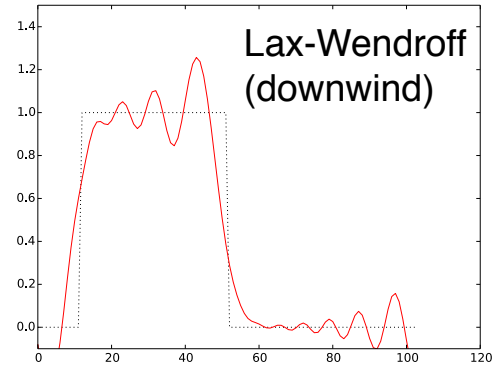
Fromm (cell-centered)

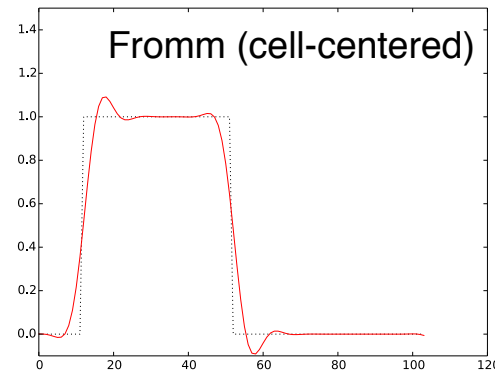
$$\sigma_{i-1} = \frac{q_i - q_{i-2}}{2\Delta x}$$

superbee flux limiter

$$\sigma_{i-1} = \max(0, \min(1, 2r), \min(2, r)) \times (q_i - q_{i-1})$$

$$r \equiv \frac{q_{i-2} - q_{i-1}}{q_i - q_{i-1}}$$





Advection on a grid

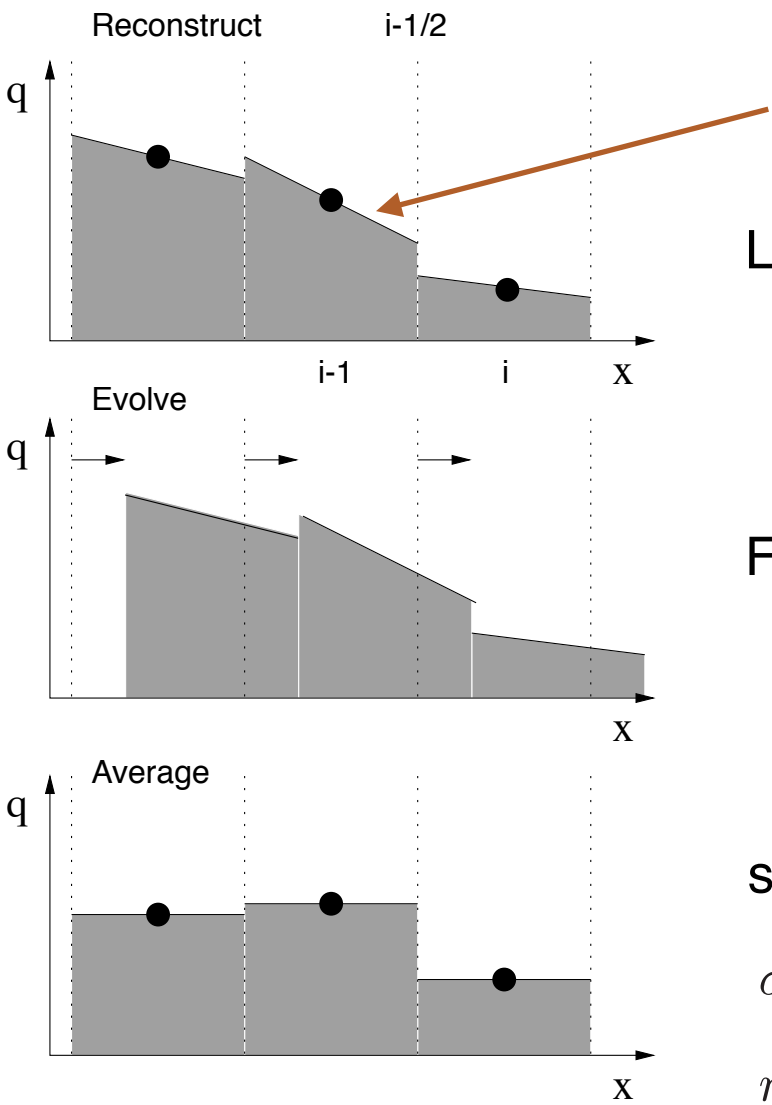


image credit: C.P. Dullemond

What should we use for the slope (for u>0)?

Lax-Wendroff (downwind)

$$\sigma_{i-1} = \frac{q_i - q_{i-1}}{\Delta x}$$

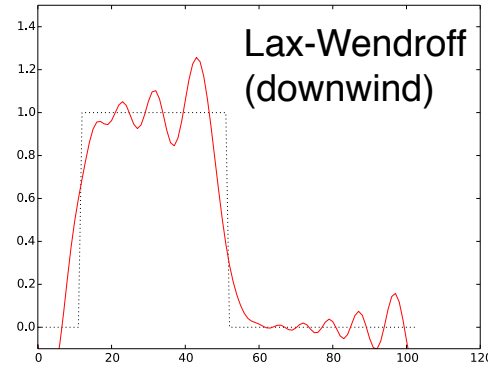
Fromm (cell-centered)

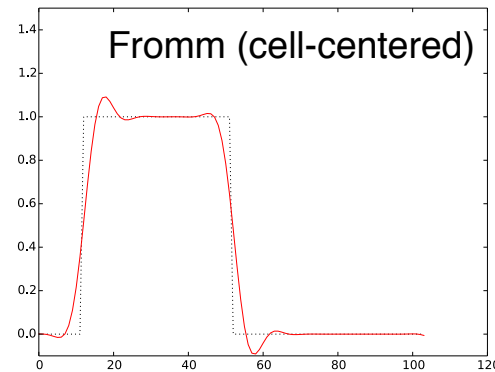
$$\sigma_{i-1} = \frac{q_i - q_{i-2}}{2\Delta x}$$

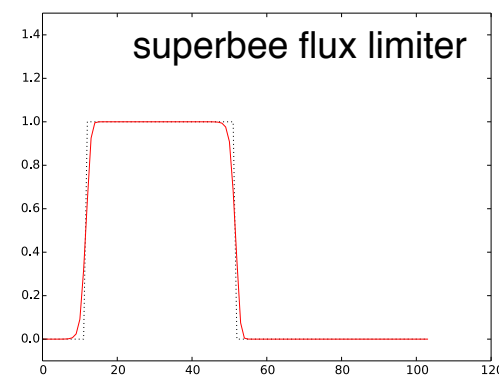
superbee flux limiter

$$\sigma_{i-1} = \max(0, \min(1, 2r), \min(2, r)) \times (q_i - q_{i-1})$$

$$r \equiv \frac{q_{i-2} - q_{i-1}}{q_i - q_{i-1}}$$







Characteristics

- With a suitable advection scheme we can solve

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0$$

with u = const.

- the full Euler equations are, however, coupled

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0$$

$$\frac{\partial (\rho \mathbf{u})}{\partial t} + \nabla \cdot (\rho \mathbf{u}\mathbf{u}) + \nabla P = \mathbf{0} \qquad P = (\gamma - 1)\rho e$$

$$\frac{\partial \rho e_{\mathrm{tot}}}{\partial t} + \nabla \cdot ((\rho e_{\mathrm{tot}} + P)\mathbf{u}) = 0$$

Characteristics

- Is it possible to decouple the equations? First we rewrite them (for simplicity in 1D) by defining

$$q_1 = \rho$$

$$q_2 = \rho u$$

$$q_3 = \rho e_{\rm tot}$$

- Using this we find

$$u = \frac{q_2}{q_1}$$

$$e = e_{\rm tot} - \frac{1}{2}u^2 = \frac{q_3}{q_1} - \frac{1}{2}\frac{q_2^2}{q_1^2}$$

$$P = (\gamma - 1)\left(q_3 - \frac{1}{2}\frac{q_2^2}{q_1}\right)$$

-> all fluid quantities can be written in terms of q_1,q_2,q_3

-> can use these expressions to replace them in the Euler equations

Characteristics

- Euler equations in terms of q_1,q_2,q_3

$$\partial_t \begin{pmatrix} q_1 \\ q_2 \\ q_3 \end{pmatrix} + \partial_x \begin{pmatrix} f_1 \\ f_2 \\ f_3 \end{pmatrix} = 0$$

conserved quantities fluxes

where

$$\begin{pmatrix} f_1 \\ f_2 \\ f_3 \end{pmatrix} = \begin{pmatrix} \rho u \\ \rho u^2 + P \\ (\rho e_{\rm tot} + P)u \end{pmatrix} = \begin{pmatrix} q_2 \\ (\gamma - 1)q_3 + \frac{3-\gamma}{2}\frac{q_2^2}{q_1} \\ \gamma \frac{q_2 q_3}{q_1} + \frac{1-\gamma}{2}\frac{q_2^3}{q_1^2} \end{pmatrix}$$

can be written more compact using the Jacobian

$$A_{k,j}(q) \equiv \frac{\partial f_k}{\partial q_j} \quad \Rightarrow \quad \partial_t q_k + A_{k,j}(q)\partial_x q_j = 0$$

Characteristics

- Let us one more time consider a simpler system first and assume A(q) = A = const.

$$\partial_t q_k + A_{k,j} \partial_x q_j = 0$$

- We can then decouple the equations by finding the eigensystem of the matrix A

$$\lambda_1, \lambda_2, \lambda_3 \qquad \mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$$

eigenvalues eigenvectors

- We can then decompose the state q in this eigenbasis

$$\mathbf{q} = \sum_m \tilde{q}_m \mathbf{e}_m$$

and find the equations for the individual components

Characteristics

- Plugging decomposition into $\partial_t q_k + A_{k,j}\partial_x q_j = 0$

$$\partial_t \tilde{q}_m \mathbf{e}_m + \mathbf{A}\partial_x \tilde{q}_m \mathbf{e}_m = 0 \quad \rightarrow \quad \mathbf{e}_m \partial_t \tilde{q}_m + \lambda_m \mathbf{e}_m \partial_x \tilde{q}_m = 0$$

and hence

$$\rightarrow \quad \partial_t \tilde{q}_m + \lambda_m \partial_x \tilde{q}_m = 0$$

- This is a simple advection equation with characteristic velocity λ_m
- We can thus solve the coupled set of equations by
  - expanding the state vector in the eigenbasis $q_k \rightarrow \tilde{q}_m$
  - advecting each $\tilde{q}_m$ with its characteristic velocity
  - recomputing the new state from the updated $\tilde{q}_m \rightarrow q_k$

Characteristics

- Each of these modes propagating with a characteristic velocity is called a ***characteristic***
- for the full Euler equations $\mathbf{A}(\mathbf{q}) \neq \mathrm{const}$

  ➡ eigenvectors depend on q (and thus position)

  ➡ no global decomposition of the state vector possible
- locally we find:

$$\lambda_- = u - \sqrt{\frac{\gamma P}{\rho}} = u - c_\mathrm{s}$$ backward travelling sound wave

$$\lambda_0 = u$$ fluid motion

$$\lambda_+ = u + \sqrt{\frac{\gamma P}{\rho}} = u + c_\mathrm{s}$$ forward travelling sound wave

Characteristics

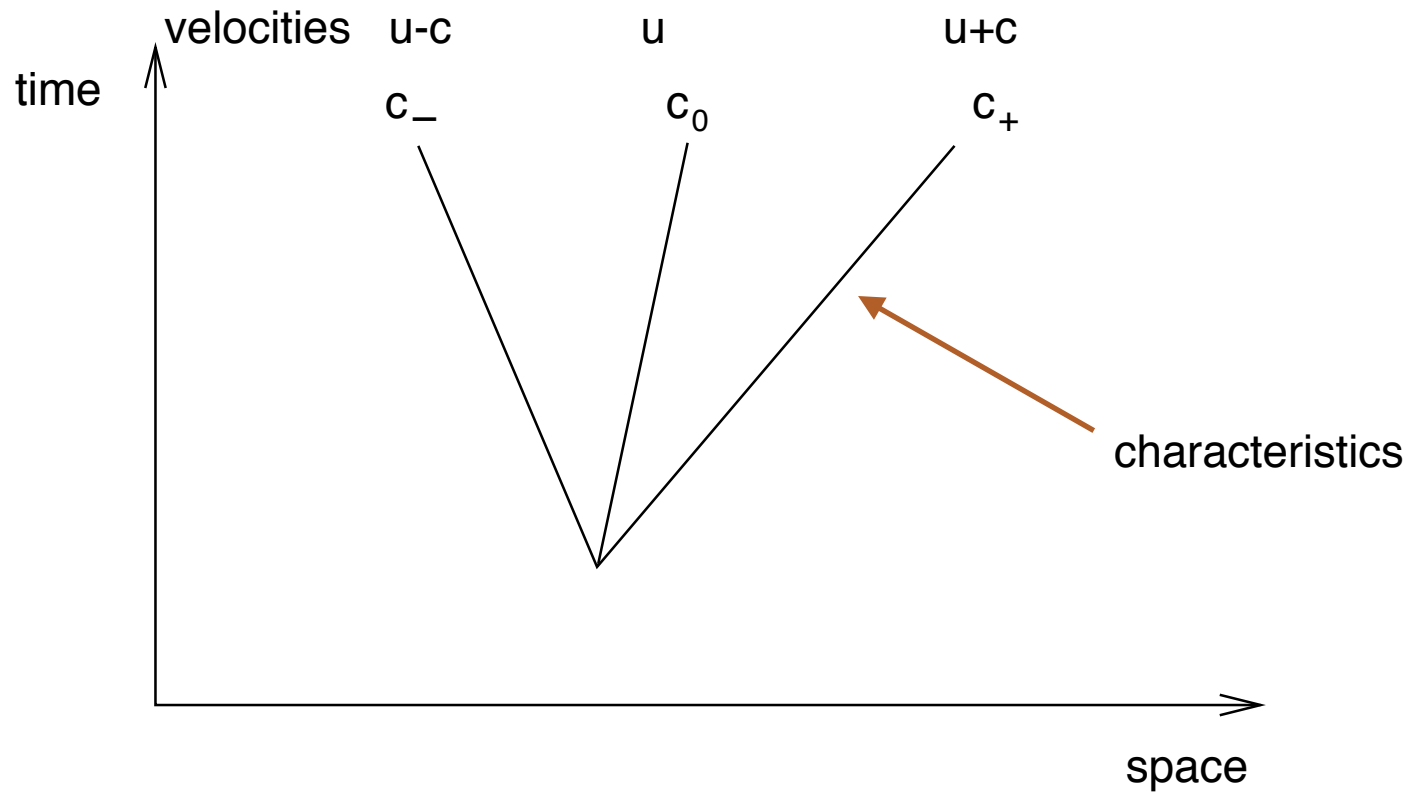


image credit: C.P. Dullemond

➡ could decompose the state vector locally at each cell interface and advect the components with their local characteristic velocities

Riemann problems

- What about shocks and contact discontinuities?

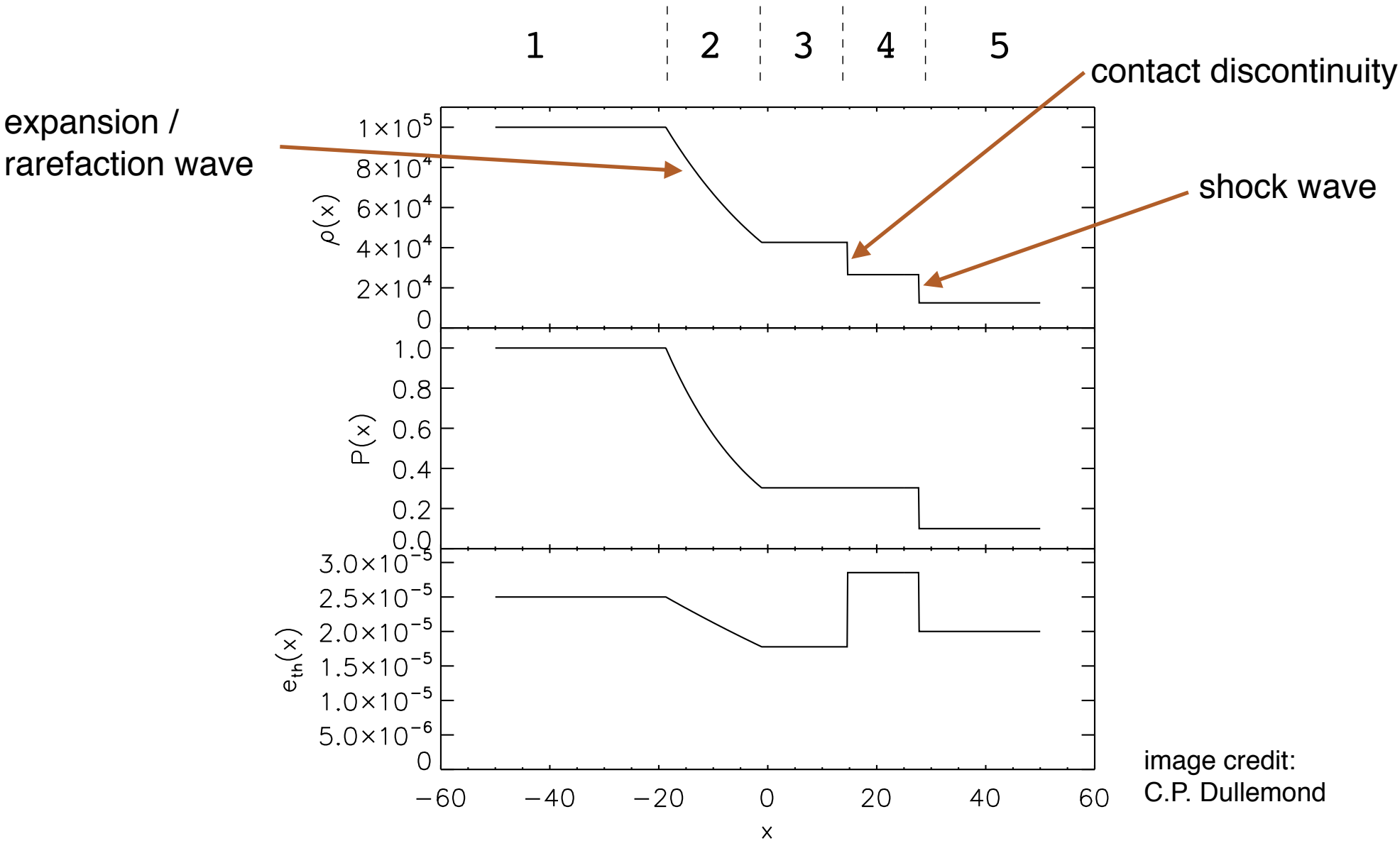
  ➡ eigensystems differ significantly on both sides

- Let's look at the full Riemann problem:

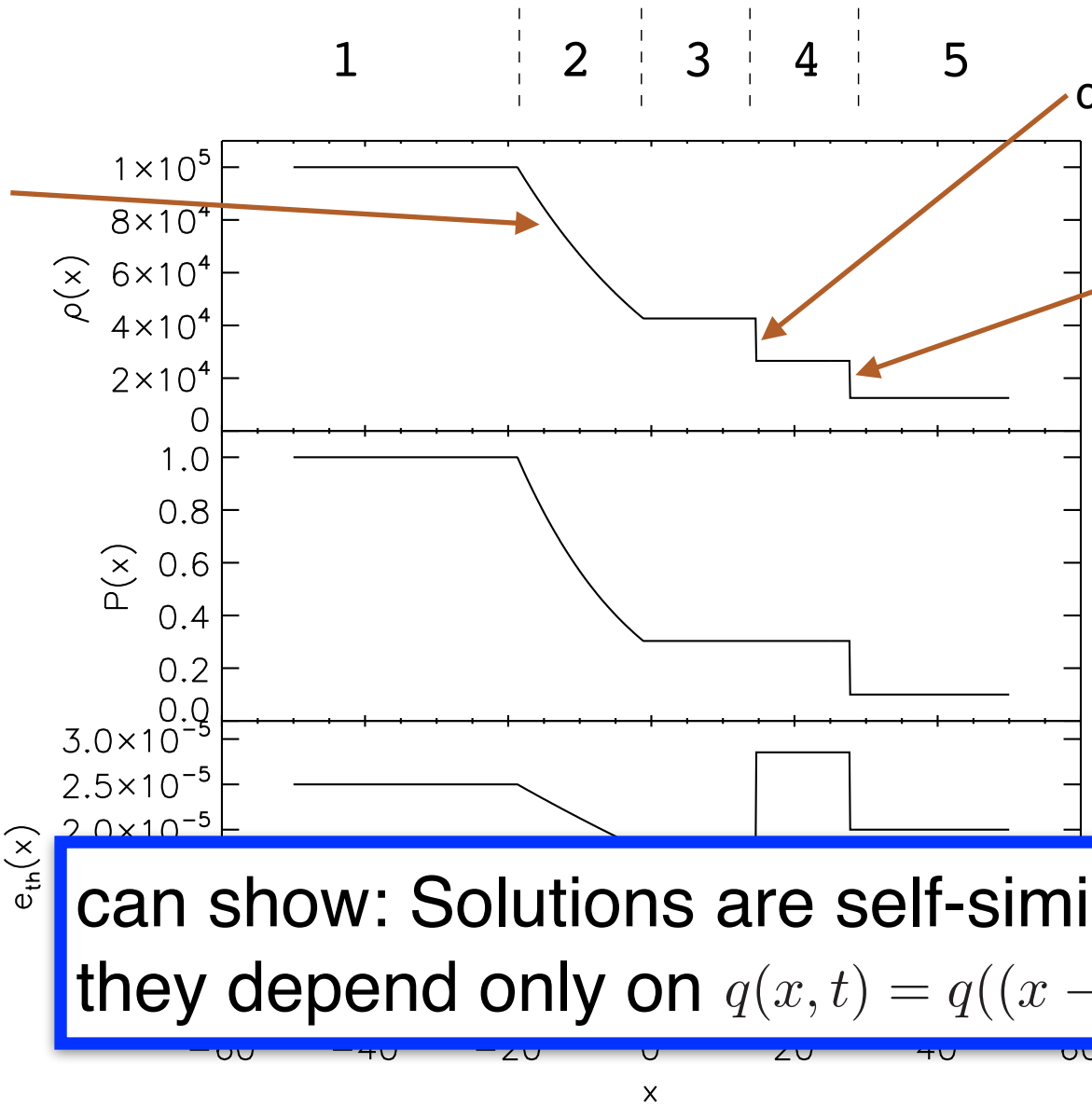
$$q(x, t = t_0) = q_{\mathrm{L}} \quad \text{for } x < x_0$$
$$q(x, t = t_0) = q_{\mathrm{R}} \quad \text{for } x > x_0$$

- e.g. for $u_L = u_R = 0$

Riemann problems



Riemann problems



can show: Solutions are self-similar, i.e. they depend only on $q(x,t) = q((x - x_0)/(t - t_0))$

Godunov's method

- assume piecewise constant fluid state
- exactly solve the Riemann problem at each interface (semi-numeric, iteratively solve algebraic equations)
- choose time step small enough that solutions of neighbouring interfaces do not overlap

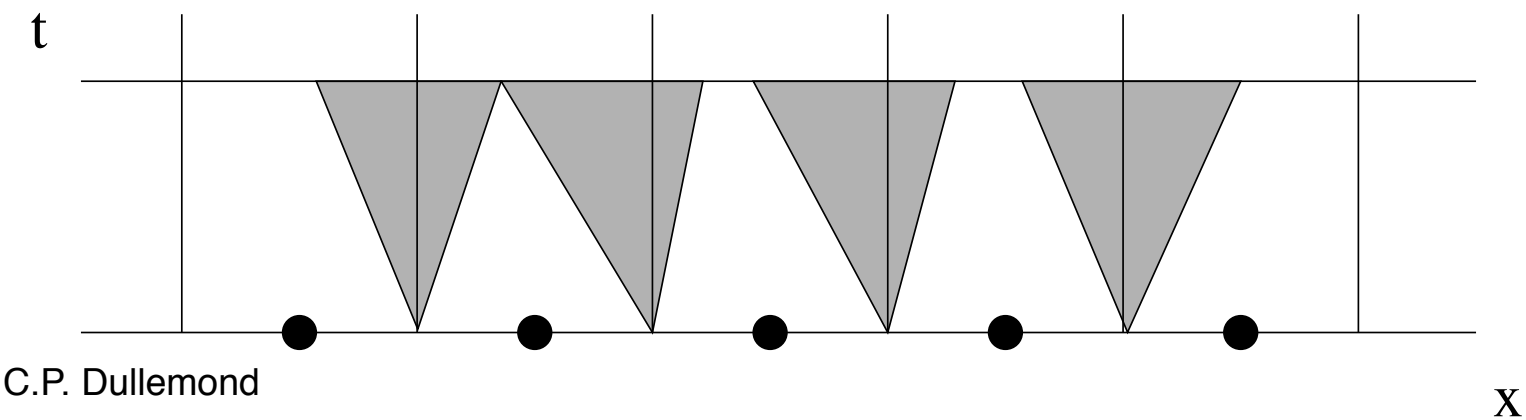


image credit: C.P. Dullemond

- calculate new average of conserved quantities at end of time step
- easy as flux f(q(x=x_0)) is constant for self-similar $q(x - x_0/(t - t_0))$

Godunov's method

- assume piecewise constant fluid state
- exactly solve the Riemann problem at each interface (semi-numeric, iteratively solve algebraic equations)
- choose time ste
  interfaces do n

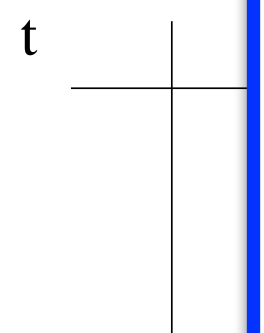


image credit: C.P. Dullemond

- calculate new a
- easy as flux f(q

for linear problems: same as advecting the components in the eigenbasis

but accounts for shocks and contact discontinuities

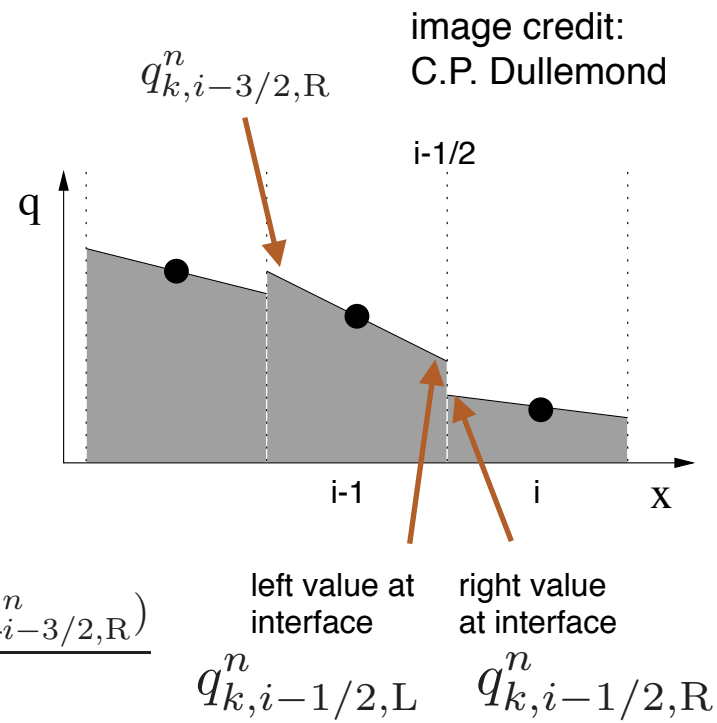
Cons: diffusive (constant fluxes correspond to donor-cell advection)

o

$t_0))$

MUSCL-Hanckock scheme

- make Godunov's method second order by
  - using a higher-order reconstruction (piecewise linear, piecewise parabolic)
  - computing the left and right q values at the interface
  - advance these values half a step in time



$$\partial_t q_k + \partial_x f_k = 0 \quad \rightarrow \quad q^{n+1/2}_{k,i-1/2,\mathrm{L}} = q^n_{k,i-1/2,\mathrm{L}} - \frac{\Delta t}{2} \frac{f_k(\mathbf{q}^n_{i-1/2,\mathrm{L}}) - f_K(\mathbf{q}^n_{i-3/2,\mathrm{R}})}{\Delta x}$$

  - use these values in the Riemann solver as if the state is constant on each side of the interface
  - used in many codes, e.g.: Ramses, Arepo

Roe's linearized Riemann solver

- exact Riemann solvers can be slow (e.g. for magneto-hydrodynamics)
- alternatively linearise the problem at each interface by setting

$$\mathbf{A}(\mathbf{q}) \quad \rightarrow \quad \mathbf{A}(\hat{\mathbf{q}})$$

$\hat{\mathbf{q}}$ suitable average value between left and right state

- a solution can then be found by decomposing the left and right states into the eigenbasis of $\mathbf{A}(\hat{\mathbf{q}})$
- and advecting the components with the corresponding characteristic velocity

➡ yields a solution in smooth parts of the flow

➡ every reasonable average value for $\hat{\mathbf{q}}$ should work there

Roe's linearized Riemann solver

- It is possible to choose the average $\hat{\mathbf{q}}$ such that the linearized Riemann solver also gives the correct propagation of contact discontinuities and shocks
- Roe average:

$$\hat{u} = \frac{\sqrt{q_\mathrm{L}} u_\mathrm{L} + \sqrt{q_\mathrm{R}} u_\mathrm{R}}{\sqrt{q_\mathrm{L}} + \sqrt{q_\mathrm{R}}}$$

$$\hat{h}_\mathrm{tot} = \frac{\sqrt{q_\mathrm{L}} h_\mathrm{tot,L} + \sqrt{q_\mathrm{R}} h_\mathrm{tot,R}}{\sqrt{q_\mathrm{L}} + \sqrt{q_\mathrm{R}}}$$

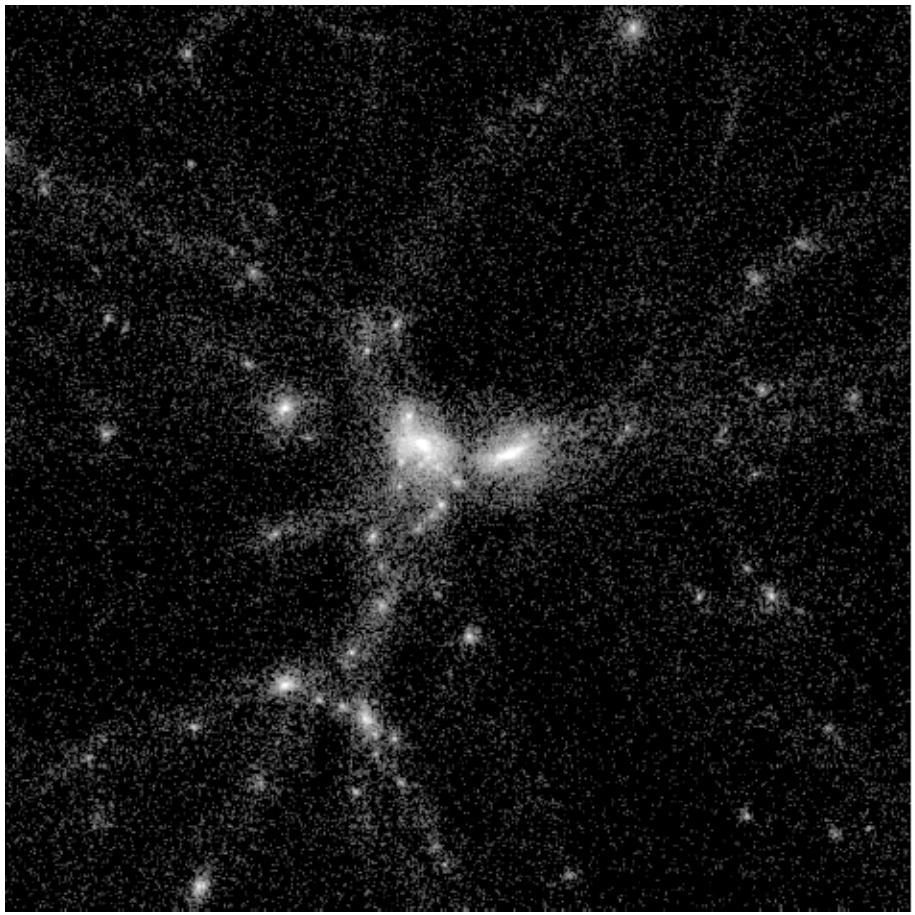
- When using this average one can show that the "jump" corresponds exactly to one eigenvector with a eigenvalue given by the correct velocity (e.g. the shock velocity).

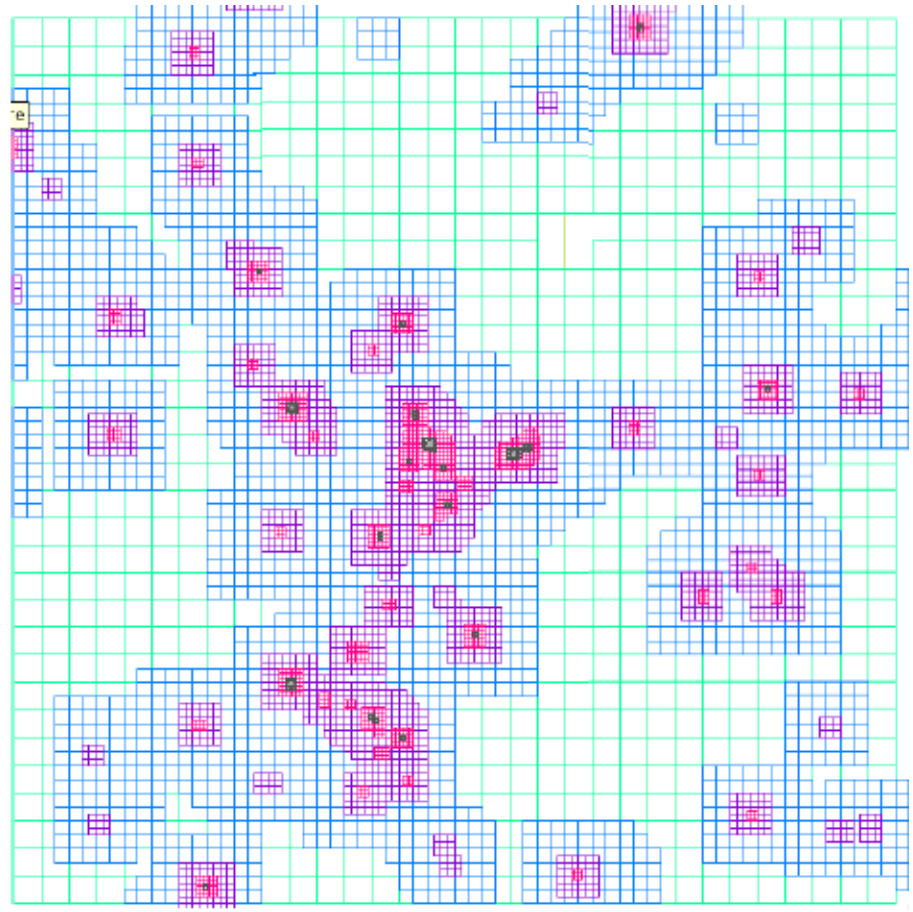
Multi-dimensional hydrodynamics

- unsplit schemes:
  - compute fluxes for all interfaces of cells
  - update cell values once per time step
- directionally split schemes:
  - apply 1D hydro scheme alternately along the different directions
  - needs less memory
  - but typically preserves e.g. spherical symmetry less well

Adaptive mesh-refinement

Eulerian codes use adaptive mesh refinement to get higher resolution in high density regions





source: http://www.deus-consortium.org/a-propos/cosmological-models/run/

Moving-mesh hydrodynamics (e.g. Arepo code)

- alternatively one can define a mesh based on a set of points using a Voronoi tessellation
- points can be allowed to move, e.g. with the fluid -> then almost Lagrangian
- need to:
  - use unspilt scheme
  - transform to frame of moving interface
  - solve Riemann problem
  - transform back

Voronoi tessellation

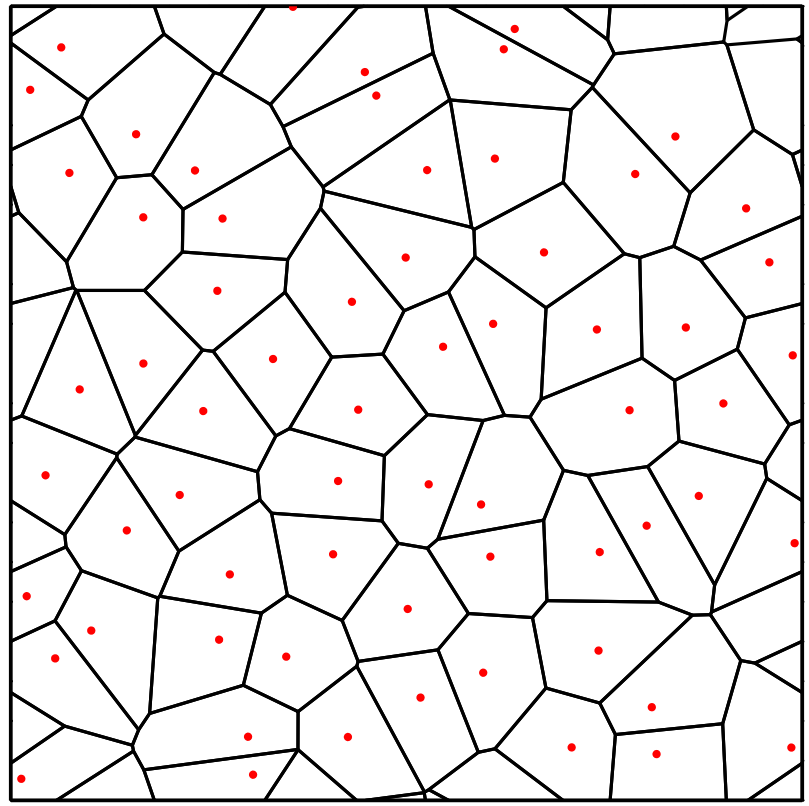
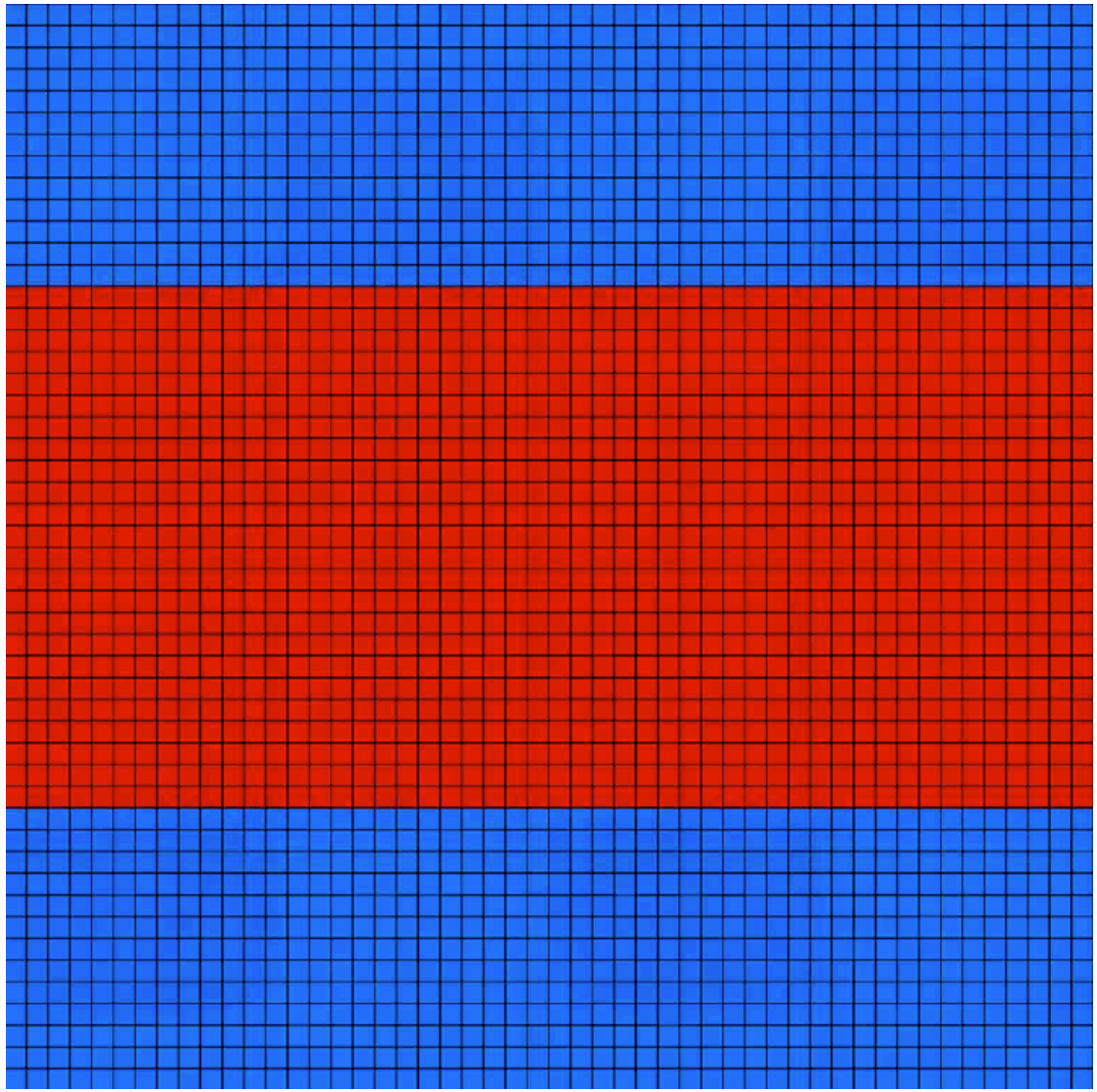


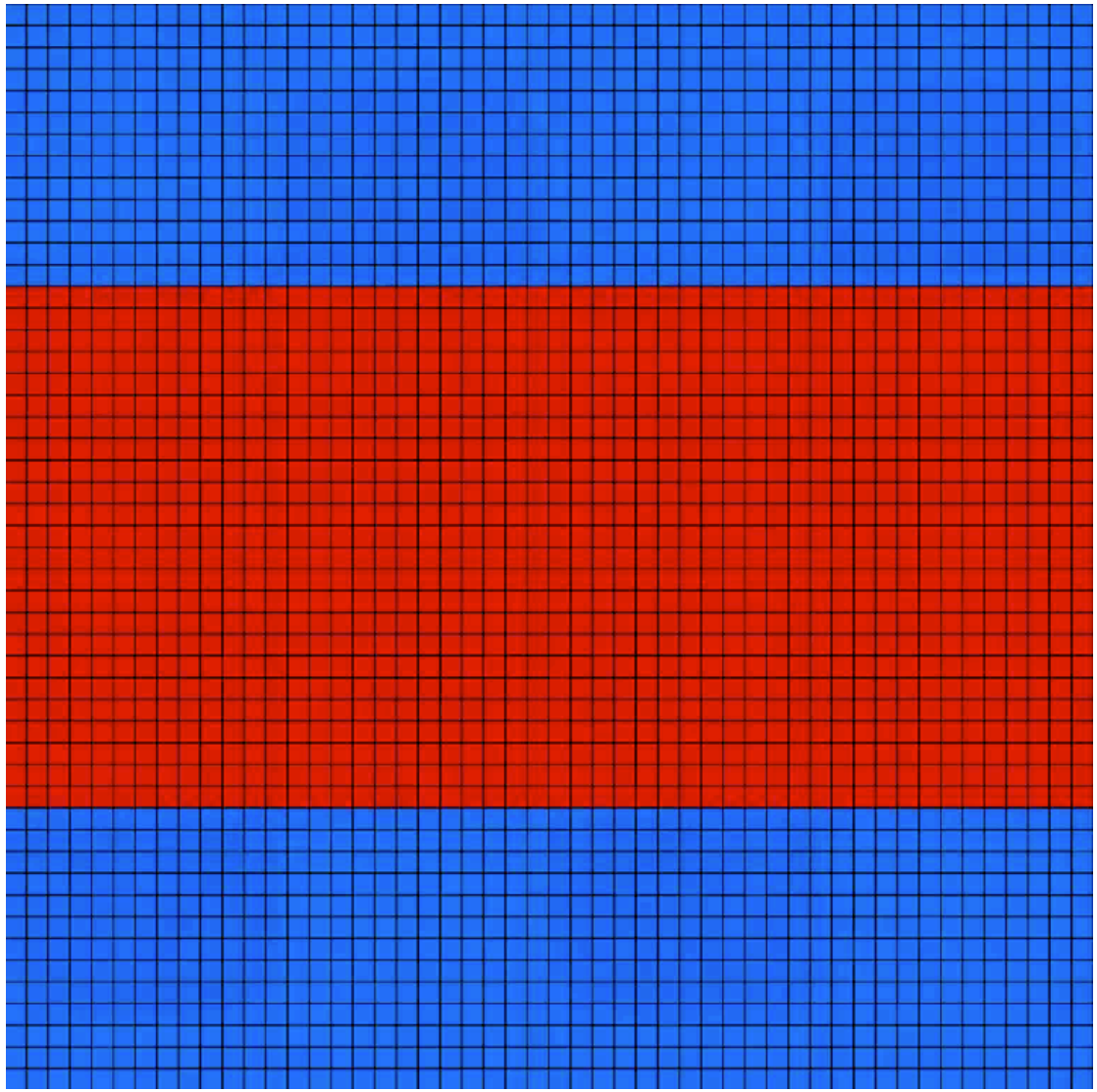
image credit: V. Springel

Moving-mesh methods



credit: V. Springel

Moving-mesh methods



credit: V. Springel

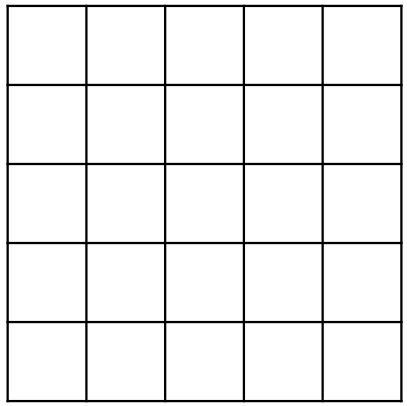
Moving-mesh methods

credit: V. Springel

Eulerian vs Lagrangian methods

Eulerian methods

discretize space
(finite-volume scheme)



use a grid fixed in space

moving-mesh

discretize space
(finite-volume scheme)

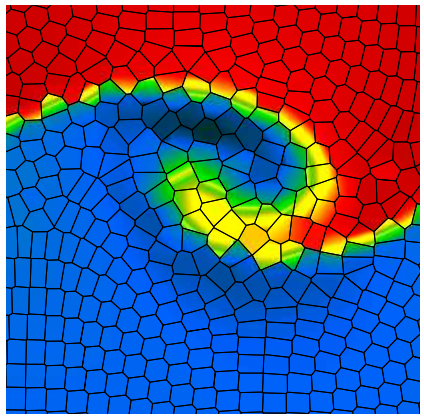
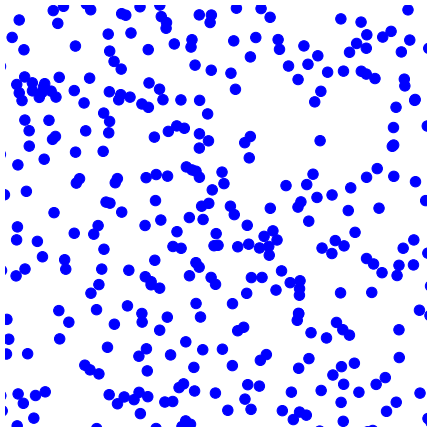


image credit: V. Springel

uses an unstructured
mesh moving with the flow

Lagrangian methods

discretize mass

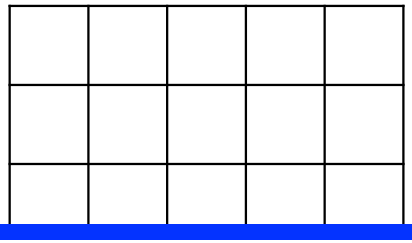


use particles for the gas (like
in n-body) which move with
the flow

Eulerian vs Lagrangian methods

Eulerian methods

discretize space
(finite-volume scheme)



Pros:

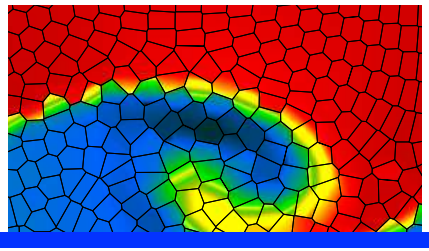
accurate hydro

Cons:

need adaptive mesh refinement to get high resolution

moving-mesh

discretize space
(finite-volume scheme)



Pros:

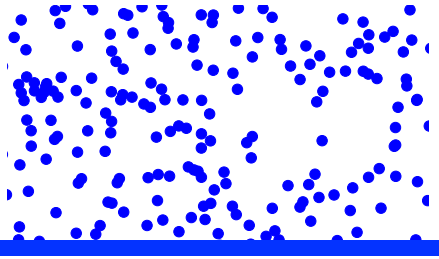
accurate hydro, automatic refinement on density, Galilean invariant by construction

Cons:

overhead (~30%) for mesh construction

Lagrangian methods

discretize mass



Pros:

automatic refinement on density, Galilean invariant by construction, conserves angular momentum (and entropy in smooth flows) exactly

Cons:

somewhat less accurate hydro (e.g. fluid instabilities, but recent improvements), slower convergence

Literature & Outlook

- Lecture script “Numerical Fluid Dynamics”, C.P. Dullemond and V. Springel, http://www.ita.uni-heidelberg.de/~dullemond/lectures/num_fluid_2012/index.shtml?lang=en

- ”Numerical Methods for Conservation Laws”, R.J. LeVeque

- “E pur si muove: Galilean-invariant cosmological hydrodynamical simulations on a moving mesh”, V. Springel, 2010, MNRAS, 401, 791

- Next lecture (Ben):
  - ***Smoothed Particle Hydrodynamics, radiative cooling, sub-resolution physics***