Monte-Carlo Tracer Particles

RUM 2017, Nice

Corentin Cadiou

September 20, 2017

IAP, CNRS

Outline

Introduction

Eulerian & Lagrangian

Eulerian code (AMR like):

- no subgrid information
- no Lagrangian history of gas

Is it possible to overcome this issue?

From passive scalar to tracers

- Where does the gas go?

From passive scalar to tracers

- Where does the gas go?
- Where does the gas come from?

From passive scalar to tracers

- Where does the gas go?
- Where does the gas come from?
- How much gas is recycled in stars? in AGNs?

From passive scalar to tracers

- Where does the gas go?
- Where does the gas come from?
- How much gas is recycled in stars? in AGNs?
- ... [TBC]

What's a tracer

What are the properties we want for tracers?

What's a tracer

What are the properties we want for tracers?

Physical properties

- Passive
- *Behave* like the gas on average
- Like individual H/He nuclei

Computational properties

- Cheap (CPU? RAM?)
- Go where the gas goes (star, sinks, grid, dust, ...)

Different methods

Velocity method

Move tracers using tri-linear interpolation of the velocity

$$v_i^p = \text{interpolation} \sum_{\text{neighbor } j} v_j \tag{1}$$

Pros

- smooth Lagrangian history (trace velocity)
- already implemented in RAMSES!

Velocity method

Move tracers using tri-linear interpolation of the velocity

$$v_i^p = \text{interpolation} \sum_{\text{neighbor } j} v_j \tag{1}$$

Pros

- smooth Lagrangian history (trace velocity)
- already implemented in RAMSES!

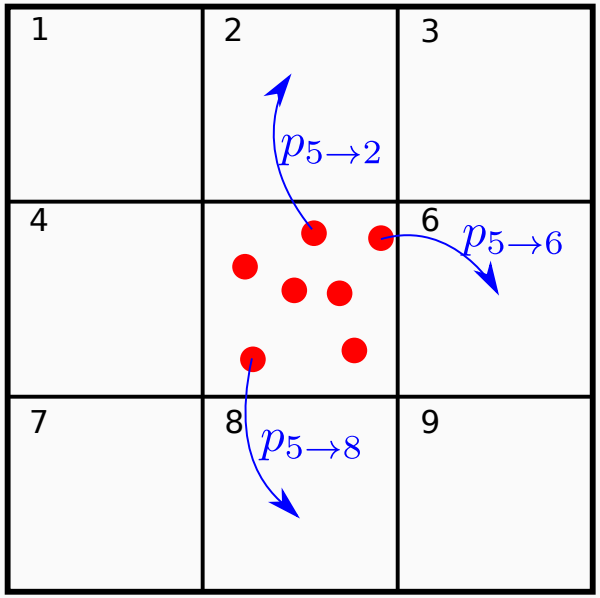
Cons

- *does not* follow the gas density: $\int \mathrm{d}S v_i^p \rho \neq \text{flux}$
- how to trace stars? AGN?
- quite CPU expensive

Monte Carlo method

Move tracers following flux

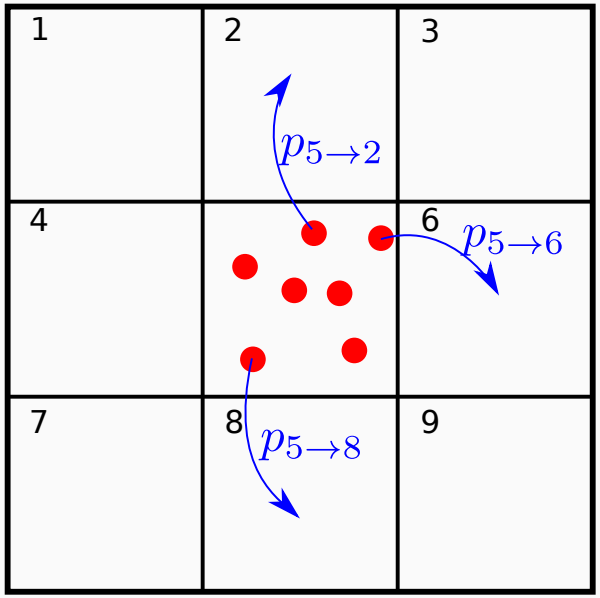
$$p_{i \to j} = \frac{\text{mass flux}_{i \to j}}{M_i}$$



Monte Carlo method

Move tracers following flux

$$p_{i\to j} = \frac{\text{mass flux}_{i\to j}}{M_i}$$



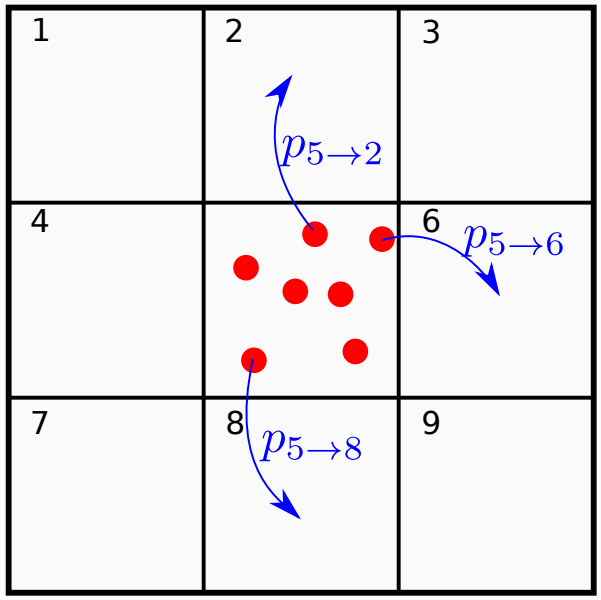
Pros

- CPU cheap
- follow gas density
- precision $\propto N_{\text{tracers}}$
- move onto stars, sinks, ...

Monte Carlo method

Move tracers following flux

$$p_{i\to j} = \frac{\text{mass flux}_{i\to j}}{M_i}$$



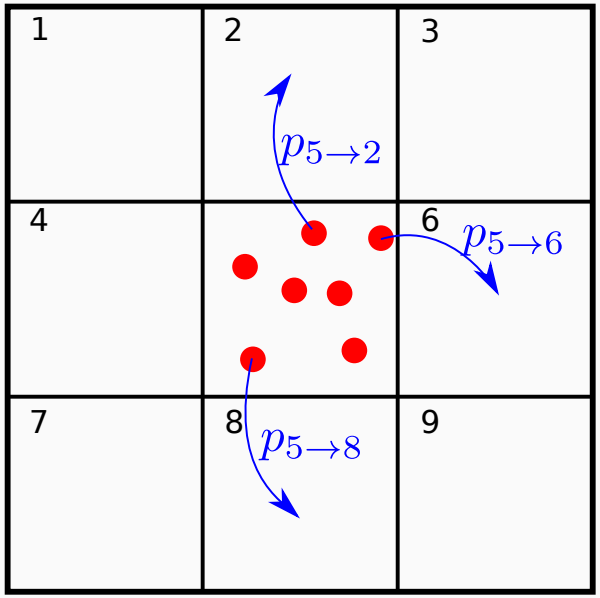
Pros

- CPU cheap
- follow gas density
- precision $\propto N_{\text{tracers}}$
- move onto stars, sinks, ...
- **now implemented!**

Monte Carlo method

Move tracers following flux

$$p_{i\to j} = \frac{\text{mass flux}_{i\to j}}{M_i}$$



Pros

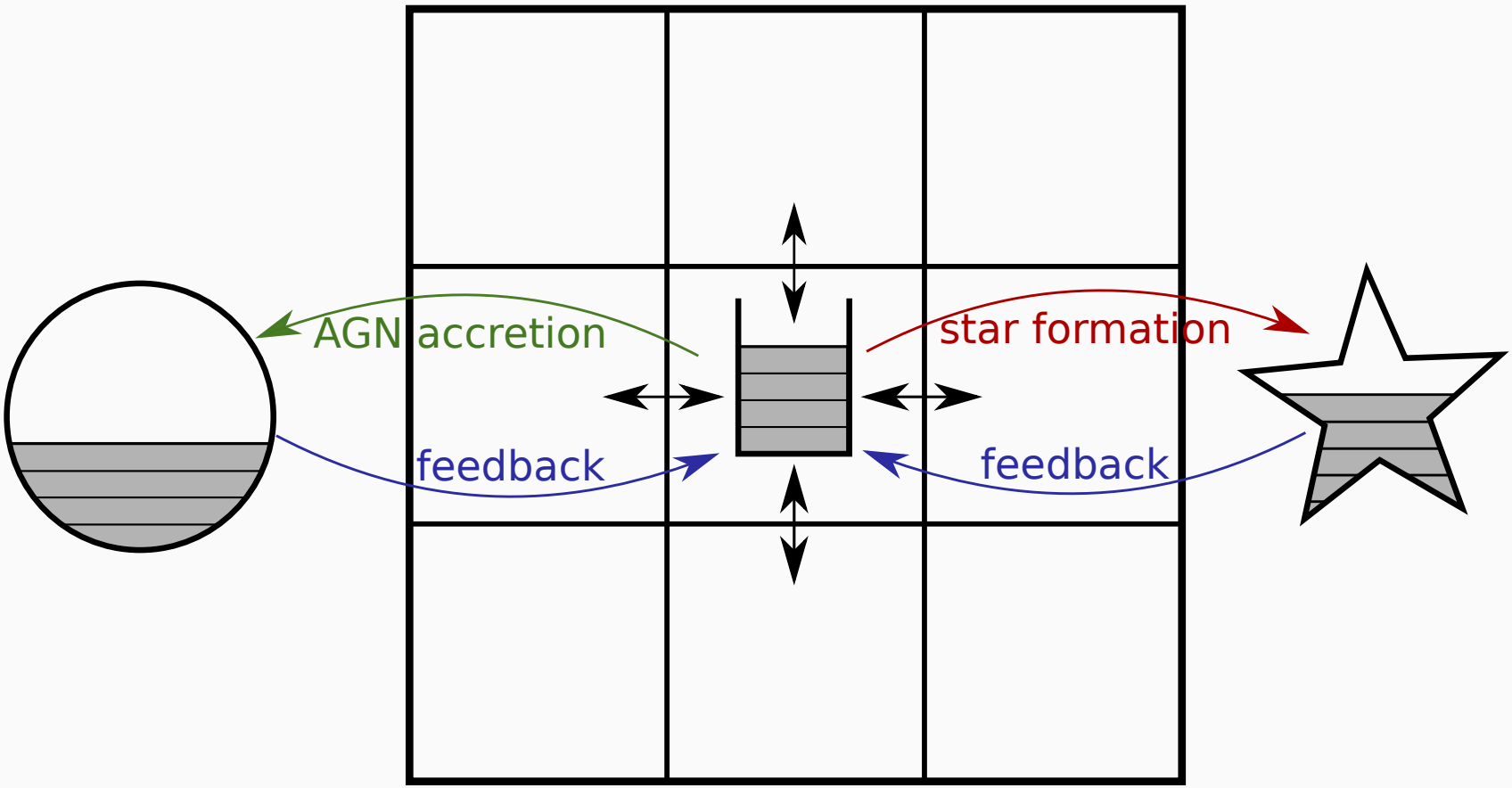
- CPU cheap
- follow gas density
- precision $\propto N_{\text{tracers}}$
- move onto stars, sinks, ...
- **now implemented!**

Cons

- RAM expensive
- noisy Lagrangian history

MC Implementation

Scheme



Equations

First and last equations: Let

$$M_{i,\text{out}} = \sum_{j \wedge i} M_{i \to j} \quad \text{if } M_{i \to j} > 0,$$

then **for all tracer particles** in cell i:

$$p_{i,\text{out}} = \frac{M_{i,\text{out}}}{M_i}, \quad \# \text{ Proba. of going out of } i \tag{2}$$

$$p_{i \to j} = \frac{M_{i \to j}}{M_{i,\text{out}}}, \quad \# \text{ Proba. of going from } i \text{ to } j \tag{3}$$

following *S.Genel et al, 13*

Algorithm

Algorithm for cell i mass M_i, neighbors j:

1. Compute outgoing mass $M_{i,\text{out}}$ and $M_{i \to j}$.

Algorithm

Algorithm for cell i mass M_i, neighbors j:

1. Compute outgoing mass $M_{i,\text{out}}$ and $M_{i\to j}$.
2. Compute outgoing proba $p_{i,\text{out}} = M_{i,\text{out}}/M_i$ and $p_{i\to j} = M_{i\to j}/M_{i,\text{out}}$.

Algorithm

Algorithm for cell i mass M_i, neighbors j:

1. Compute outgoing mass $M_{i,\text{out}}$ and $M_{i\to j}$.
2. Compute outgoing proba $p_{i,\text{out}} = M_{i,\text{out}}/M_i$ and $p_{i\to j} = M_{i\to j}/M_{i,\text{out}}$.
3. For each particle:
   3.1 Draw random number r_j.
   3.2 Select particles $r_j < p_{i,\text{out}}$.

Algorithm

Algorithm for cell i mass M_i, neighbors j:

1. Compute outgoing mass $M_{i,\text{out}}$ and $M_{i\to j}$.
2. Compute outgoing proba $p_{i,\text{out}} = M_{i,\text{out}}/M_i$ and $p_{i\to j} = M_{i\to j}/M_{i,\text{out}}$.
3. For each particle:
   - 3.1 Draw random number r_j.
   - 3.2 Select particles $r_j < p_{i,\text{out}}$.
4. For each selected particle:
   - 4.1 Draw random number r'_j.
   - 4.2 If $r'_j < p_{i\to j}$, move to cell j.

Algorithm

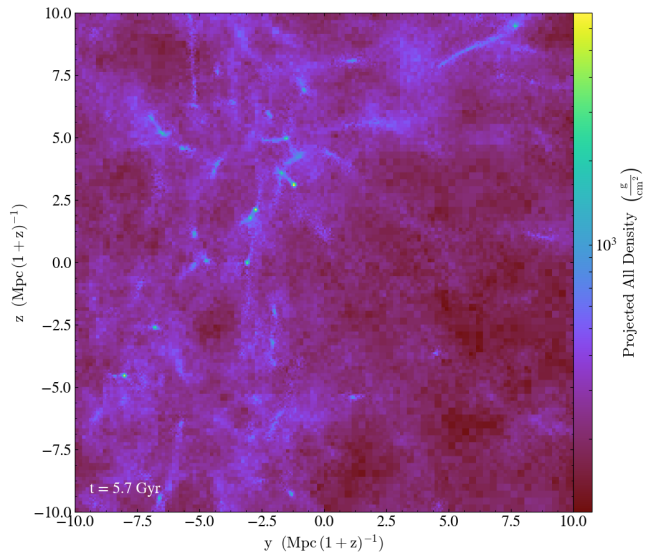
Algorithm for cell i mass M_i, neighbors j:

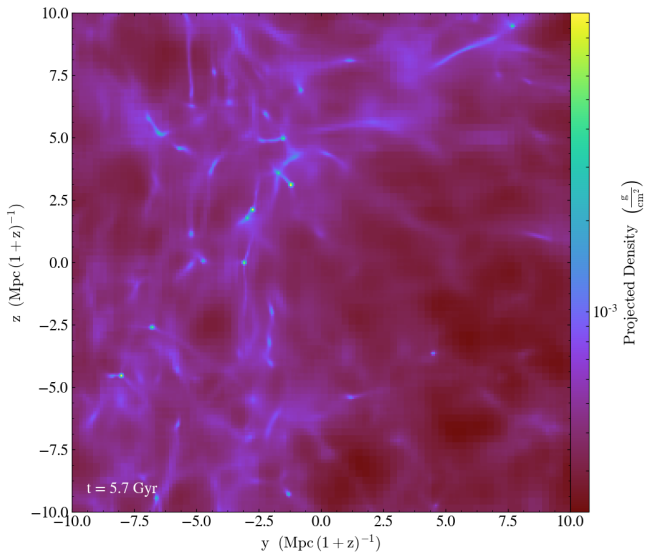
1. Compute outgoing mass $M_{i,\text{out}}$ and $M_{i\to j}$.
2. Compute outgoing proba $p_{i,\text{out}} = M_{i,\text{out}}/M_i$ and $p_{i\to j} = M_{i\to j}/M_{i,\text{out}}$.
3. For each particle:
   - 3.1 Draw random number r_j.
   - 3.2 Select particles $r_j < p_{i,\text{out}}$.
4. For each selected particle:
   - 4.1 Draw random number r'_j.
   - 4.2 If $r'_j < p_{i\to j}$, move to cell j.

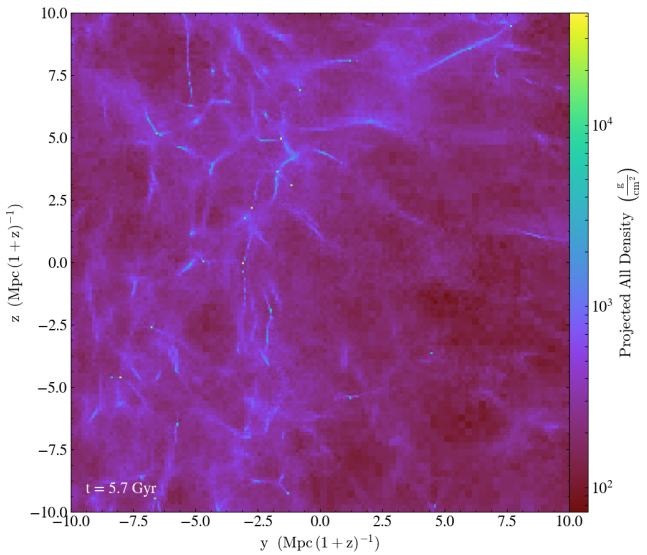
Small flux limit: $N_{\text{moved}} \sim$ Poisson distribution(p)

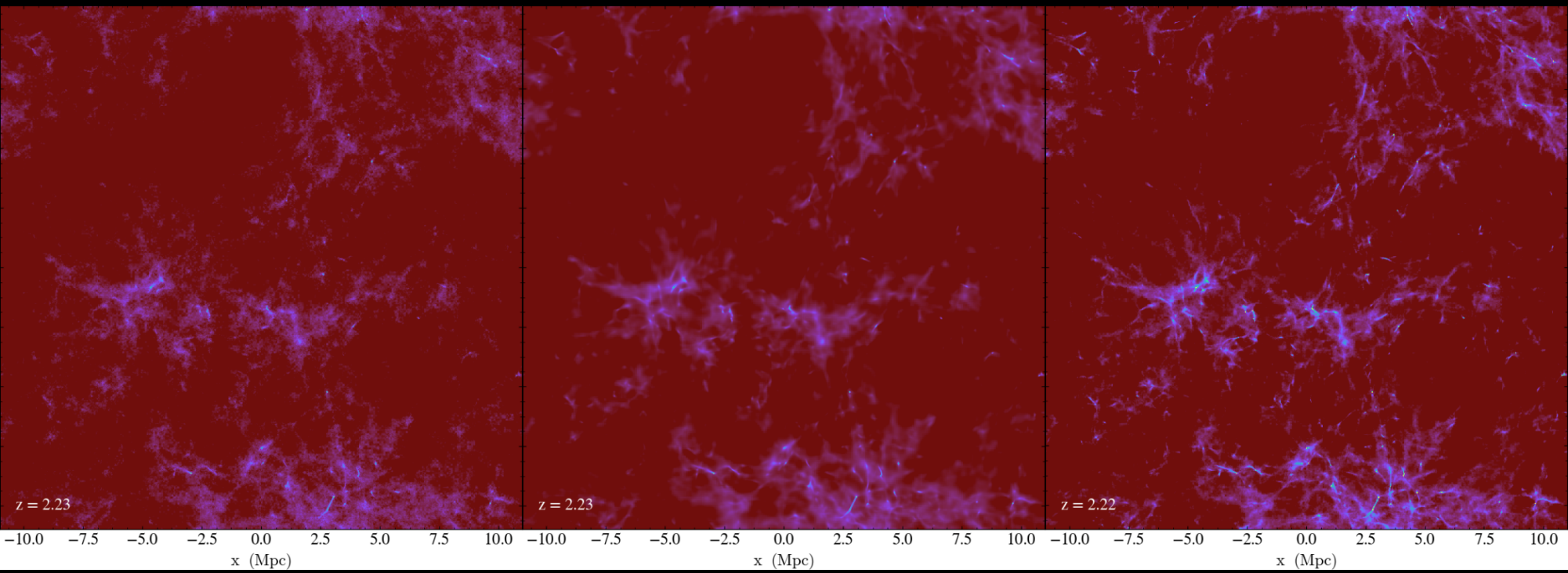
Is it working?

Cosmo

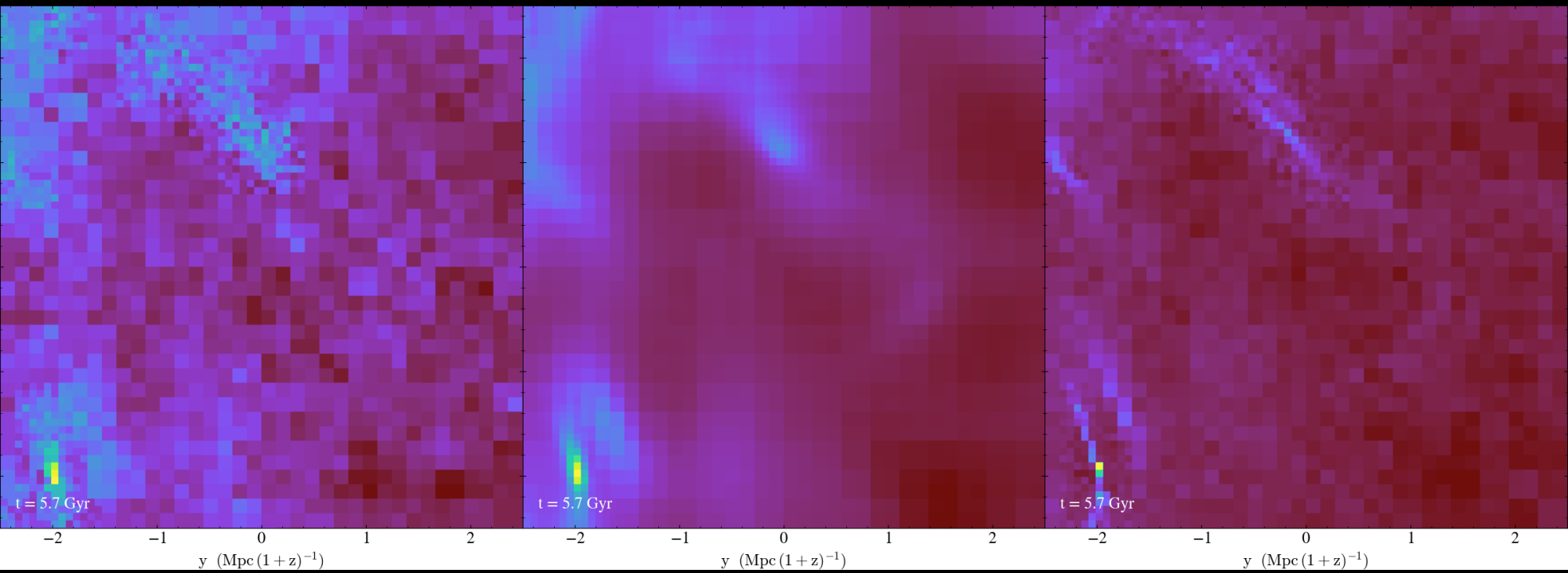

t = 5.7 Gyr
z (Mpc (1 + z)^−1)
y (Mpc (1 + z)^−1)
Projected All Density (g/cm^2)


t = 5.7 Gyr
z (Mpc (1 + z)^−1)
y (Mpc (1 + z)^−1)
Projected Density (g/cm^2)

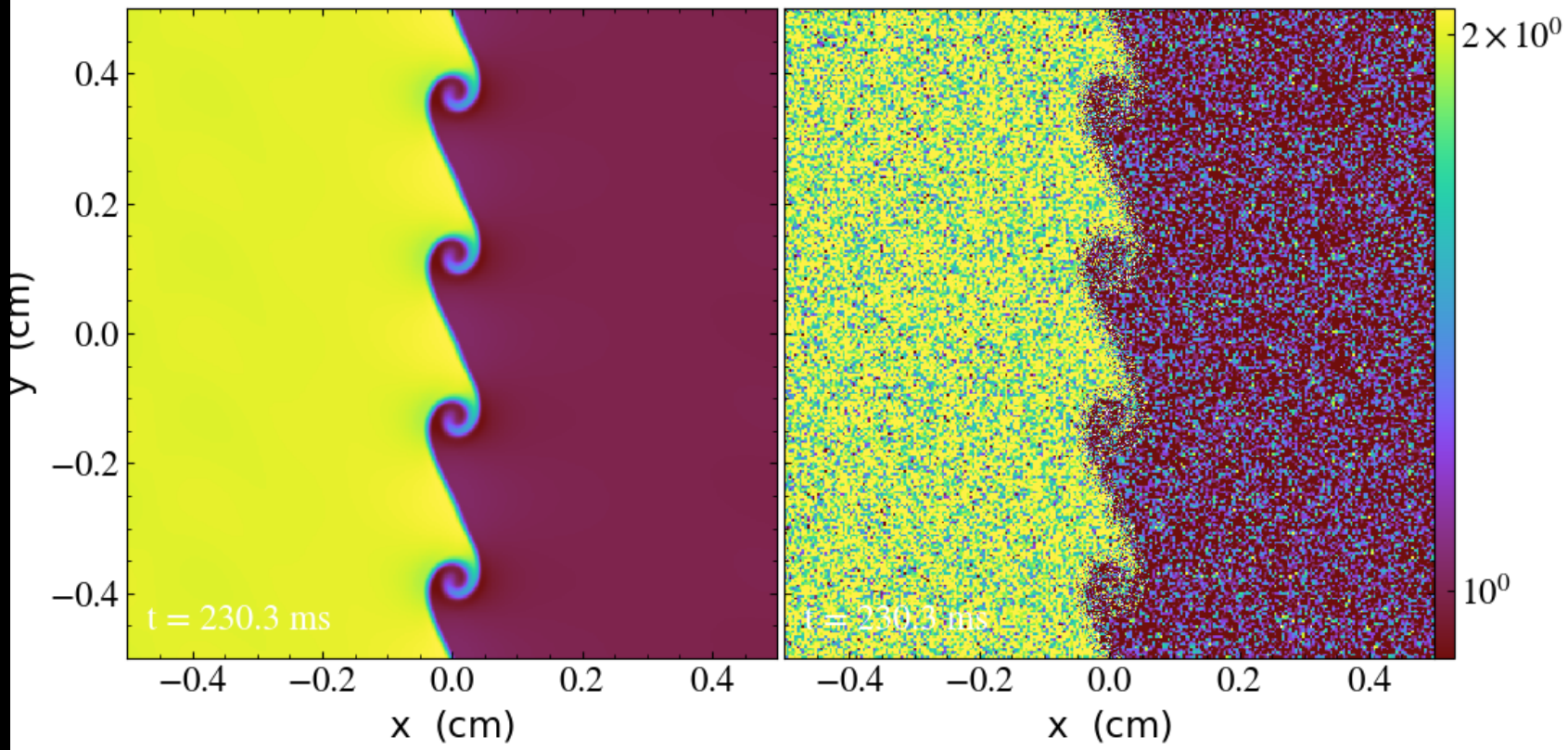

t = 5.7 Gyr
z (Mpc (1 + z)^−1)
y (Mpc (1 + z)^−1)
Projected All Density (g/cm^2)

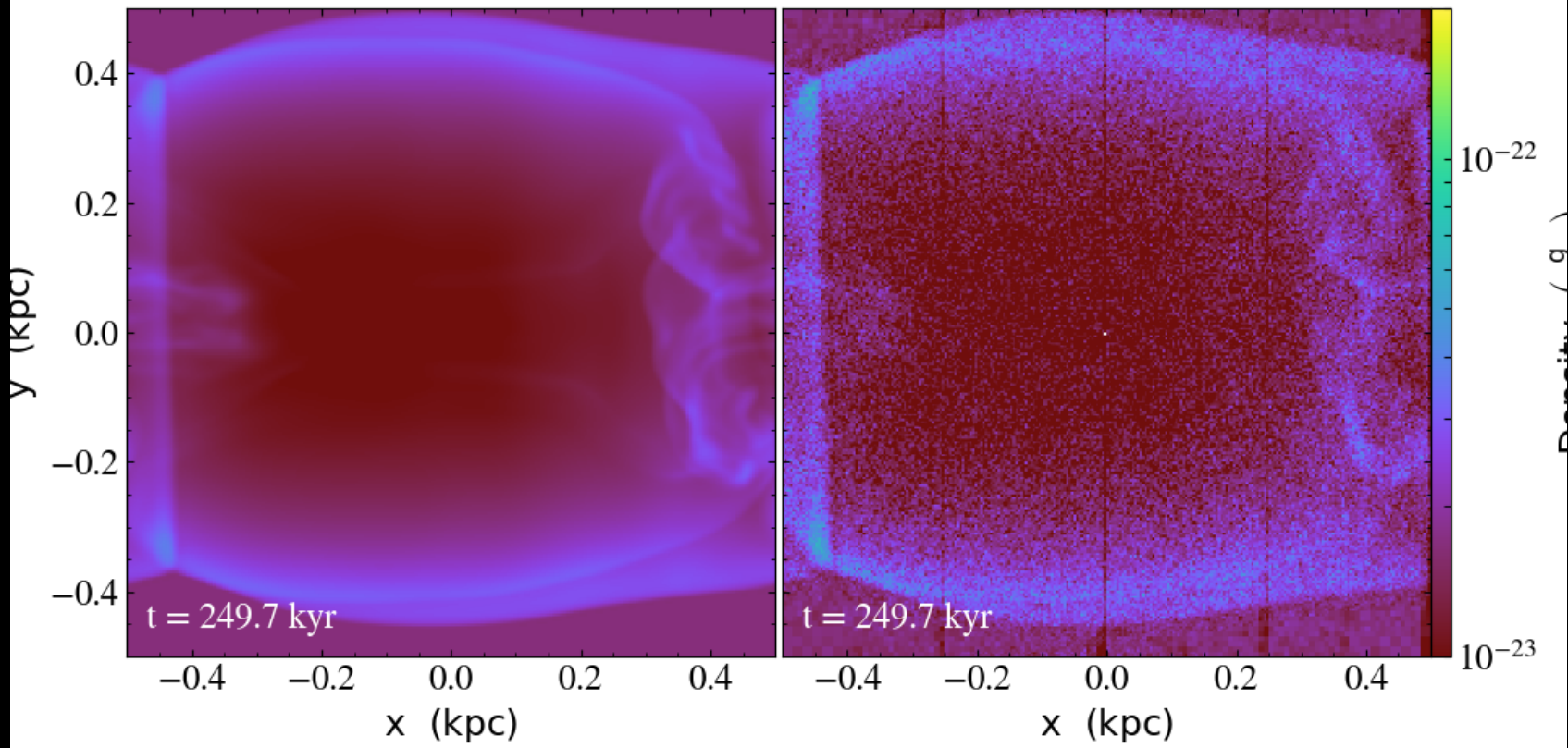


Cosmological Simulation, DM+hydro, left to right: MC tracer, gas and velocity tracers



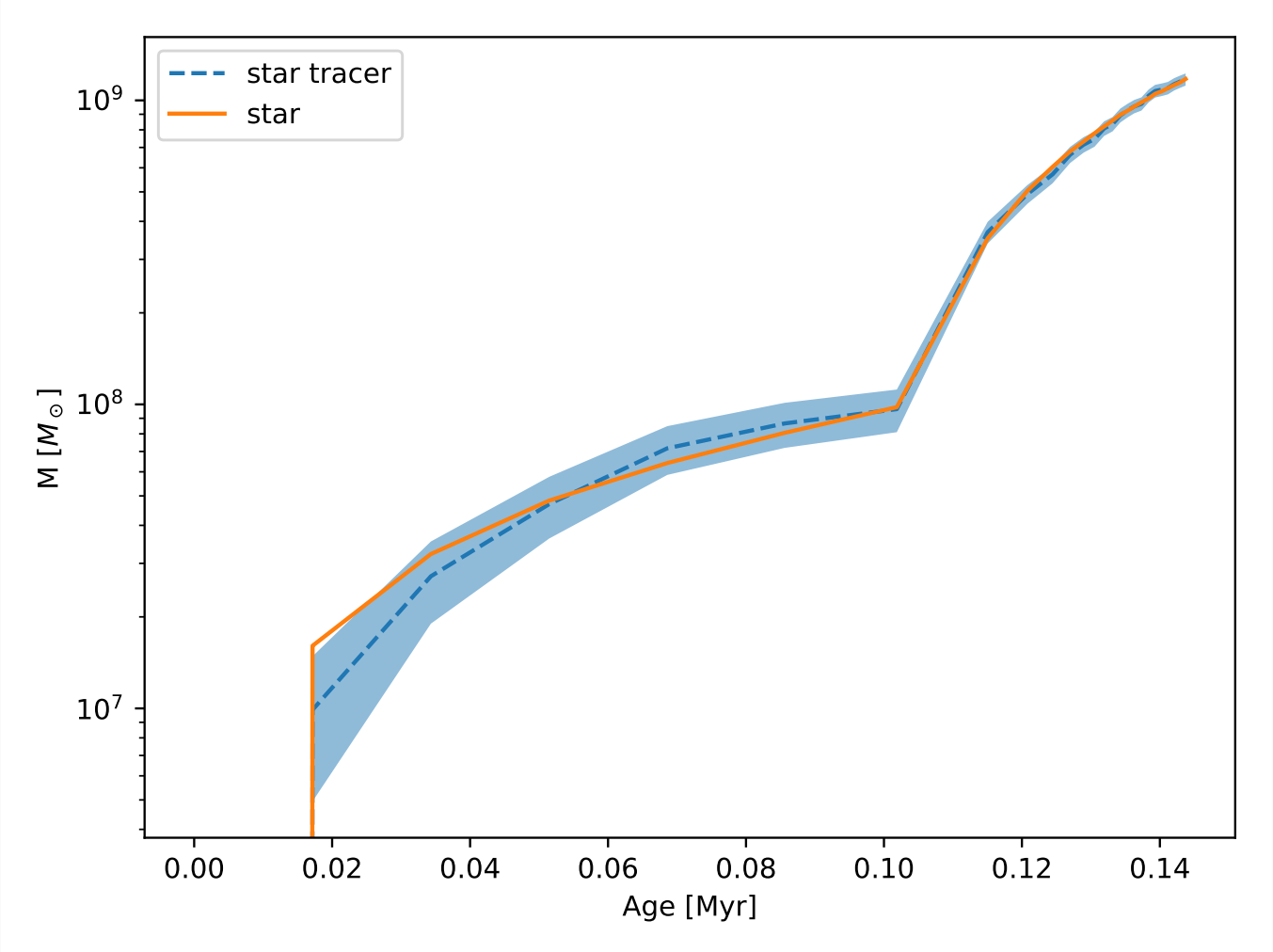
Cosmological Simulation, DM+hydro, left to right: MC tracer, gas and velocity tracers


0.4
0.2
0.0
−0.2
−0.4
y (cm)
t = 230.3 ms
t = 230.3 ms
−0.4
−0.2
0.0
0.2
0.4
x (cm)
x (cm)
2×10^0
10^0


0.4
0.2
0.0
−0.2
−0.4
y (kpc)
t = 249.7 kyr
t = 249.7 kyr
−0.4
−0.2
0.0
0.2
0.4
x (kpc)
x (kpc)
10^{-22}
10^{-23}

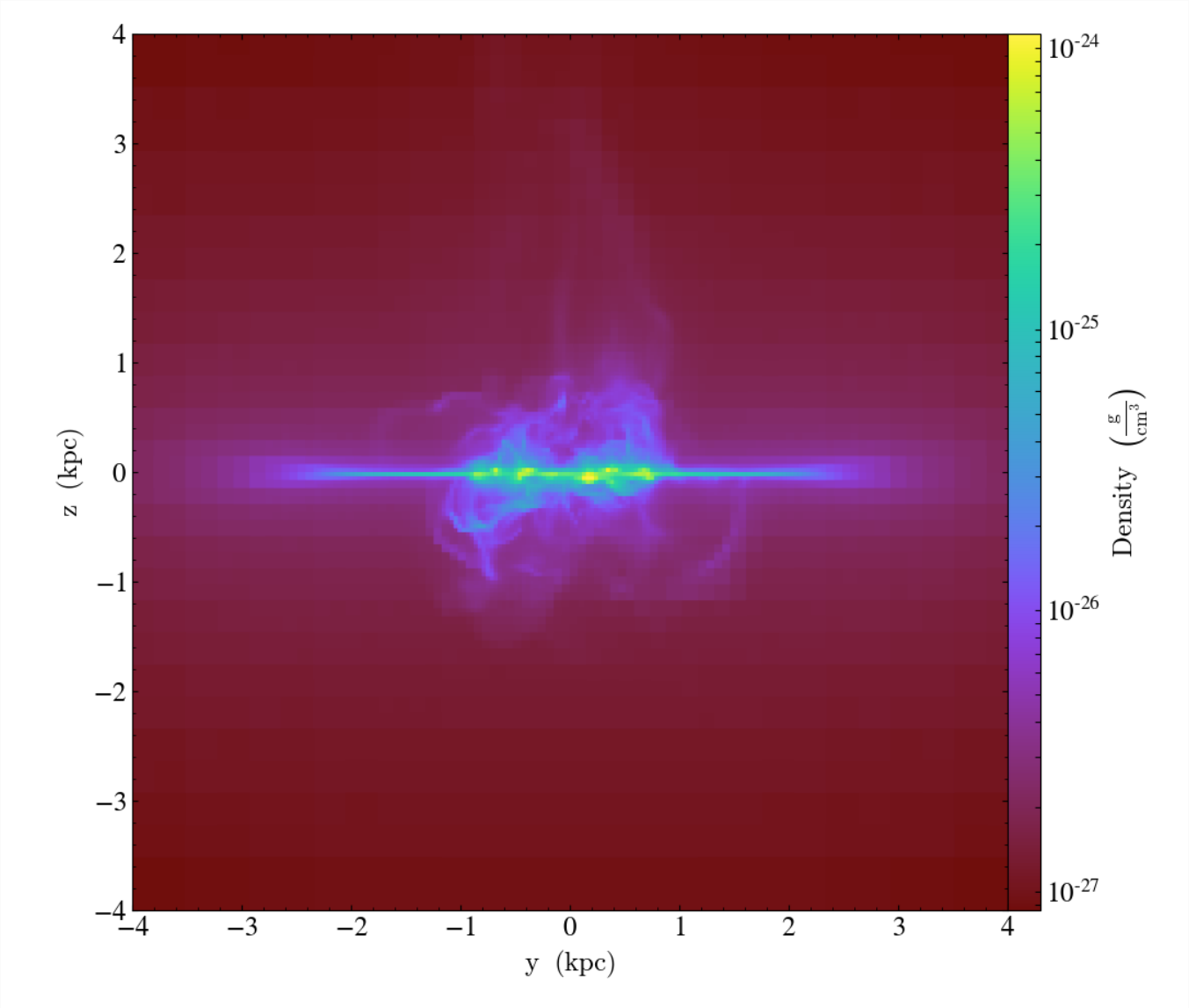
Star formation

- SF recipy: mechanical feedback
- homogeneous density



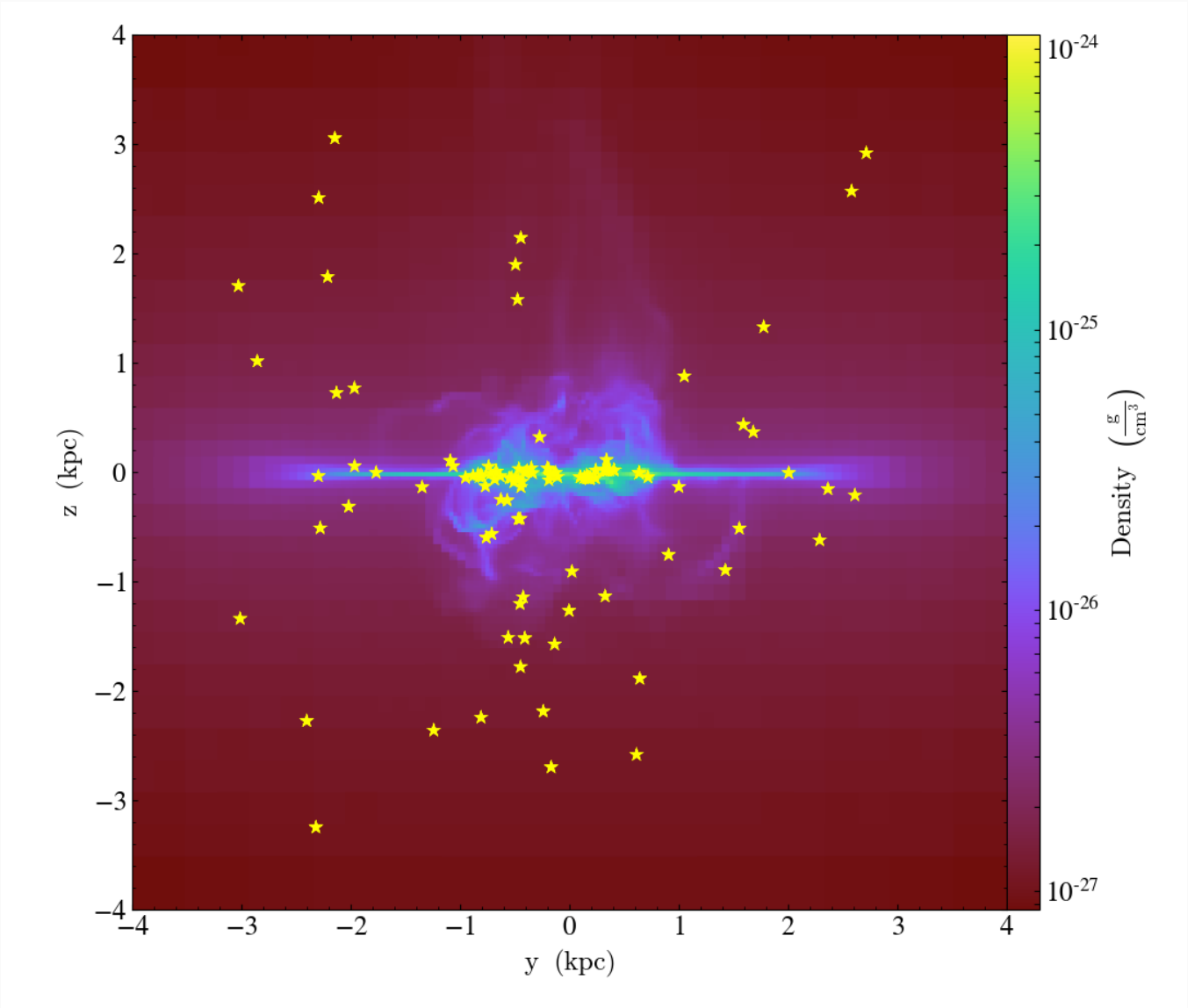
Small galaxy

Gas only



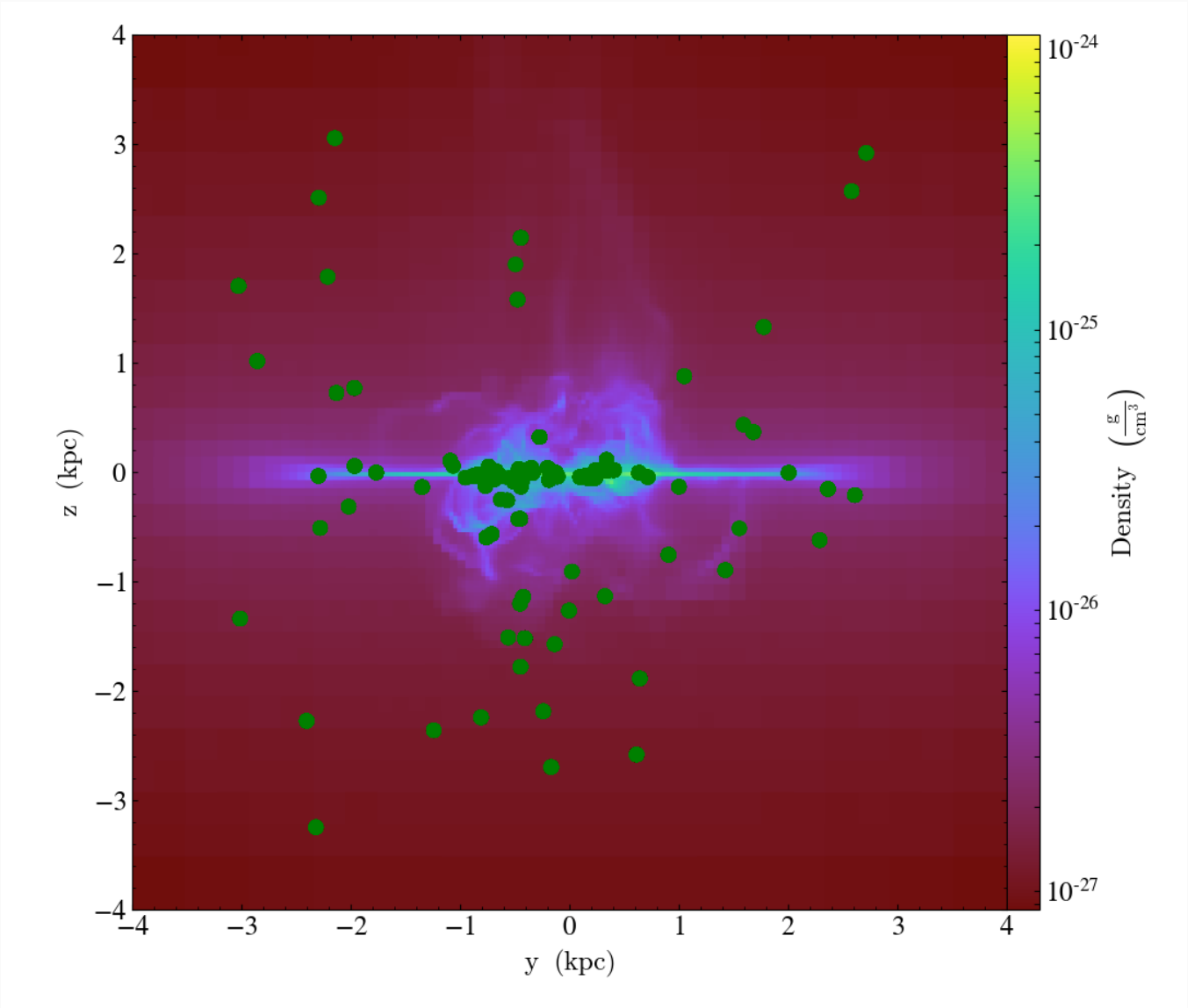
Small galaxy

Gas and stars



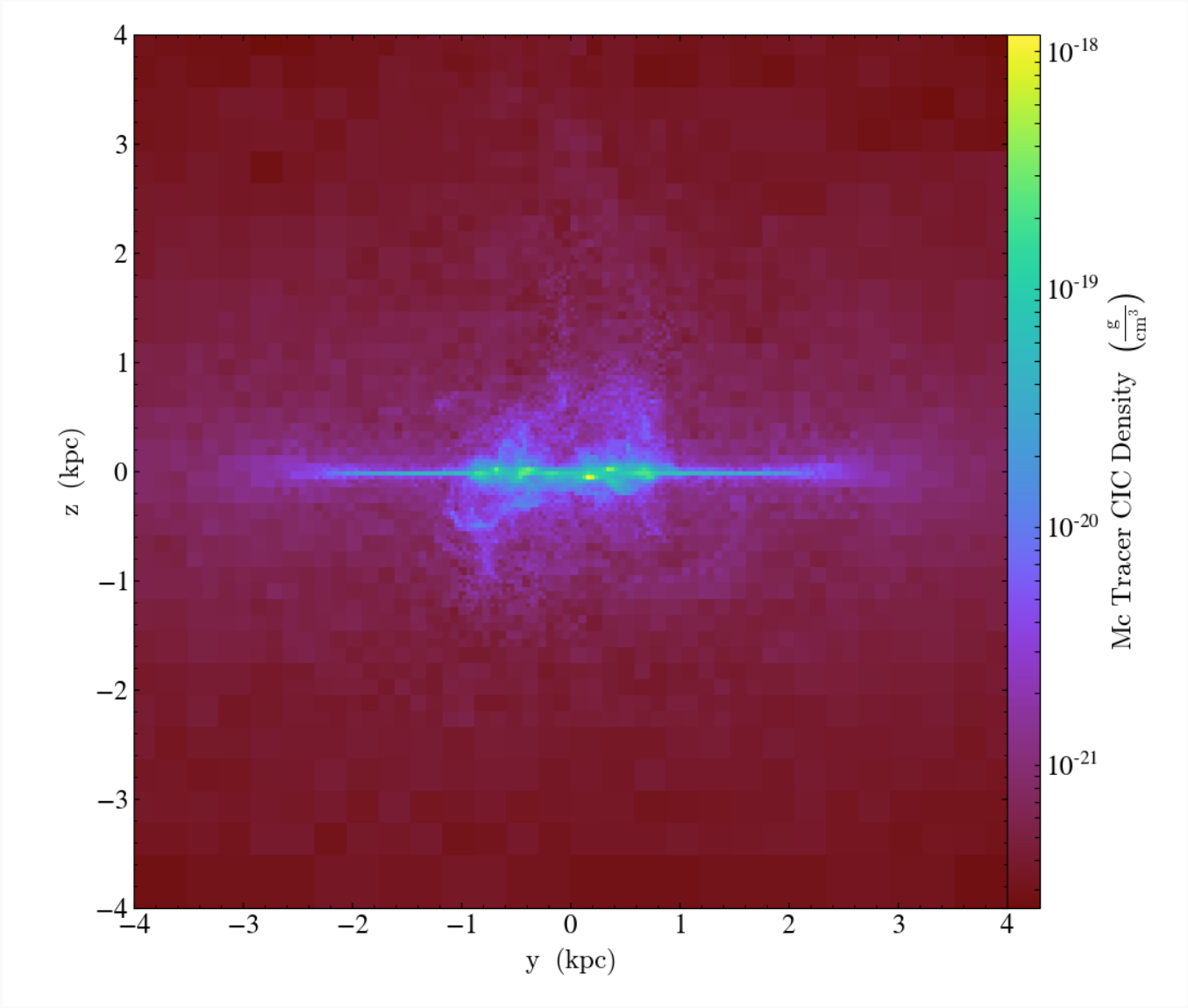
Small galaxy

Gas and star tracers



Small galaxy

MC tracers



Discussion

Room for improvement

TODO & wishlist:

- Get AGN feedback done (WIP).
- Other SN feedback.
- Quantify diffusion (esp. high flux limit)
- Explore other MC algorithms.

Room for improvement

TODO & wishlist:

- Get AGN feedback done (WIP).
- Other SN feedback.
- Quantify diffusion (esp. high flux limit)
- Explore other MC algorithms.
- Get users!

"SAV" at corentin.cadiou@iap.fr

Thank you!

Questions?

Advertisement

YT!

yt now supports:

- sinks
- RT
- custom particle + fluid fields
- BSD license (permissive)

http://yt-project.com