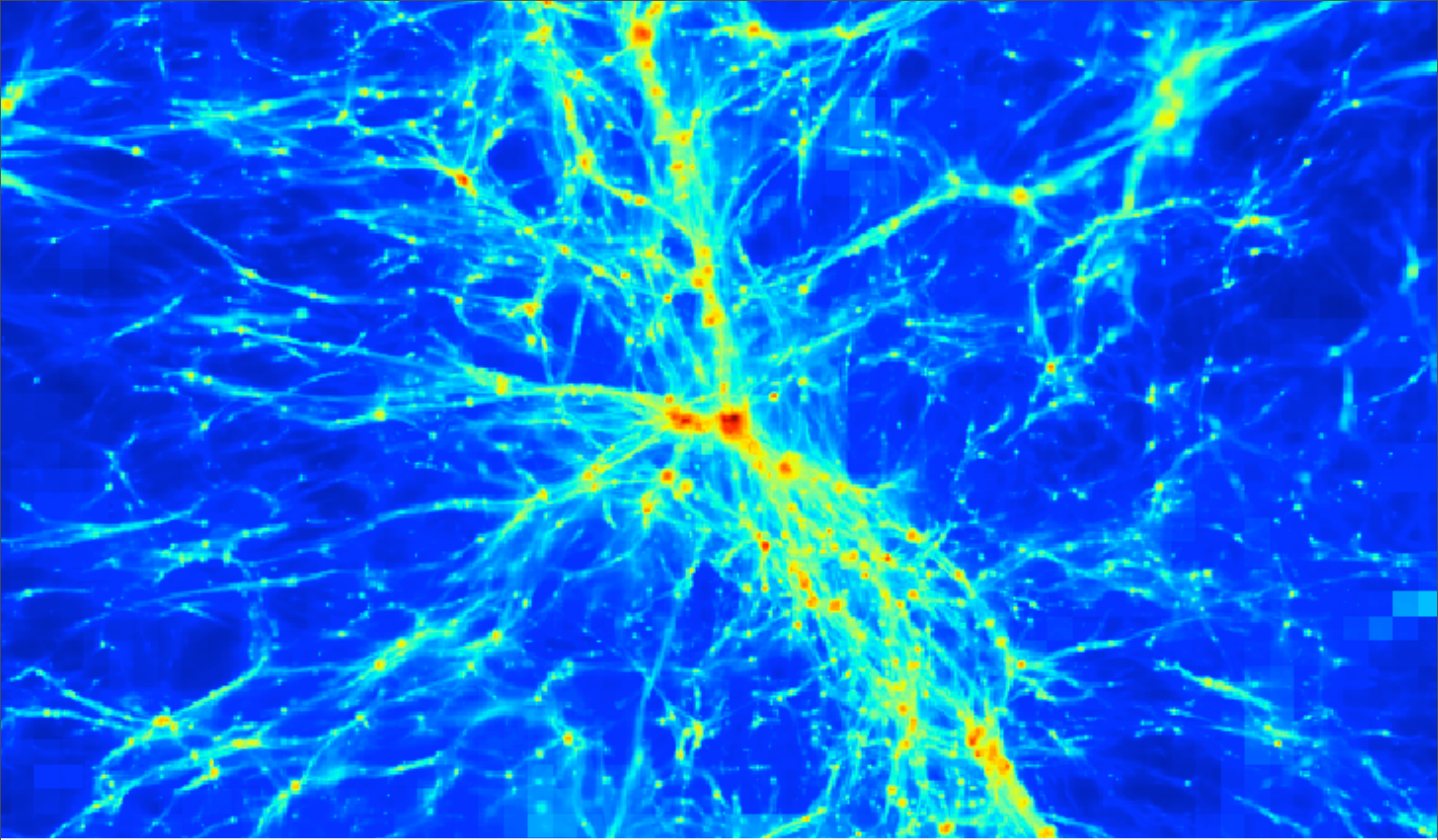


Advanced Cosmological Simulations

John Wise (Georgia Tech)

Enzo Workshop

北海道大学 – 19 Oct 2013

Outline

- We will consider additional physics in Thursday's AMR (no nested grids) cosmology simulation.
- Refresher on the hydro + N-body run performed on Thursday with an adiabatic equation of state.
- Adding radiative cooling and a ultraviolet background
- Adding star formation and feedback (only supernovae)
- Adding radiative feedback from stars

Hydrodynamics with an Adiabatic Equation of State

- On Thursday, everyone ran a 32^3 AMR simulation with hydrodynamics and N-body dynamics.

- Go to the run directory. For example

  - `cd ~/sapporo_cosmo/sapporo_cosmo_nbody`

- Let's analyze it.

  - Projections

  - Phase plots in density and temperature

Hydrodynamics with an Adiabatic Equation of State

- `source ~/yt-x86_64/bin/activate`
- We will be using the yt script, `anyl.py`

```
ts = TimeSeriesData.from_filenames(fname)
for pf in ts.piter():
    test_pic_name = "pics/%s_Projection_x_%s_Density.png" % (pf, fields[0])
    if os.path.exists(test_pic_name): continue
    for dim in 'xyz':
        p = ProjectionPlot(pf, dim, fields, weight_field="Density", center=[0.5]*3)
        for f in fields:
            if f in zlim.keys():
                p.set_zlim(f, zlim[f][0], zlim[f][1])
            if f in cmap.keys():
                p.set_cmap(f, cmap[f])
        p.annotate_text((1.1, 1.08), 'z = %.2f' % pf.current_redshift,
                        text_args={'ha': 'right', 'va': 'top'})
        p.save("pics/%s" % pf)
        del p
    pc = PlotCollection(pf)
    ad = pf.h.all_data()
```

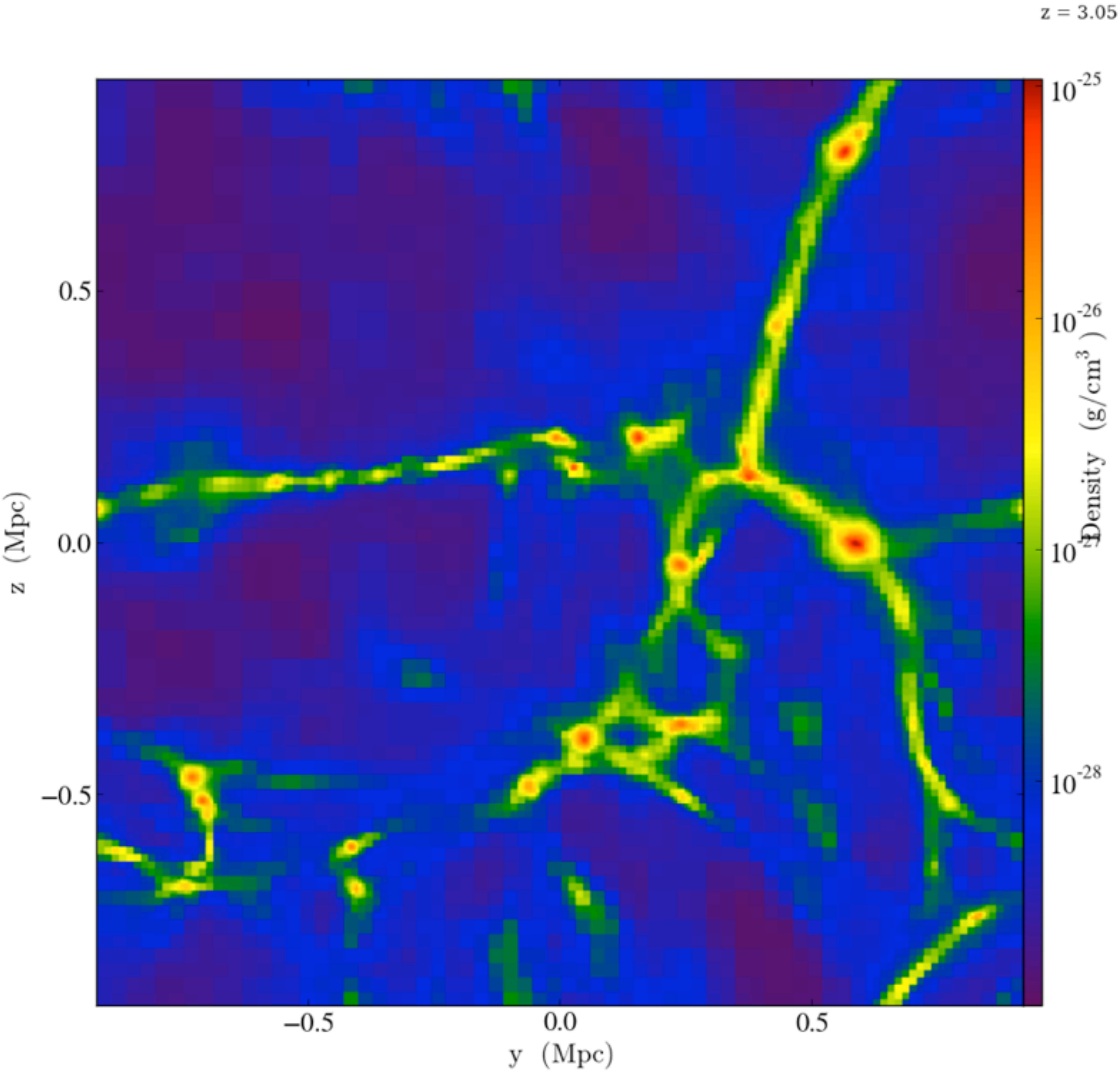
Hydrodynamics with an Adiabatic Equation of State

- source ~/yt-x86_64/bin/activate

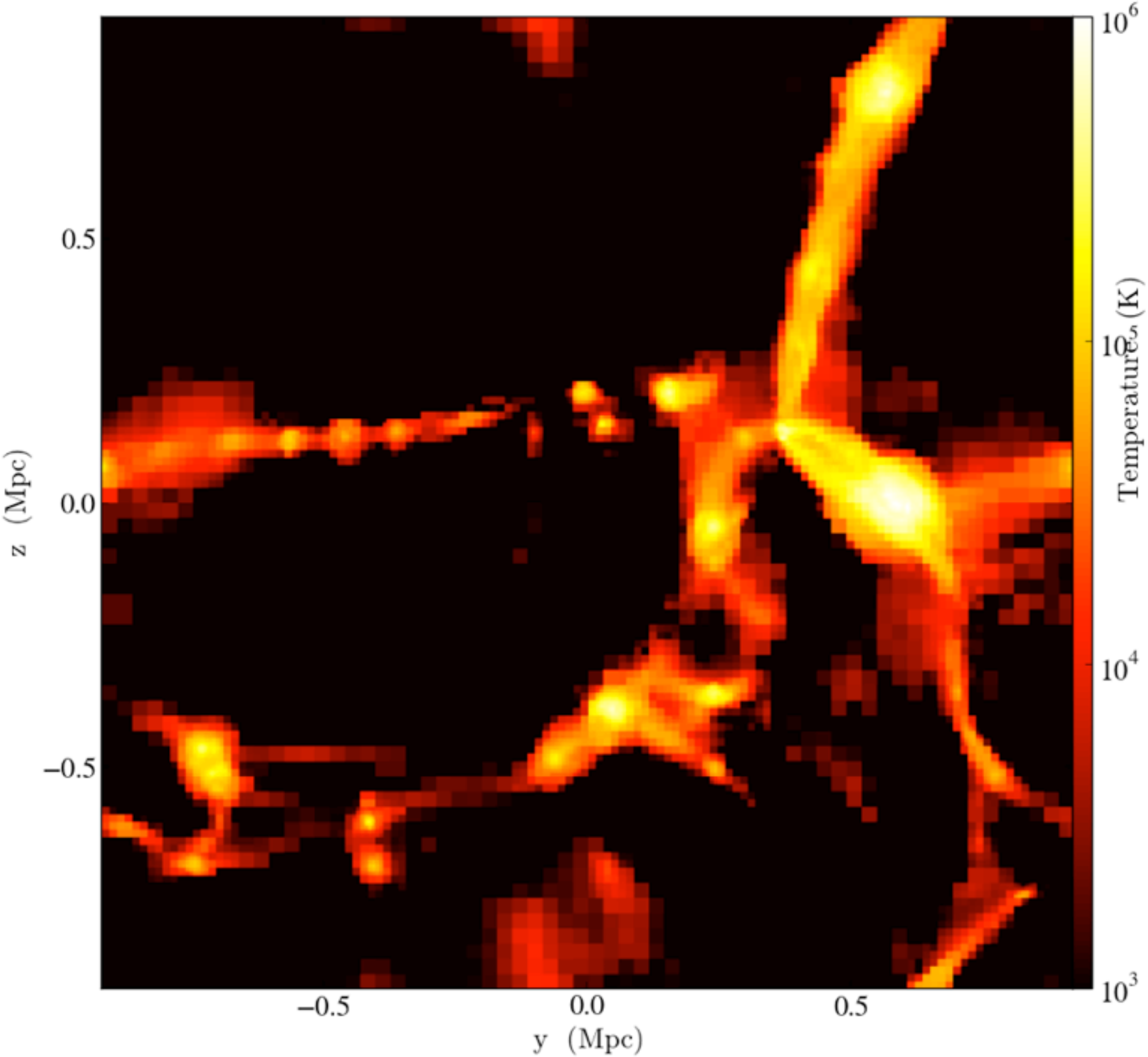
```
ts = TimeSeriesData.from_filenames(fname)
for pf in ts.piter():
    test_pic_name = "pics/%s_Projection_x_%s_Density.png" % (pf, fields[0])
    if os.path.exists(test_pic_name): continue
    for dim in 'xyz':
        p = ProjectionPlot(pf, dim, fields, weight_field="Density", center=[0.5]*3)
        for f in fields:
            if f in zlim.keys():
                p.set_zlim(f, zlim[f][0], zlim[f][1])
            if f in cmap.keys():
                p.set_cmap(f, cmap[f])
        p.annotate_text((1.1, 1.08), 'z = %.2f' % pf.current_redshift,
                        text_args={'ha': 'right', 'va': 'top'})
        p.save("pics/%s" % pf)
        del p
    pc = PlotCollection(pf)
    ad = pf.h.all_data()
    pc.add_phase_object(ad, ['Density', 'Temperature', 'CellMassMsun'], weight=None)
    pc.save("pics/%s" % pf)
    del pc
    pf.h.clear_all_data()
```

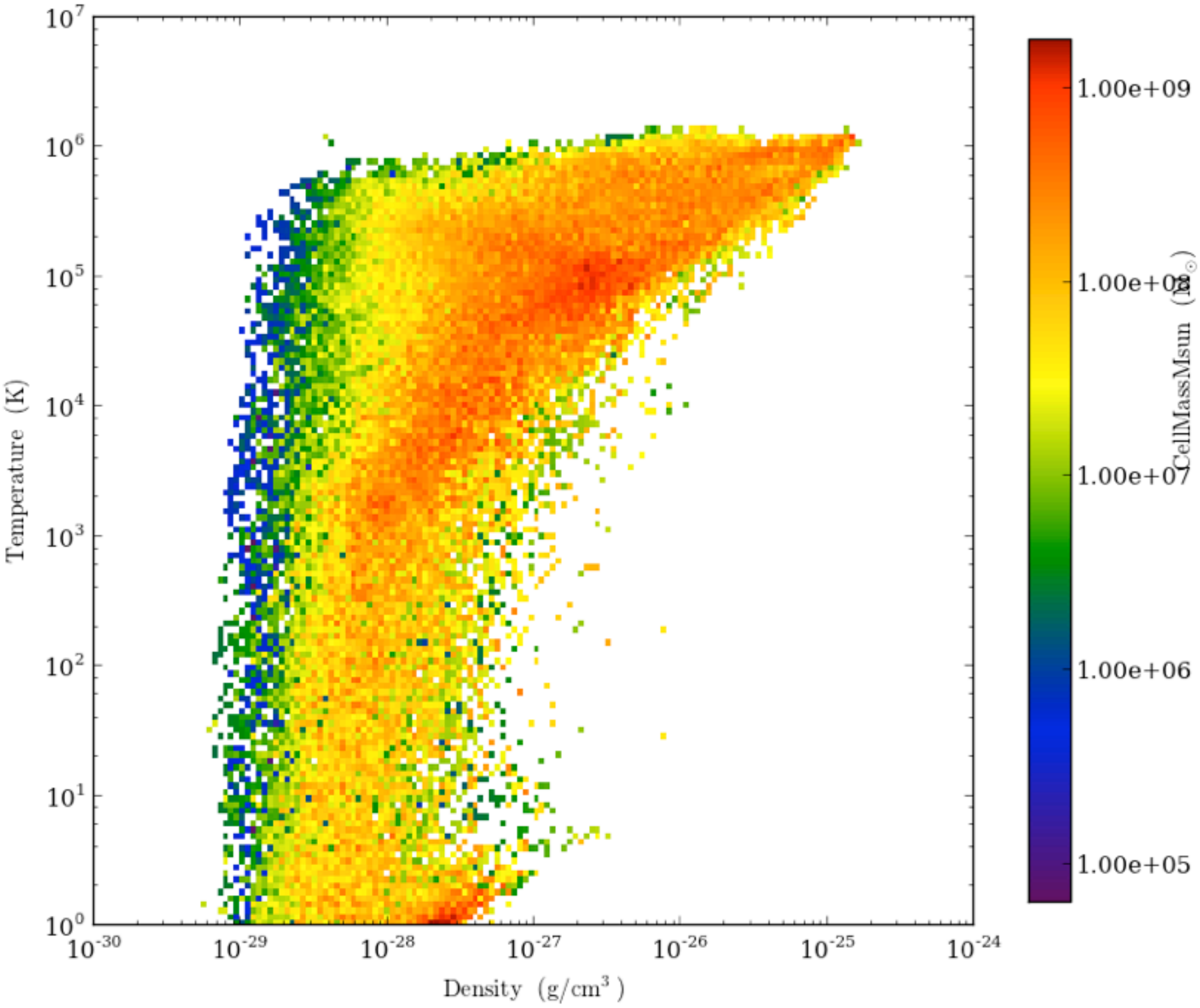
Hydrodynamics with an Adiabatic Equation of State

- Run the yt script for the last dataset. For example,
  - `python ../anyl.py DD0049/DD0049`
- Creates the projections and phase plots and places them in `pics/`
- Note: I have written the script so that if **no argument is given**, all datasets are analyzed.
  - `python ../anyl.py`


z = 3.05
z (Mpc)
0.5
0.0
−0.5
−0.5
0.0
0.5
y (Mpc)
Density (g/cm^3)
10^{-25}
10^{-26}
10^{-28}

z = 3.05





Temperature (K)
Density (g/cm³)
CellMassMsun (M☉)
1.00e+09
1.00e+08
1.00e+07
1.00e+06
1.00e+05

Radiative Cooling and Ultraviolet Background

- Now we can add more physics to the adiabatic simulation.

`~guest009/cooling/cooling.enzo`

- Add radiative cooling and non-equilibrium chemistry.

```
RadiativeCooling          = 1
MultiSpecies              = 1
```

- Add optically-thin ultraviolet background

```
#
# UV background parameters
#
RadiationFieldType = 15   // 0 = None, 15 = Haardt-Madau 2012
```

```
RadiationRedshiftOn           = 7.000000
RadiationRedshiftOff          = 0.000000
RadiationRedshiftFullOn       = 6.000000
```

Cosmology Simulations

```
RadiationFieldRedshift        = 0.000000
```

Non-cosmology Simulations

Different Ultraviolet Backgrounds

Radiation Parameters

Background Radiation Parameters

`RadiationFieldType` (external)

This integer parameter specifies the type of radiation field that is to be used. Except for `RadiationFieldType` = 9, which should be used with `MultiSpecies` = 2, UV backgrounds can currently only be used with `MultiSpecies` = 1 (i.e. no molecular H support). The following values are used. Default: 0

```
1  - Haardt & Madau spectrum with q_alpha = 1.5
2  - Haardt & Madau spectrum with q_alpha = 1.8
3  - Modified Haardt & Madau spectrum to match observations
     (Kirkman & Tytler 2005).
4  - Haardt & Madau spectrum with q_alpha = 1.5 supplemented with an X-ray Compton heating
     background from Madau & Efstathiou (see astro-ph/9902080)
9  - Constant molecular H2 photo-dissociation rate
10 - Internally computed radiation field using the algorithm of Cen & Ostriker
11 - Same as previous, but with very, very simple optical shielding fudge
12 - Haardt & Madau spectrum with q_alpha = 1.57
15 - Haardt & Madau 2012.  See Table 3 in '2012ApJ...746..125H <http://adsabs.harvard.edu/abs/201
```

This integer parameter specifies the type of radiation field that is to be used. Except for `RadiationFieldType` = 9, which should be used with `MultiSpecies` = 2, UV backgrounds can currently only be used with `MultiSpecies` = 1 (i.e. no molecular H support). The following values are used. Default: 0

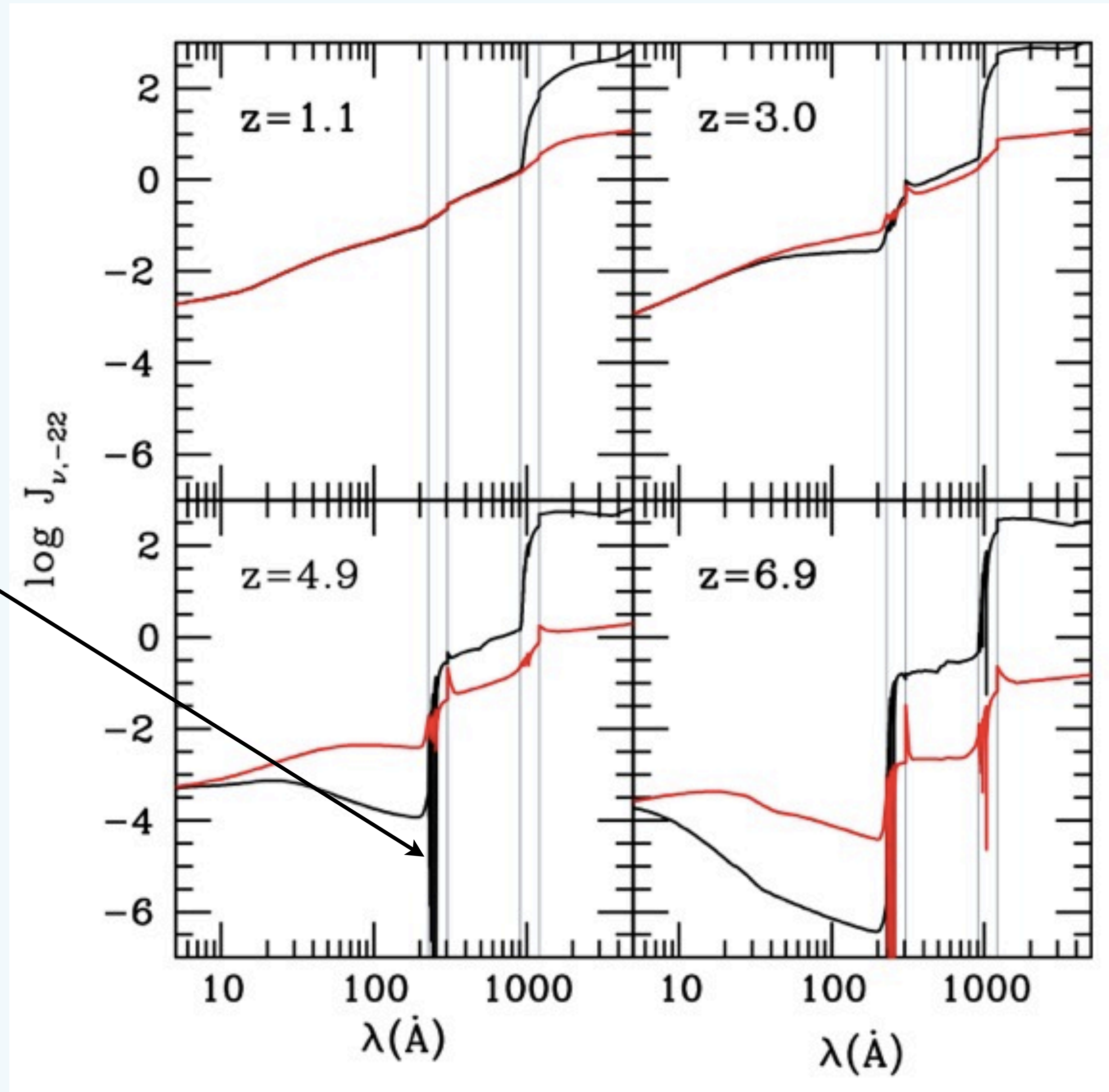
```
1  - Haardt & Madau spectrum with q_alpha = 1.5
2  - Haardt & Madau spectrum with q_alpha = 1.8
3  - Modified Haardt & Madau spectrum to match observations
     (Kirkman & Tytler 2005).
4  - Haardt & Madau spectrum with q_alpha = 1.5 supplemented with an X-ray Compton heating
     background from Madau & Efstathiou (see astro-ph/9902080)
9  - Constant molecular H2 photo-dissociation rate
10 - Internally computed radiation field using the algorithm of Cen & Ostriker
11 - Same as previous, but with very, very simple optical shielding fudge
12 - Haardt & Madau spectrum with q_alpha = 1.57
15 - Haardt & Madau 2012.  See Table 3 in '2012ApJ...746..125H <http://adsabs.harvard.edu/abs/201
```

Ultraviolet Background (Haardt & Madau 2012)

QSO only

QSO + Galaxies

Includes Lyman series lines



Ultraviolet Background (Haardt & Madau 2012)

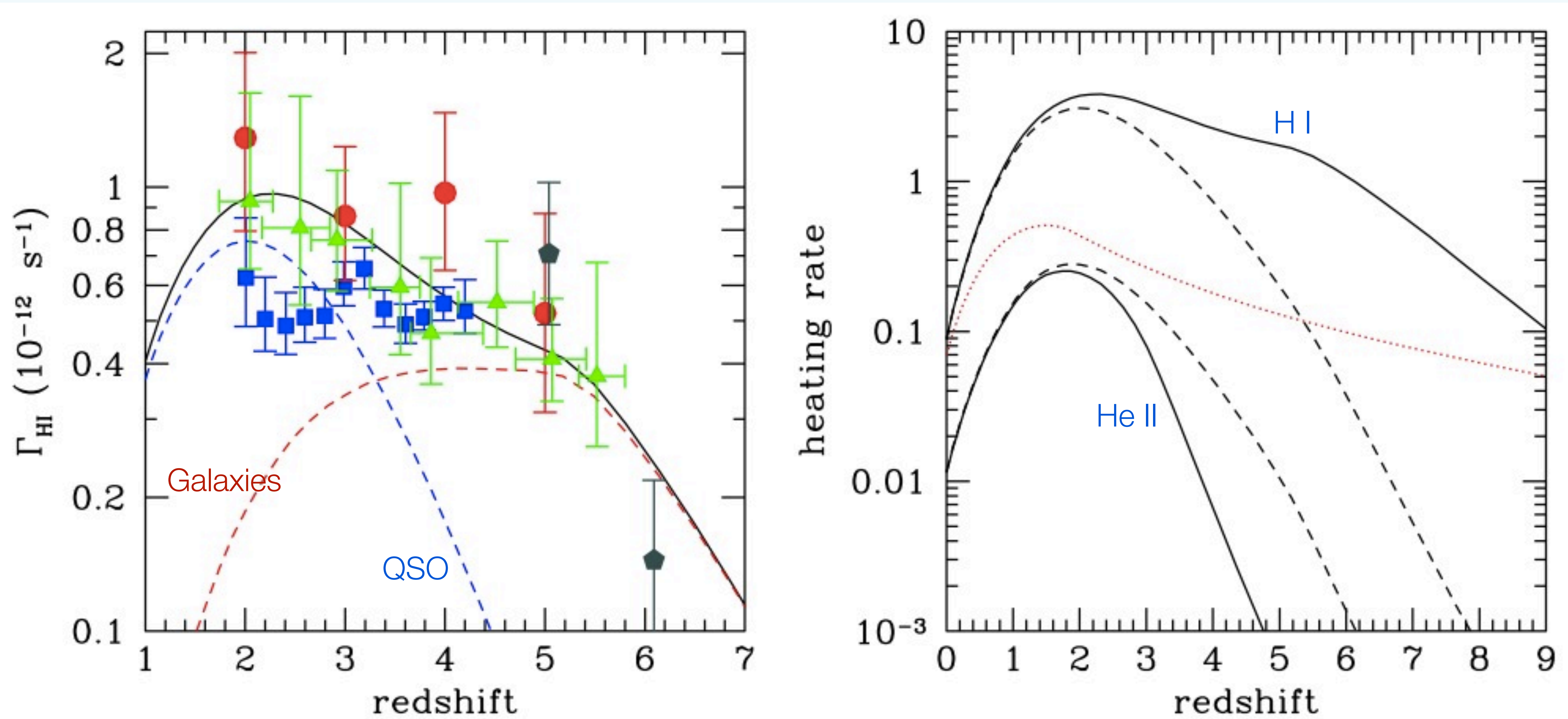


Photo-ionization Rate

Photo-heating Rate

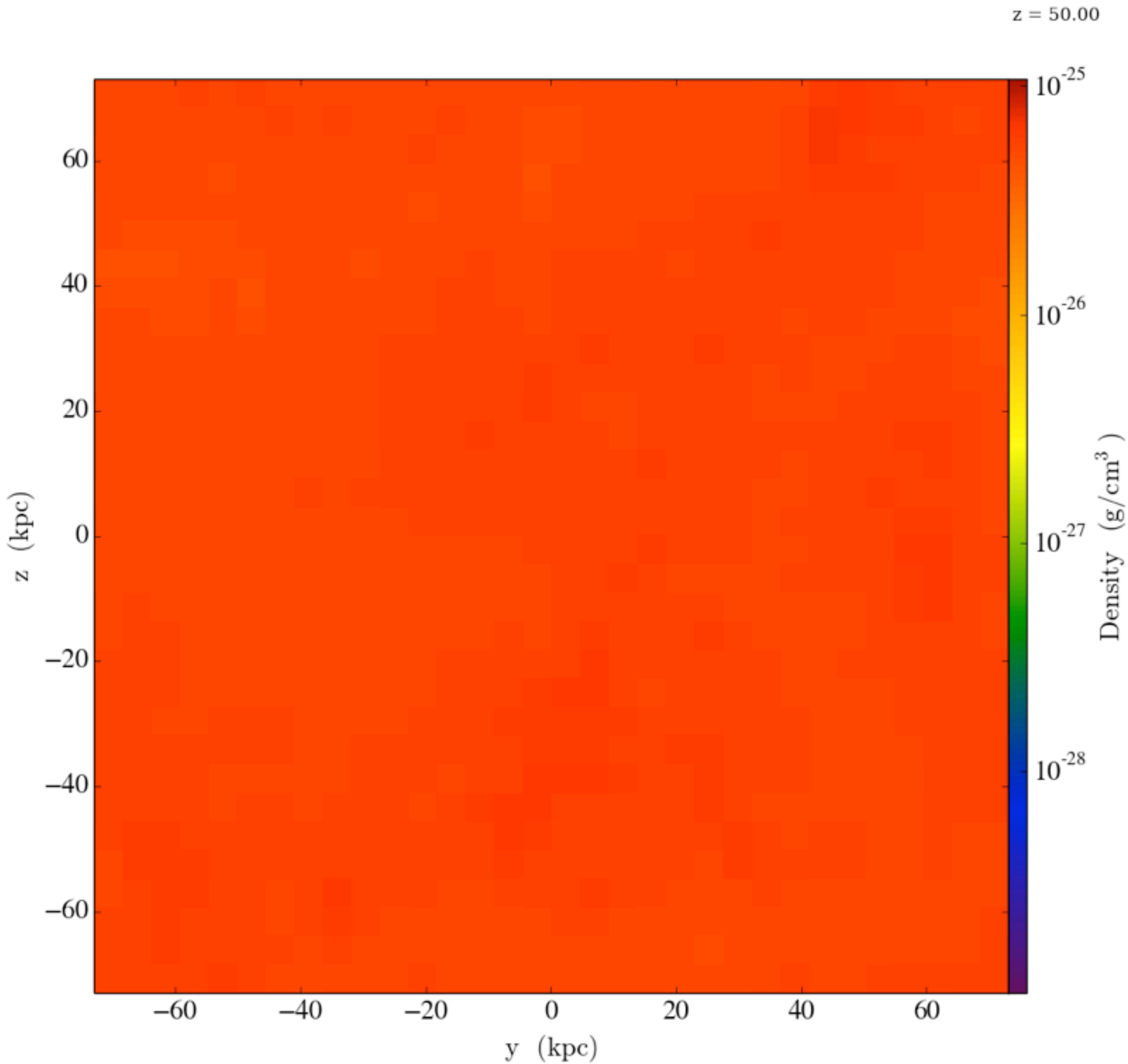
Radiative Cooling and Ultraviolet Background

- This simulation takes 50 minutes to run on my laptop with 2 cores.
- Requires the tabulated UV background in the run directory.
    - ~/enzo-stable/input/hm12_photorates.dat
- If you want to run the simulation later, you can copy the parameter file and initial conditions files from the adiabatic simulation

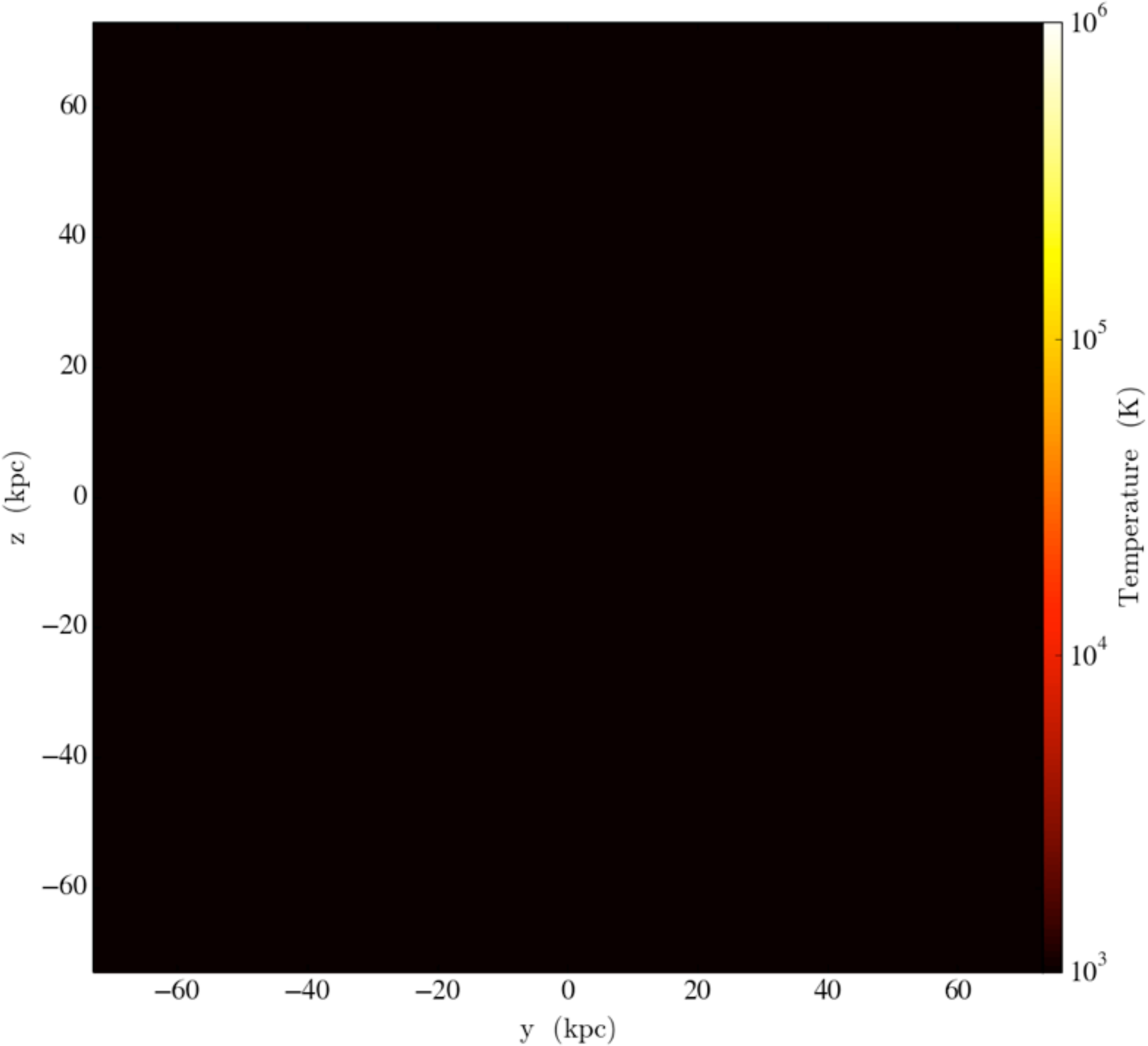
```
cd ~/sapporo_cosmo
mkdir cooling
cp sapporo_cosmo_nbody/Grid* sapporo_cosmo_nbody/Particle* cooling
cd cooling
cp ~guest009/cooling.enzo .
cp <where enzo is>/enzo.exe .
./enzo.exe -d cooling.enzo
```

Radiative Cooling and Ultraviolet Background

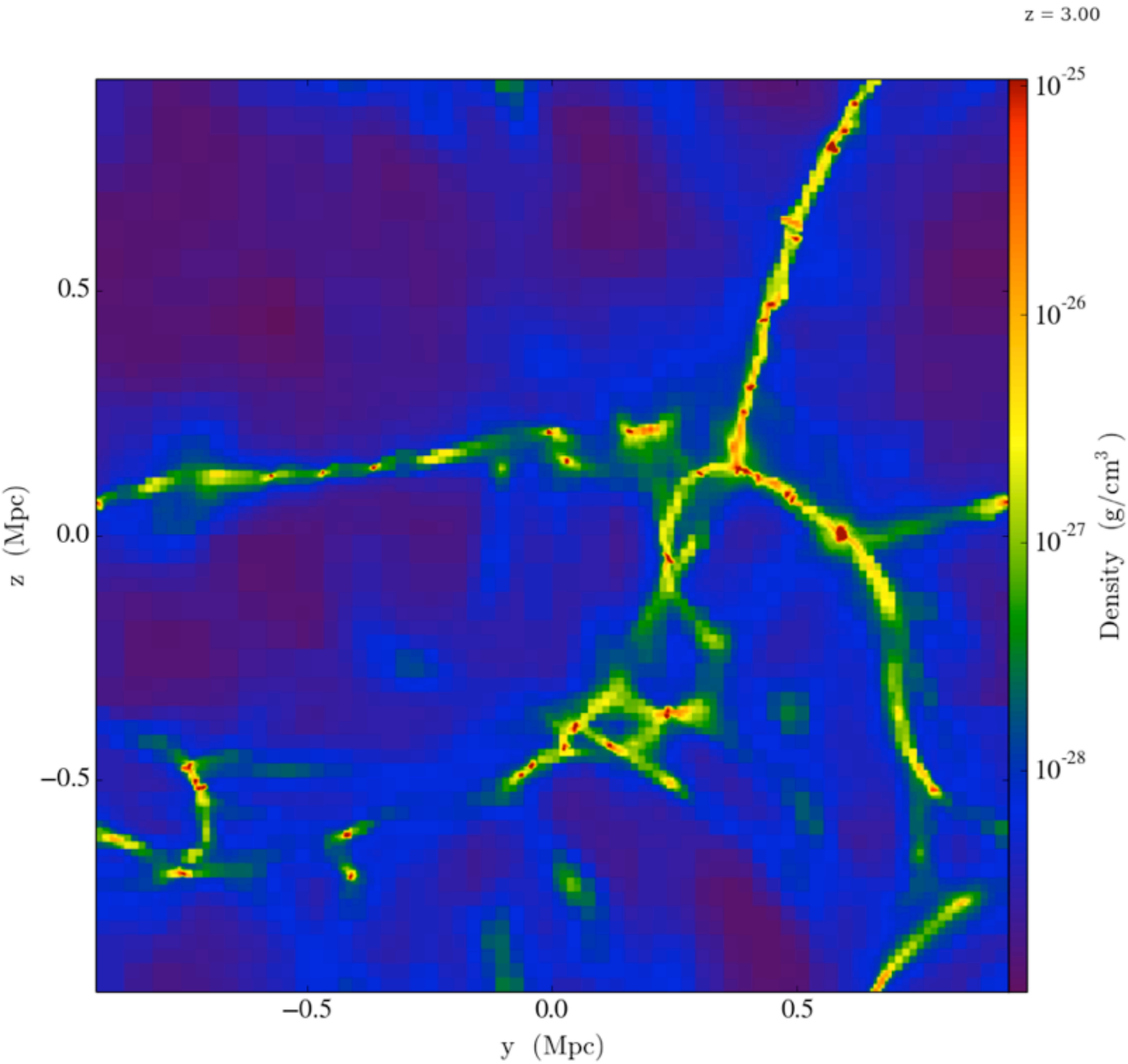
- I have copied the last output to conival. You can copy it to your directory
    - `cp -r ~guest009/cooling/DD0050 .`
- Let's inspect it!

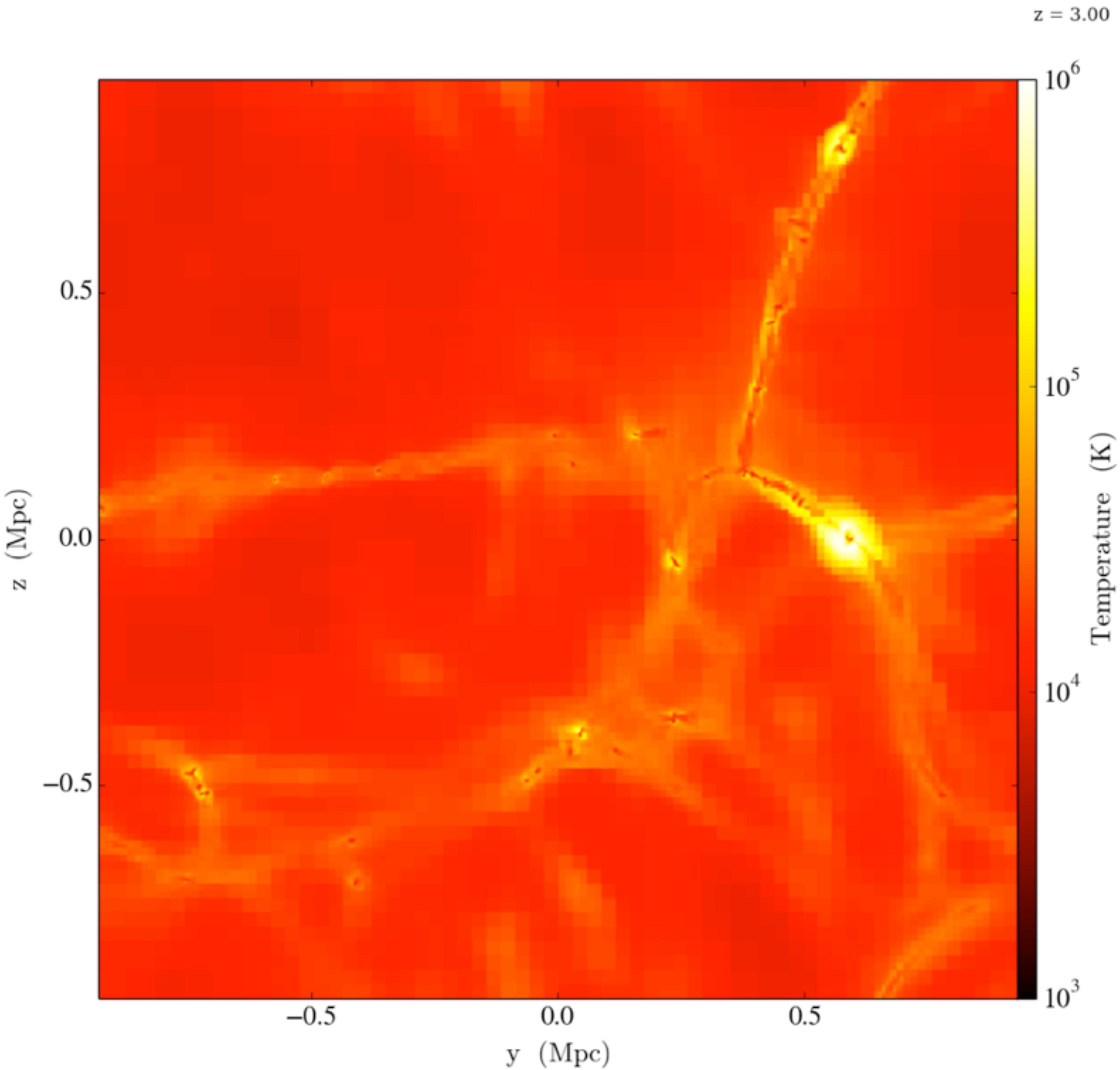

z = 50.00
z (kpc)
y (kpc)
60
40
20
0
−20
−40
−60
10^{-25}
10^{-26}
10^{-27}
10^{-28}
Density (g/cm^3)

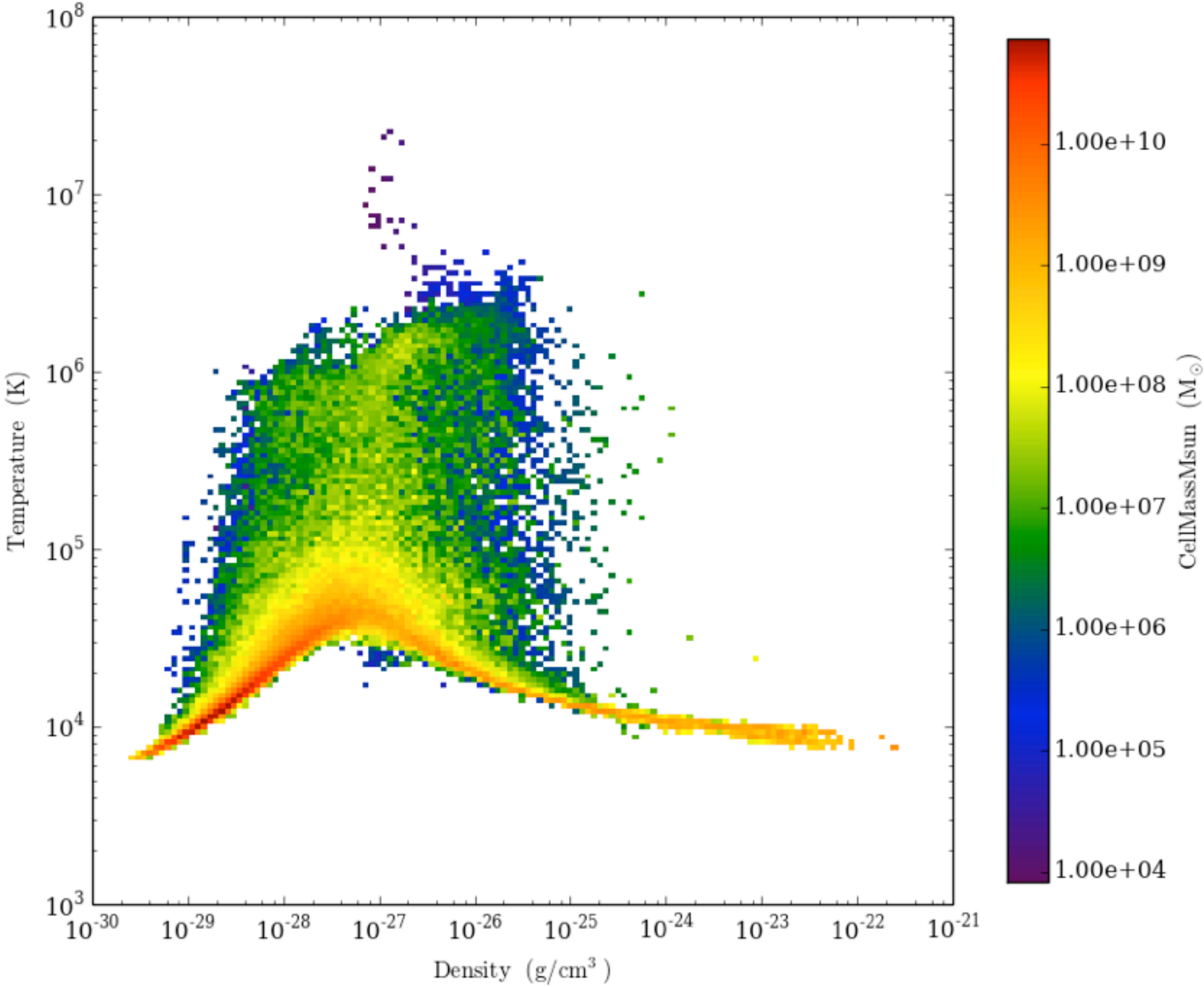
z = 50.00



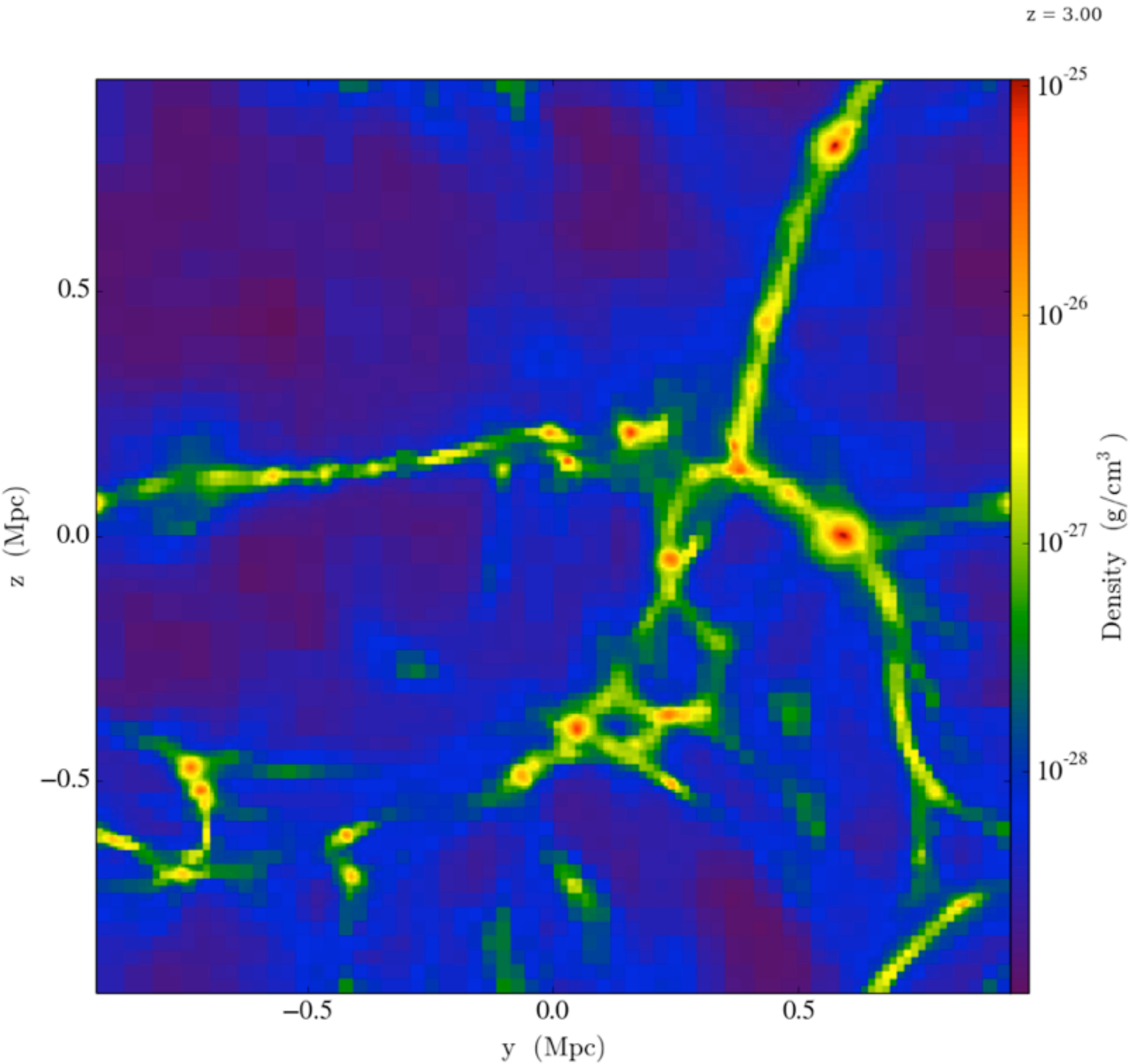
60
40
20
0
−20
−40
−60
z (kpc)
−60
−40
−20
0
20
40
60
y (kpc)
10^6
10^5
10^4
10^3
Temperature (K)


z = 3.00
0.5
0.0
−0.5
z (Mpc)
−0.5
0.0
0.5
y (Mpc)
10^{-25}
10^{-26}
10^{-27}
10^{-28}
Density (g/cm^3)

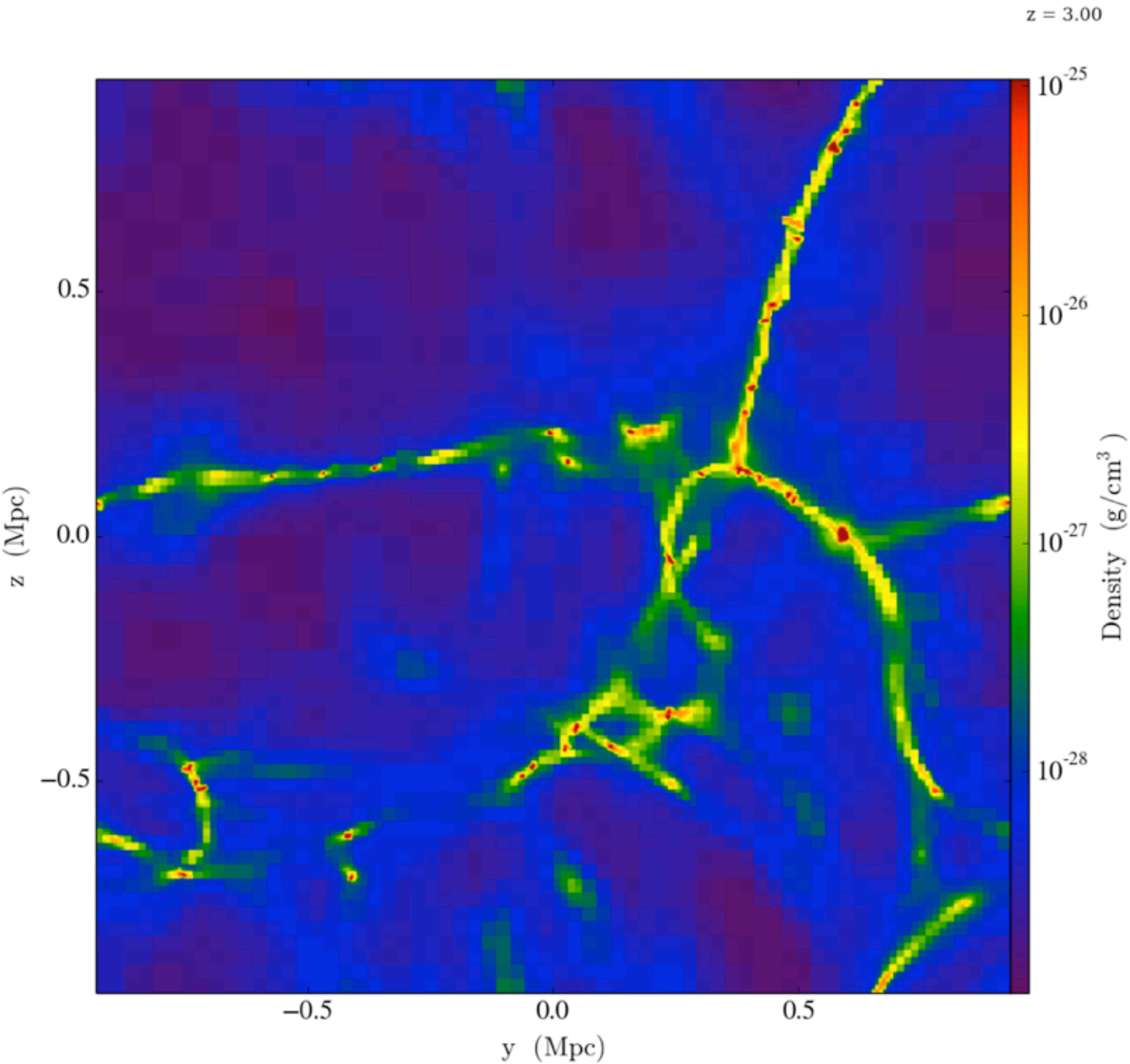

z = 3.00
0.5
0.0
−0.5
z (Mpc)
−0.5
0.0
0.5
y (Mpc)
10^6
10^5
10^4
10^3
Temperature (K)



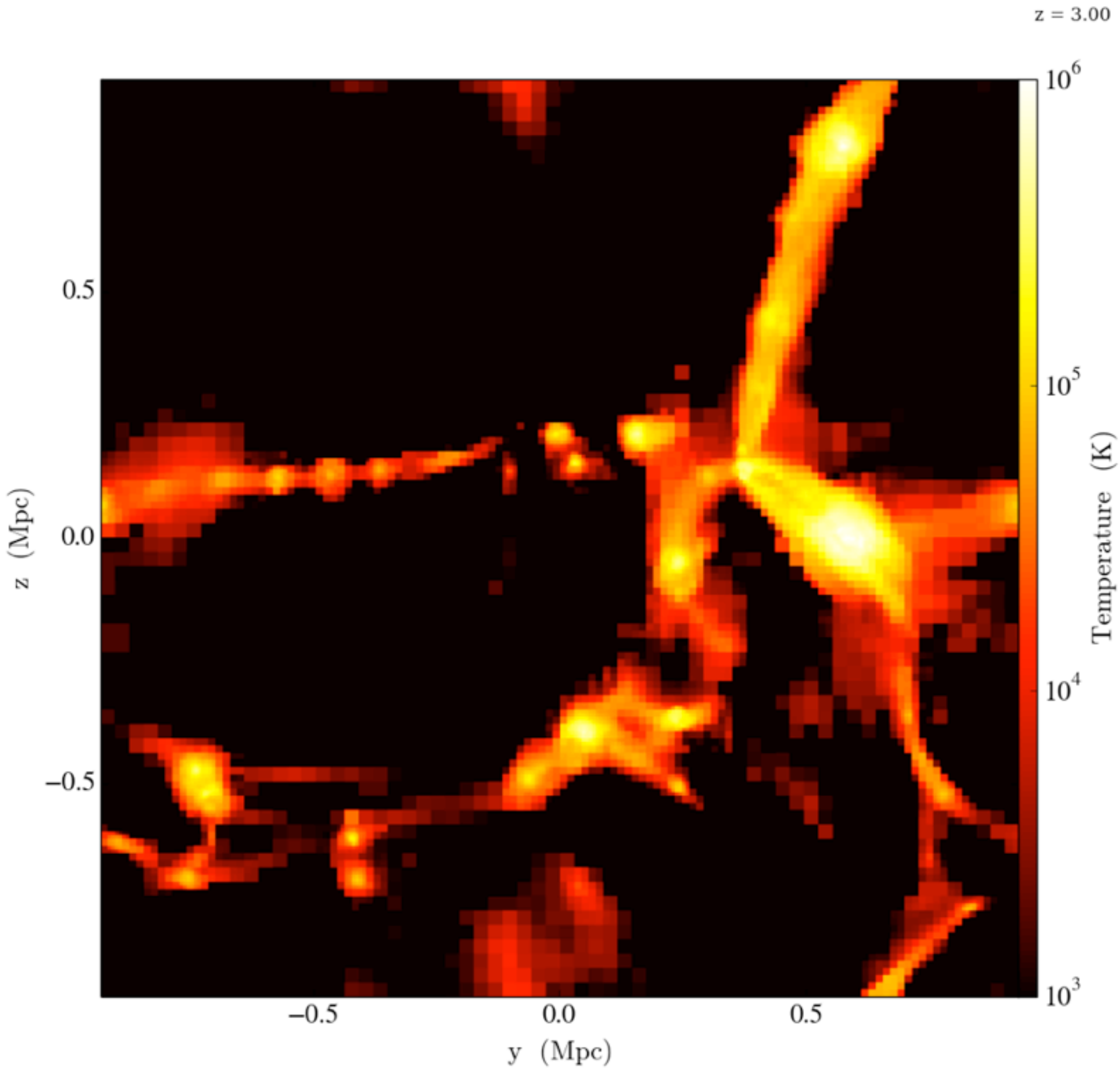
Temperature (K)
Density (g/cm³)
CellMassMsun (M☉)
10^8
10^7
10^6
10^5
10^4
10^3
10^-30
10^-29
10^-28
10^-27
10^-26
10^-25
10^-24
10^-23
10^-22
10^-21
1.00e+10
1.00e+09
1.00e+08
1.00e+07
1.00e+06
1.00e+05
1.00e+04



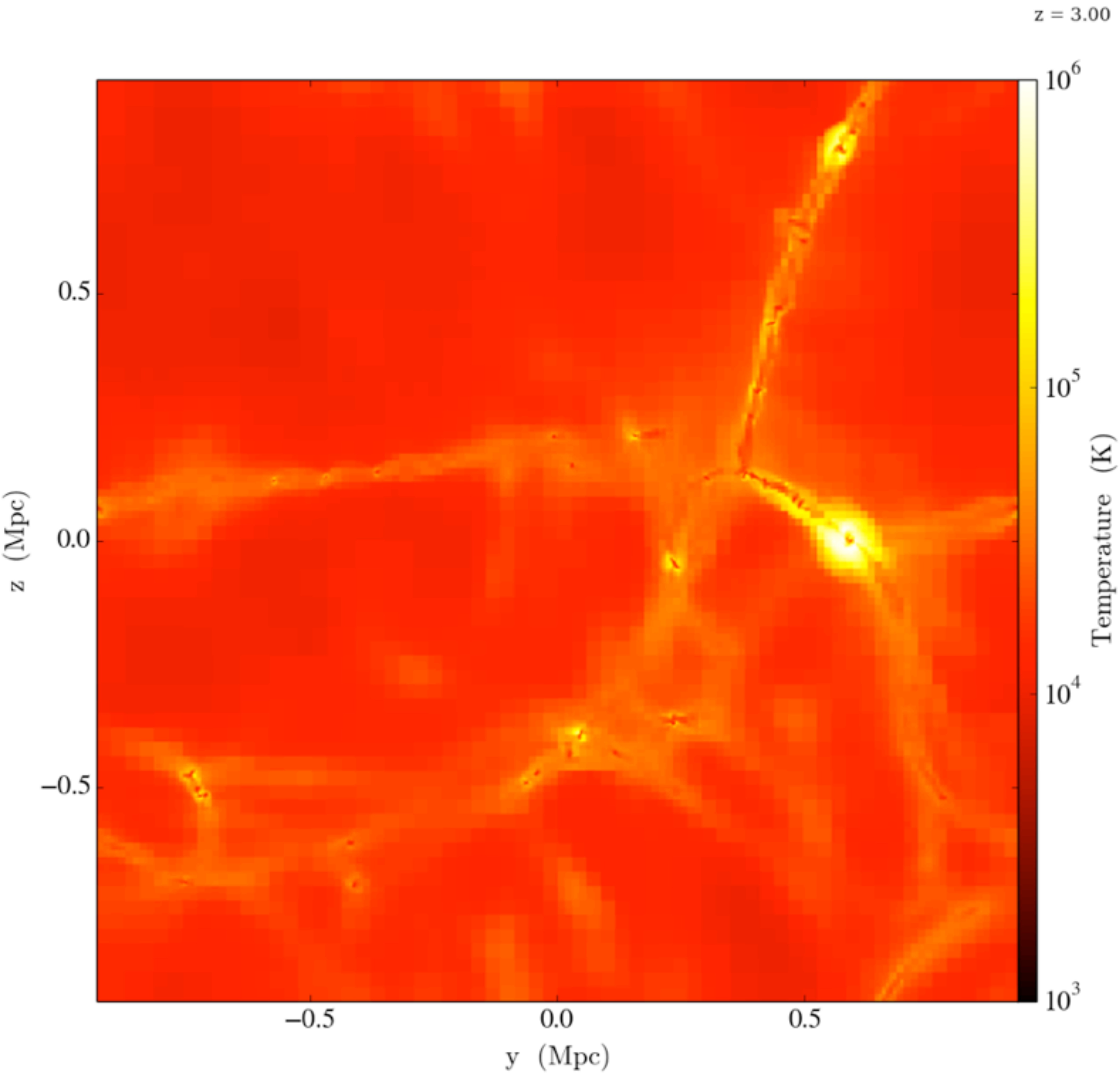
z = 3.00
0.5
0.0
−0.5
z (Mpc)
−0.5
0.0
0.5
y (Mpc)
10^{-25}
10^{-26}
10^{-27}
10^{-28}
Density (g/cm^3)



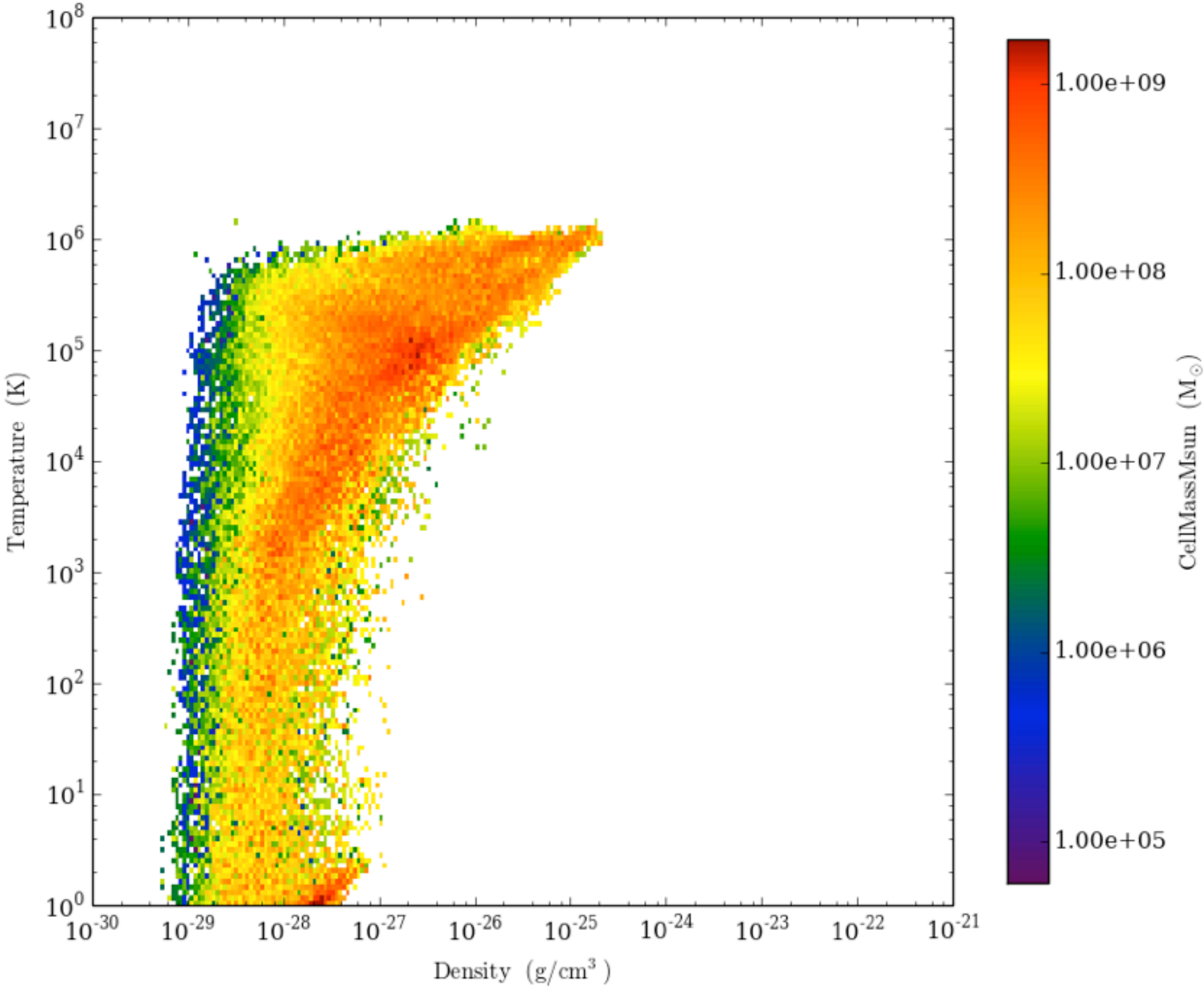
z = 3.00
0.5
0.0
−0.5
z (Mpc)
−0.5
0.0
0.5
y (Mpc)
10^{-25}
10^{-26}
10^{-27}
10^{-28}
Density (g/cm^3)



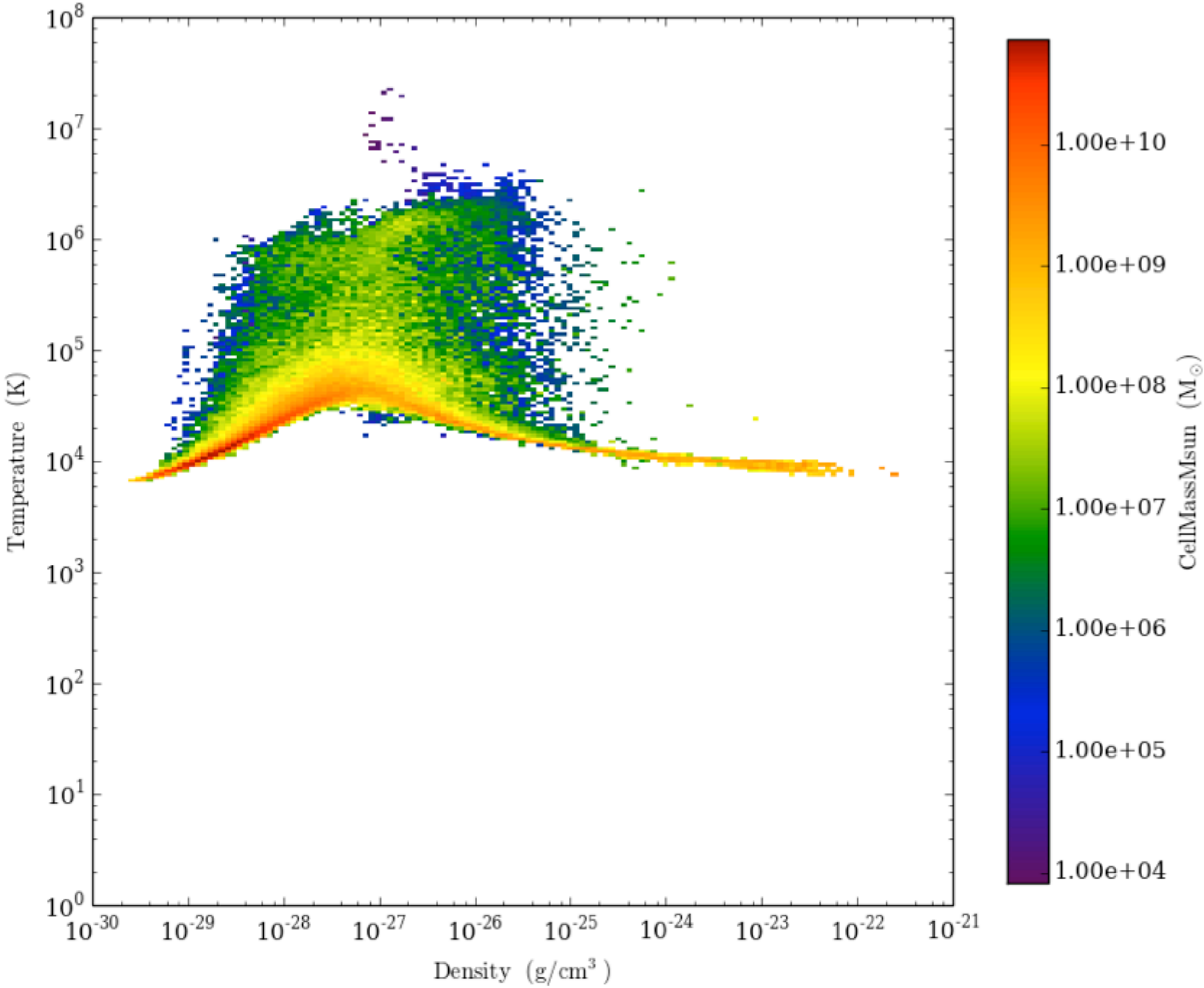
z = 3.00
0.5
0.0
−0.5
z (Mpc)
−0.5
0.0
0.5
y (Mpc)
10^6
10^5
10^4
10^3
Temperature (K)



z = 3.00
z (Mpc)
0.5
0.0
−0.5
y (Mpc)
−0.5
0.0
0.5
Temperature (K)
10^6
10^5
10^4
10^3



Temperature (K)
Density (g/cm³)
CellMassMsun (M⊙)
1.00e+09
1.00e+08
1.00e+07
1.00e+06
1.00e+05



Temperature (K)
Density (g/cm³)
CellMassMsun (M☉)
1.00e+10
1.00e+09
1.00e+08
1.00e+07
1.00e+06
1.00e+05
1.00e+04

Radiative Cooling and Ultraviolet Background

- I have also uploaded a IPython notebook to my directory.
  - `cp ~guest009/cooling/Cooling.ipynb .`

+ Star Formation and Supernova Feedback

- Let's add more physics! Star formation and supernova feedback.

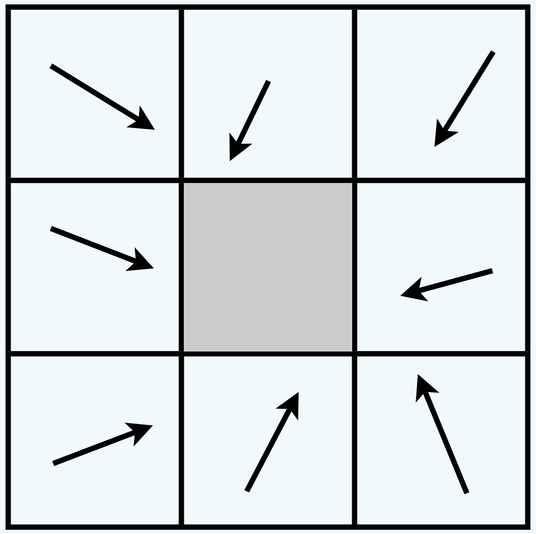
- Star Formation (original formulation: Cen & Ostriker 1992)

  - Overdense: $\rho > \rho_{\rm SF}$

  - Converging flow: $\nabla \cdot \mathbf{v} < 0$

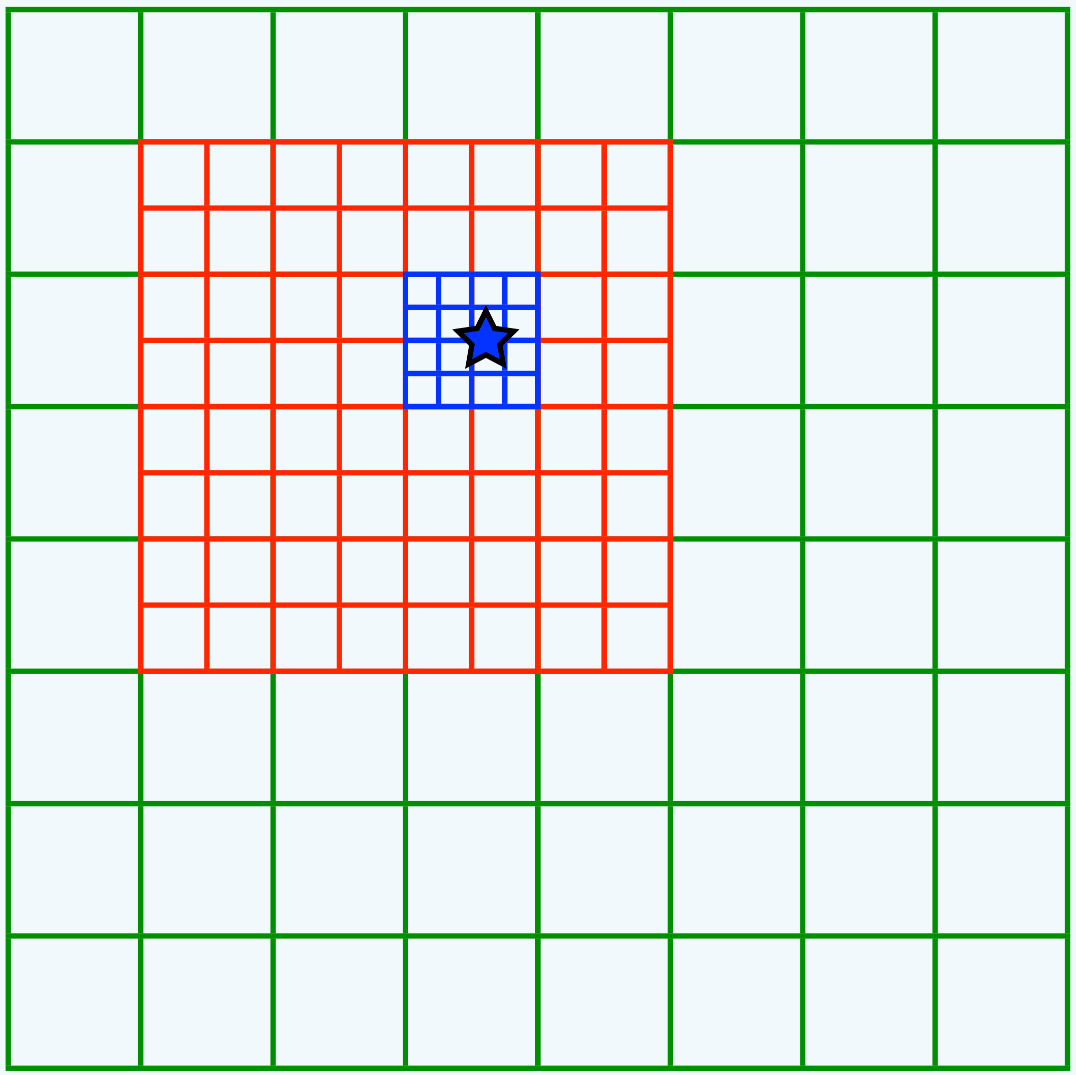
  - Cooling: $t_{\rm cool} < t_{\rm dyn}$

  - Gravitational unstable (originally used, but not in the specific algorithm we'll be using): $M_{\rm cell} > M_J$



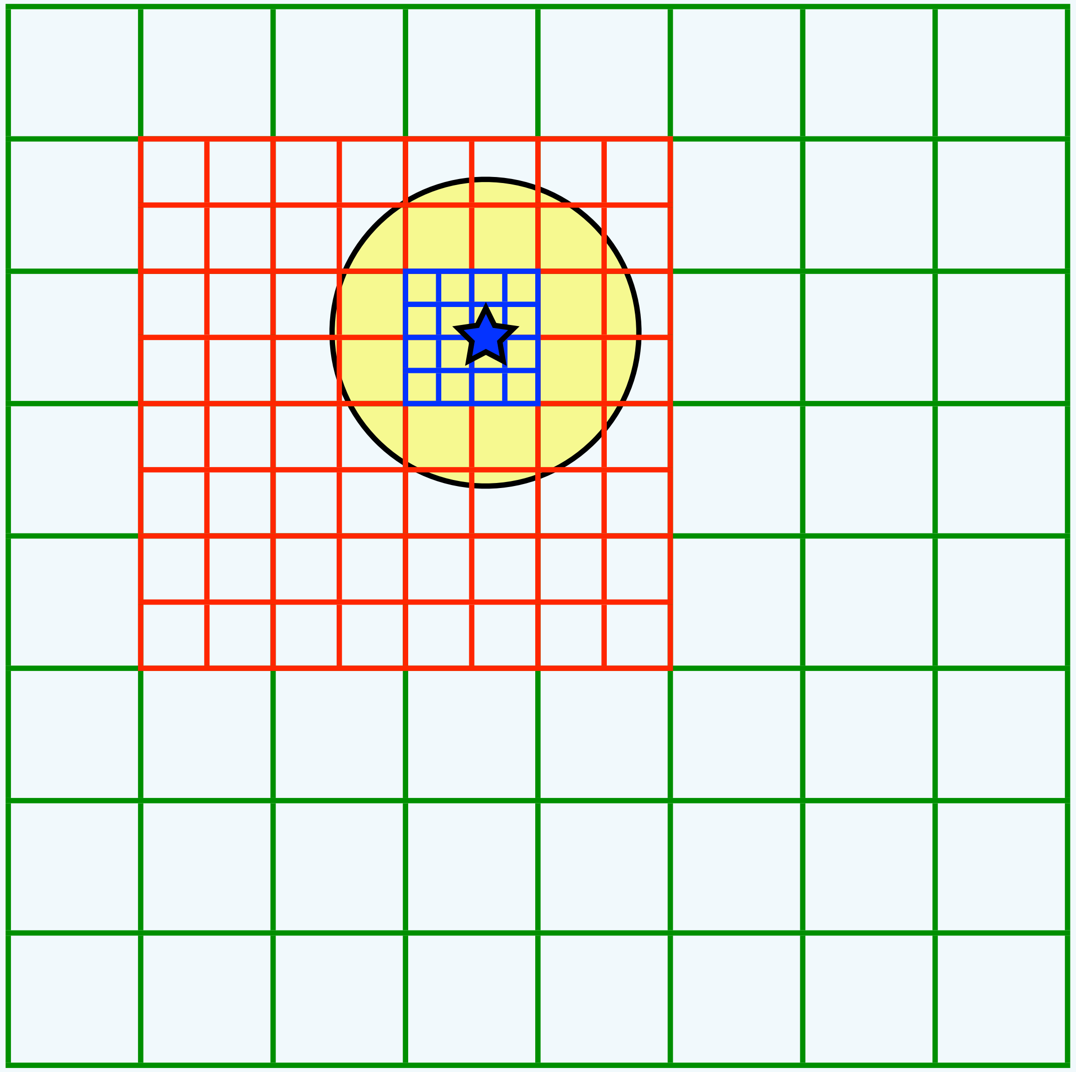
+ Star Formation and Supernova Feedback

- Supernova feedback is modeled with thermal energy injection



+ Star Formation and Supernova Feedback

- Supernova feedback is modeled with thermal energy injection



+ Star Formation and Supernova Feedback

- You can find the parameter file in
  - `sapporo_cosmo/stars-uvb.enzo`
- Star formation and feedback.
  - Method 5 → $2^5 = 32$

```
StarParticleCreation                  = 32
StarParticleFeedback                  = 32

PopIIIOverDensityThreshold            = -0.1
PopIIIMetalCriticalFraction           = 1e-30

StarClusterUseMetalField              = 1
StarClusterUnresolvedModel            = 1
StarClusterMinDynamicalTime           = 1e+07
StarClusterIonizingLuminosity         = 3e+46
StarClusterSNEnergy                   = 1.25e+49
StarClusterSNRadius                   = 100
StarClusterFormEfficiency             = 0.07
StarClusterMinimumMass                = 1e7
```

+ Star Formation and Supernova Feedback

`StarParticleCreation` (external)

This parameter is bitwise so that multiple types of star formation routines can be used in a single simulation. For example if methods 1 and 3 are desired, the user would specify 10 ($2^1 + 2^3$), or if methods 1, 4 and 7 are wanted, this would be 146 ($2^1 + 2^4 + 2^7$). Default: 0

```
0  - Cen & Ostriker (1992)
1  - Cen & Ostriker (1992) with stocastic star formation
2  - Global Schmidt Law / Kravstov et al. (2003)
3  - Population III stars / Abel, Wise & Bryan (2007)
4  - Sink particles: Pure sink particle or star particle with wind feedback depending on
     choice for HydroMethod / Wang et al. (2009)
5  - Radiative star clusters  / Wise & Cen (2009)
6  - [reserved for future use]
7  - Cen & Ostriker (1992) with no delay in formation
8  - Springel & Hernquist (2003)
9  - Massive Black Hole (MBH) particles insertion by hand / Kim et al. (2010)
10 - Population III stellar tracers
11 - Molecular hydrogen regulated star formation
```

+ Star Formation and Supernova Feedback

- Star formation and feedback.
  Method 5 → $2^5 = 32$

- Method 3 (Population III stars) and method 5 use the same minimum overdensity. **Negative** number means units in cm^{-3}. **Positive** number is in code units.

- Critical metallicity to transition from Population III to Population II. For this simulation, we only want to consider **Pop II stars**, so we set to a tiny_number.

```
StarParticleCreation                = 32
StarParticleFeedback                = 32

PopIIIOverDensityThreshold          = -0.1
PopIIIMetalCriticalFraction         = 1e-30

StarClusterUseMetalField            = 1
StarClusterUnresolvedModel          = 1
StarClusterMinDynamicalTime         = 1e+07
StarClusterIonizingLuminosity       = 3e+46
StarClusterSNEnergy                 = 1.25e+49
StarClusterSNRadius                 = 100
StarClusterFormEfficiency           = 0.07
StarClusterMinimumMass              = 1e7
```

+ Star Formation and Supernova Feedback

- Use metal feedback in supernova.
- Use Cen & Ostriker prescription for time-dependent feedback.
- Dynamical time (→ avg. density) of a sphere that accretes onto the star particle.
- Ionizing photon luminosity (in units of photons / s / M$_\odot$)
- Supernova thermal energy (in units of erg / M$_\odot$)
- Radius of sphere where the energy is injected (in units of pc)

```
StarParticleCreation                 = 32
StarParticleFeedback                 = 32

PopIIIOverDensityThreshold           = -0.1
PopIIIMetalCriticalFraction          = 1e-30

StarClusterUseMetalField             = 1
StarClusterUnresolvedModel           = 1
StarClusterMinDynamicalTime          = 1e+07
StarClusterIonizingLuminosity        = 3e+46
StarClusterSNEnergy                  = 1.25e+49
StarClusterSNRadius                  = 100
StarClusterFormEfficiency            = 0.07
StarClusterMinimumMass               = 1e7
```

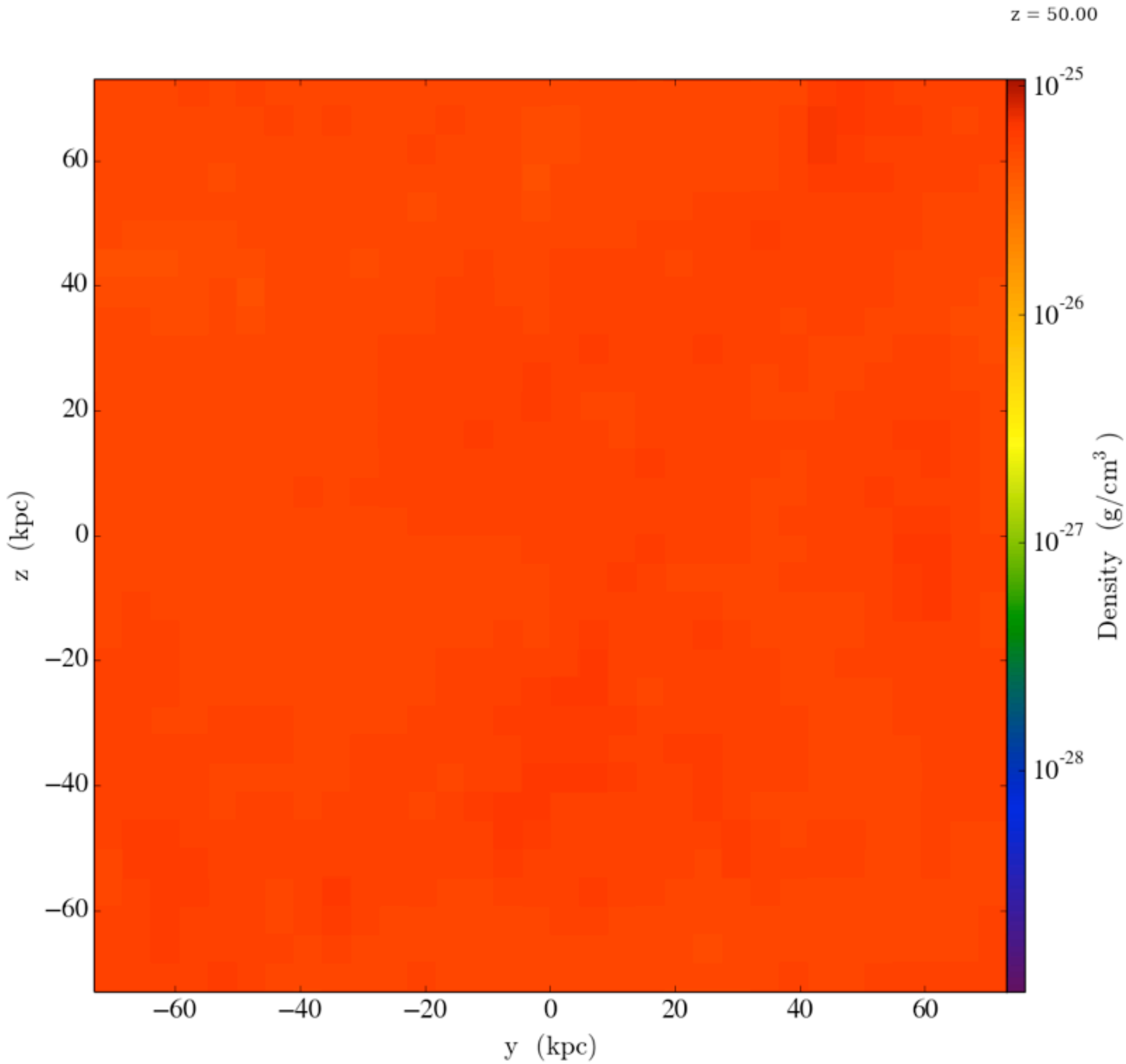
+ Star Formation and Supernova Feedback

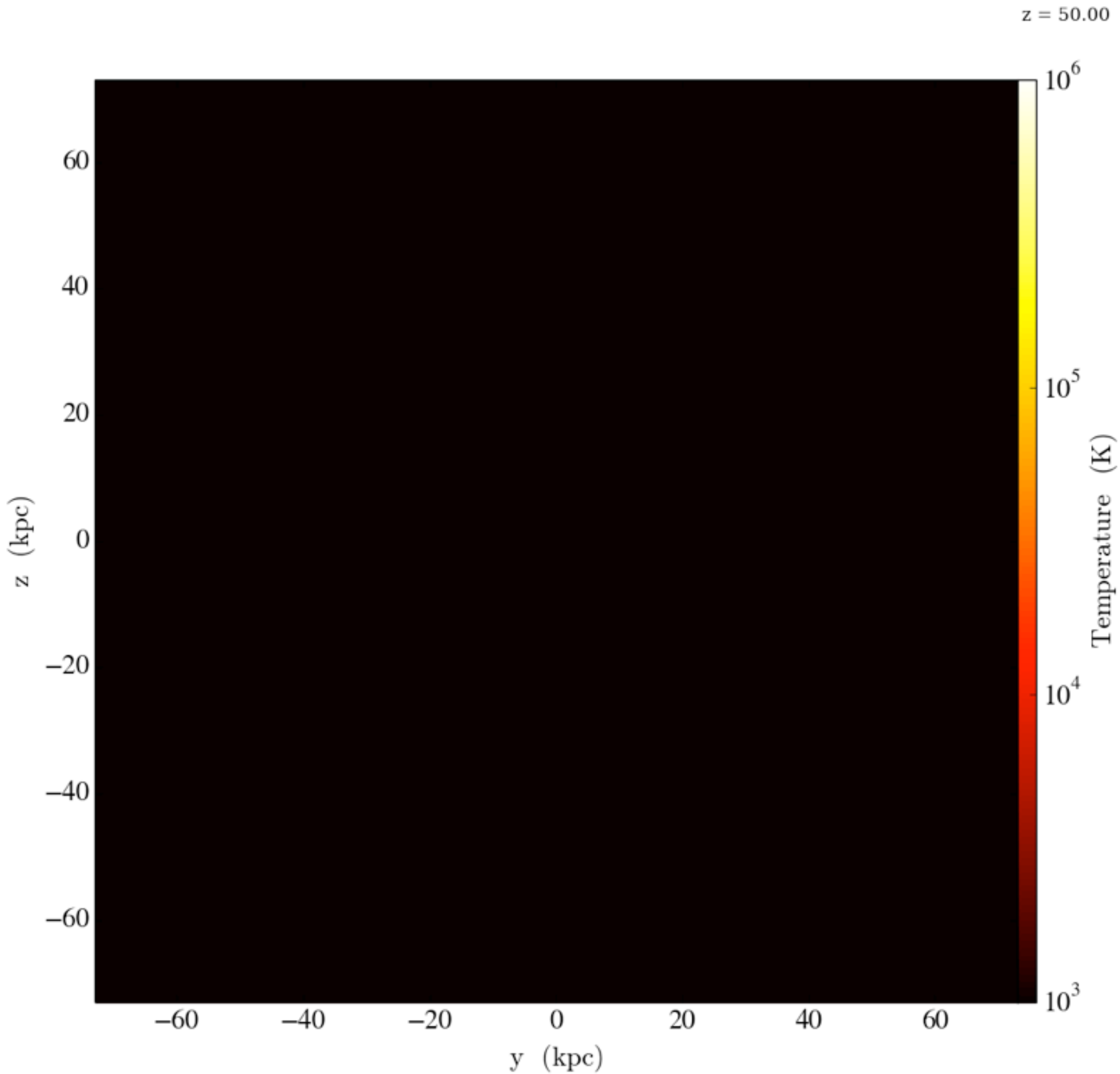
- Mass fraction of cold gas inside the sphere that is deposited into the star particles.
- Minimum mass (in units of $M_\odot$) of star particles

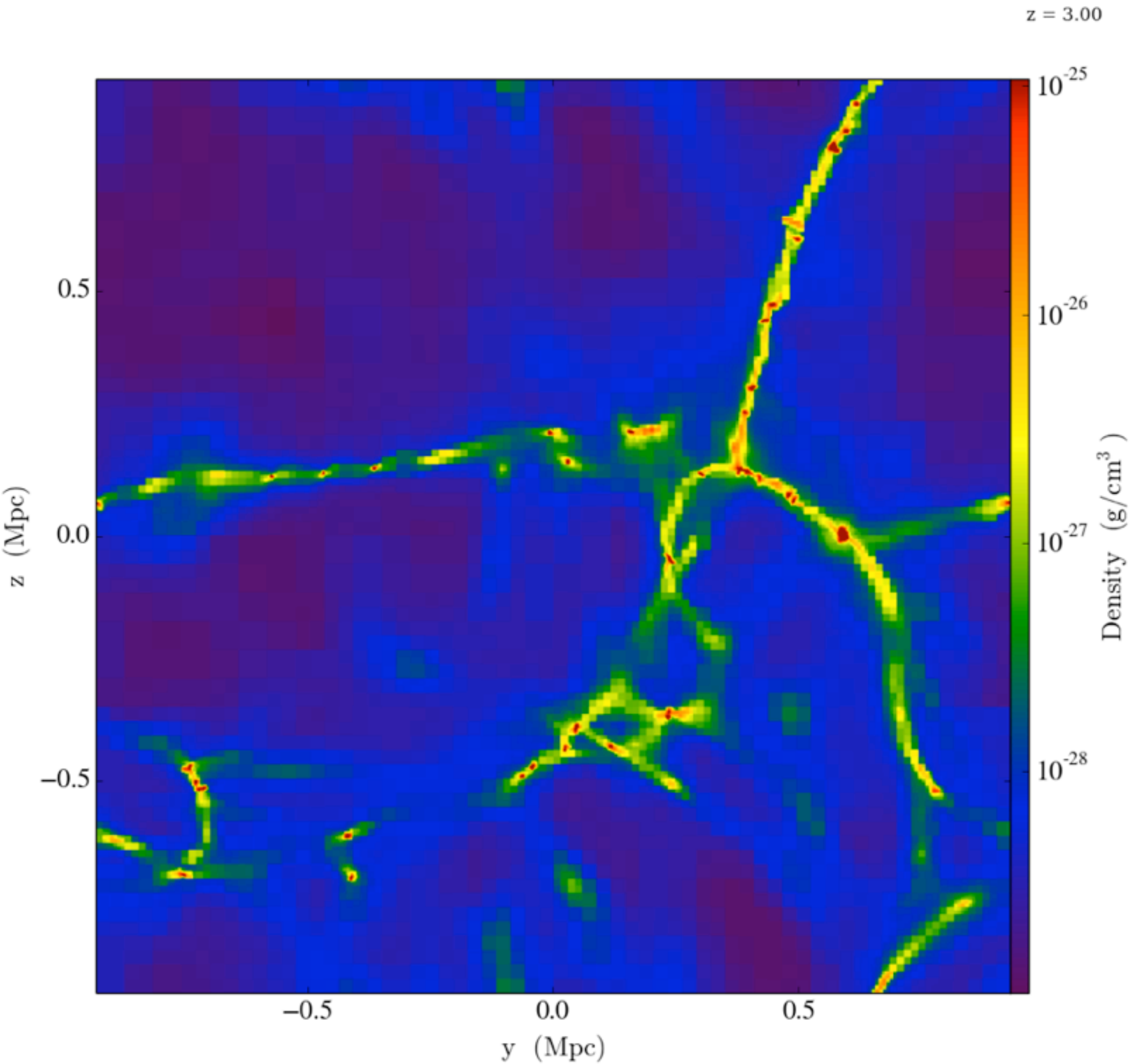
```
StarParticleCreation                = 32
StarParticleFeedback                = 32

PopIIIOverDensityThreshold          = -0.1
PopIIIMetalCriticalFraction         = 1e-30

StarClusterUseMetalField            = 1
StarClusterUnresolvedModel          = 1
StarClusterMinDynamicalTime         = 1e+07
StarClusterIonizingLuminosity       = 3e+46
StarClusterSNEnergy                 = 1.25e+49
StarClusterSNRadius                 = 100
StarClusterFormEfficiency           = 0.07
StarClusterMinimumMass              = 1e7
```

+ Star Formation and Supernova Feedback

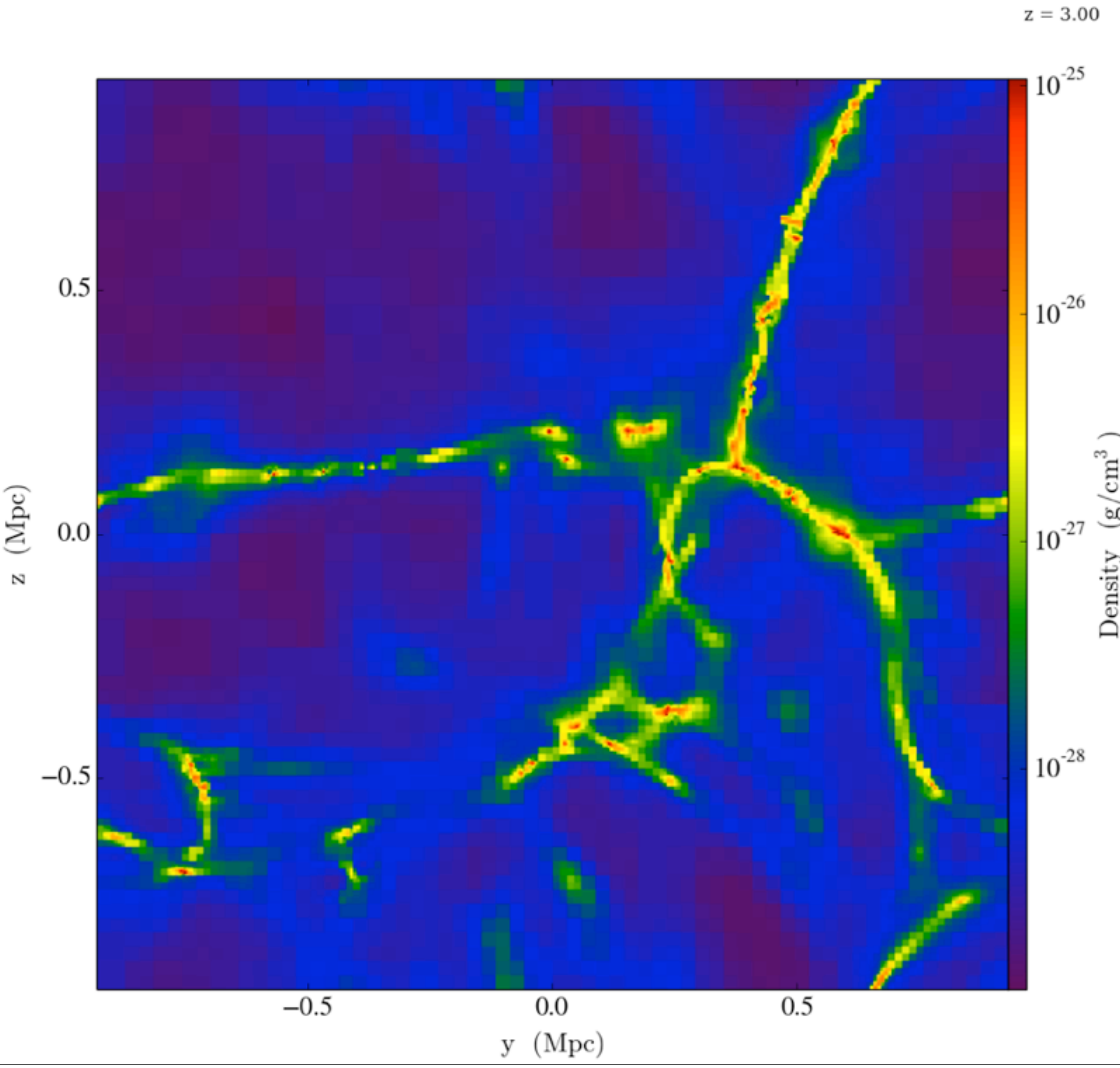
- Again, this simulation takes some time to complete. About an hour.
- I have uploaded the last output at redshift 3 and IPython notebook to
  - `~guest009/SNe/DD0050`
  - `~guest009/SNe/Supernovae.ipynb`
- Let's inspect it!

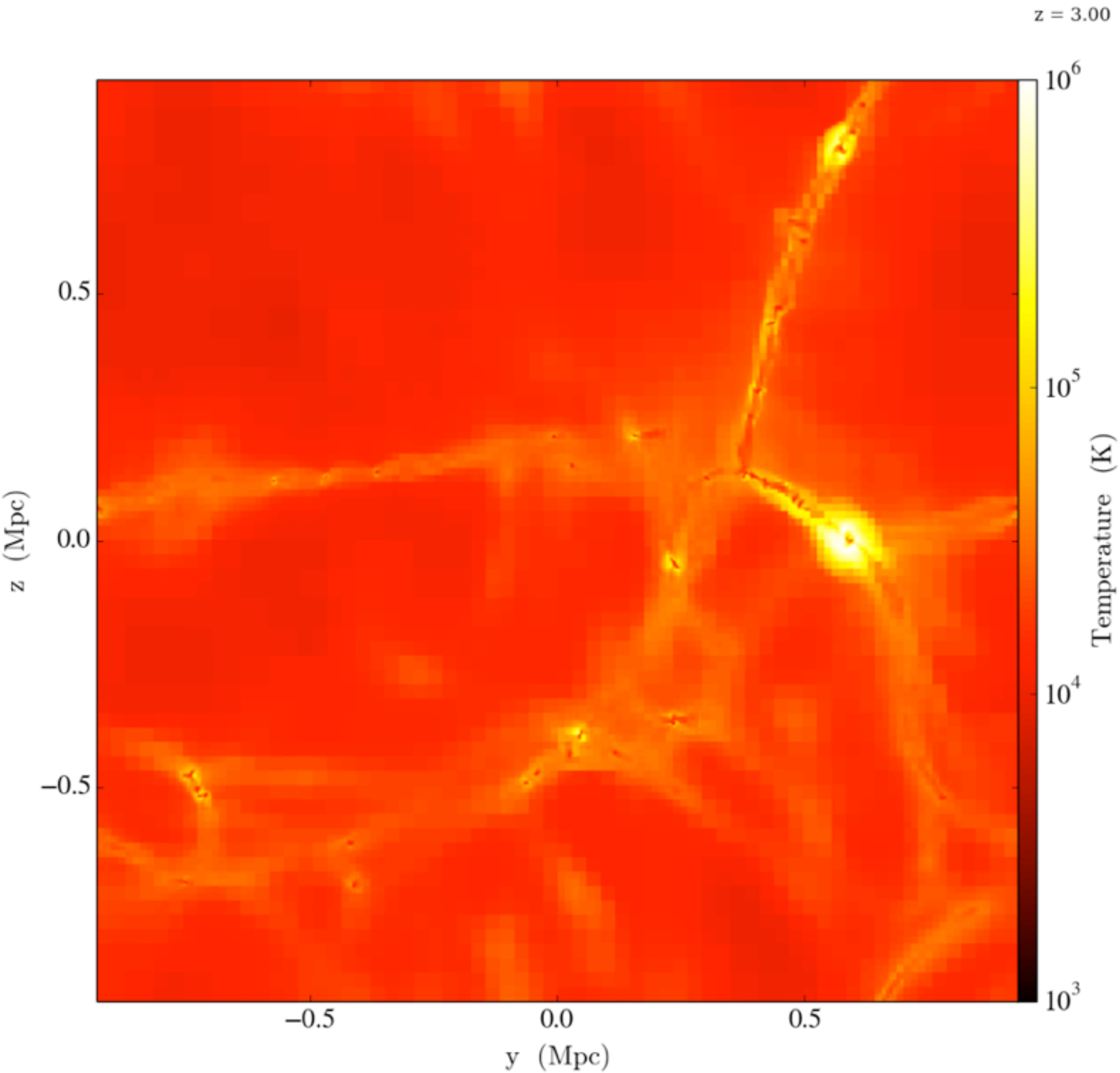

z = 50.00
60
40
20
0
−20
−40
−60
z (kpc)
−60
−40
−20
0
20
40
60
y (kpc)
10^{-25}
10^{-26}
10^{-27}
10^{-28}
Density (g/cm^3)


z = 50.00
60
40
20
0
−20
−40
−60
z (kpc)
−60
−40
−20
0
20
40
60
y (kpc)
10^6
10^5
10^4
10^3
Temperature (K)

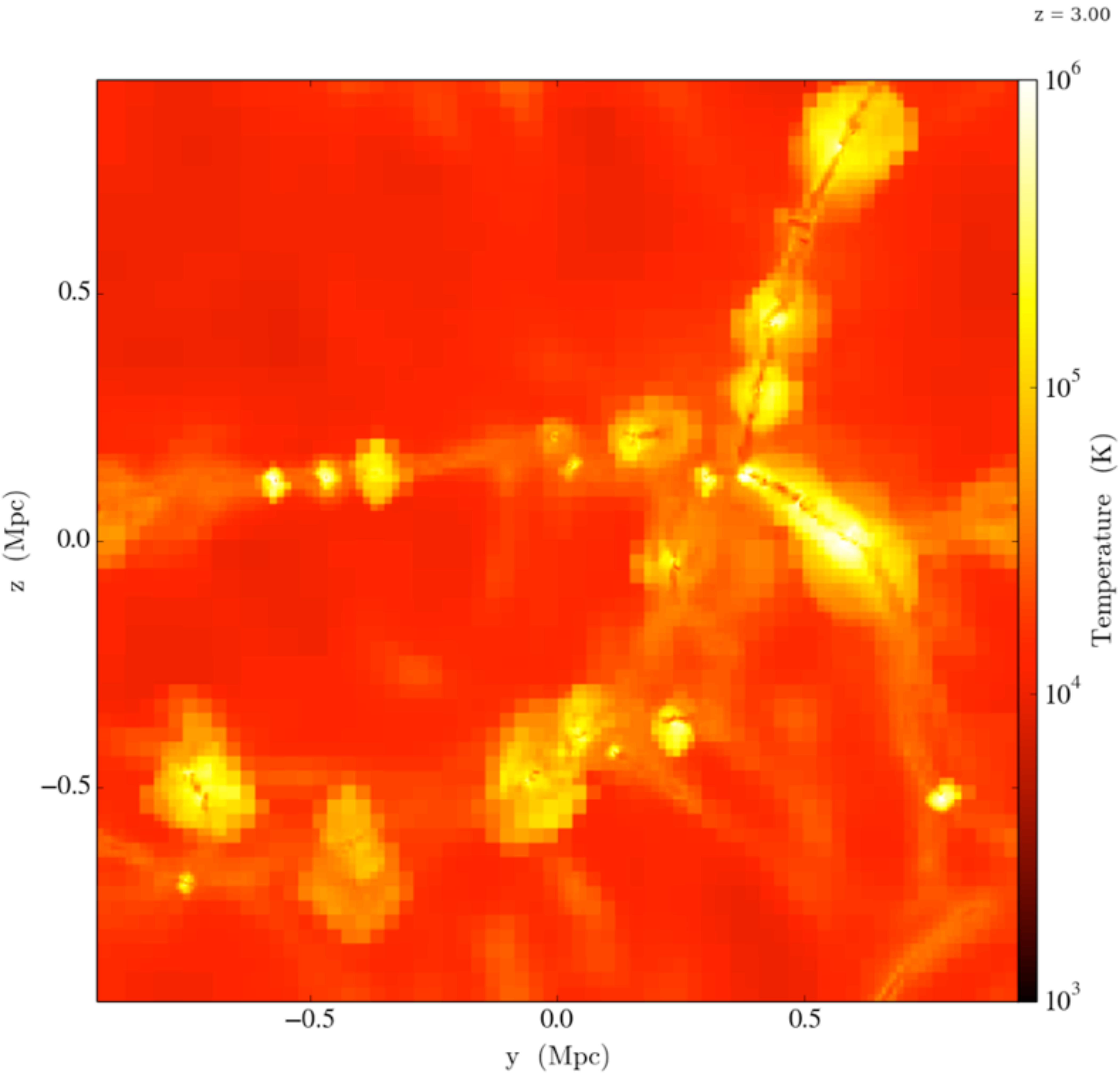

z = 3.00
0.5
0.0
−0.5
z (Mpc)
−0.5
0.0
0.5
y (Mpc)
10^{-25}
10^{-26}
10^{-27}
10^{-28}
Density (g/cm^3)

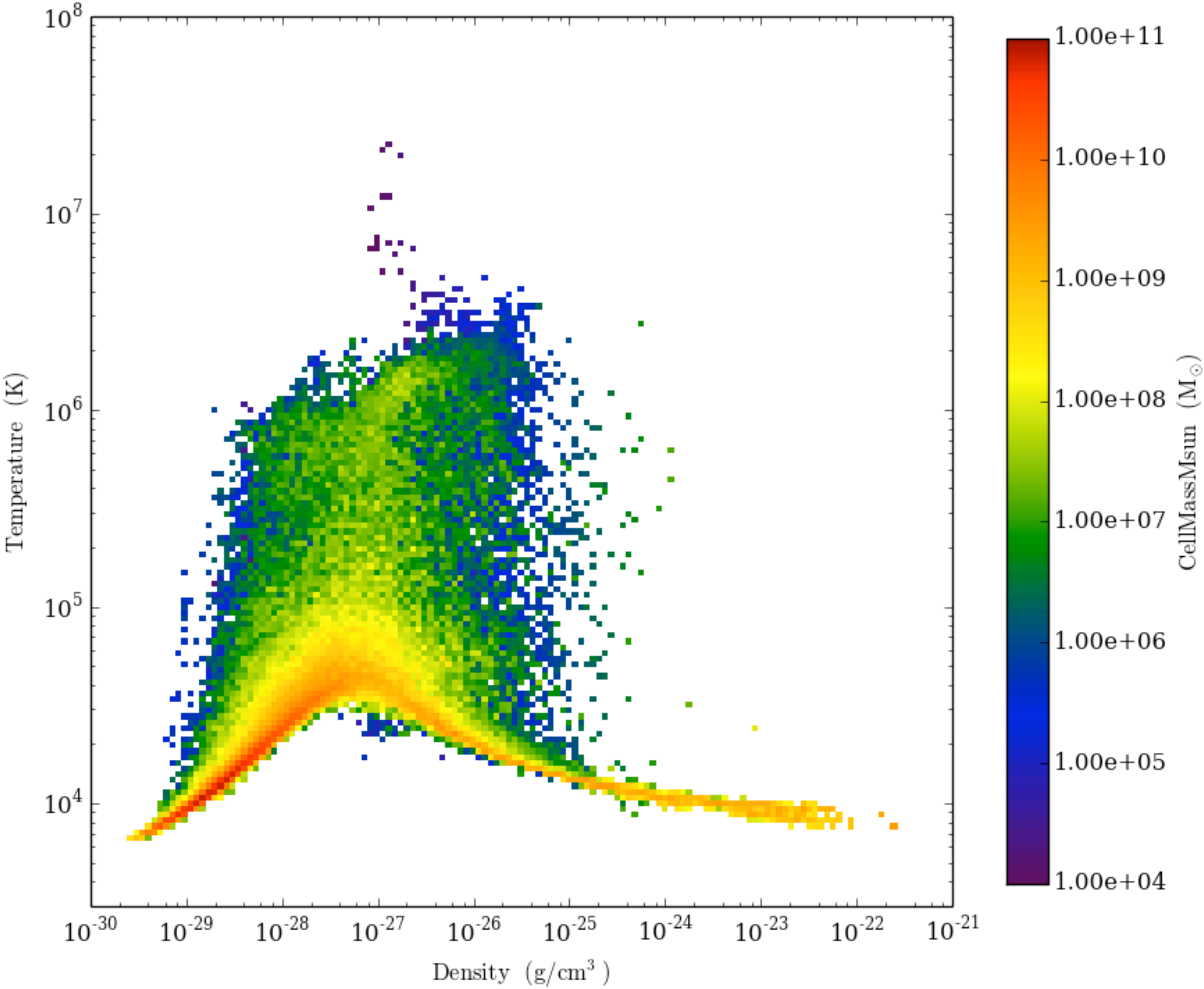
30


z = 3.00
0.5
0.0
−0.5
z (Mpc)
−0.5
0.0
0.5
y (Mpc)
10^{-25}
10^{-26}
10^{-27}
10^{-28}
Density (g/cm^3)

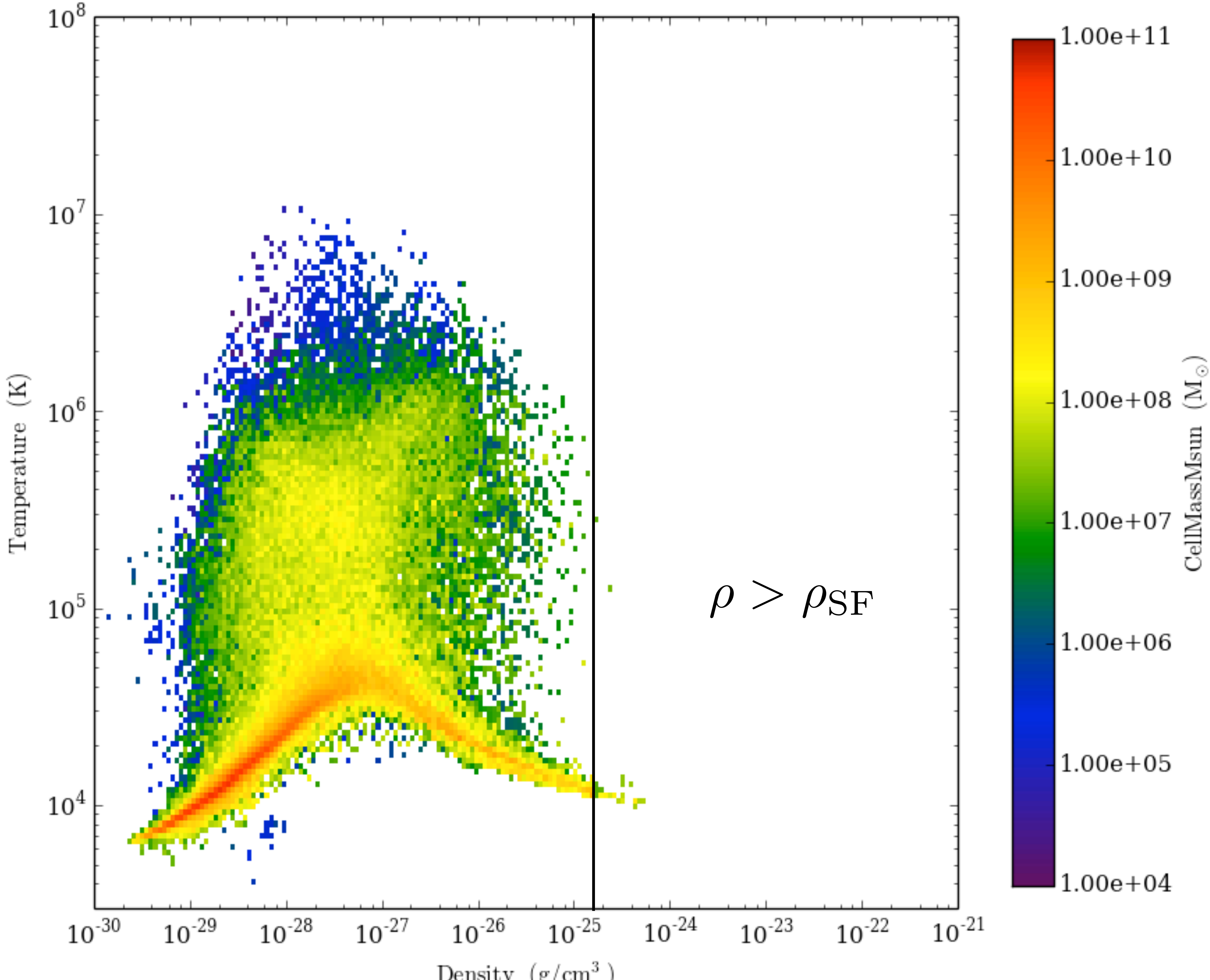


z = 3.00
0.5
0.0
−0.5
z (Mpc)
−0.5
0.0
0.5
y (Mpc)
10^6
10^5
10^4
10^3
Temperature (K)


z = 3.00
0.5
0.0
−0.5
z (Mpc)
−0.5
0.0
0.5
y (Mpc)
10^6
10^5
10^4
10^3
Temperature (K)



10^8
10^7
10^6
10^5
10^4
Temperature (K)
10^{-30} 10^{-29} 10^{-28} 10^{-27} 10^{-26} 10^{-25} 10^{-24} 10^{-23} 10^{-22} 10^{-21}
Density (g/cm^3)
1.00e+11
1.00e+10
1.00e+09
1.00e+08
1.00e+07
1.00e+06
1.00e+05
1.00e+04
CellMassMsun ($M_\odot$)



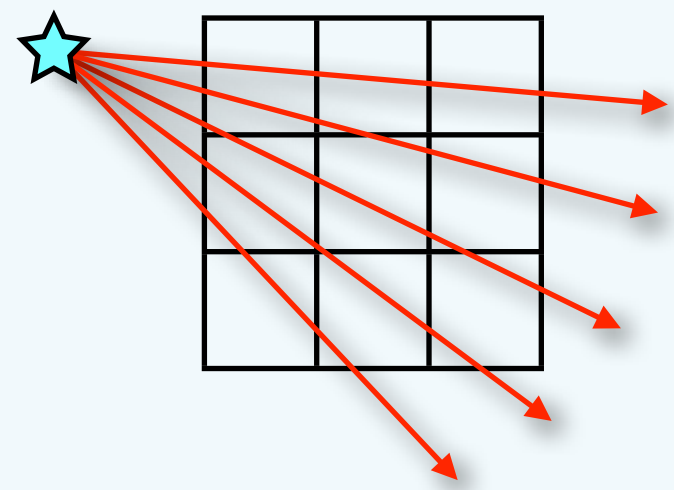
Temperature (K)
Density (g/cm³)
CellMassMsun (M☉)
1.00e+11
1.00e+10
1.00e+09
1.00e+08
1.00e+07
1.00e+06
1.00e+05
1.00e+04
$\rho > \rho_{SF}$

+ Radiative Feedback

- Enzo has two prescriptions to solve the radiative transfer equation:
  - Adaptive ray tracing
  - Flux limited diffusion
- This allows for an inhomogenous radiation field with spatially dependent absorption and emission coefficients.
- Can be used in conjunction with a radiation background.

+ Radiative Feedback

- Enzo has two prescriptions to solve the radiative transfer equation:
  - Adaptive ray tracing
  - Flux limited diffusion



- This allows for an inhomogenous radiation field with spatially dependent absorption and emission coefficients.
- Can be used in conjunction with a radiation background.

Cosmological Radiative Transfer Equation

n := normal vector
a := scale factor
ā := a/a_{em}
H := Hubble factor
ν := frequency

$$I_\nu \equiv I(\nu, \mathbf{x}, \Omega, t)$$

$$\frac{1}{c}\,\frac{\partial I_\nu}{\partial t} + \frac{\hat{n}\cdot\nabla I_\nu}{\bar{a}} - \frac{H}{c}\left(\nu\frac{\partial I_\nu}{\partial \nu} - 3I_\nu\right) = -\kappa_\nu I_\nu + j_\nu$$

Cosmological Radiative Transfer Equation

n := normal vector
a := scale factor
ā := a/a_{em}
H := Hubble factor
ν := frequency

$$I_\nu \equiv I(\nu, \mathbf{x}, \Omega, t)$$

$$\frac{1}{c}\frac{\partial I_\nu}{\partial t} + \frac{\hat{n}\cdot\nabla I_\nu}{\bar{a}} - \frac{H}{c}\left(\nu\frac{\partial I_\nu}{\partial \nu} - 3I_\nu\right) = -\kappa_\nu I_\nu + j_\nu$$

Propagation &
Cosmic Expansion

Redshifting

Cosmological Dilution

Absorption

Emission

Simplifications – “Local” Approximation

1. Short timesteps (ā = 1)
2. Ignore cosmological redshift and dilution (may become important >50 Mpc)

$$\frac{1}{c}\frac{\partial I_\nu}{\partial t} + \frac{\hat{n}\cdot\nabla I_\nu}{\bar{a}} - \frac{H}{c}\left(\nu\frac{\partial I_\nu}{\partial \nu} - 3I_\nu\right) = -\kappa_\nu I_\nu + j_\nu$$

Propagation & Cosmic Expansion

Redshifting

Cosmological Dilution

Absorption

Emission

Simplifications – “Local” Approximation

1. Short timesteps (ā = 1)
2. Ignore cosmological redshift and dilution (may become important >50 Mpc)

$$\frac{1}{c}\frac{\partial I_\nu}{\partial t} + \frac{\hat{n}\cdot\nabla I_\nu}{} - \frac{H}{c}\left(\nu\frac{\partial I_\nu}{\partial \nu} - 3I_\nu\right) = -\kappa_\nu I_\nu + j_\nu$$

Propagation & Cosmic Expansion

Redshifting

Cosmological Dilution

Absorption

Emission

Simplifications – “Local” Approximation

1. Short timesteps (ā = 1)
2. Ignore cosmological redshift and dilution (may become important >50 Mpc)

$$\frac{1}{c}\frac{\partial I_\nu}{\partial t} + \frac{\hat{n}\cdot\nabla I_\nu}{} - \qquad\qquad) = -\kappa_\nu I_\nu + j_\nu$$

Propagation & Cosmic Expansion

Redshifting

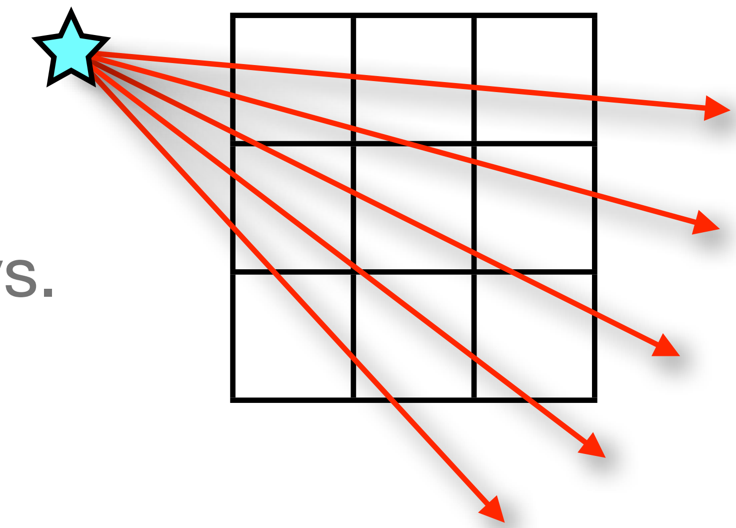
Cosmological Dilution

Absorption

Emission

RT Equation along a Ray

- Consider point sources of radiation



- Initially, the radiation flux is split equally among all rays.

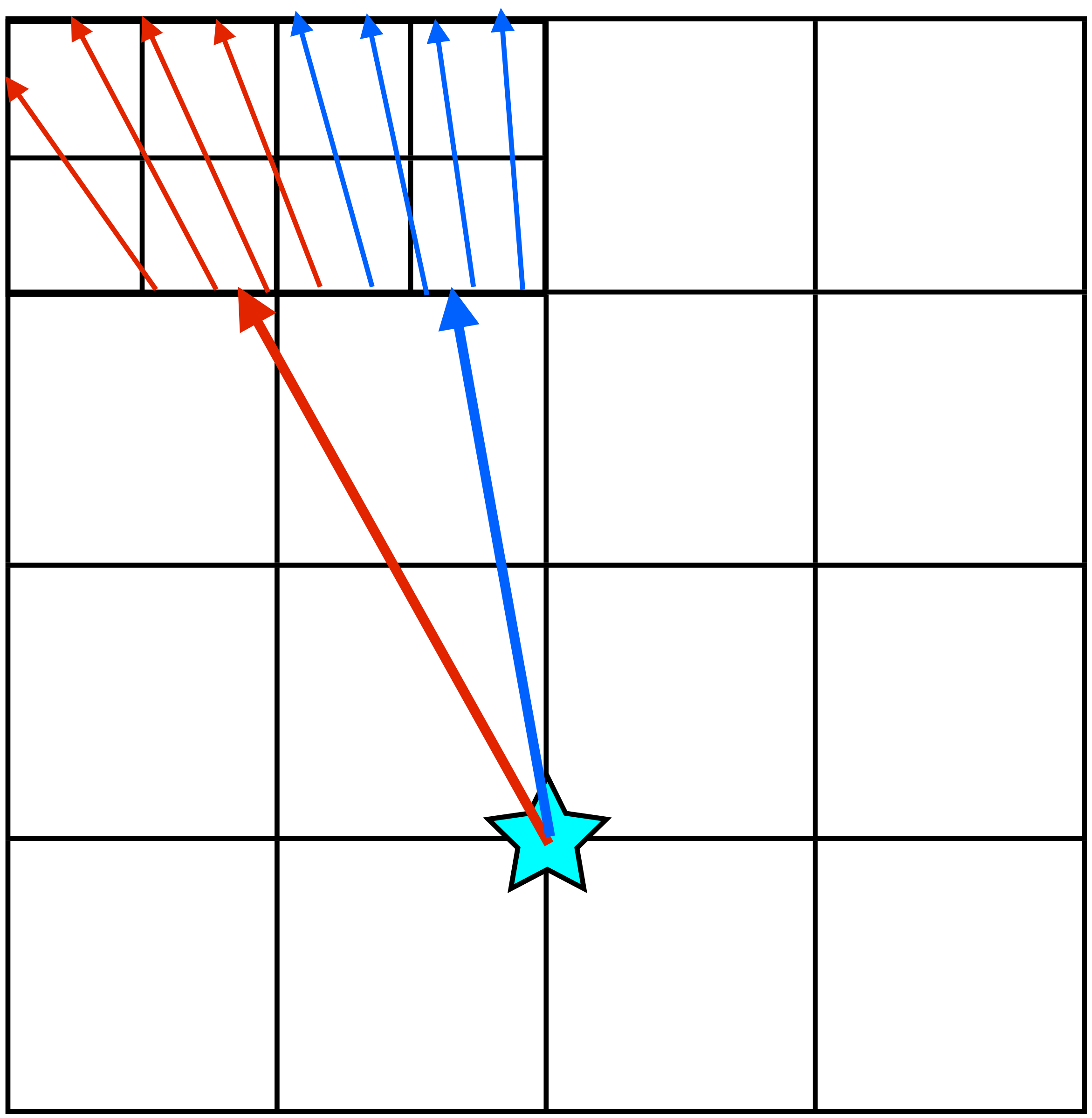
$$\frac{1}{c}\frac{\partial P}{\partial t} + \frac{\partial P}{\partial r} = -\kappa P$$

- P := photon flux in the ray

Adaptive Ray Tracing (Enzo+Moray)

Abel & Wandelt (2002)
Wise & Abel (2011)

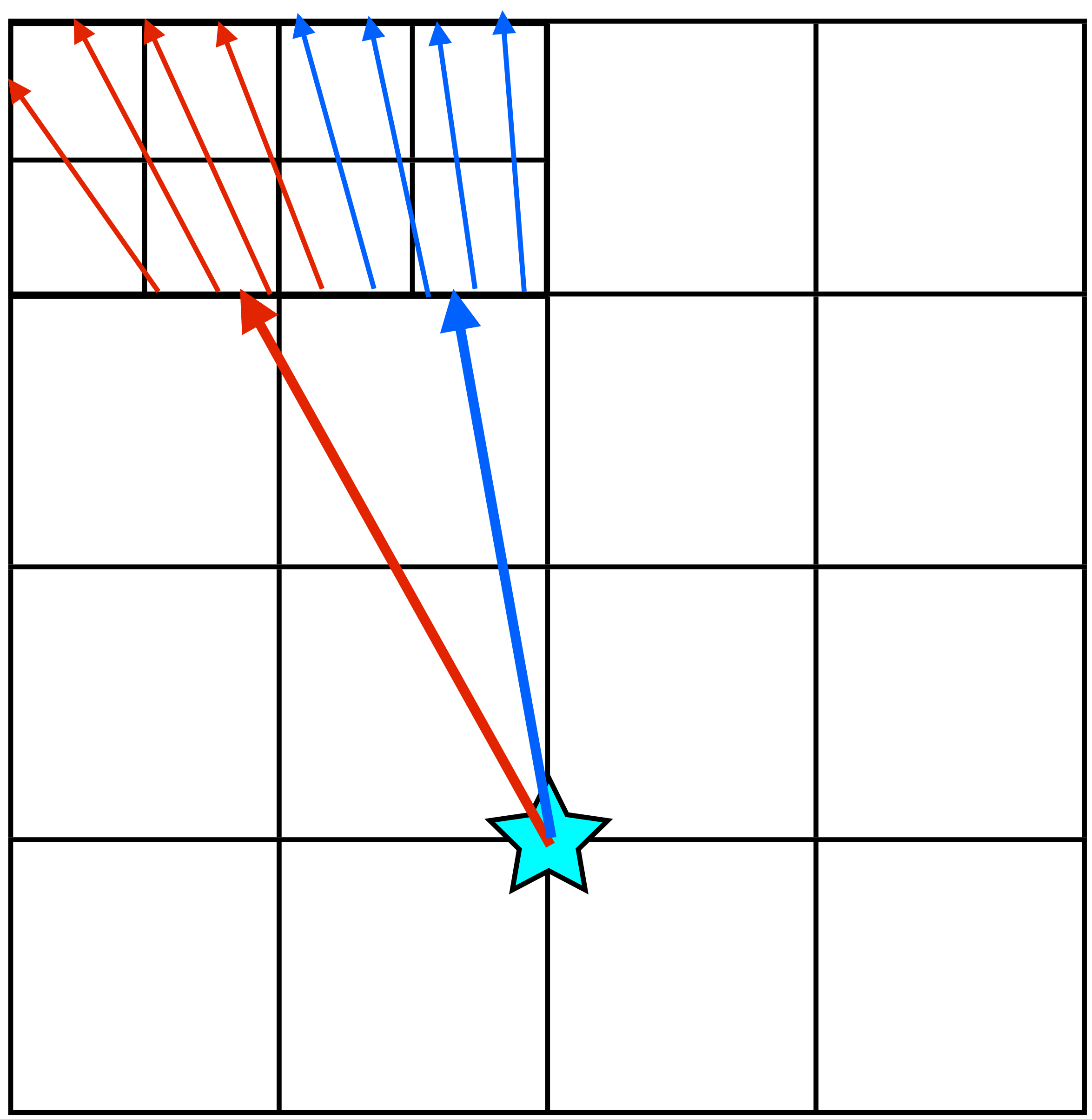
- Ray directions and splitting based on HEALPix (Gorski et al. 2005)
- Coupled with (magneto-) hydrodynamics of Enzo
- Rays are split into 4 child rays when the solid angle is large compared to the cell face area
- Well-suited for AMR
- Can calculate the photo-ionization rates so that the method is photon conserving.
- MPI/OpenMP hybrid parallelized.



Adaptive Ray Tracing (Enzo+Moray)

Abel & Wandelt (2002)
Wise & Abel (2011)

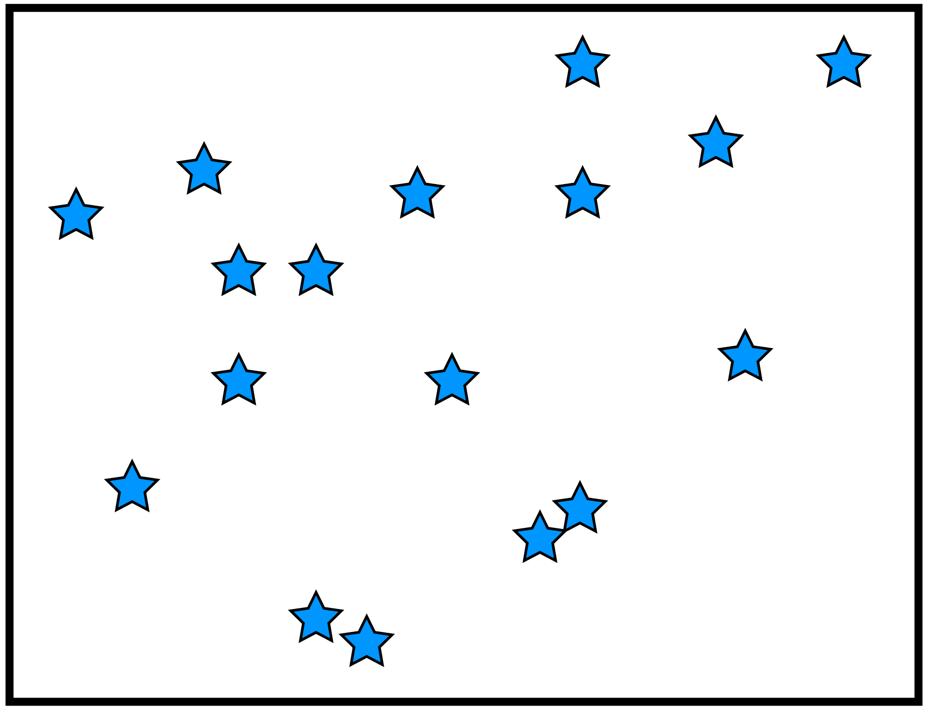
- H + He ionization (heating)
- X-rays (secondary ionizations)
- Lyman-Werner transfer (based on Draine & Bertoldi shielding function)
- Choice between energy discretization and general spectral shapes (column density lookup tables, see C^2-Ray)
- See Mirocha+ (2012) for optimized choices for energy bins.
- Radiation pressure from continuum
- Choice between c = Ac, ∞
- Can delete a ray when its flux drops below some fraction of the UVB for local UV feedback.



Overcoming $O(N_{\mathrm{star}})$ :: Ray / Source Merging

Okamoto et al. (2011)
Wise & Abel (in prep)

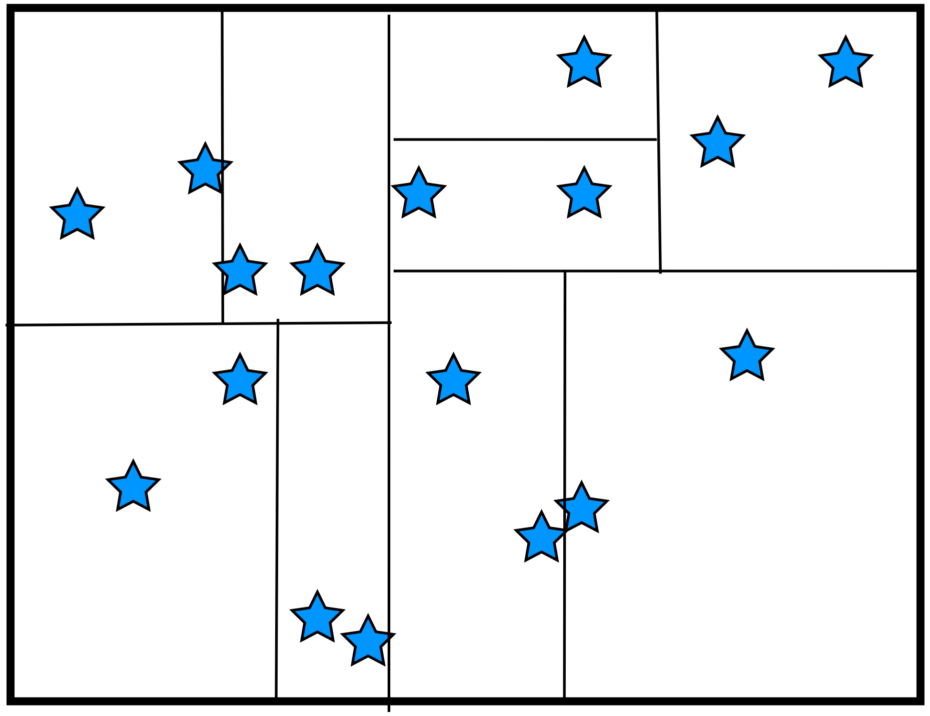
- Sources are grouped on a binary tree.
- On each leaf, a "super-source" is created that has the center of luminosity.
- After the ray travel ~3-5 times the source separation, the rays merge.
- Recursive.
- Have run simulations with 25k point sources.



OVERCOMING O(N_{STAR}) :: RAY / SOURCE MERGING

Okamoto et al. (2011)
Wise & Abel (in prep)

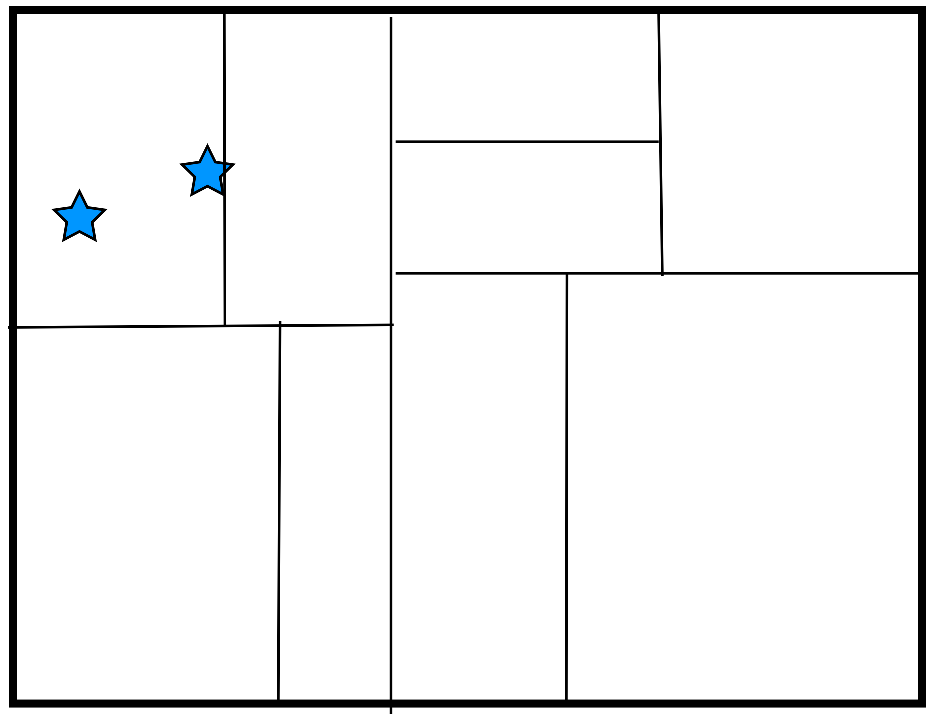
- Sources are grouped on a binary tree.
- On each leaf, a "super-source" is created that has the center of luminosity.
- After the ray travel ~3-5 times the source separation, the rays merge.
- Recursive.
- Have run simulations with 25k point sources.



OVERCOMING O(N_{STAR}) :: RAY / SOURCE MERGING

Okamoto et al. (2011)
Wise & Abel (in prep)

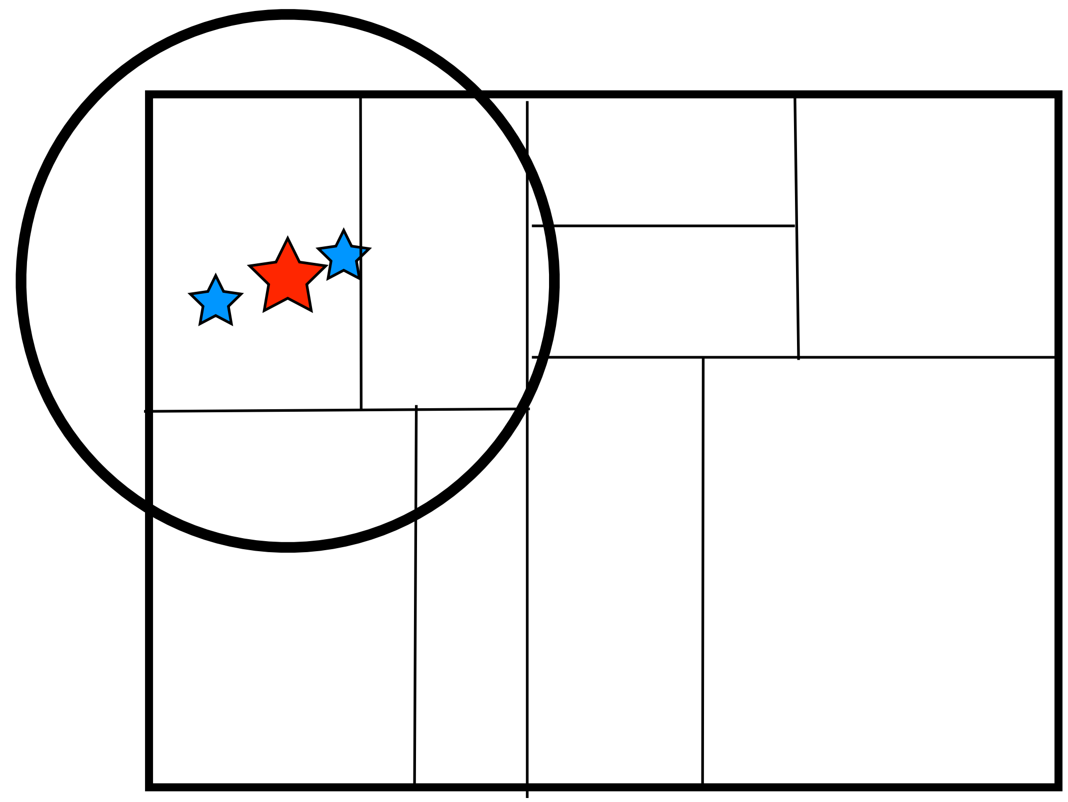
- Sources are grouped on a binary tree.
- On each leaf, a "super-source" is created that has the center of luminosity.
- After the ray travel ~3-5 times the source separation, the rays merge.
- Recursive.
- Have run simulations with 25k point sources.



OVERCOMING O(N_{STAR}) :: RAY / SOURCE MERGING

Okamoto et al. (2011)
Wise & Abel (in prep)

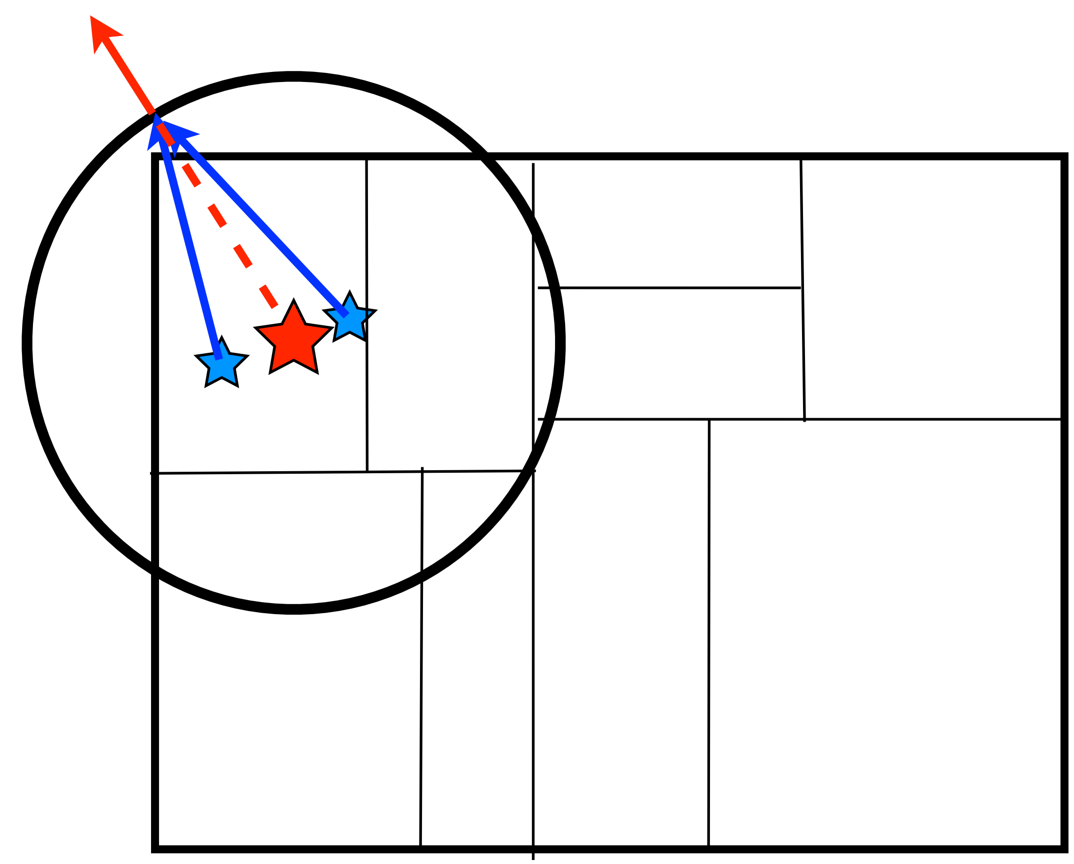
- Sources are grouped on a binary tree.
- On each leaf, a "super-source" is created that has the center of luminosity.
- After the ray travel ~3-5 times the source separation, the rays merge.
- Recursive.
- Have run simulations with 25k point sources.



Overcoming O(N_{STAR}) :: Ray / Source Merging

Okamoto et al. (2011)
Wise & Abel (in prep)

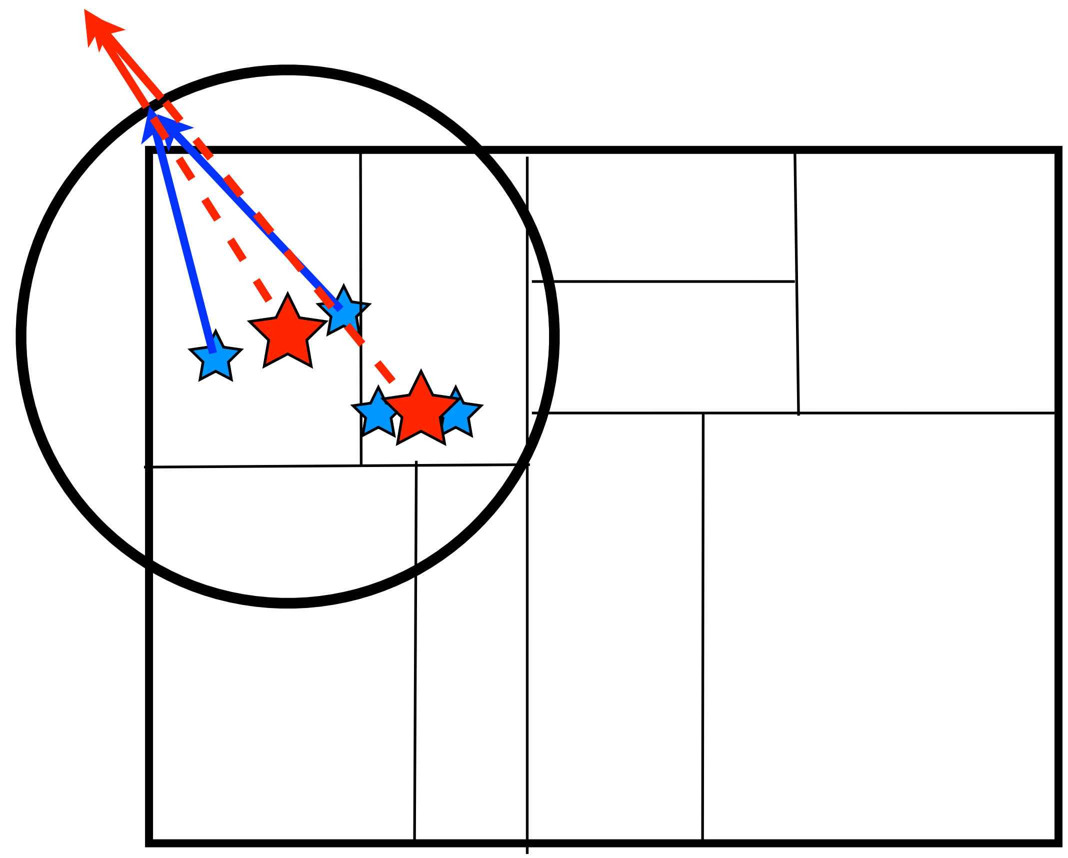
- Sources are grouped on a binary tree.
- On each leaf, a "super-source" is created that has the center of luminosity.
- After the ray travel ~3-5 times the source separation, the rays merge.
- Recursive.
- Have run simulations with 25k point sources.



OVERCOMING O(N_{STAR}) :: RAY / SOURCE MERGING

Okamoto et al. (2011)
Wise & Abel (in prep)

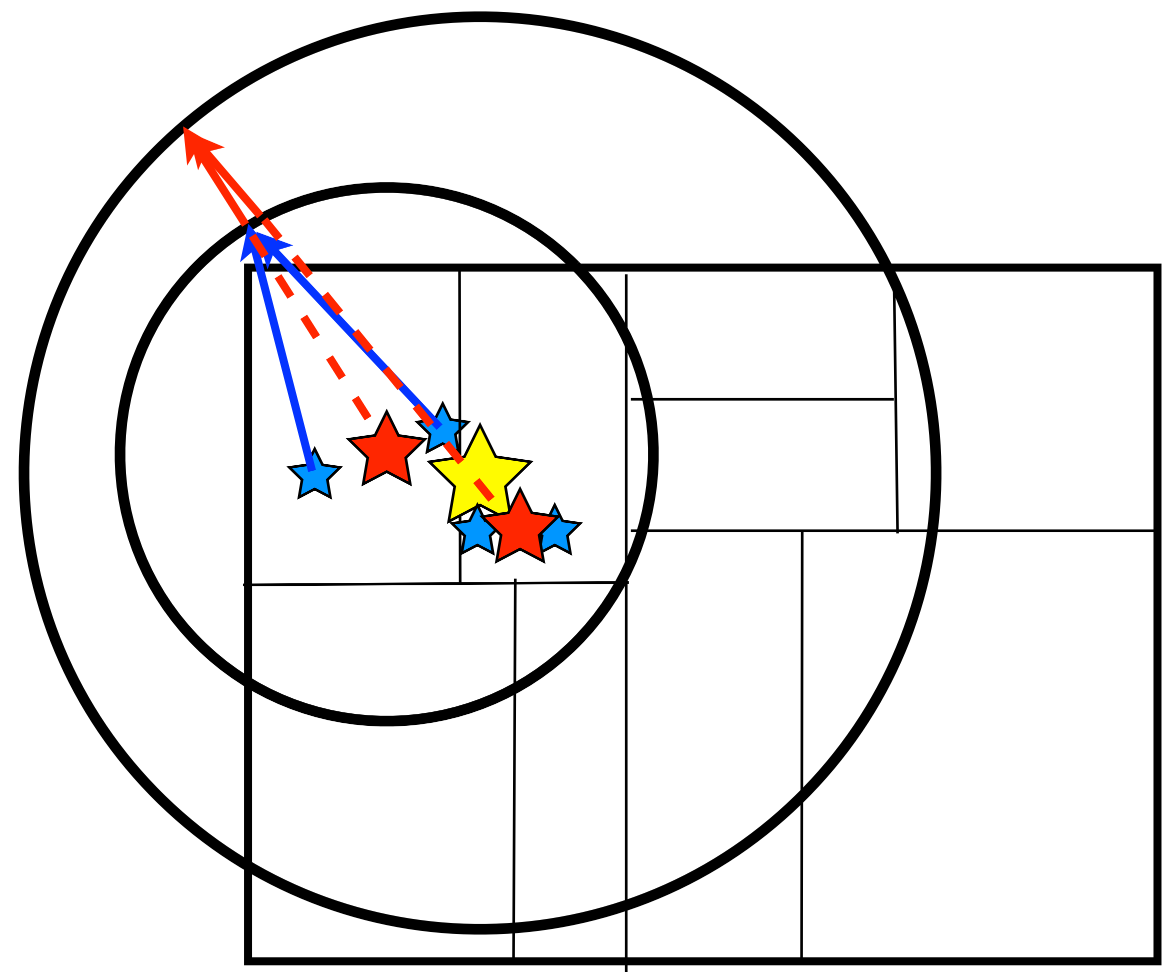
- Sources are grouped on a binary tree.
- On each leaf, a "super-source" is created that has the center of luminosity.
- After the ray travel ~3-5 times the source separation, the rays merge.
- Recursive.
- Have run simulations with 25k point sources.



Overcoming $O(N_{STAR})$ :: Ray / Source Merging

Okamoto et al. (2011)
Wise & Abel (in prep)

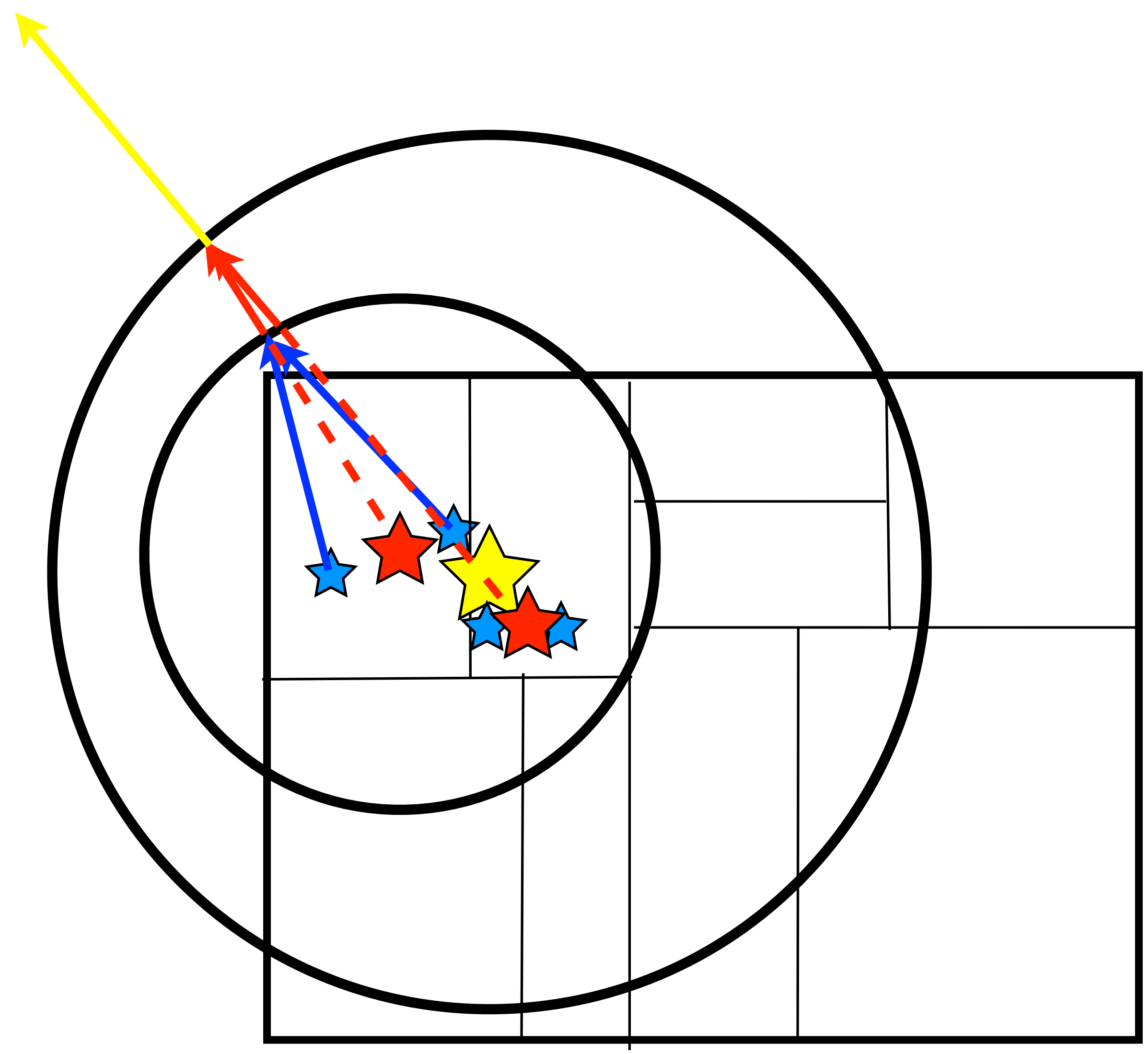
- Sources are grouped on a binary tree.
- On each leaf, a "super-source" is created that has the center of luminosity.
- After the ray travel ~3-5 times the source separation, the rays merge.
- Recursive.
- Have run simulations with 25k point sources.



Overcoming O(N_{STAR}) :: Ray / Source Merging

Okamoto et al. (2011)
Wise & Abel (in prep)

- Sources are grouped on a binary tree.
- On each leaf, a "super-source" is created that has the center of luminosity.
- After the ray travel ~3-5 times the source separation, the rays merge.
- Recursive.
- Have run simulations with 25k point sources.



+ Radiative Feedback

- Radiative transfer ON
- Minimum rays per cell (angular resolution)
- Hydrogen photo-ionization only
- Radiation periodic boundary
- Ray merging ON
- Ray merging radius (in units of separation of source pairs)

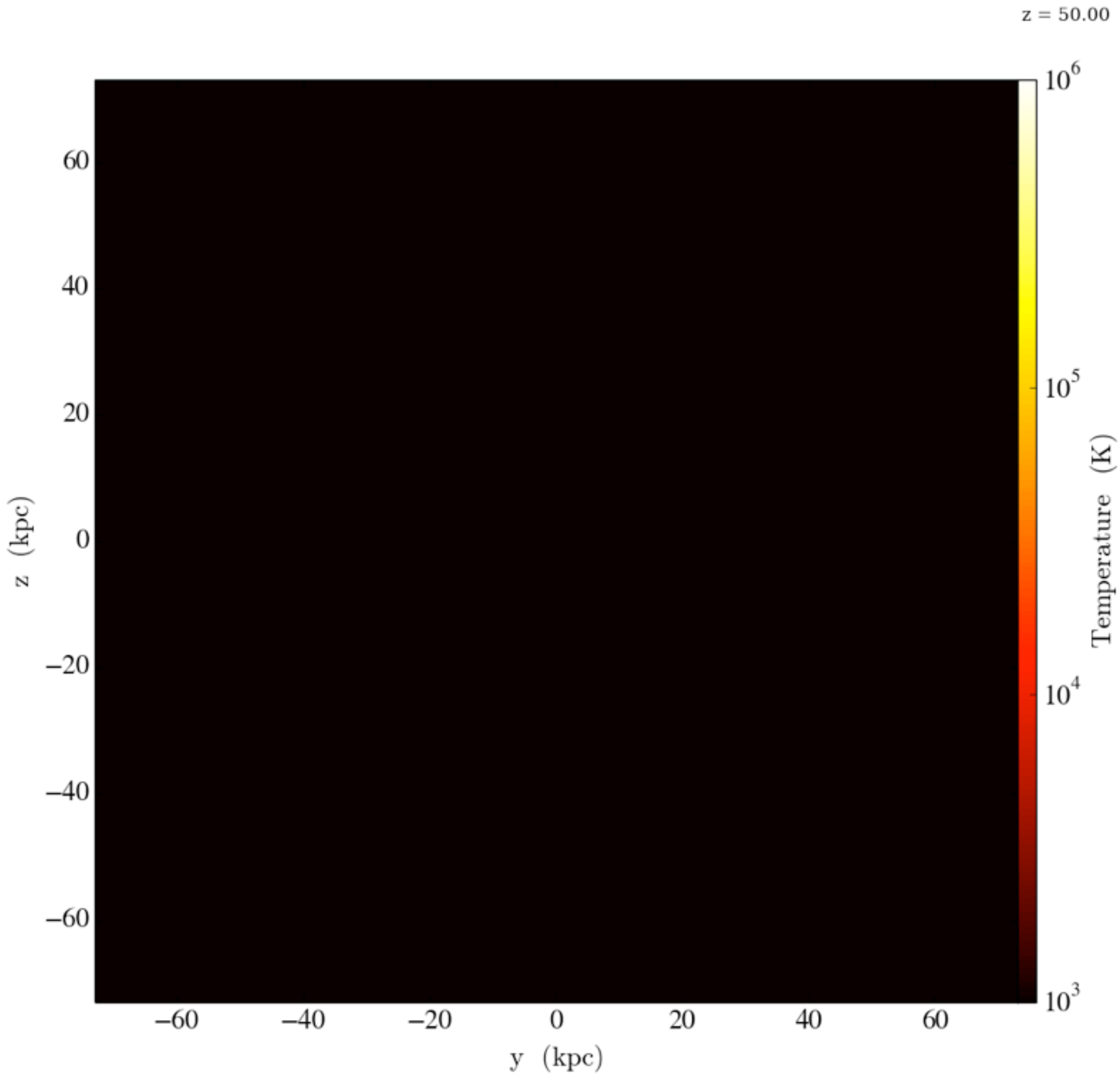
`sapporo_cosmo/stars-rt.enzo`

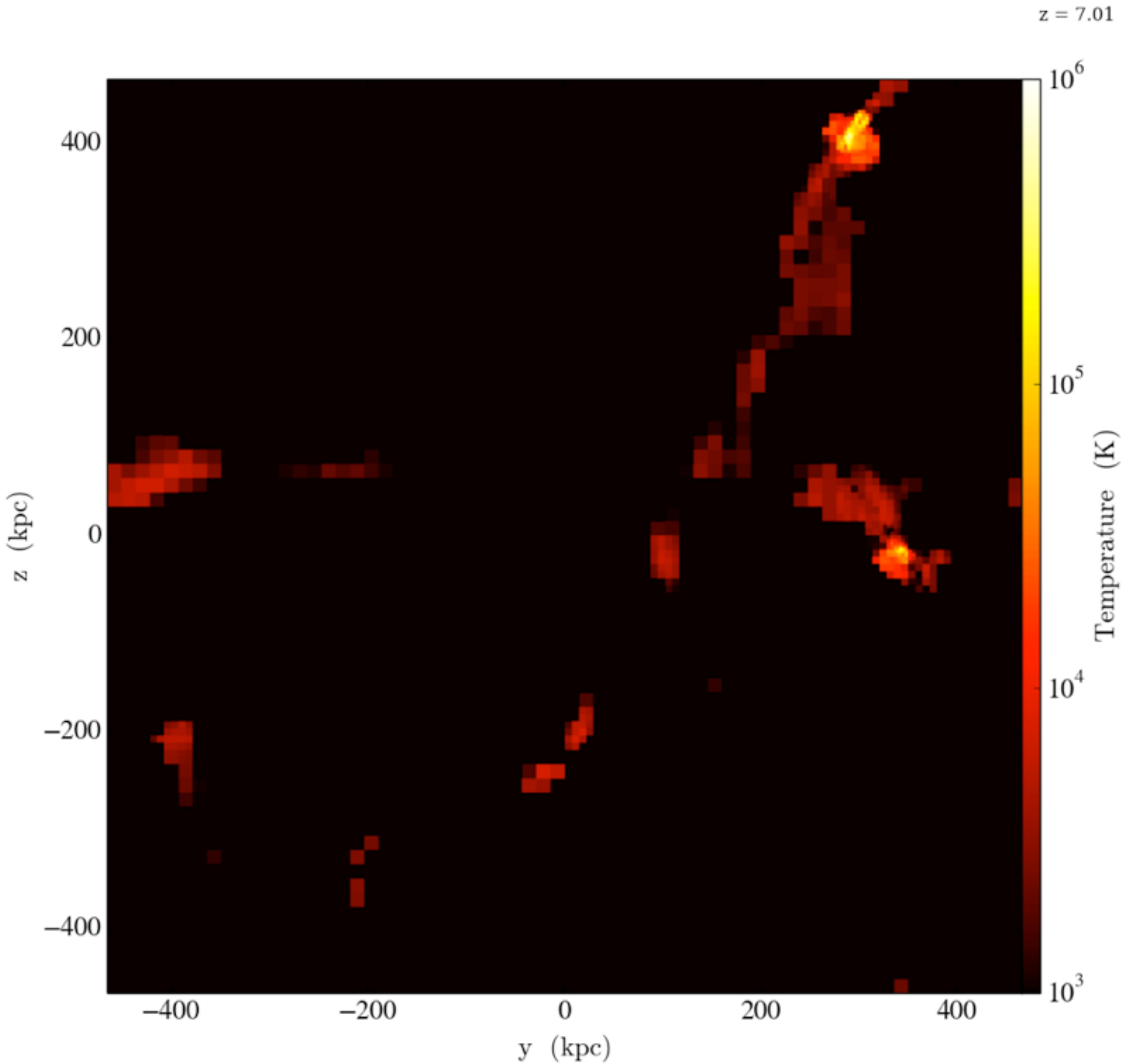
```
#
# Star formation and feedback parameters
#
RadiativeTransfer = 1
RadiativeTransferRaysPerCell              = 3.100000
RadiativeTransferFluxBackgroundLimit      = 0.01
RadiativeTransferInitialHEALPixLevel      = 1
RadiativeTransferHydrogenOnly             = 1
RadiativeTransferOpticallyThinH2          = 0
RadiativeTransferPeriodicBoundary         = 1
RadiativeTransferAdaptiveTimestep         = 1
RadiativeTransferSourceClustering         = 1
RadiativeTransferPhotonMergeRadius        = 3.0
```

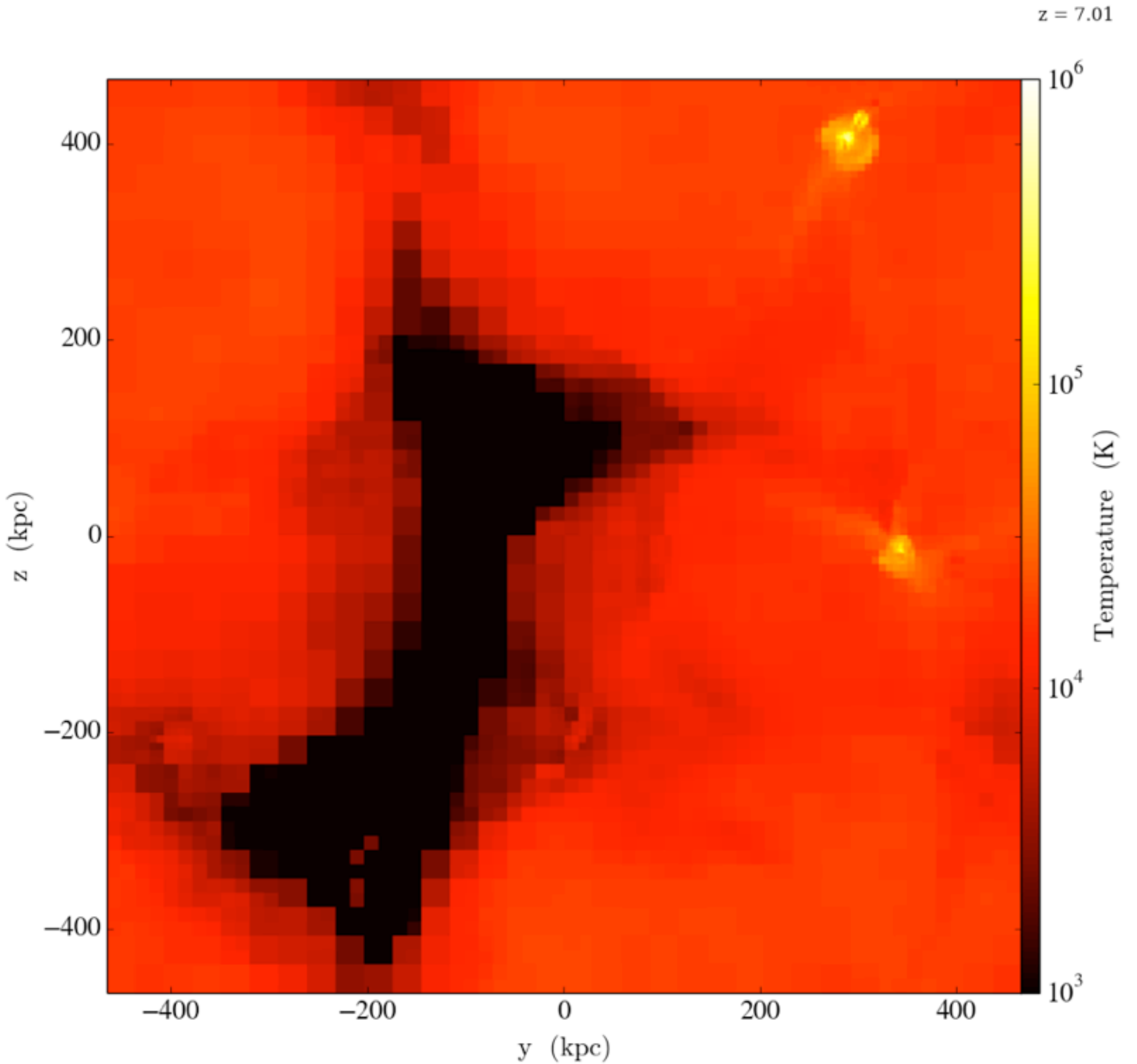
+ Radiative Transfer

- This simulation only runs to z = 7, so we can run this simulation.

```
cd ~/sapporo_cosmo
mkdir RT
cp stars-rt.enzo RT
cp sapporo_cosmo_nbody/Grid* sapporo_cosmo_nbody/Particle* RT
cp ~/enzo-stable/input/hm12_photorates.dat RT
cd RT
cp ~/enzo-stable/src/enzo/enzo.exe .
./enzo.exe -d stars-rt.enzo
```


z = 50.00
60
40
20
0
−20
−40
−60
z (kpc)
−60
−40
−20
0
20
40
60
y (kpc)
10^6
10^5
10^4
10^3
Temperature (K)


z = 7.01
400
200
0
−200
−400
z (kpc)
−400
−200
0
200
400
y (kpc)
10^6
10^5
10^4
10^3
Temperature (K)



z = 7.01
400
200
0
−200
−400
z (kpc)
−400
−200
0
200
400
y (kpc)
10^6
10^5
10^4
10^3
Temperature (K)

Summary

- Today we have covered some advanced topics in cosmology simulations.
- Usually when doing research, it is best to introduce physics progressively to understand the effect of each physical process.
- We have compared the same cosmological volume with the following physics.
    - Adiabatic equation of state
    - + Radiative cooling (H, He) and an ultraviolet radiation background
    - + Star formation and supernova feedback
    - + Stellar radiative feedback, using adaptive ray tracing