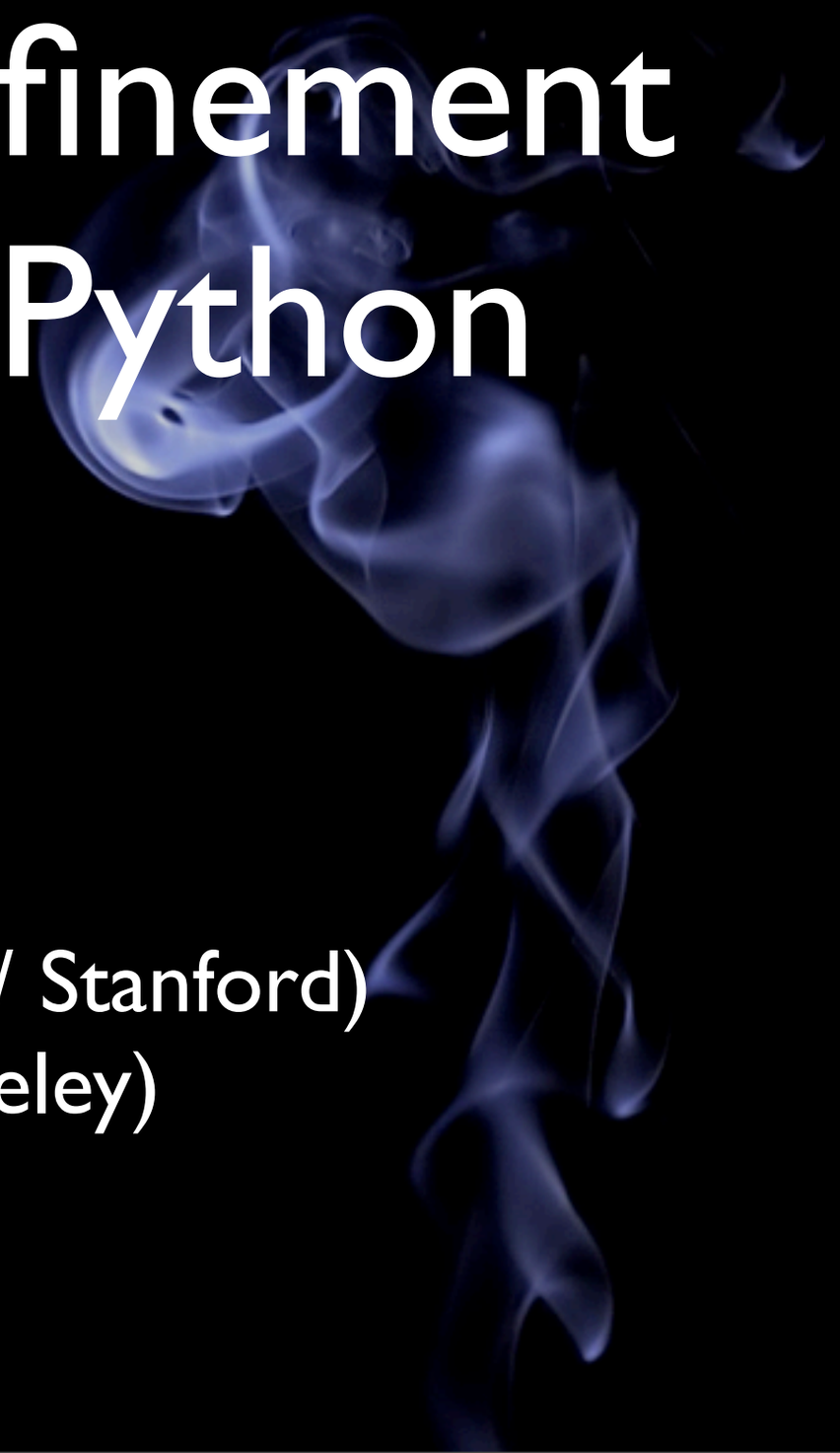
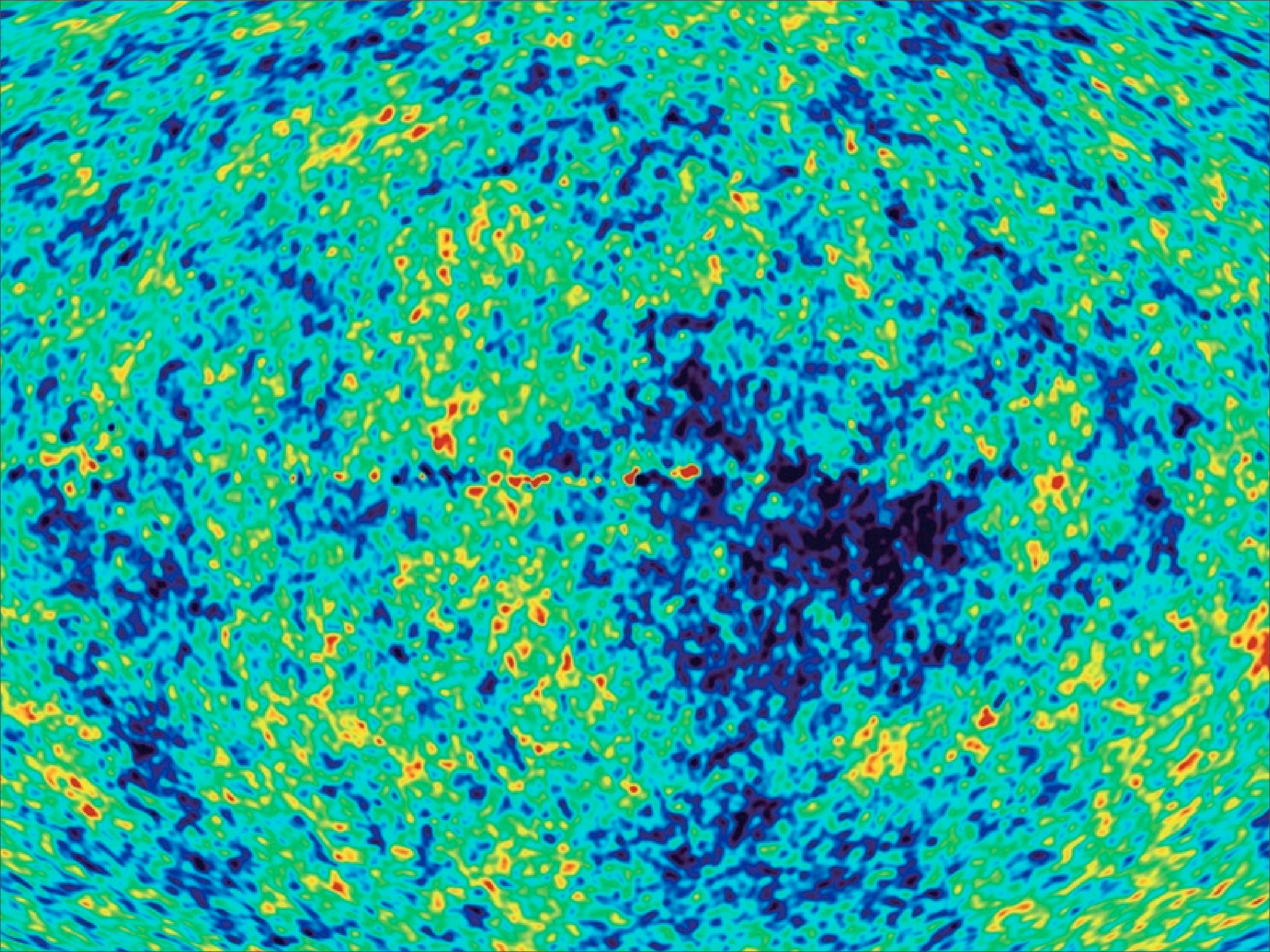
Adaptive Mesh Refinement Data Analysis in Python



Matthew Turk (KIPAC / SLAC / Stanford)
with Jeff Oishi (UC Berkeley)
SciPy 2008



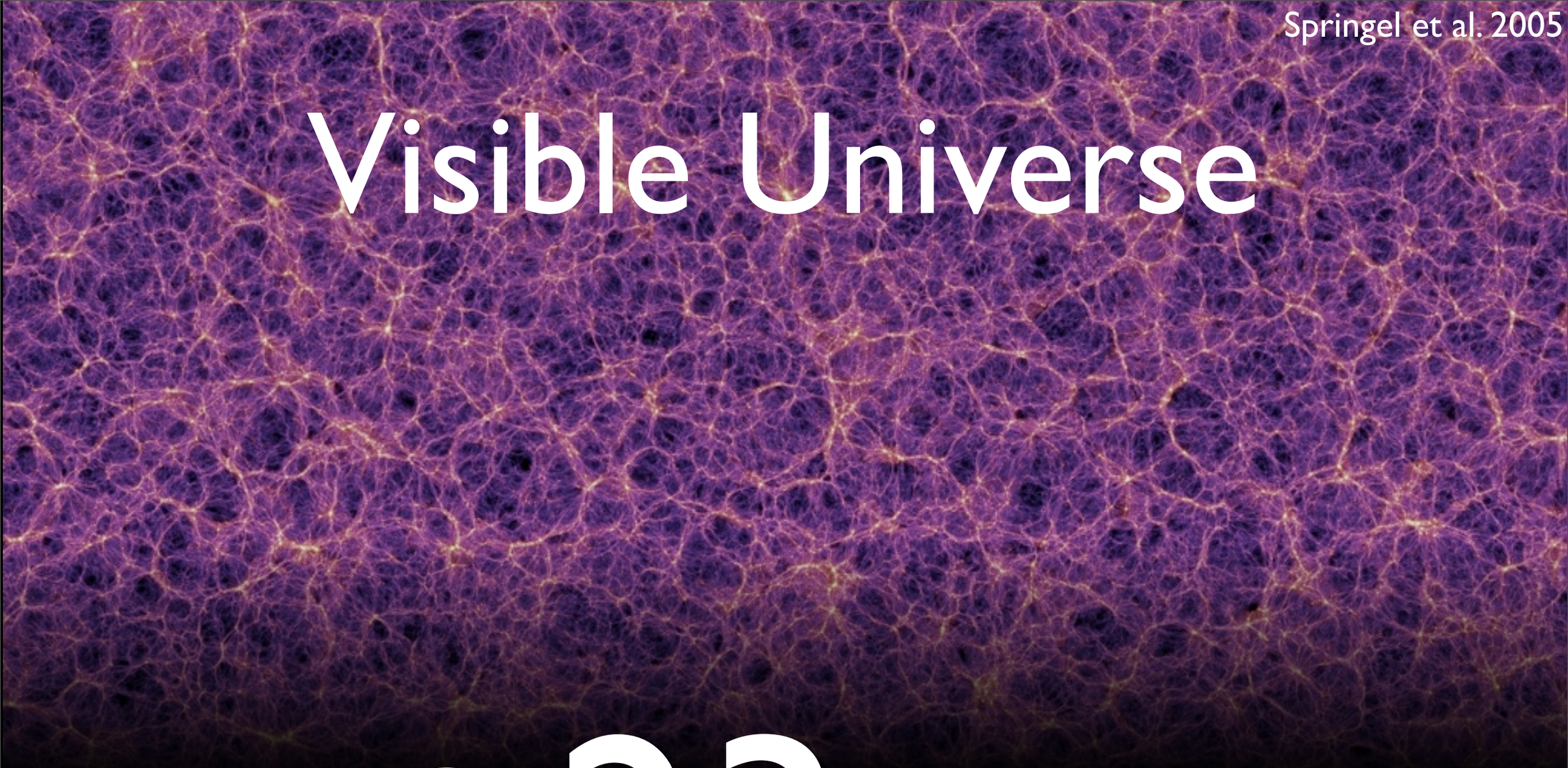
“Nothing”

“Nothing”

“Nothing” → Stuff.

Size

Springel et al. 2005
Visible Universe



10^{23} km

HST Image
Galaxy Cluster

10^{19} km

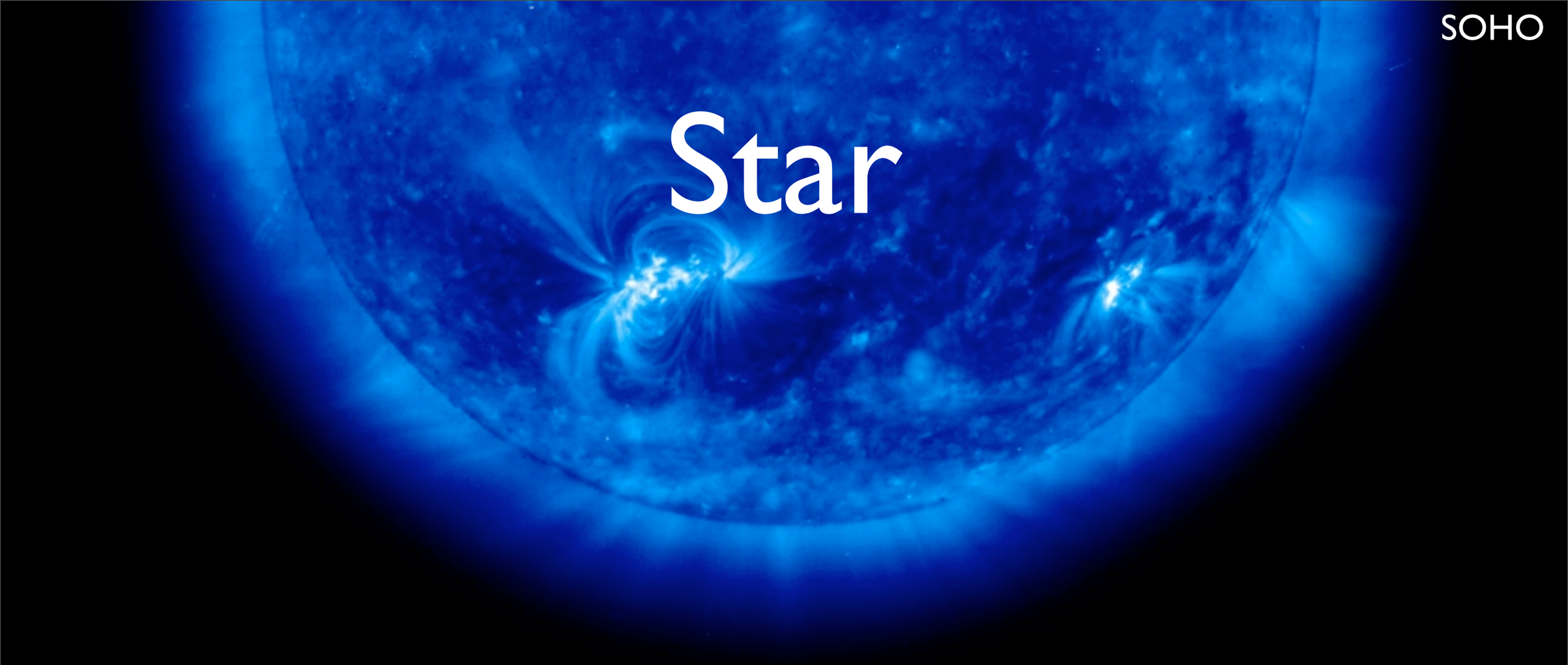
Galaxy
NASA/JPL-Caltech/R. Hurt

10^{17} km

Matthew T. Russell
Star Cluster

10^{15} km

Star



10^6 km

Earth

6000 km

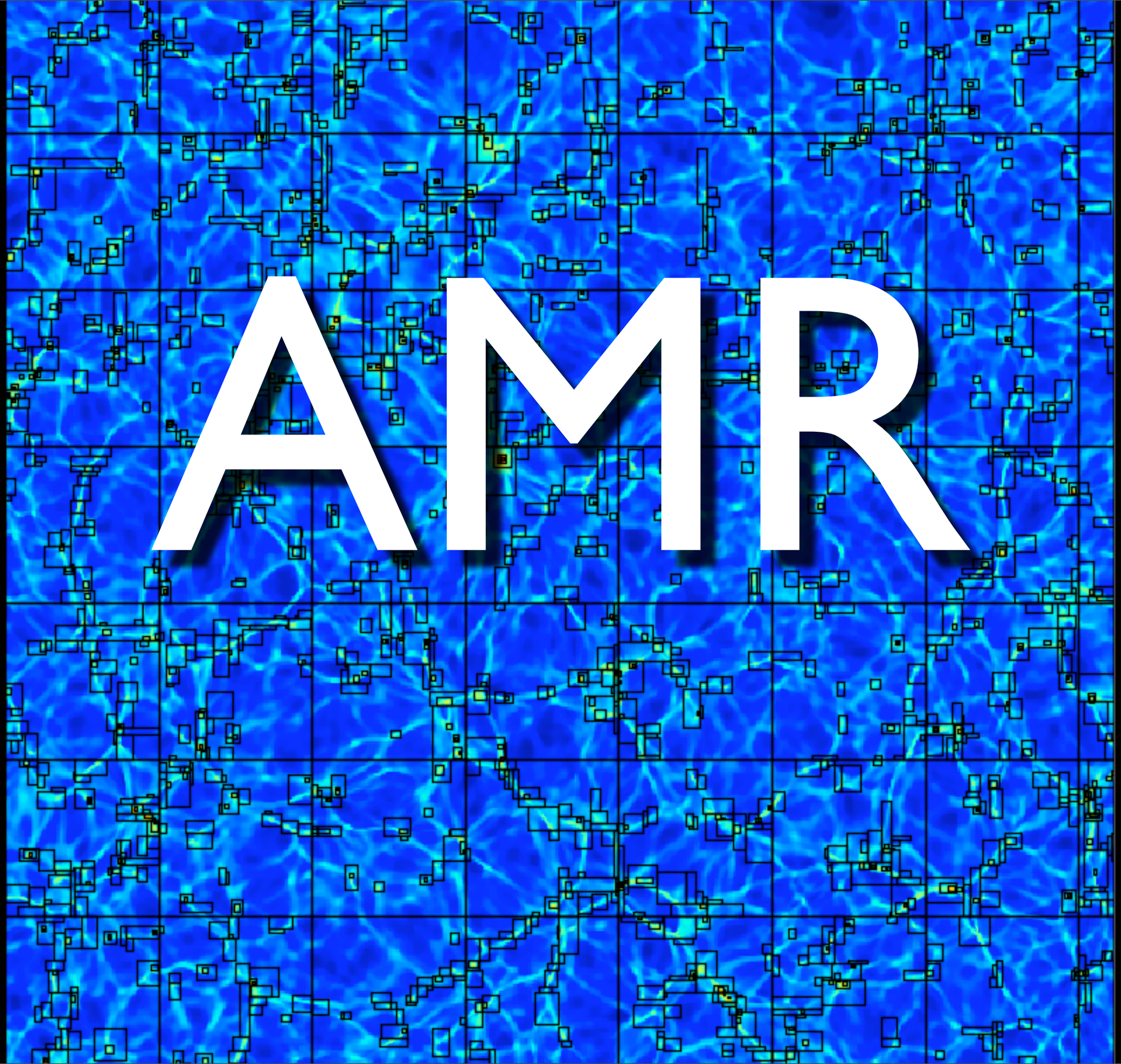
RENO
THE BIGGEST LITTLE CITY IN THE WORLD
EVERYONE IS
A GUARANTEE
WINNER
LUCKY
FOREST
2ND FLOOR

RENO
THE BIGGEST LITTLE CITY IN THE WORLD
EVERYONE IS
A GUARANTEE
WINNER
LUCKY
FOREST
2ND FLOOR

835 km

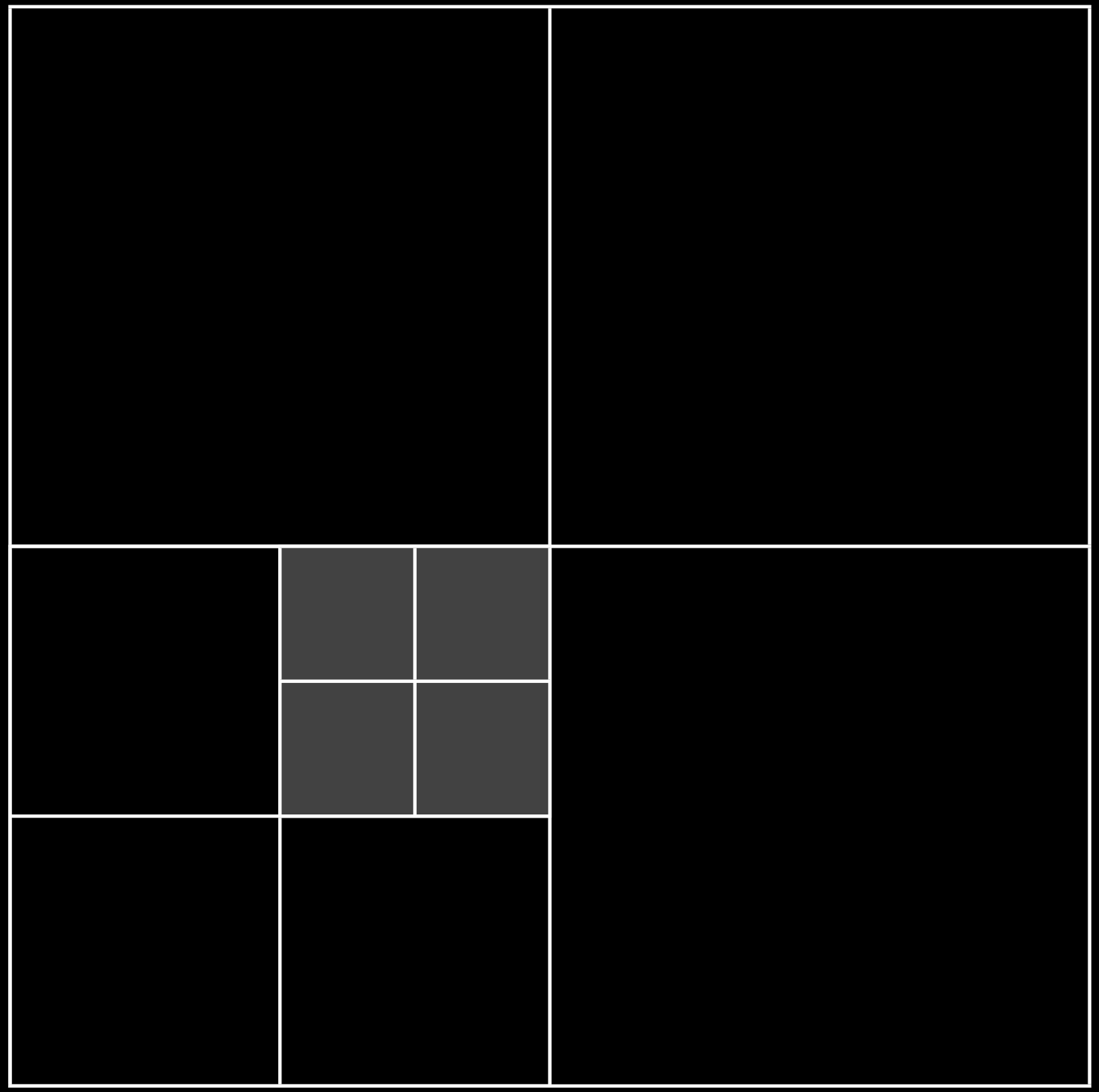
You & Me

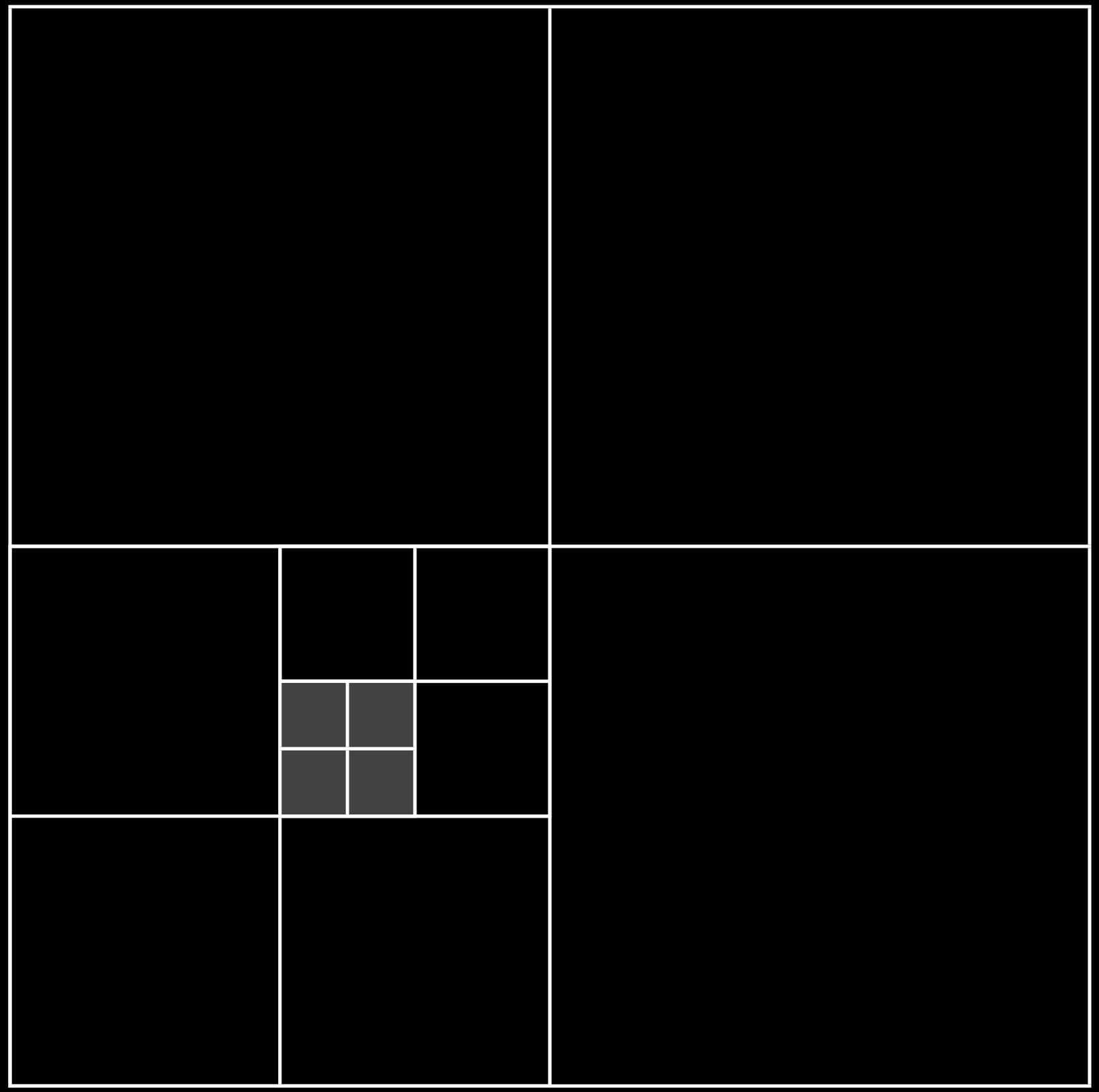
.002 km

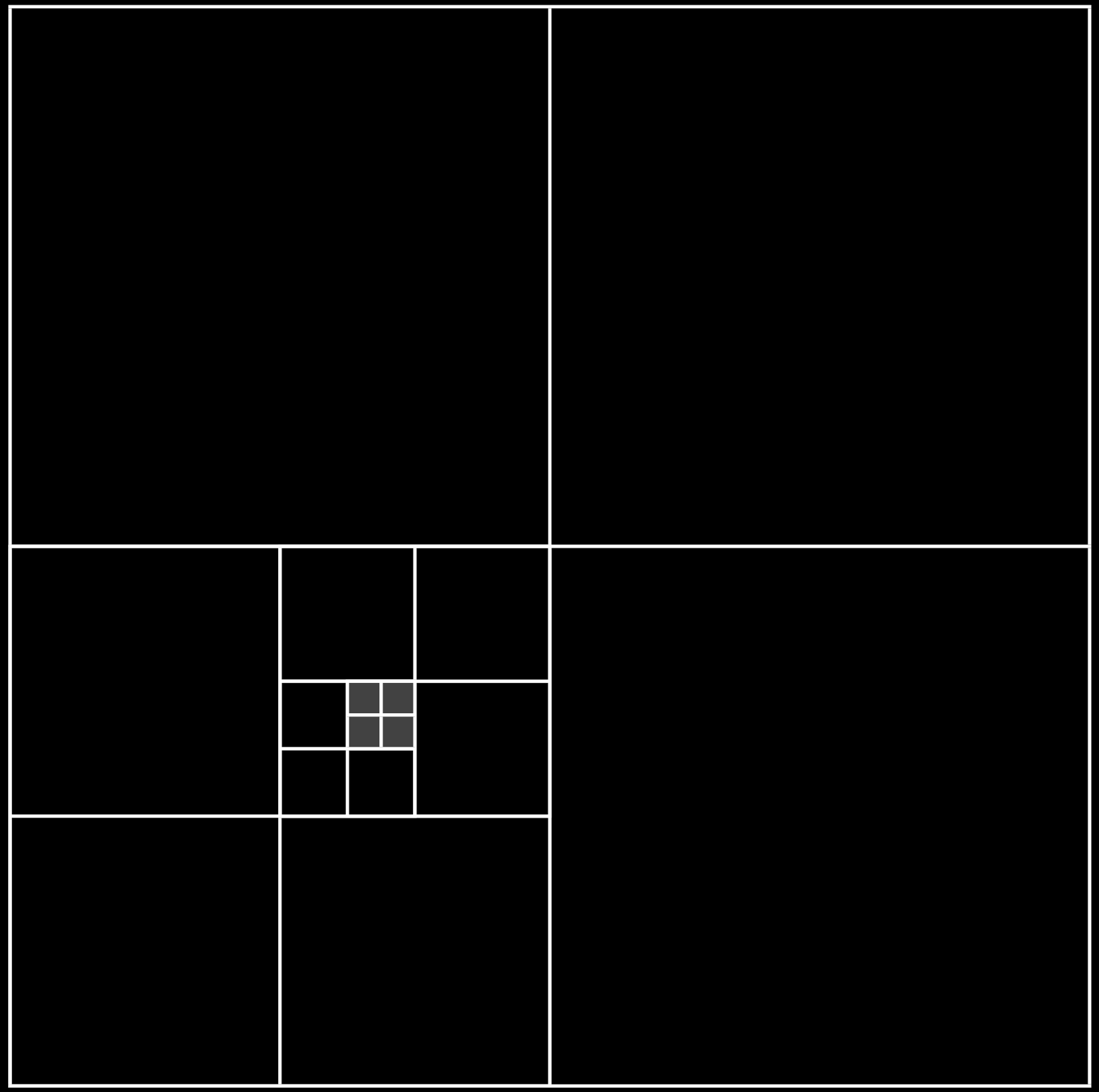

AMR

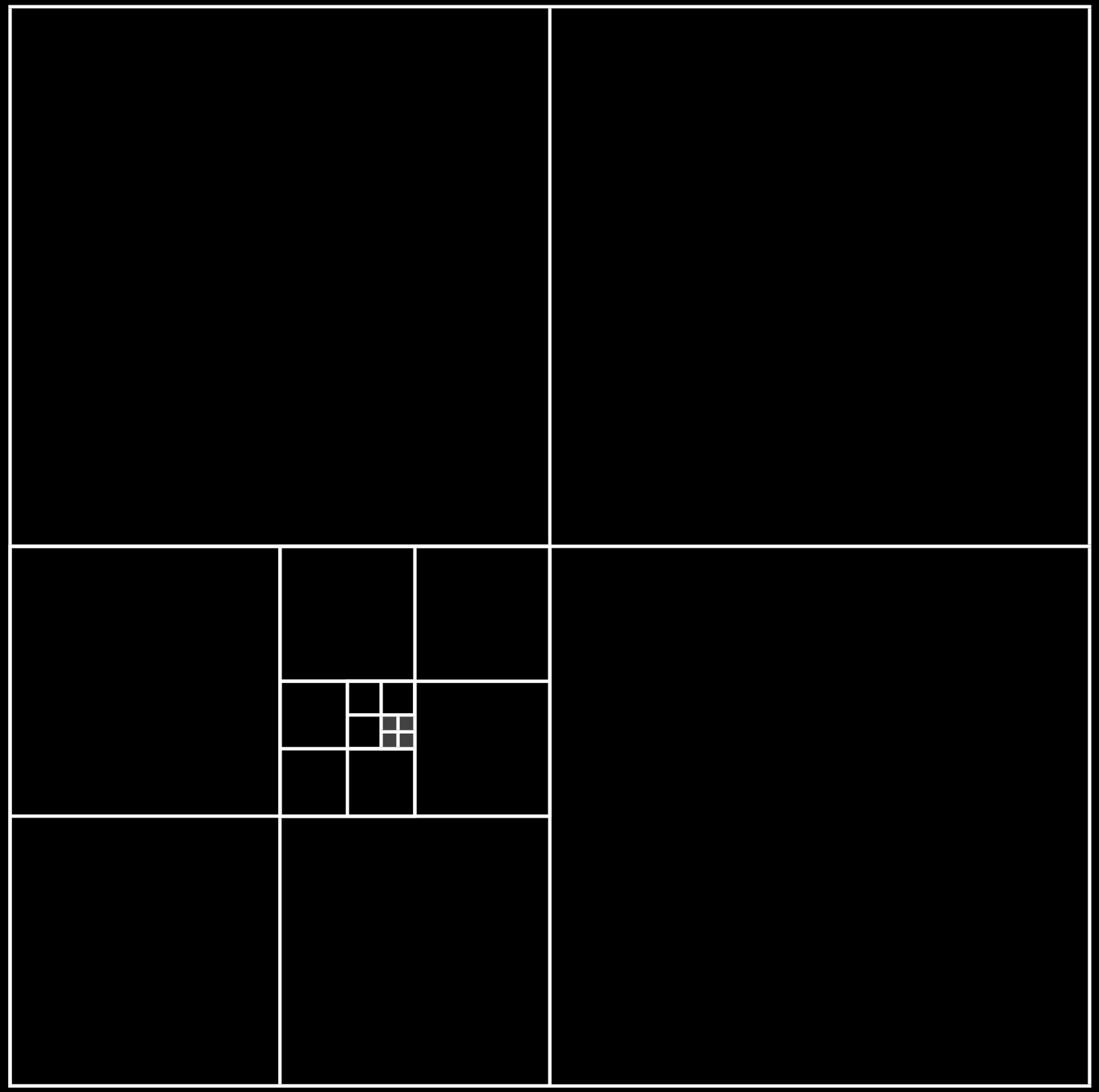
Universe

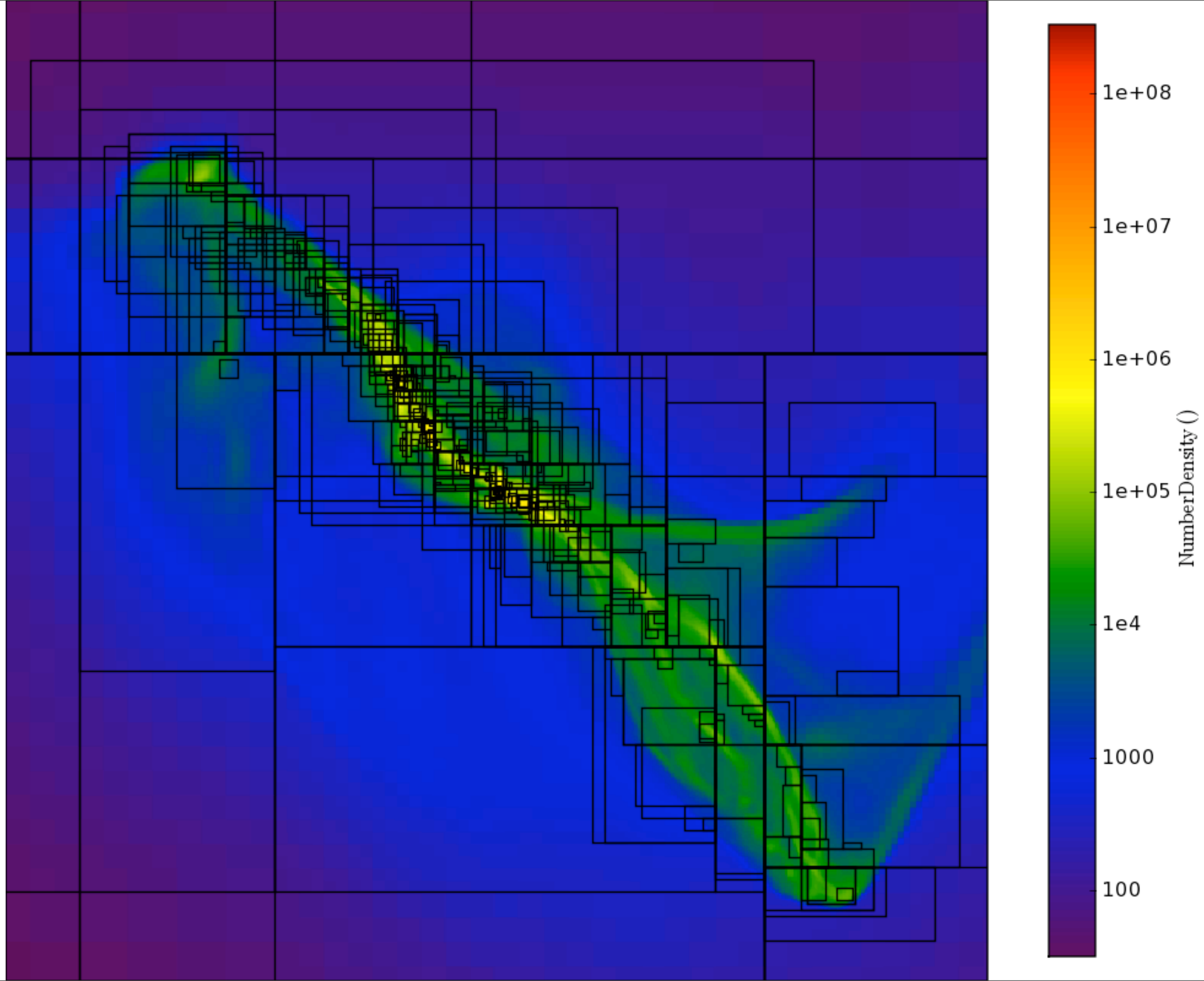
BORING	BORING
INTERESTING	BORING










1e+08
1e+07
1e+06
1e+05
1e4
1000
100
NumberDensity ()

enzo

Adaptive Mesh Refinement (patch-based)
N-body gravity, DM and Stars
Magnetohydrodynamics
Radiative Cooling
Radiative Transfer
12-species chemistry model

Black holes through galaxy clusters!

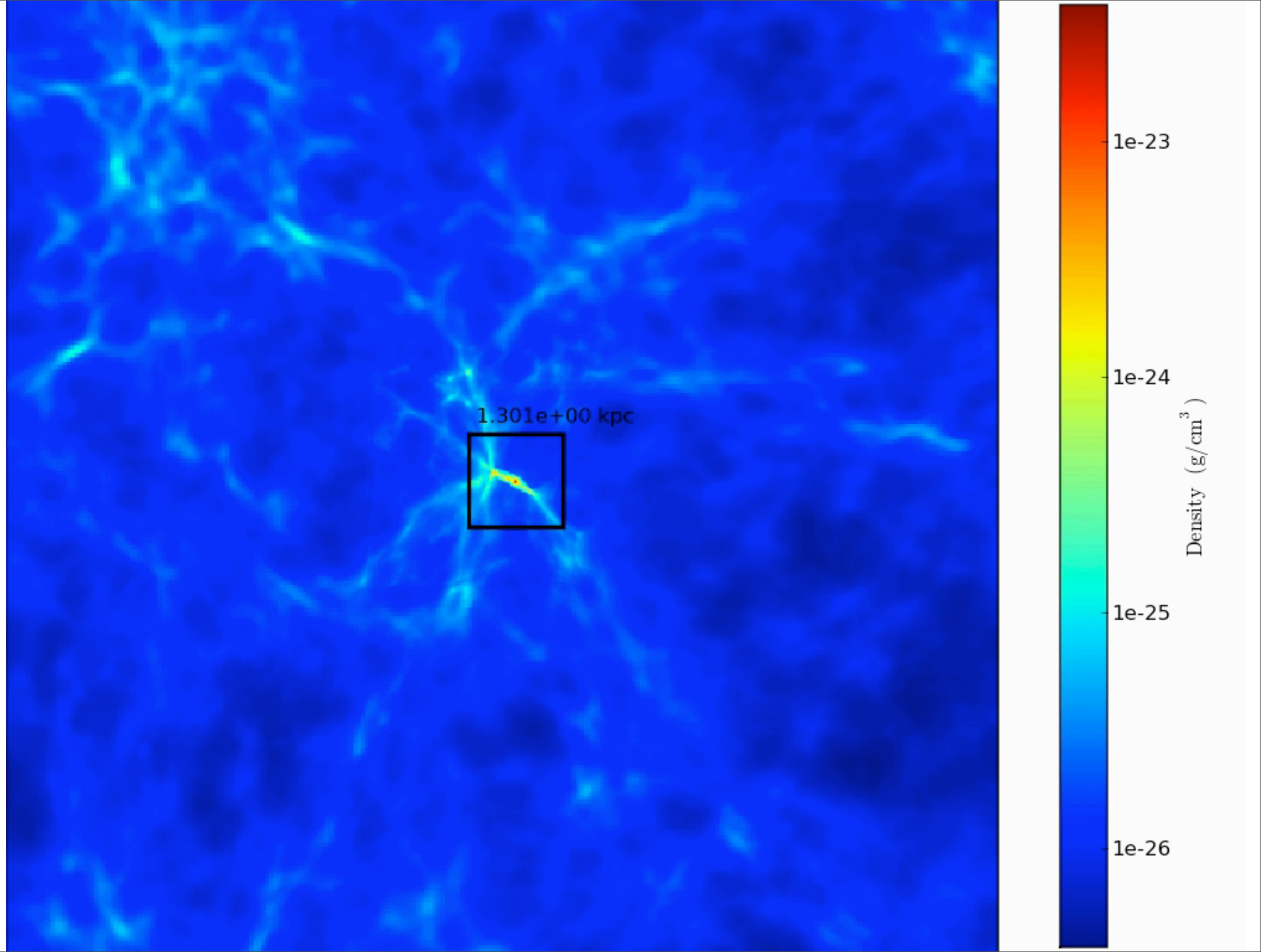
Developed and used at UCSD, Stanford, Columbia, CITA, Cambridge, ...

enzo

Adaptive Mesh Refinement (patch-based)
N-body gravity, DM and Stars
Magnetohydrodynamics
Radiative Cooling
Radiative Transfer
12-species chemistry model

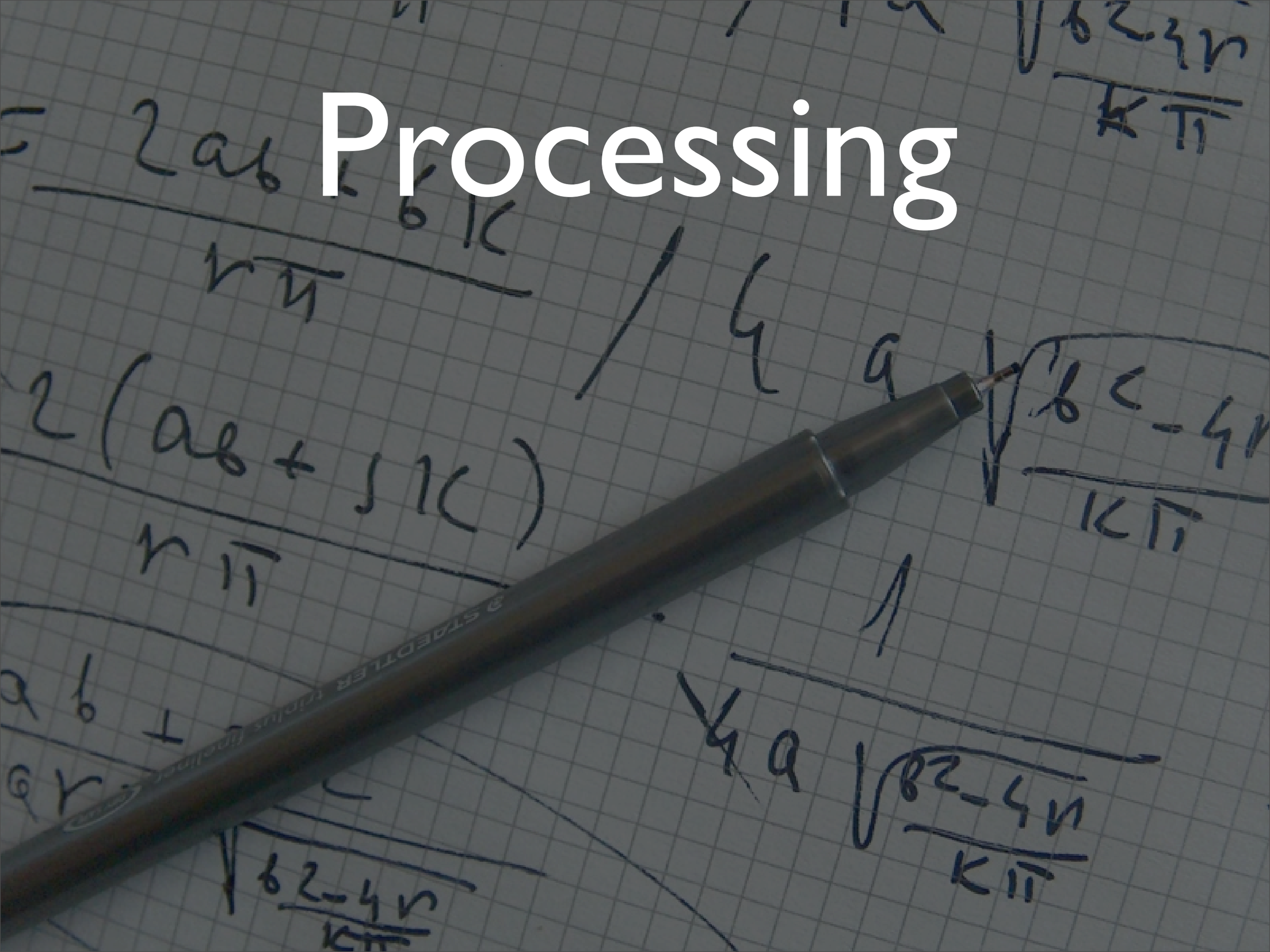
Black holes through galaxy clusters!

Developed and used at UCSD, Stanford, Columbia, CITA, Cambridge, ...


1.301e+00 kpc
1e-23
1e-24
1e-25
1e-26
Density (g/cm^3)

Requirements

Processing



Speed

Plotting

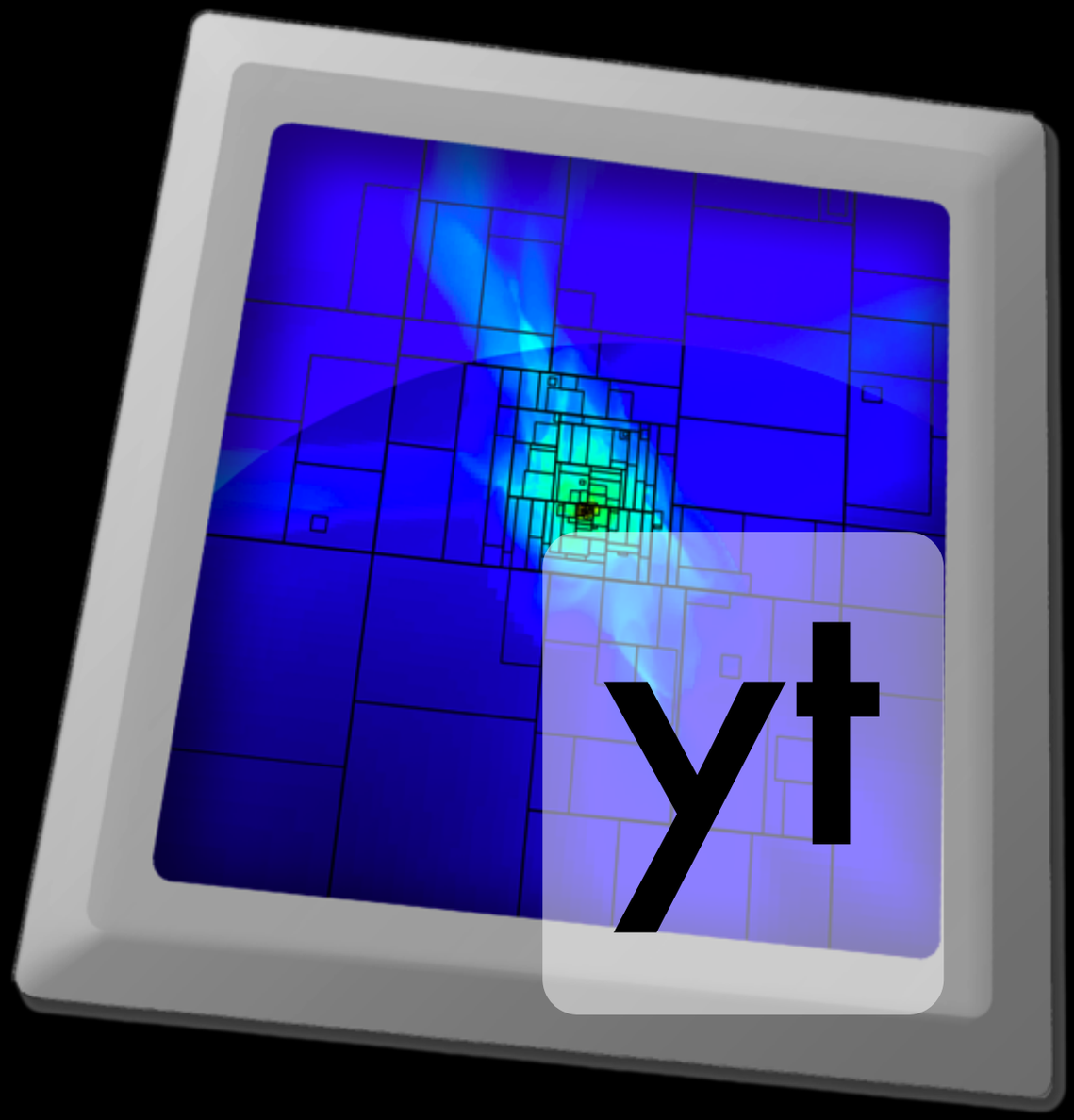
Stablility

Stable-ility

Free.

Free.

(completely.)

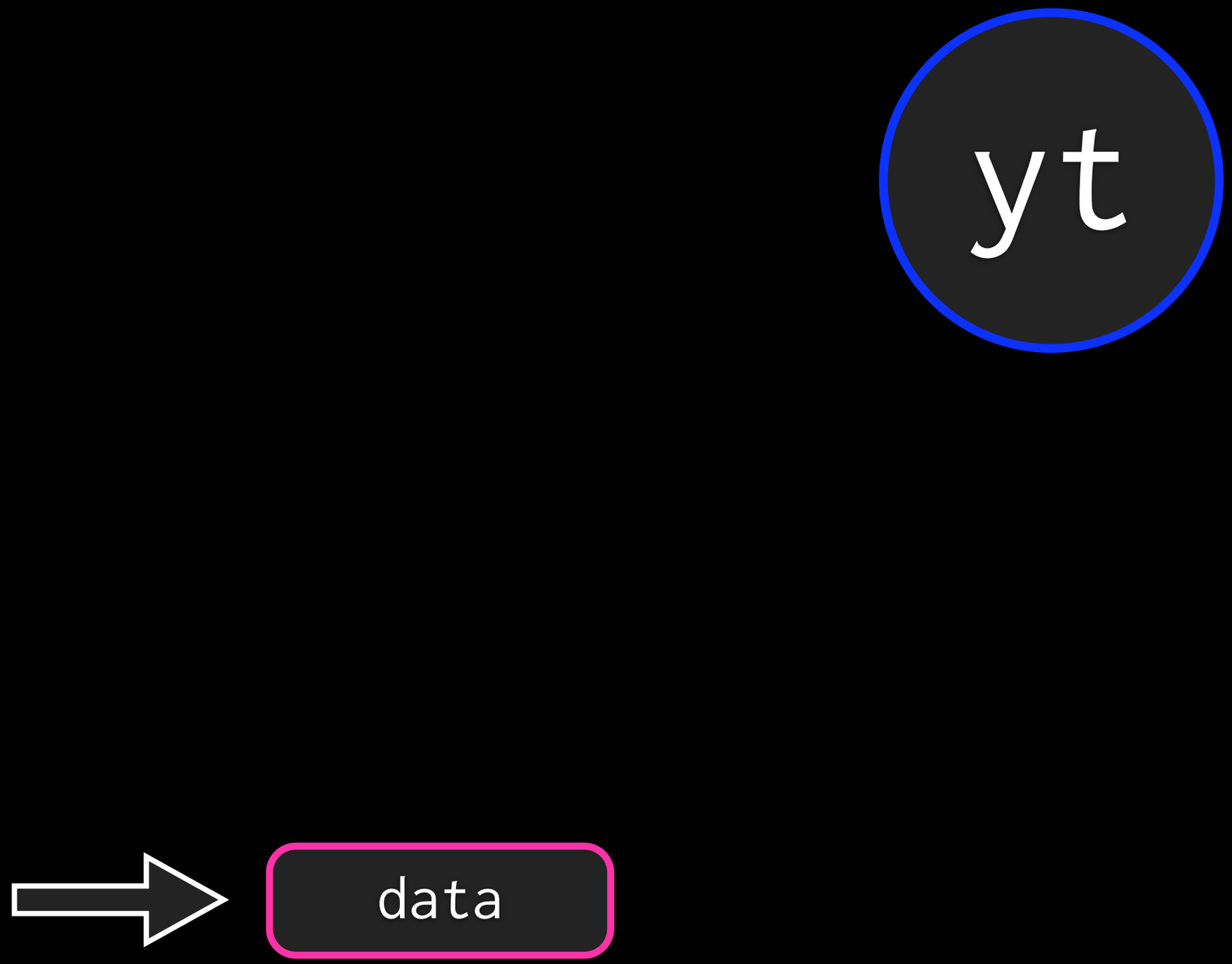

yt

Two Years Old

enzo

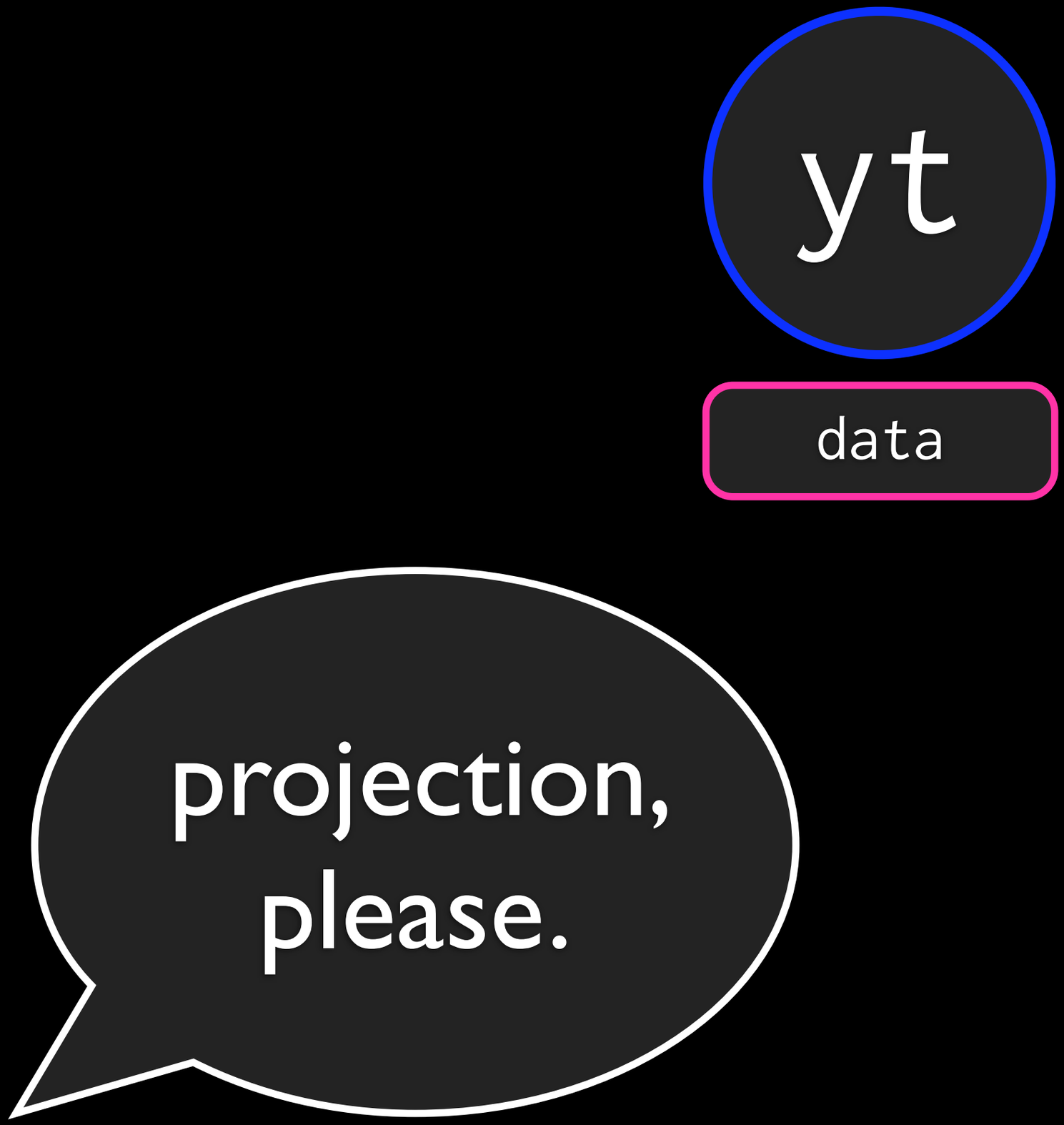
yt

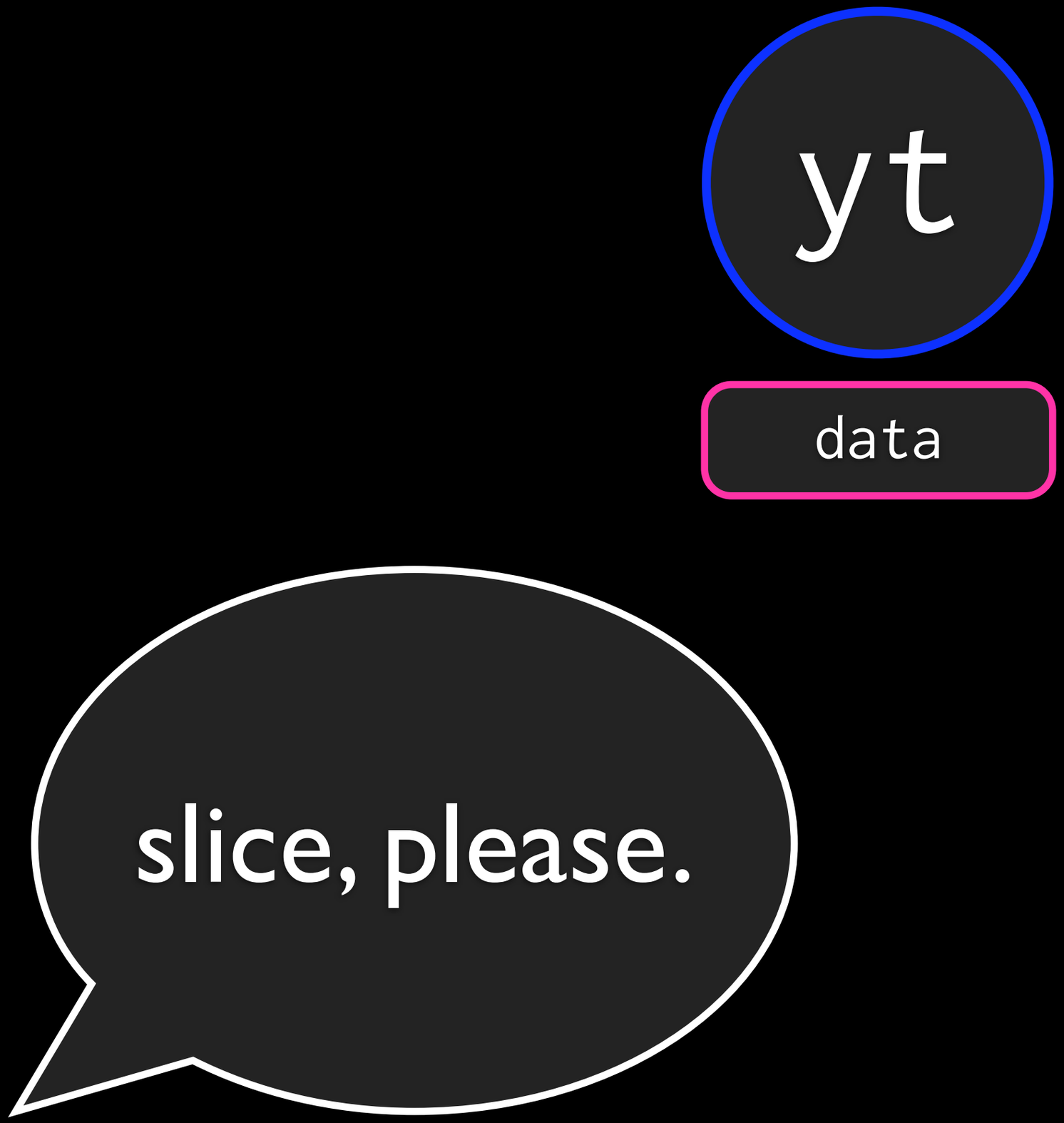
enzo

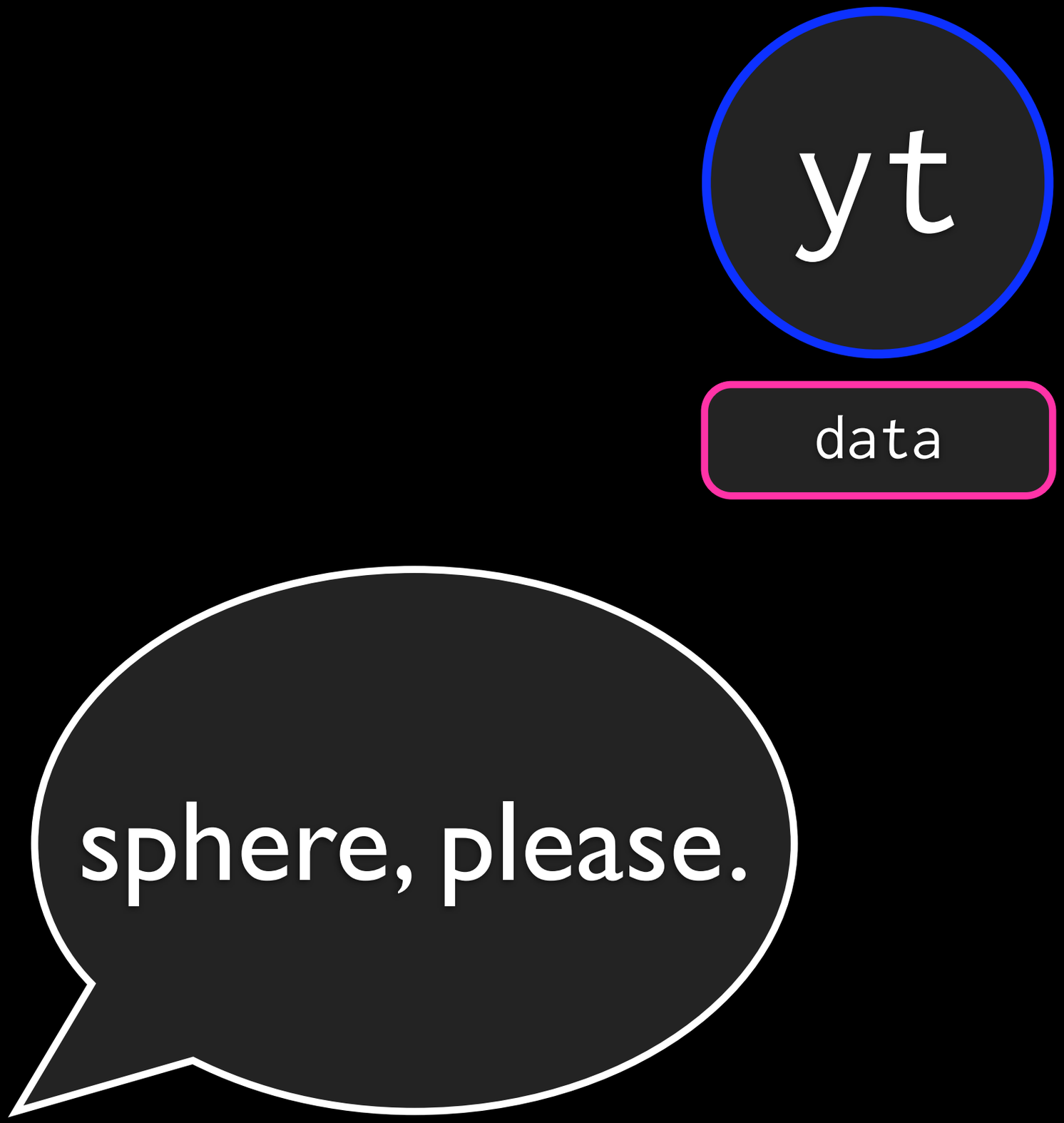


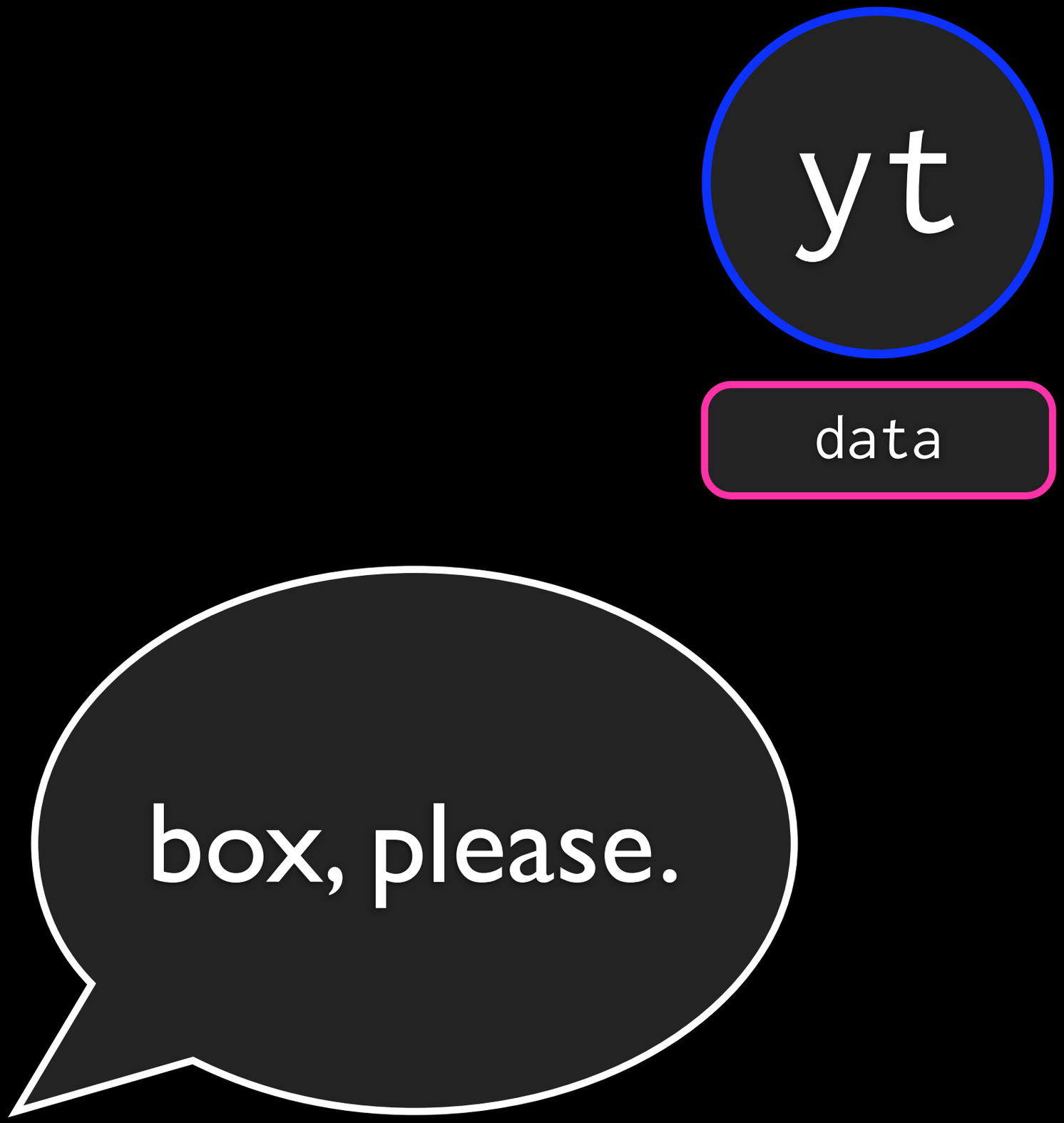
enzo

yt
data


yt
data
projection,
please.


yt
data
slice, please.

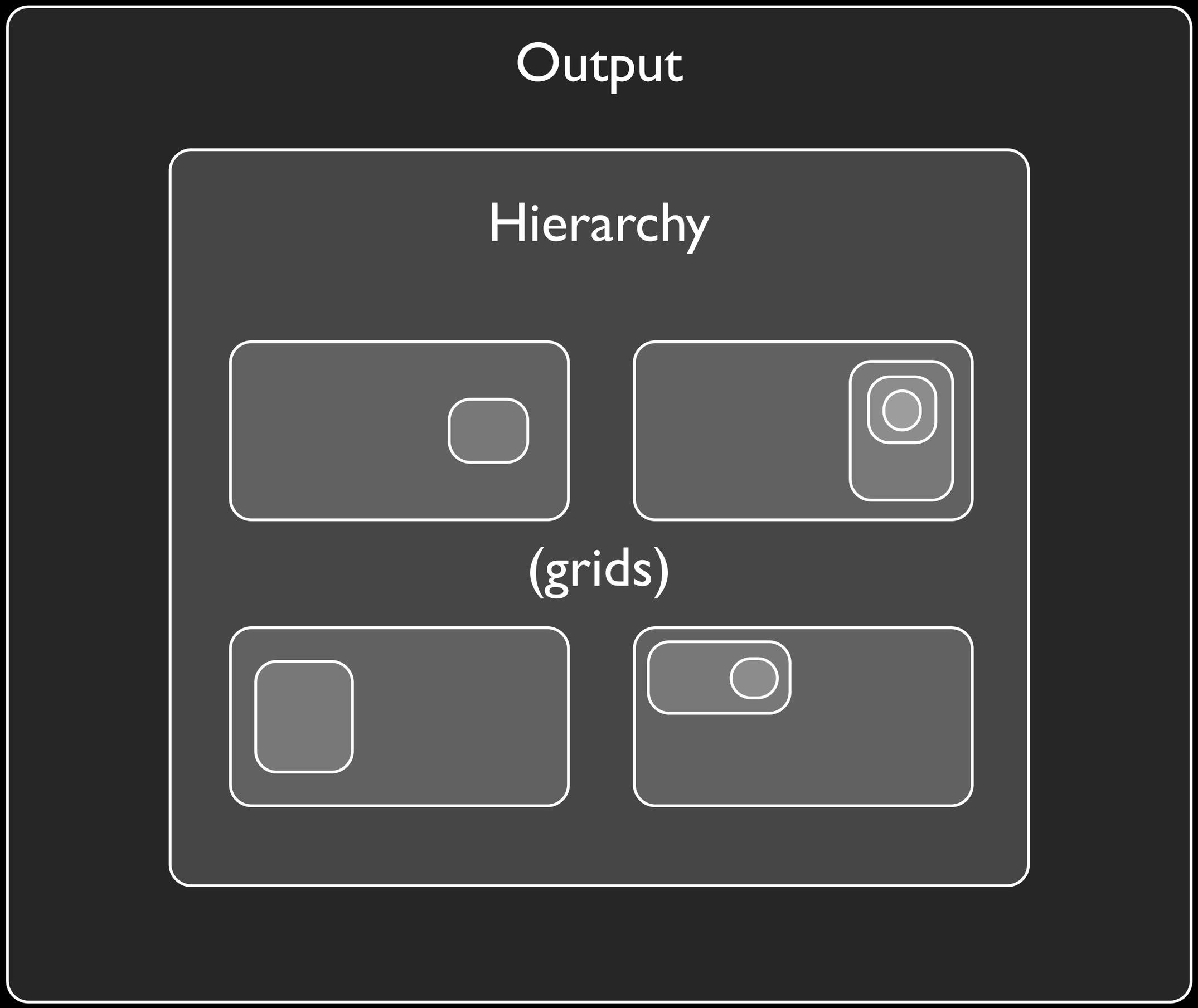

yt
data
sphere, please.

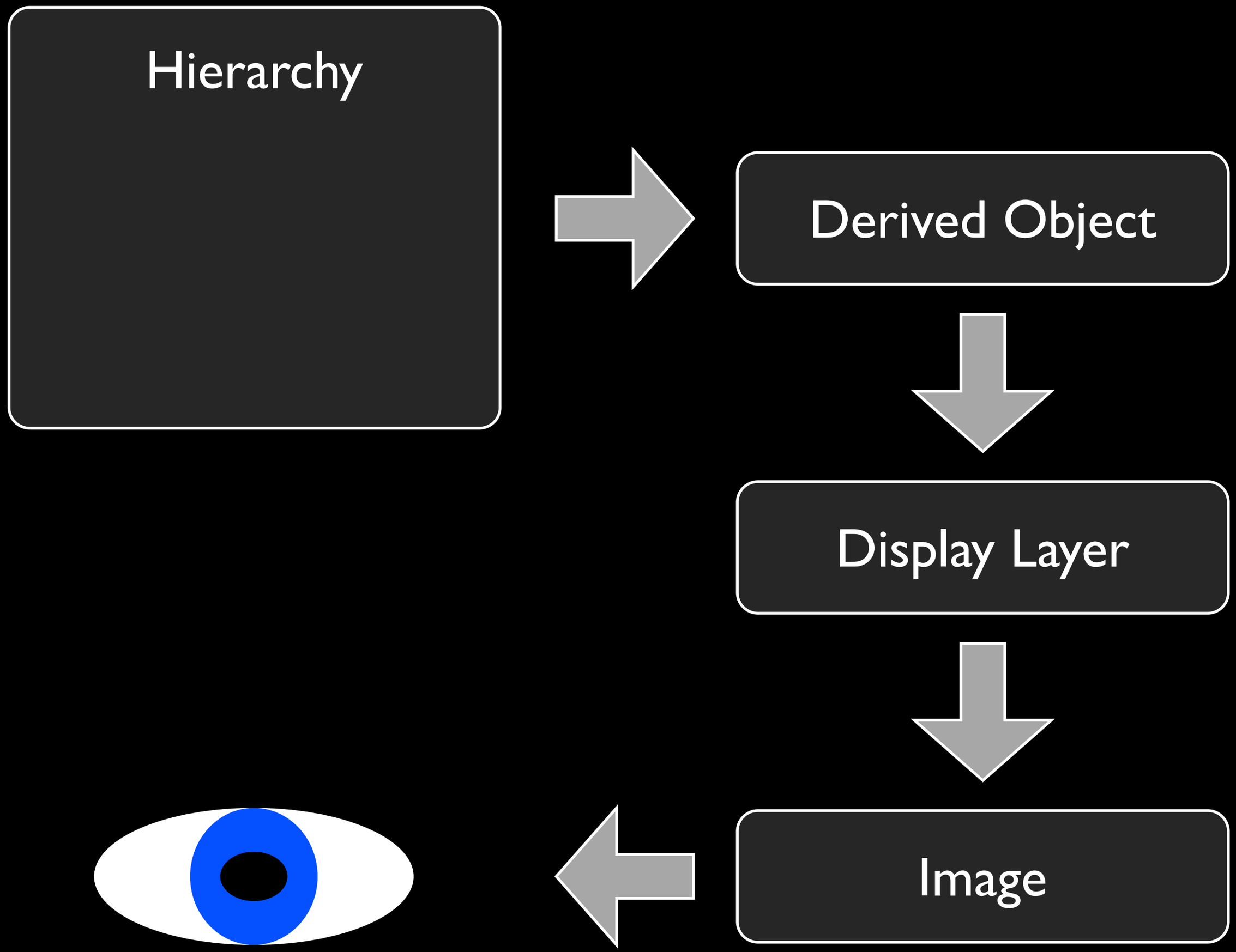

yt
data
box, please.

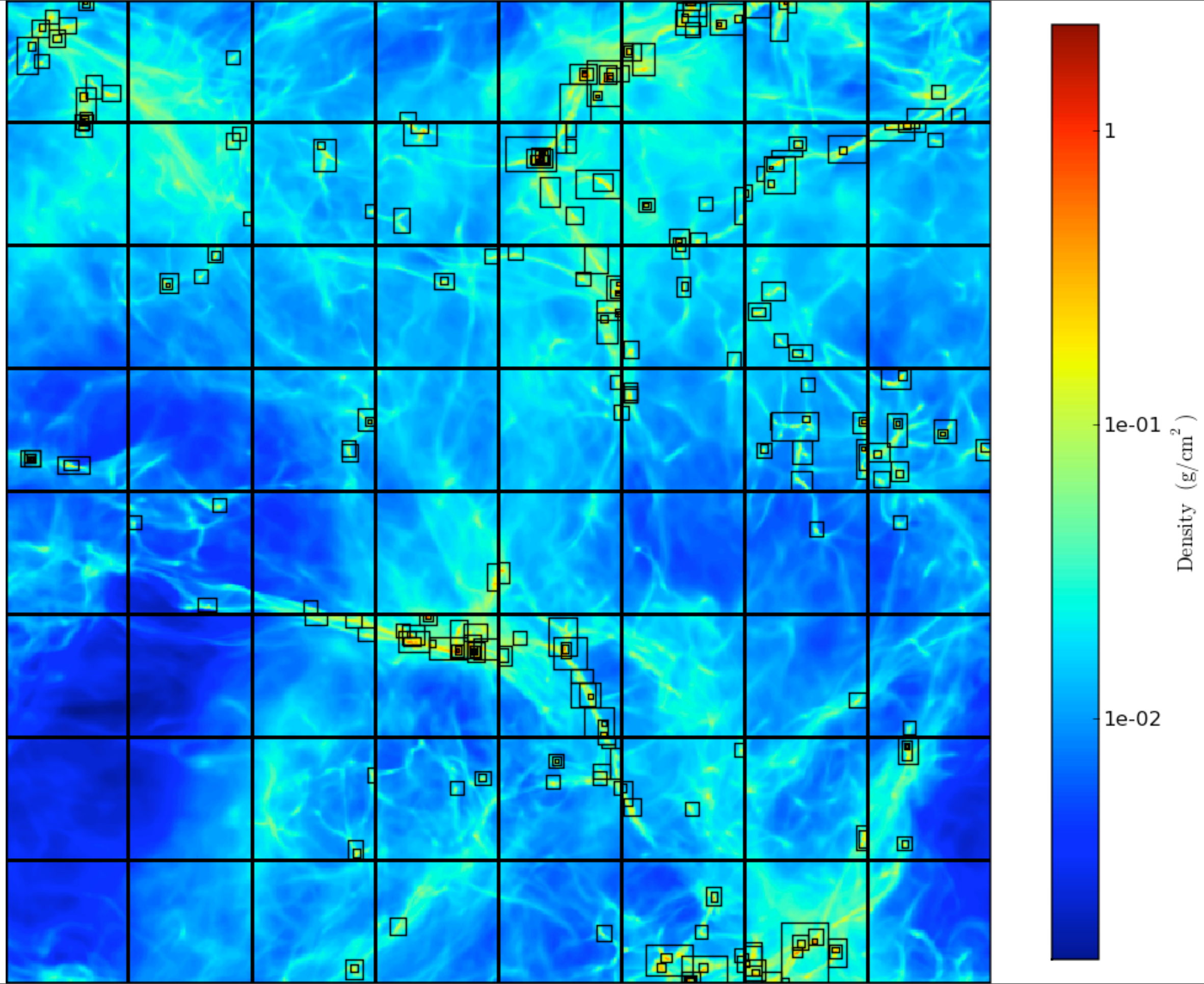
Output

Output

Hierarchy


Output
Hierarchy
(grids)


Hierarchy
Derived Object
Display Layer
Image


1
1e-01
1e-02
Density (g/cm^2)

What is distinguishing?

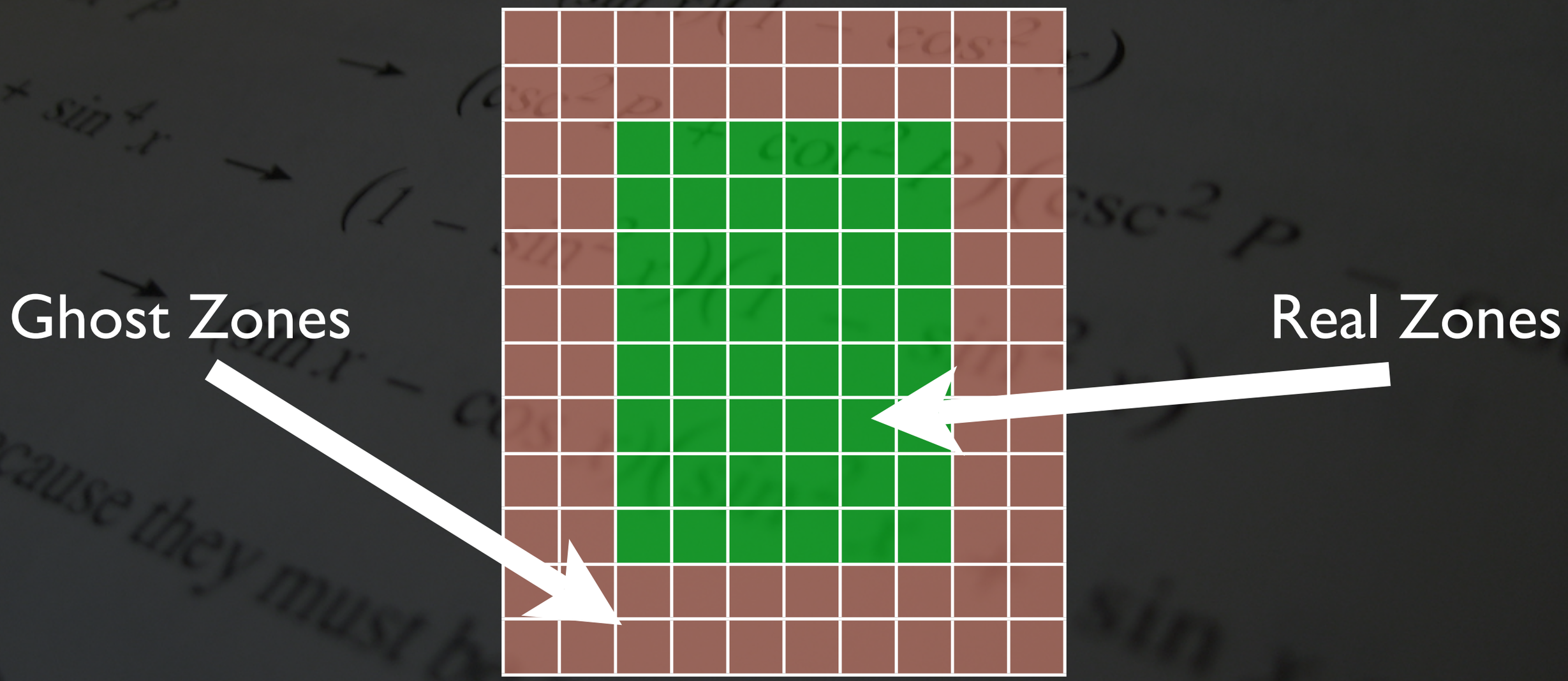
Data Object Protocol

```
>>> grid[“Density”]
>>> sphere[“RadialVelocity”]
>>> disk[“Temperature”]
>>> sphere.quantities[“CenterOfMass”]()
```

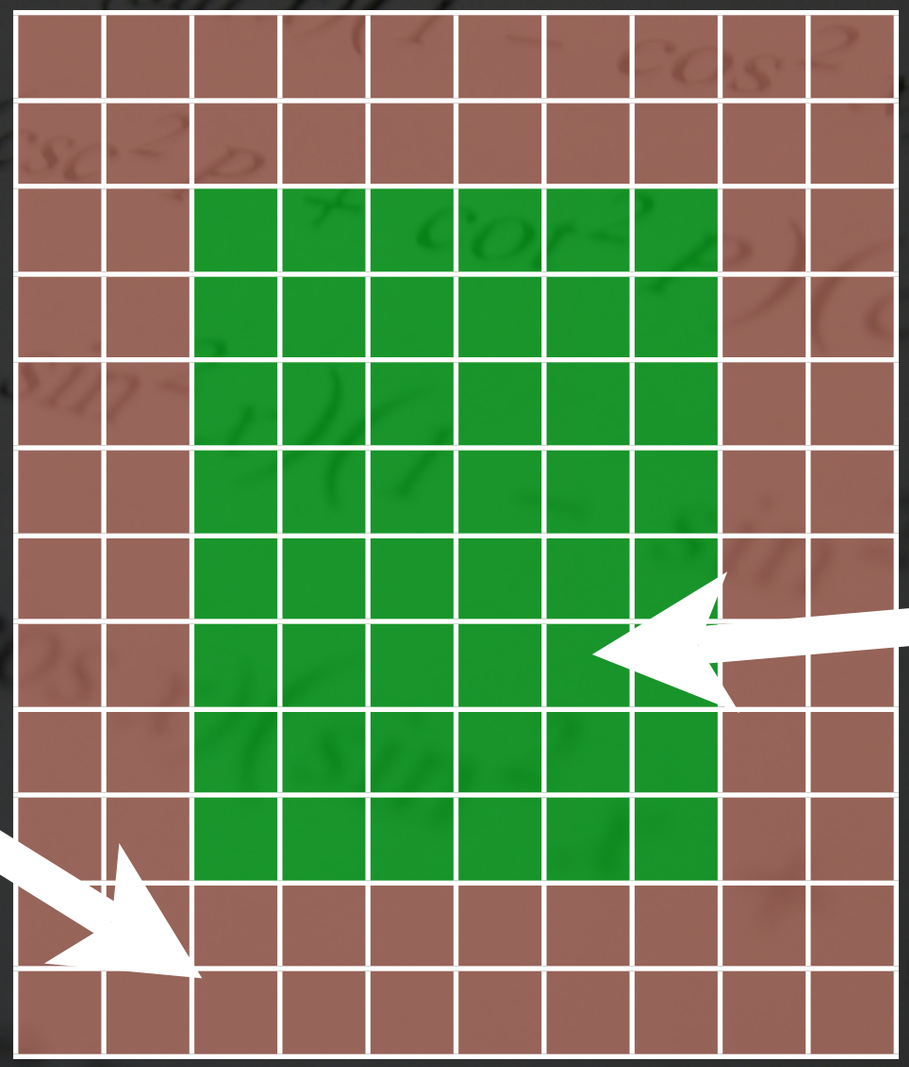
Derived Fields

```
def HydrogenNuclei(field, data):
    return (data[“HI_Density”] +
            data[“HII_Density”]) / m_h

add_field(“HydrogenNuclei”,
          function=HydrogenNuclei,
          units = r“\rm{cm}^{-3}”)
```

Derived Fields



Derived Fields



Finite difference stencils!

Units

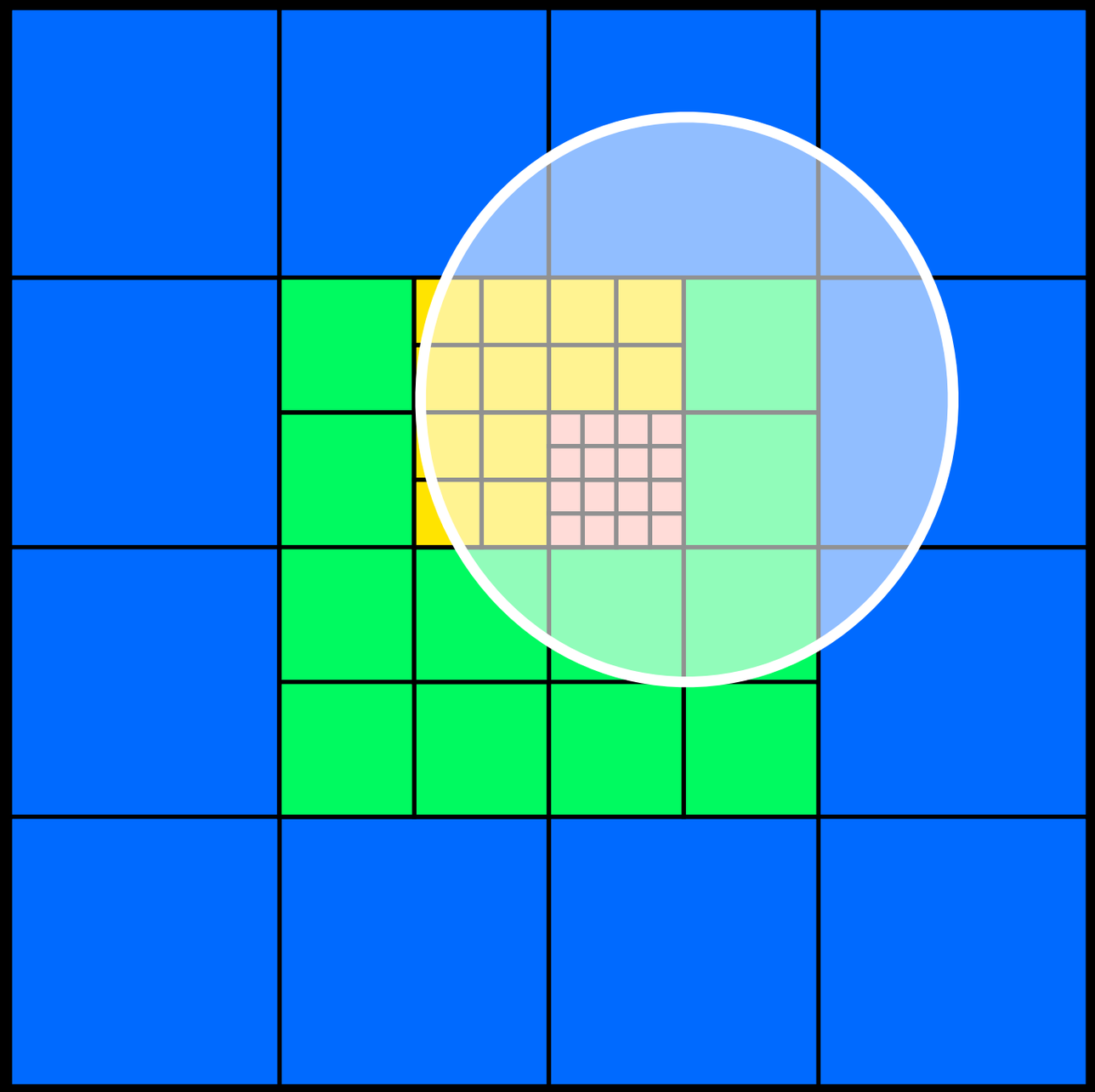
Units

CGS

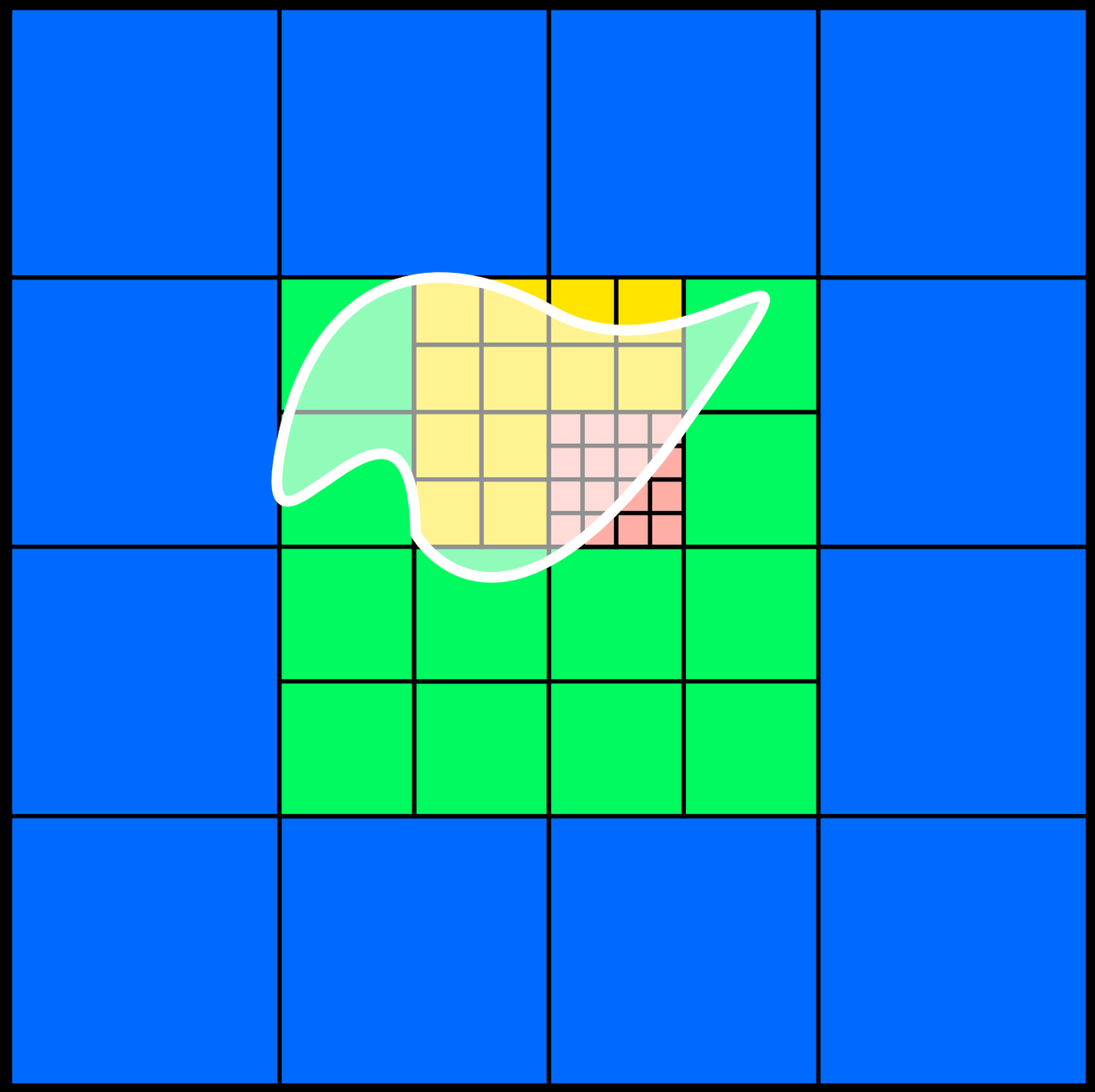
Analysis

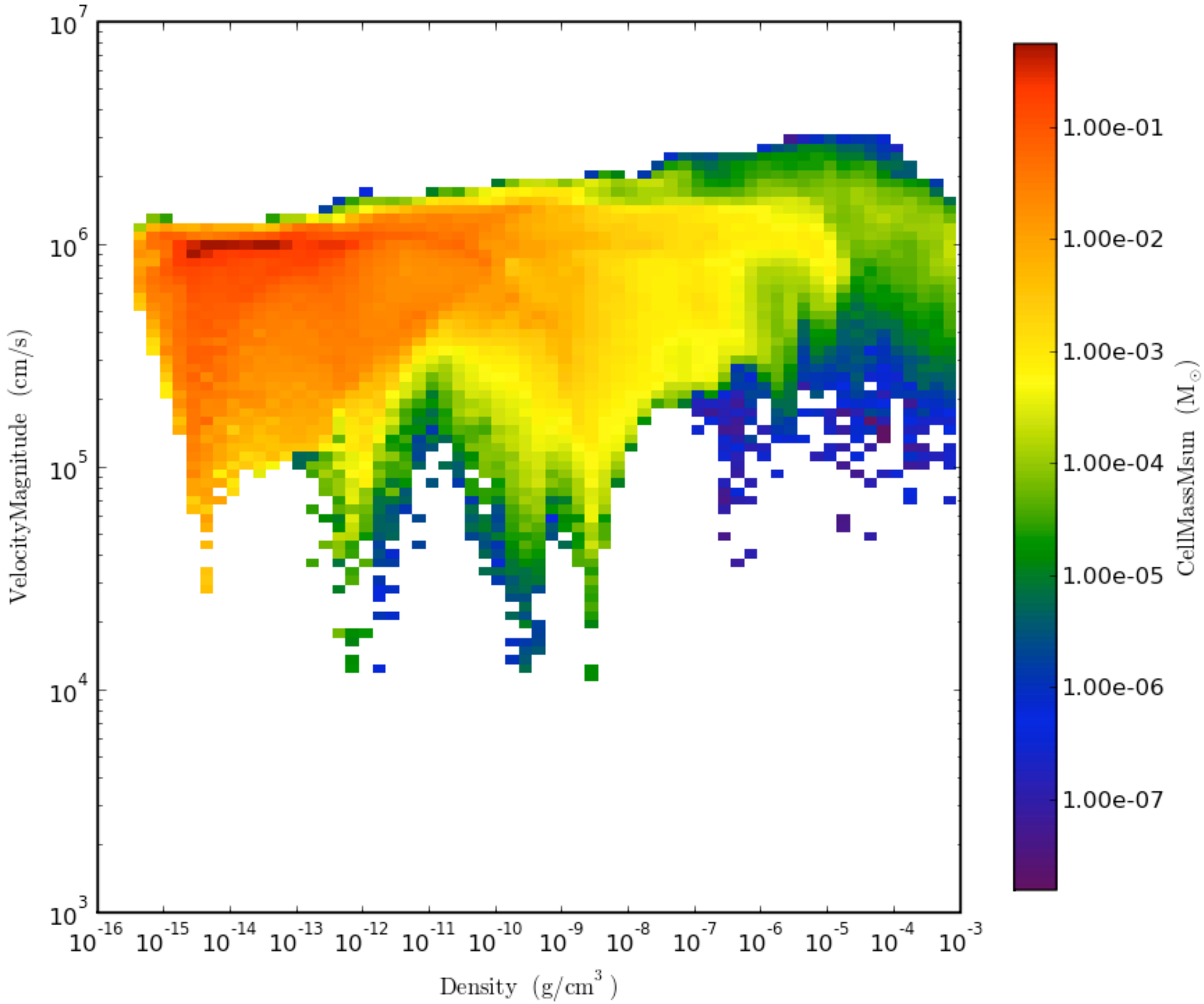
Data Selection
Profiling
Halo Finding
Contour Finding
Derived Fields

Profiles

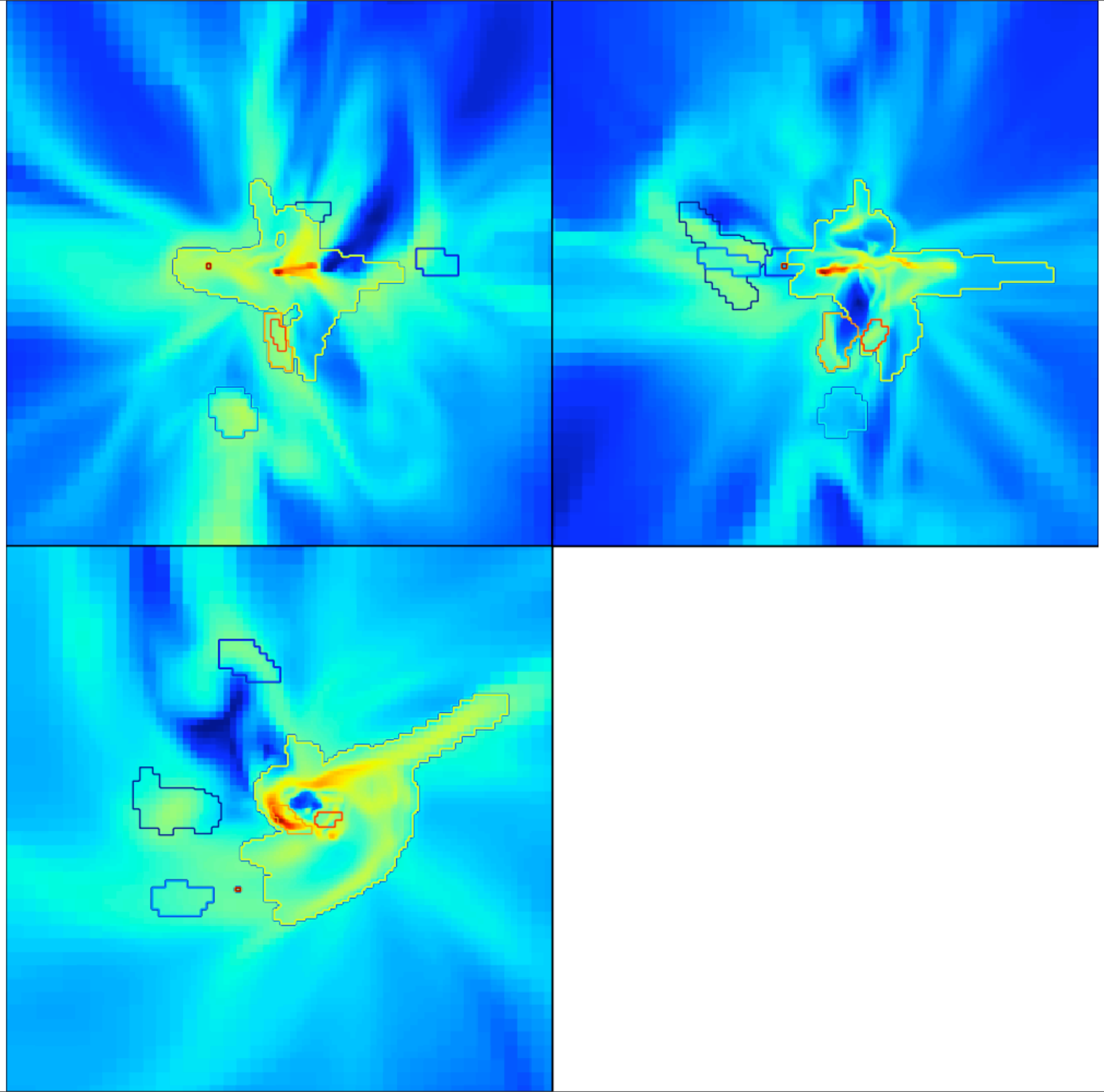


Profiles



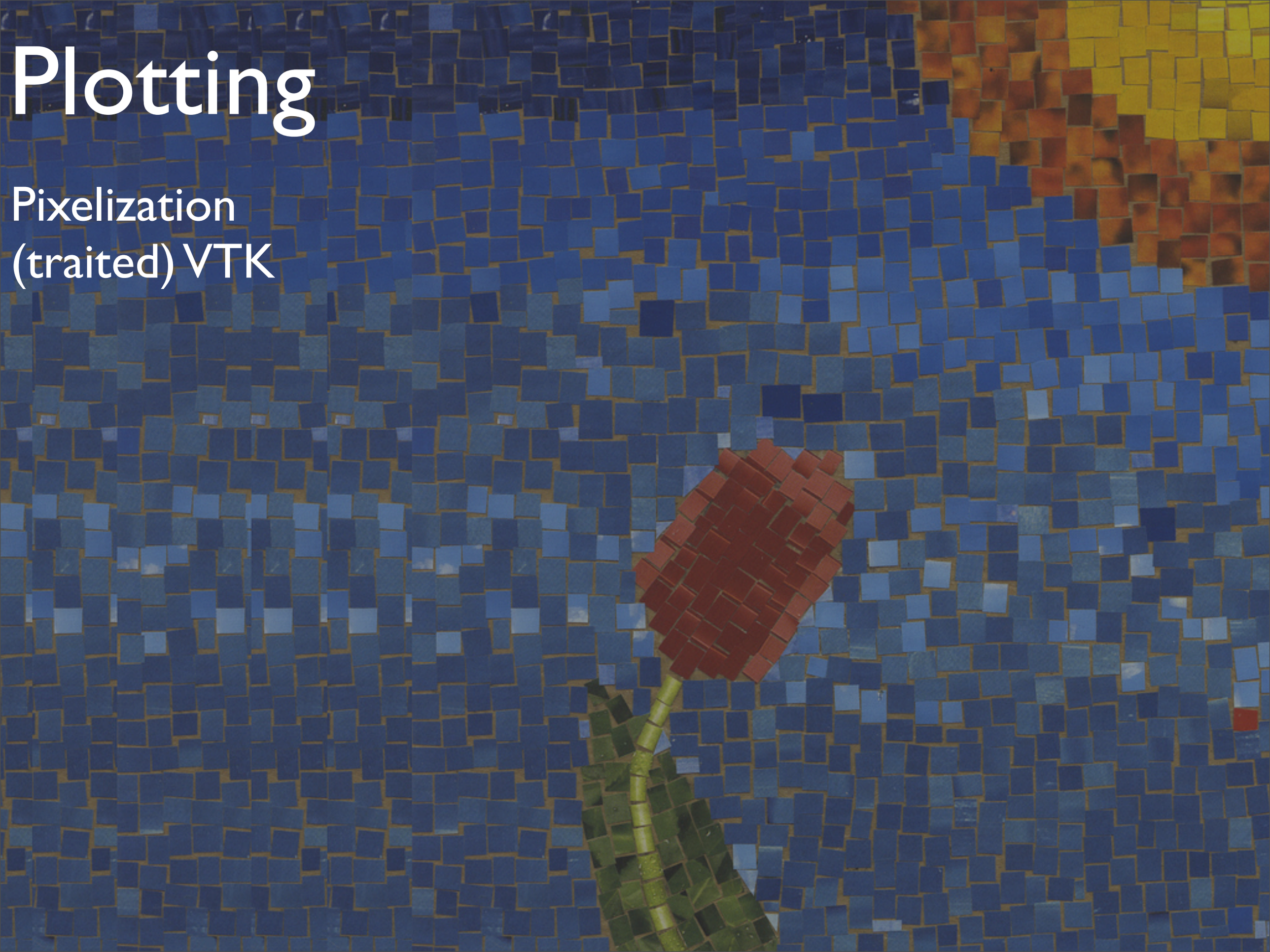

VelocityMagnitude (cm/s)
Density (g/cm^3)
CellMassMsun (M$_\odot$)
10^7
10^6
10^5
10^4
10^3
10^{-16} 10^{-15} 10^{-14} 10^{-13} 10^{-12} 10^{-11} 10^{-10} 10^{-9} 10^{-8} 10^{-7} 10^{-6} 10^{-5} 10^{-4} 10^{-3}
1.00e-01
1.00e-02
1.00e-03
1.00e-04
1.00e-05
1.00e-06
1.00e-07

Topologically Connected Sets

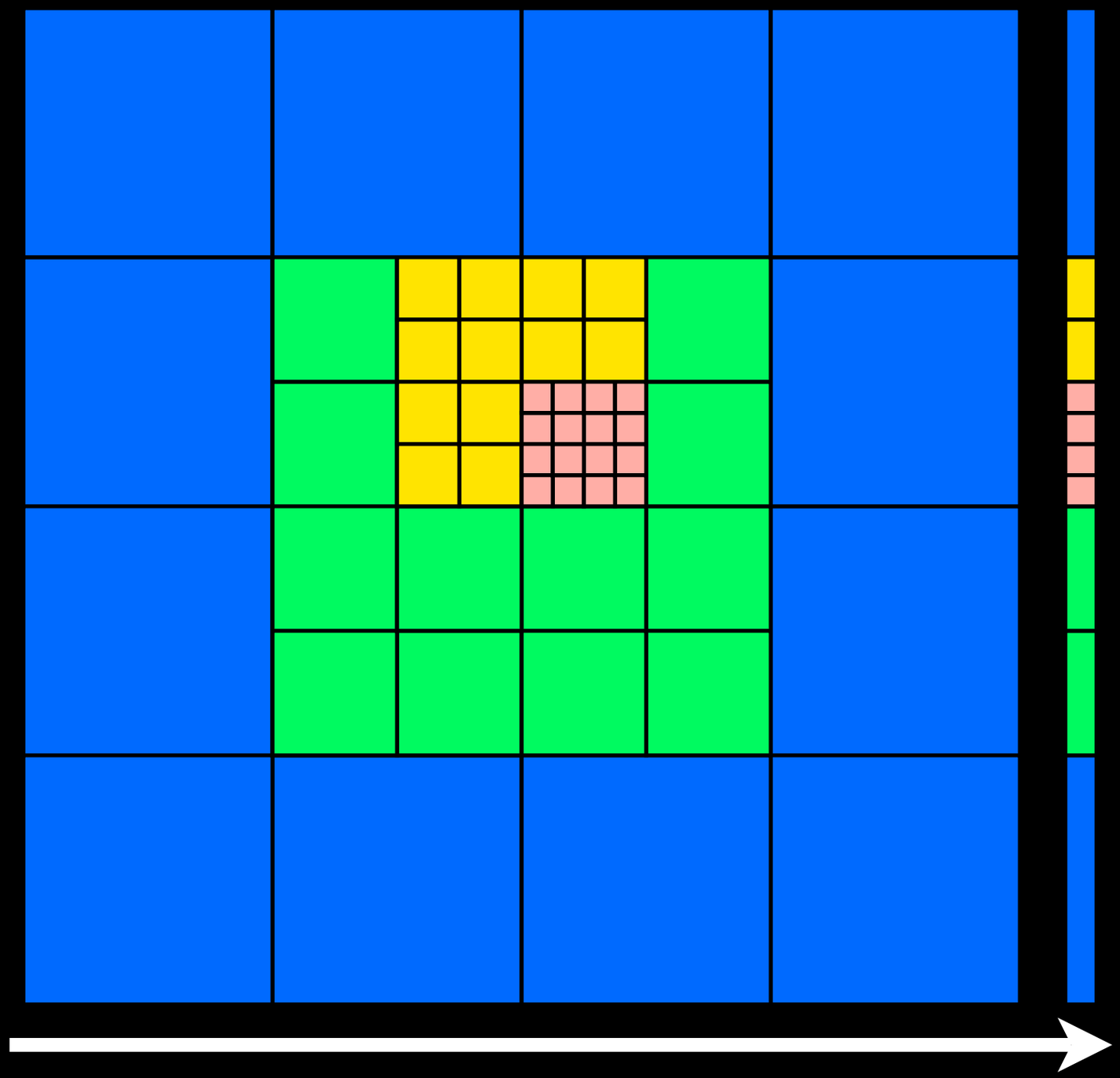


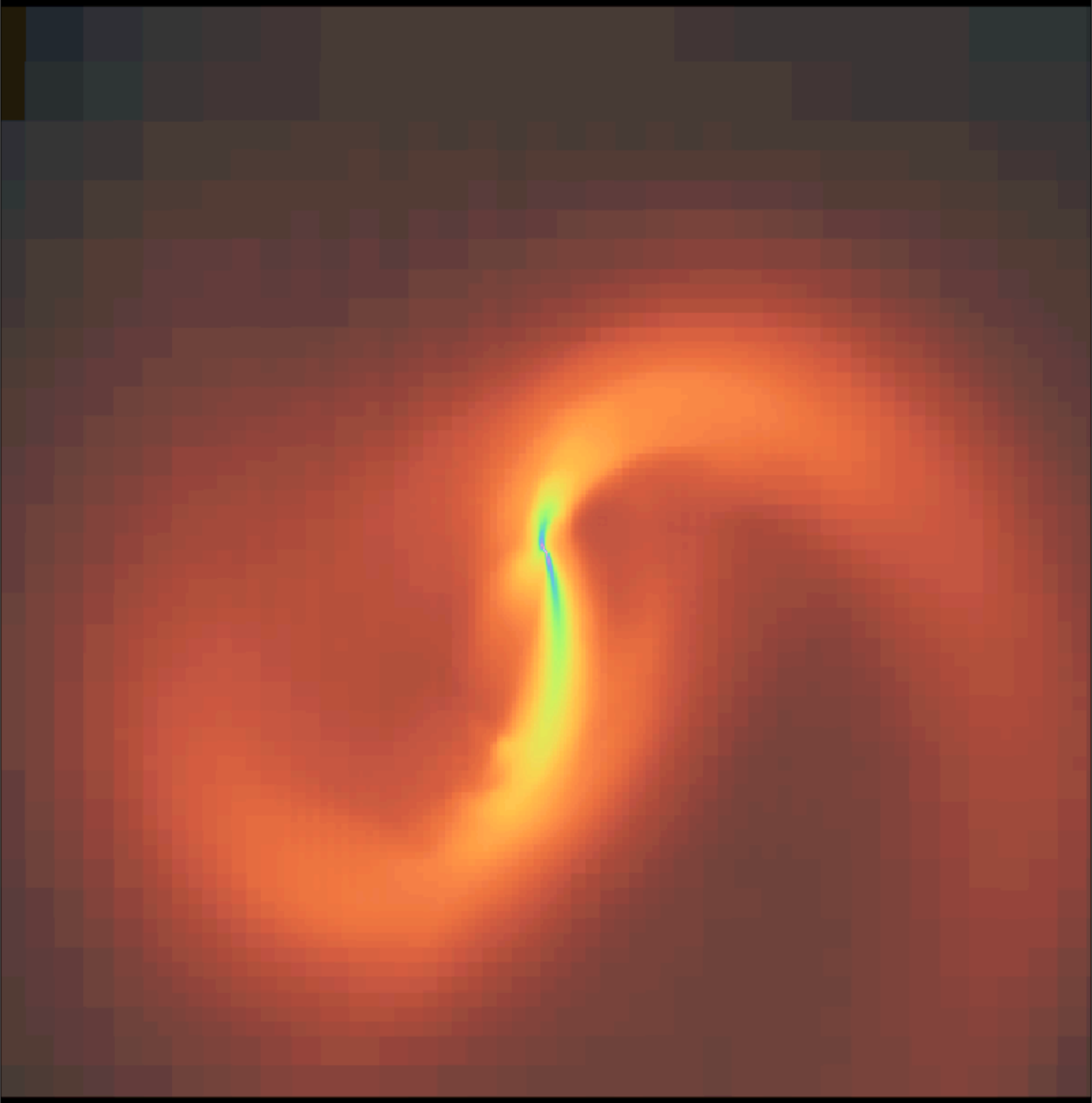
Plotting

Pixelization
(traited) VTK

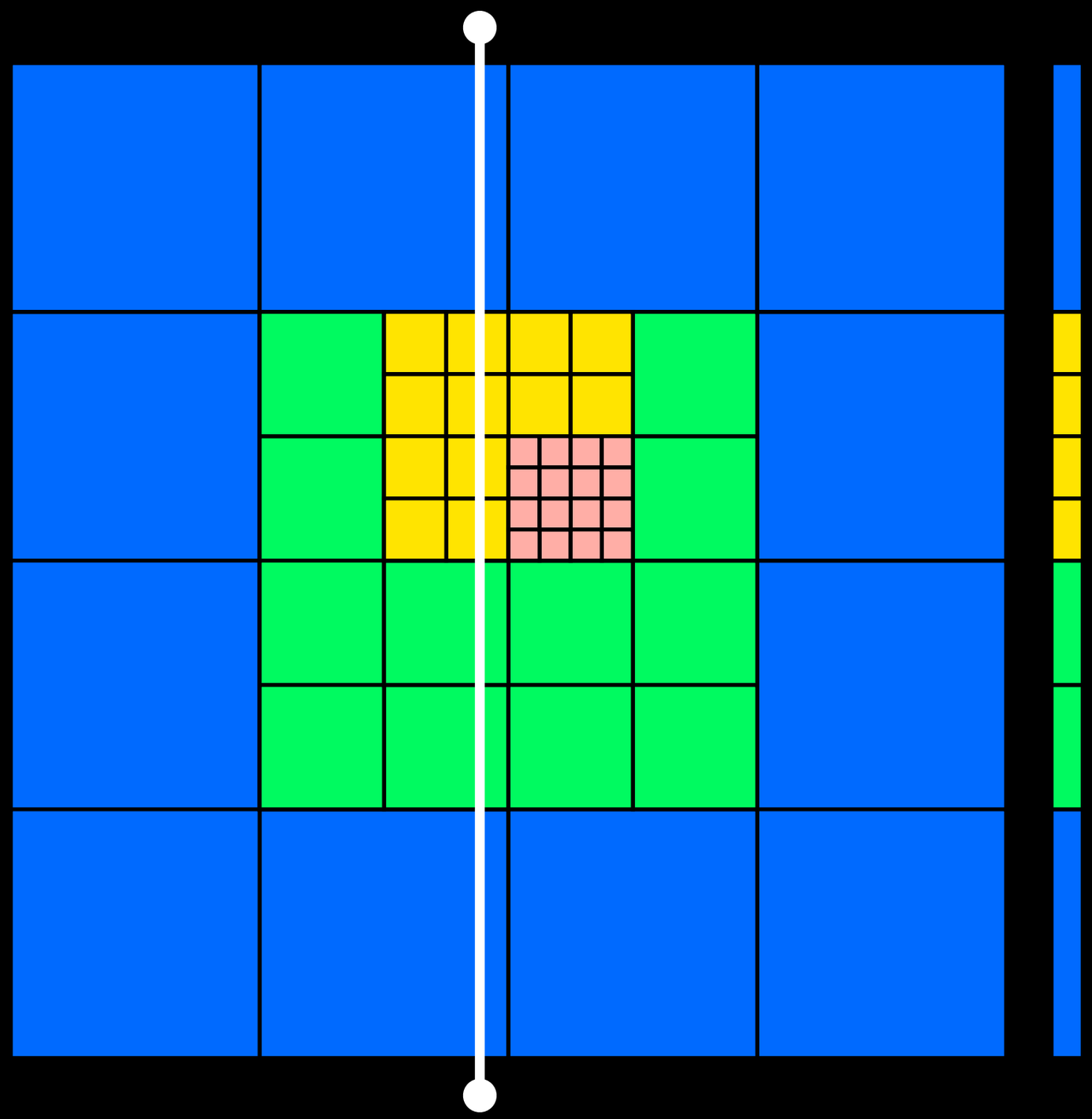


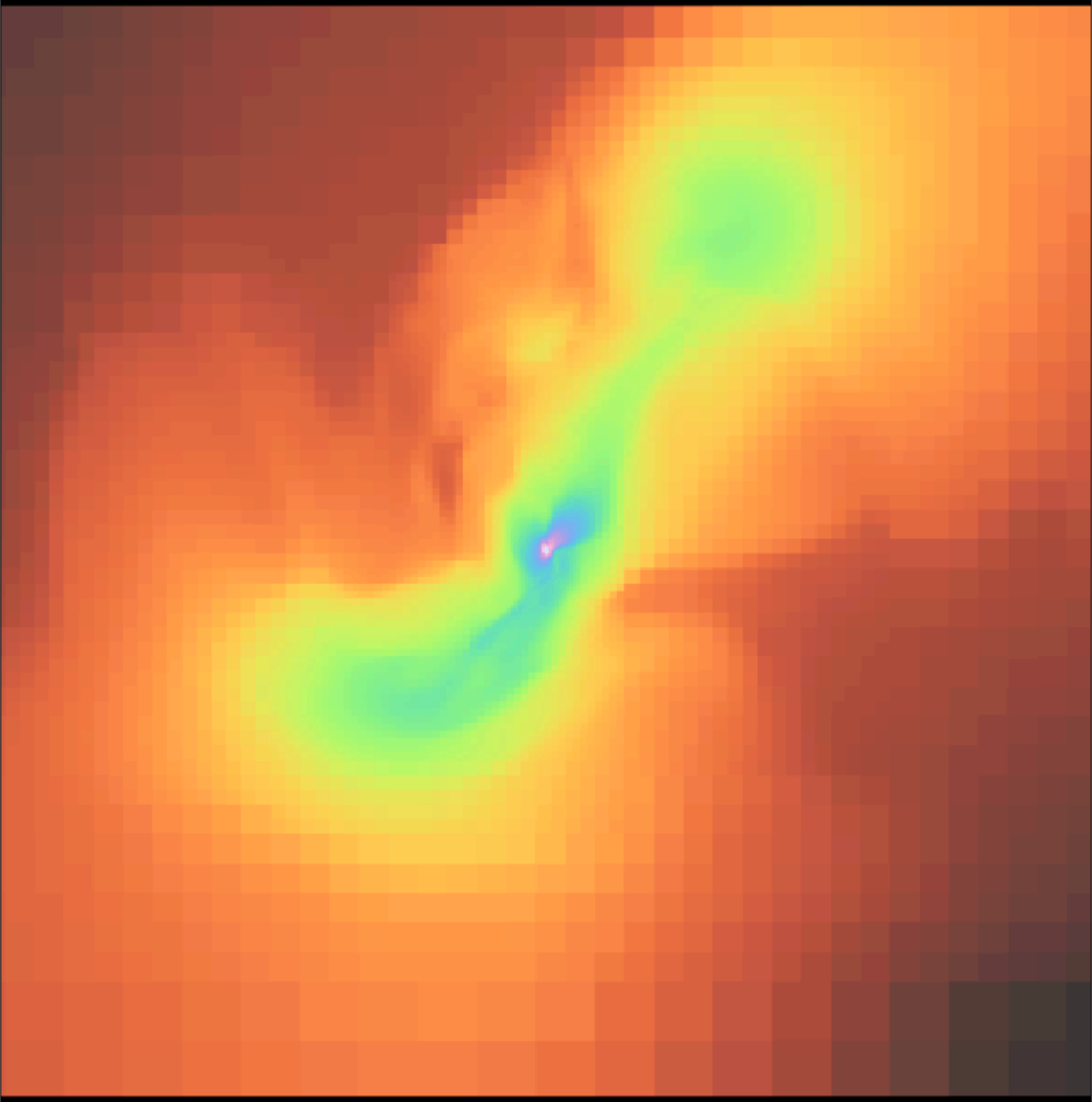
Projections

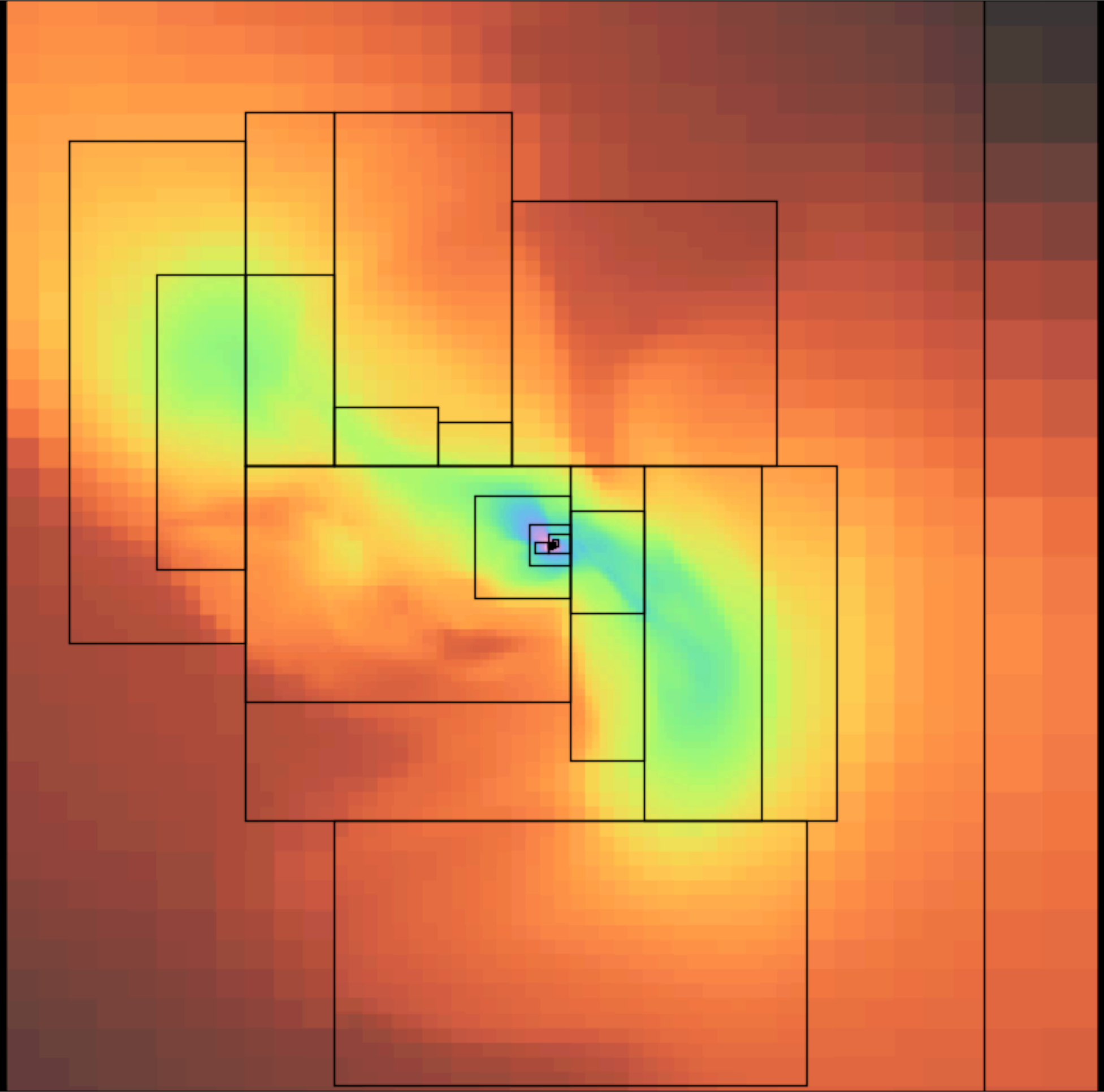




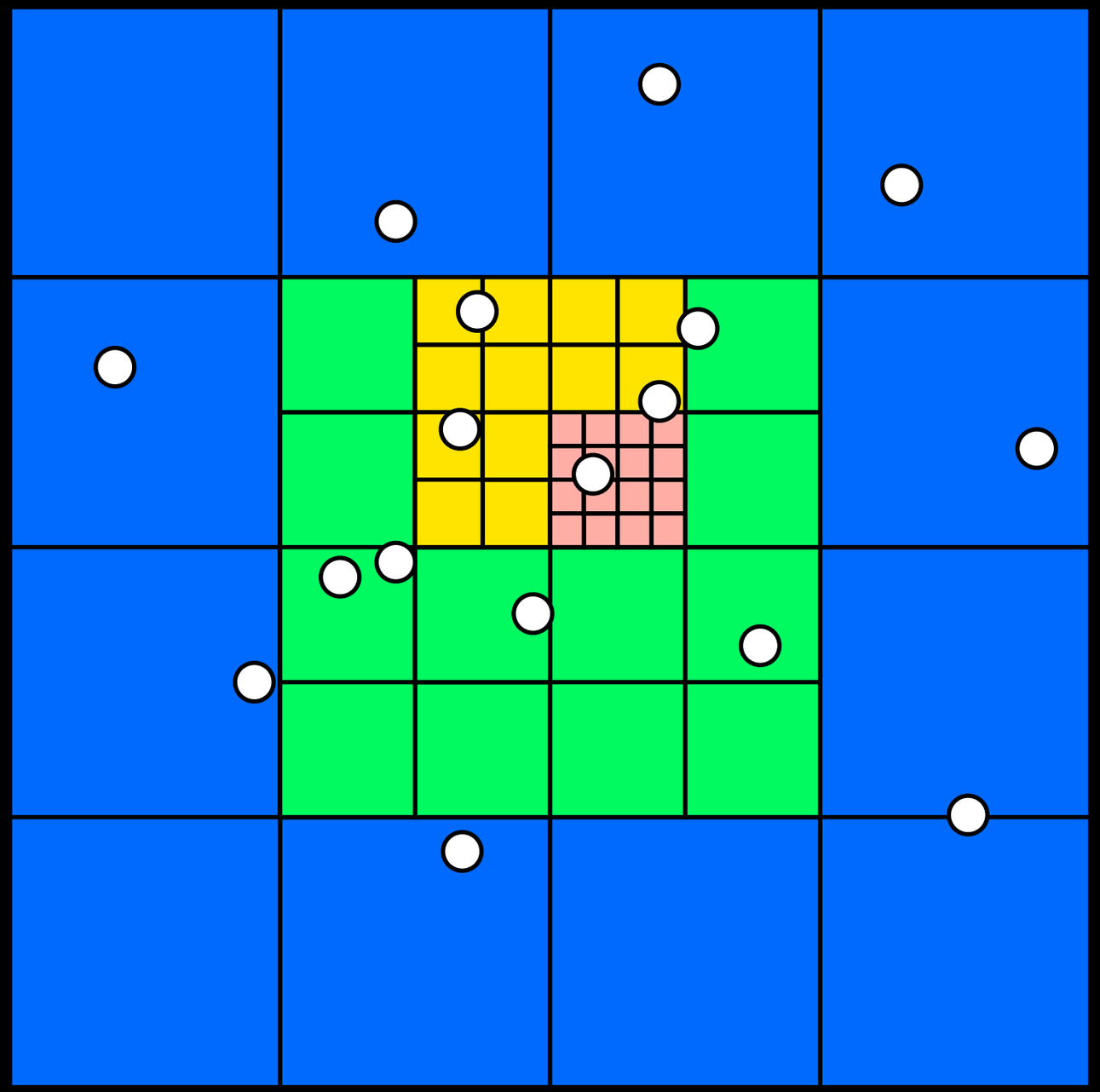
Slices

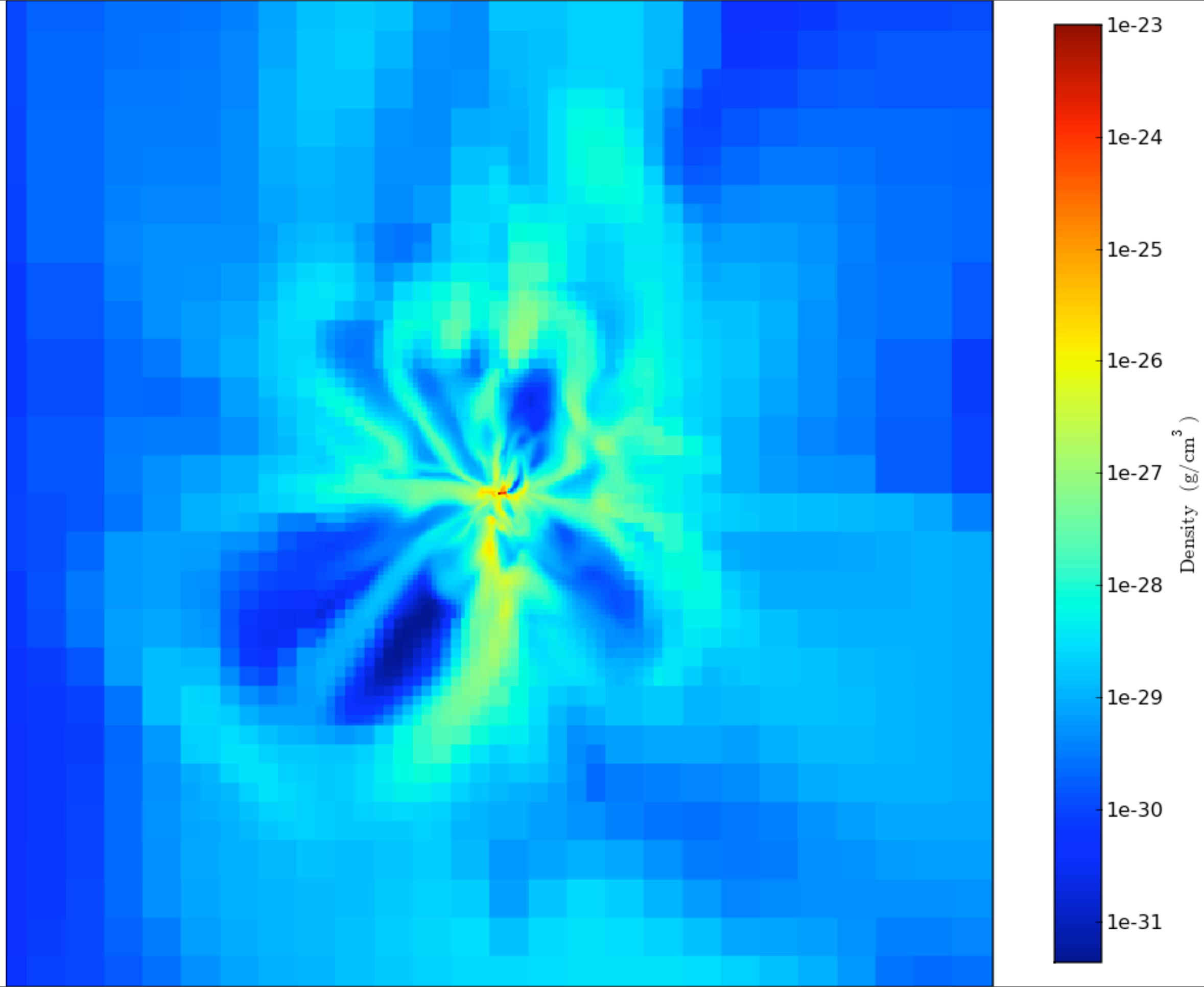


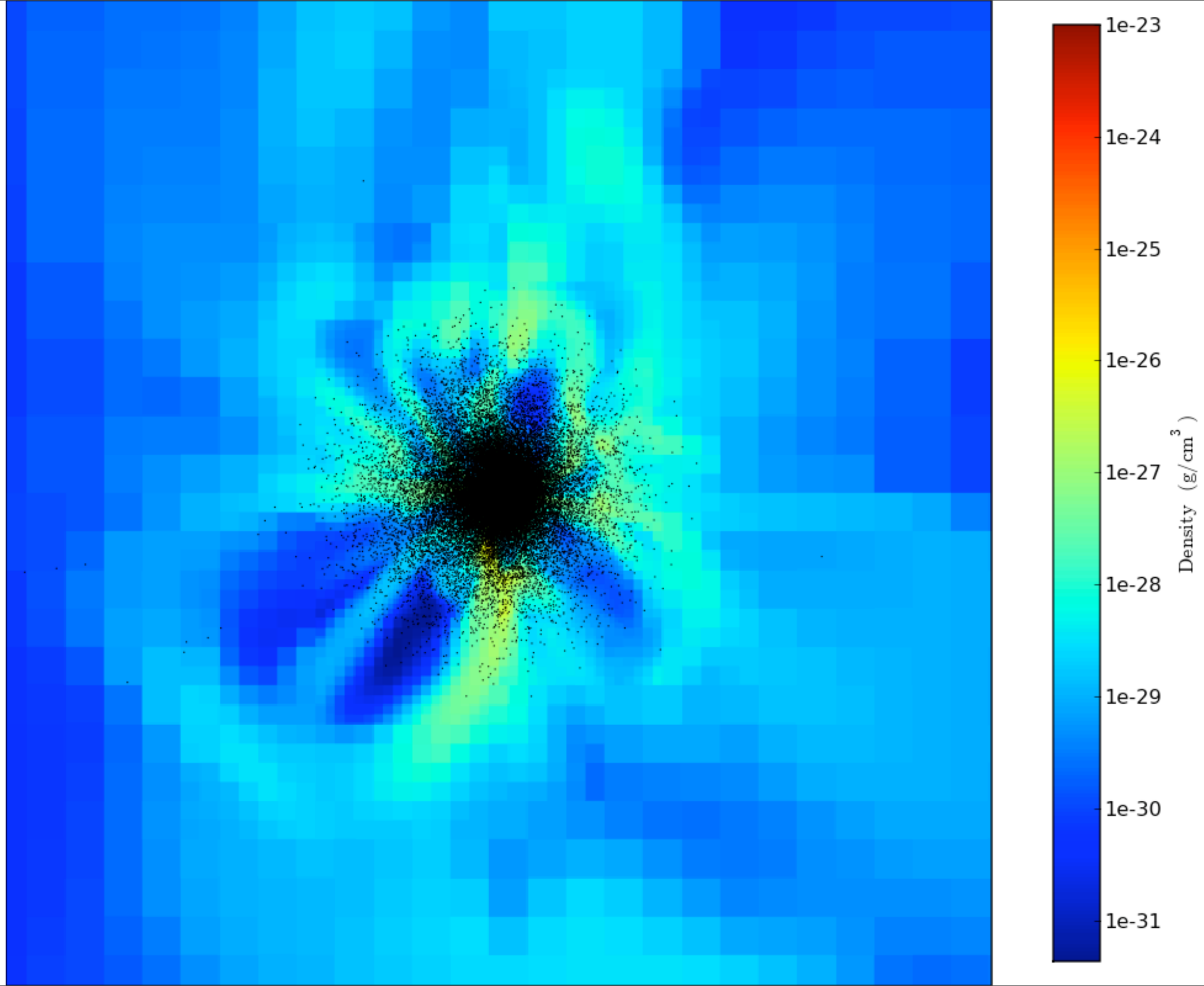




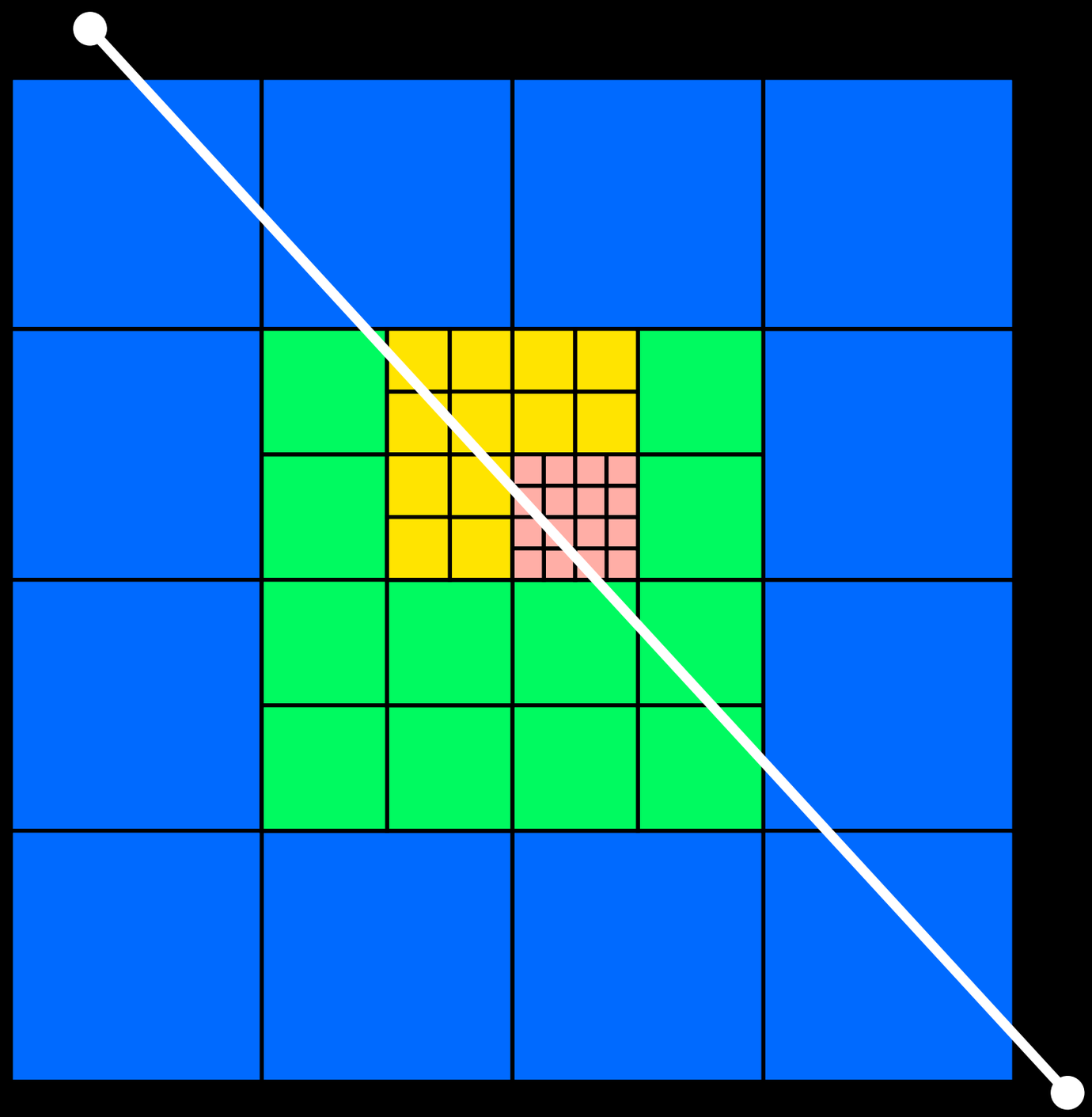
Particles

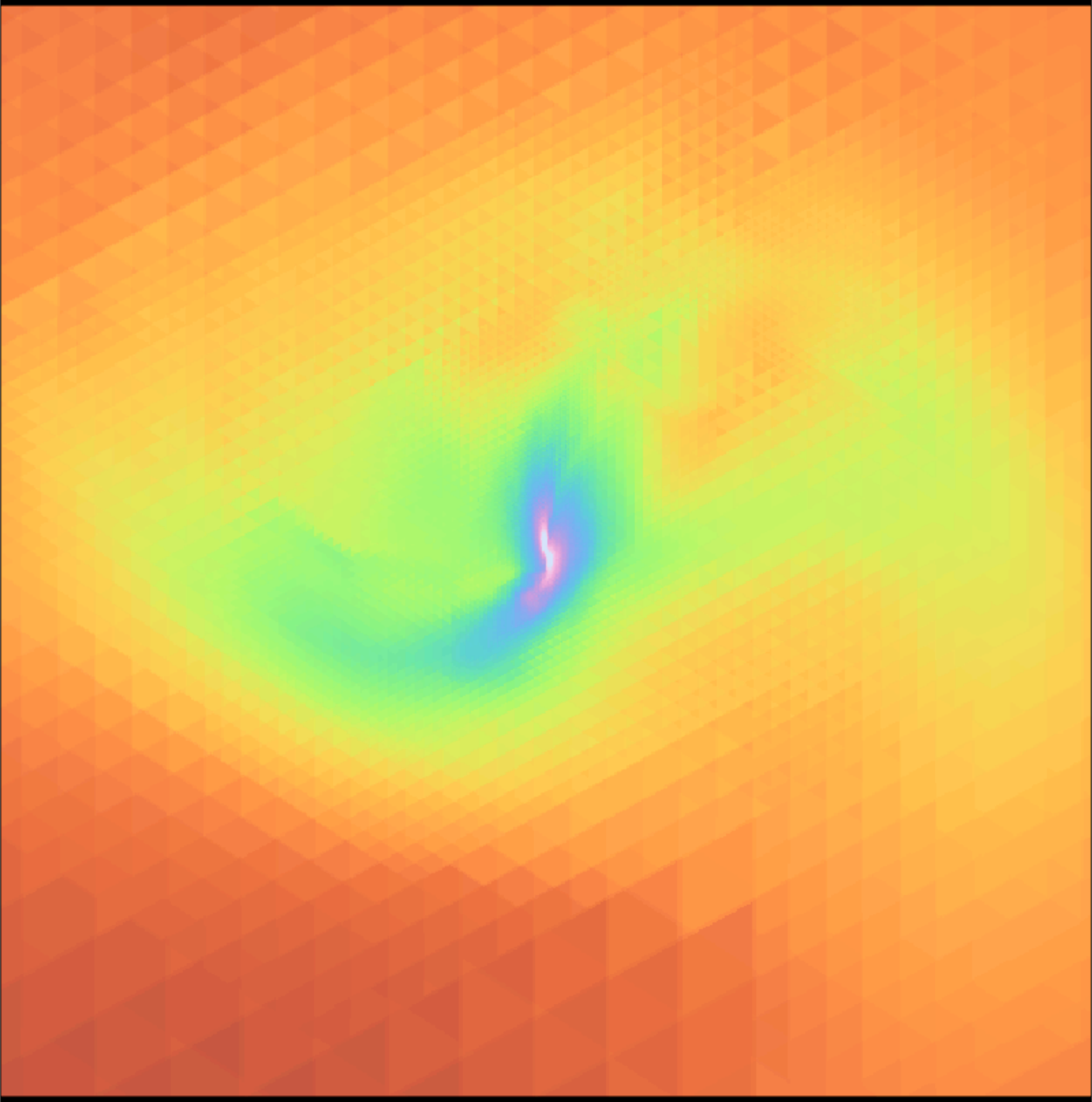


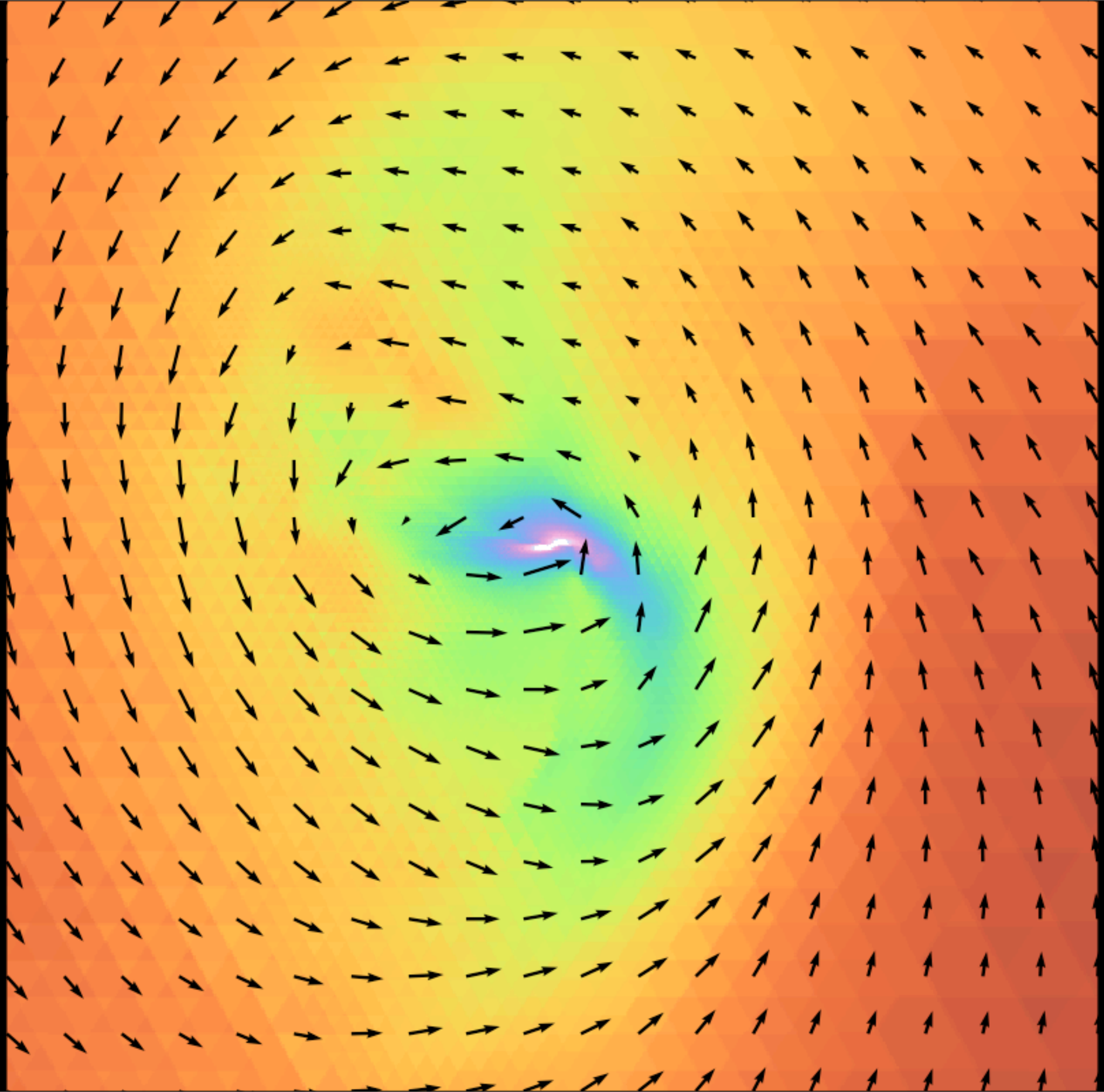

1e-23
1e-24
1e-25
1e-26
1e-27
1e-28
1e-29
1e-30
1e-31
Density (g/cm^3)

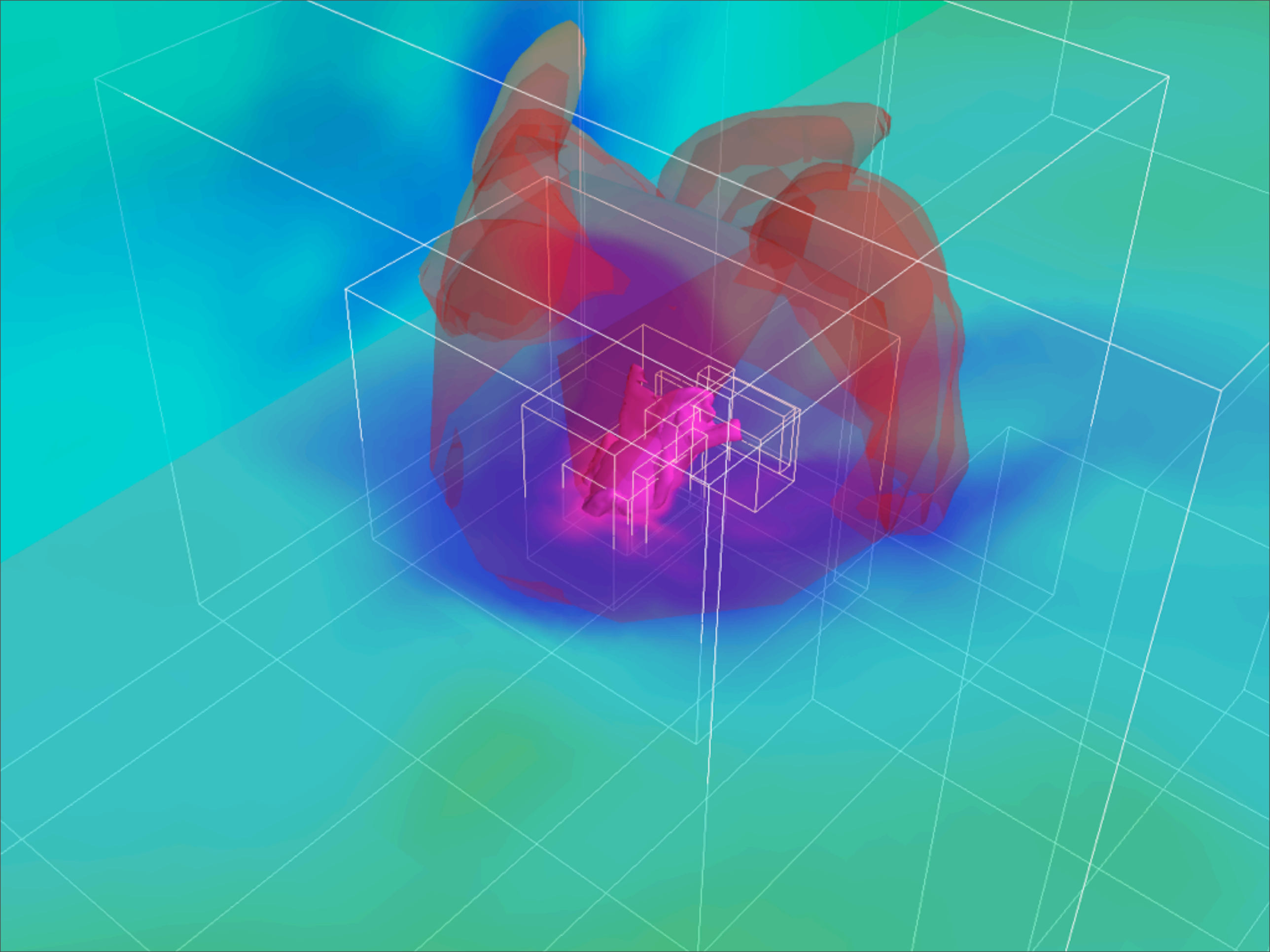

1e-23
1e-24
1e-25
1e-26
1e-27
1e-28
1e-29
1e-30
1e-31
Density (g/cm^3)

Oblique Slices









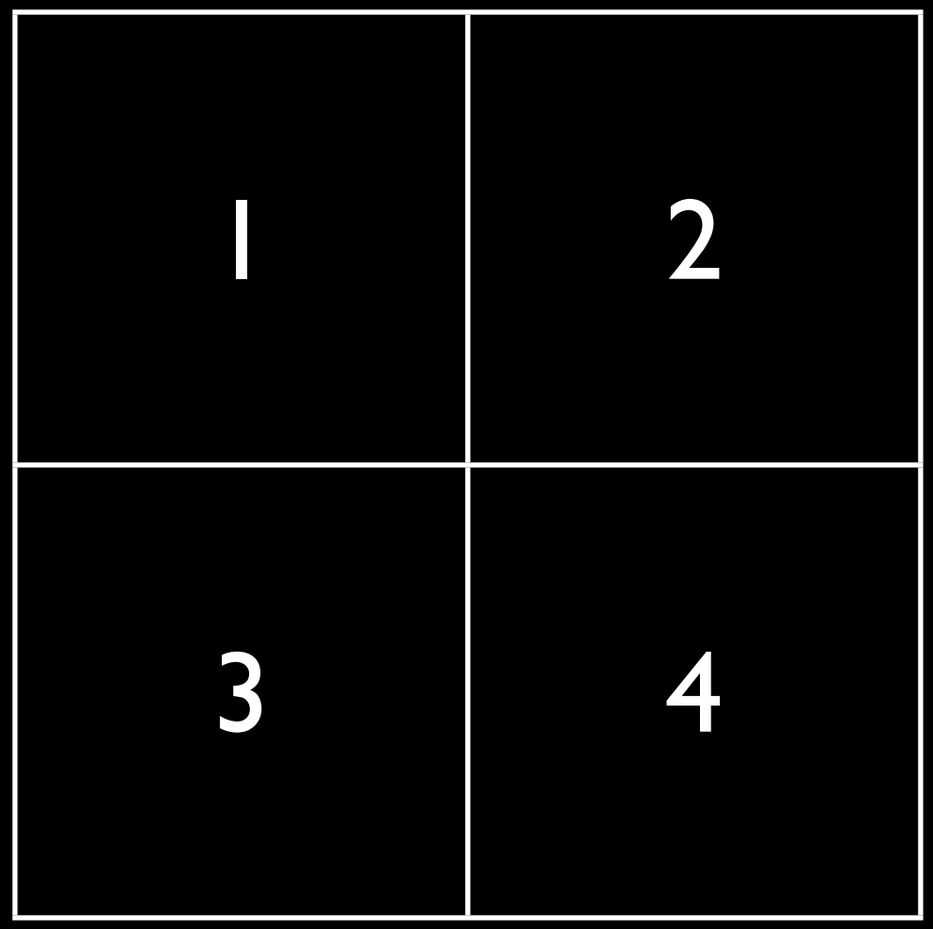
HDF5 Wrappers

Discovery
Slicing
Reading

Parallel Analysis

Parallel Analysis Interface

`partition_hierarchy_2d()`



Parallel Analysis Interface

```
get_grids()
```

GridIterator

ParallelGridIterator

```
>>> pf = EnzoStaticOutput(“my_data”)
>>> pc = PlotCollection(pf)

>>> pc.add_projection(“Density”, 0)
>>> pc.save(“my_data”)
```

yesterday morning:

```
>>> pf = EnzoStaticOutput(“my_data”)
>>> pc = PlotCollection(pf)

>>> pc.add_projection(“Density”, 0)
>>> pc.save(“my_data”)
```

yesterday morning:

```
>>> pf = EnzoStaticOutput(“my_data”)
>>> pc = PlotCollection(pf)

>>> pc.add_projection(“Density”, 0)
>>> pc.save(“my_data”)
```

this morning:

```
>>> “my_data” | projector_x
>>> “my_data” | slicer(field=“Temperature”)
```

Preparing for Release

Documentation

Table Of Contents

yt Overview
Indices and tables

Next topic

Introduction

This Page

Show Source

Quick search

Go

yt Overview

yt is a toolkit designed to analyze, manage and plot adaptive mesh refinement data from the Enzo code.

Contents:

- Introduction
  - History
  - What yt is and is not
  - What functionality does yt offer?
- Getting Started
  - Maintained Installations
  - Binary Packages
  - Installing From Source
- The Quick Guide to yt
  - Starting Python
  - Opening Your Data File
  - Making Plots
  - Saving Plots
  - A Few More Plots
- The Tutorial
  - Getting Started

Documentation

Previous topic
Making Movies

Next topic
Extracting Data

This Page
Show Source

Quick search
Go

Making Timeseries Plots

Note: Unlike the other scripts in this cookbook, these do not serve as standalone executables. You must edit them with your data information.

I don't think anyone will argue that timeseries data is important. It is possible to create it with yt; however, a few things should be noted.

Python is notorious for leaking memory via circular references – if you have a variable *var1* that references variable *var2*, when one is deleted, is the other? Efforts have been made to identify and remove all leaking references, but if references are created in your code that generates the timeseries data, you may find that more memory is used than is desired.

The process of generating time series data is fairly simple: you iterate over the datasets, and generate a value for each. (`cookbook_timeseries_max_dens.py`)

```
from yt.mods import *

max_rho = []
max_pos = []
times = []
for i in range(30):
    pf = lagos.EnzoStaticOutput("my_output%04i" % (i))
    v, c = pf.h.find_max("Density")
    max_rho.append(v)
    max_pos.append(c)
```

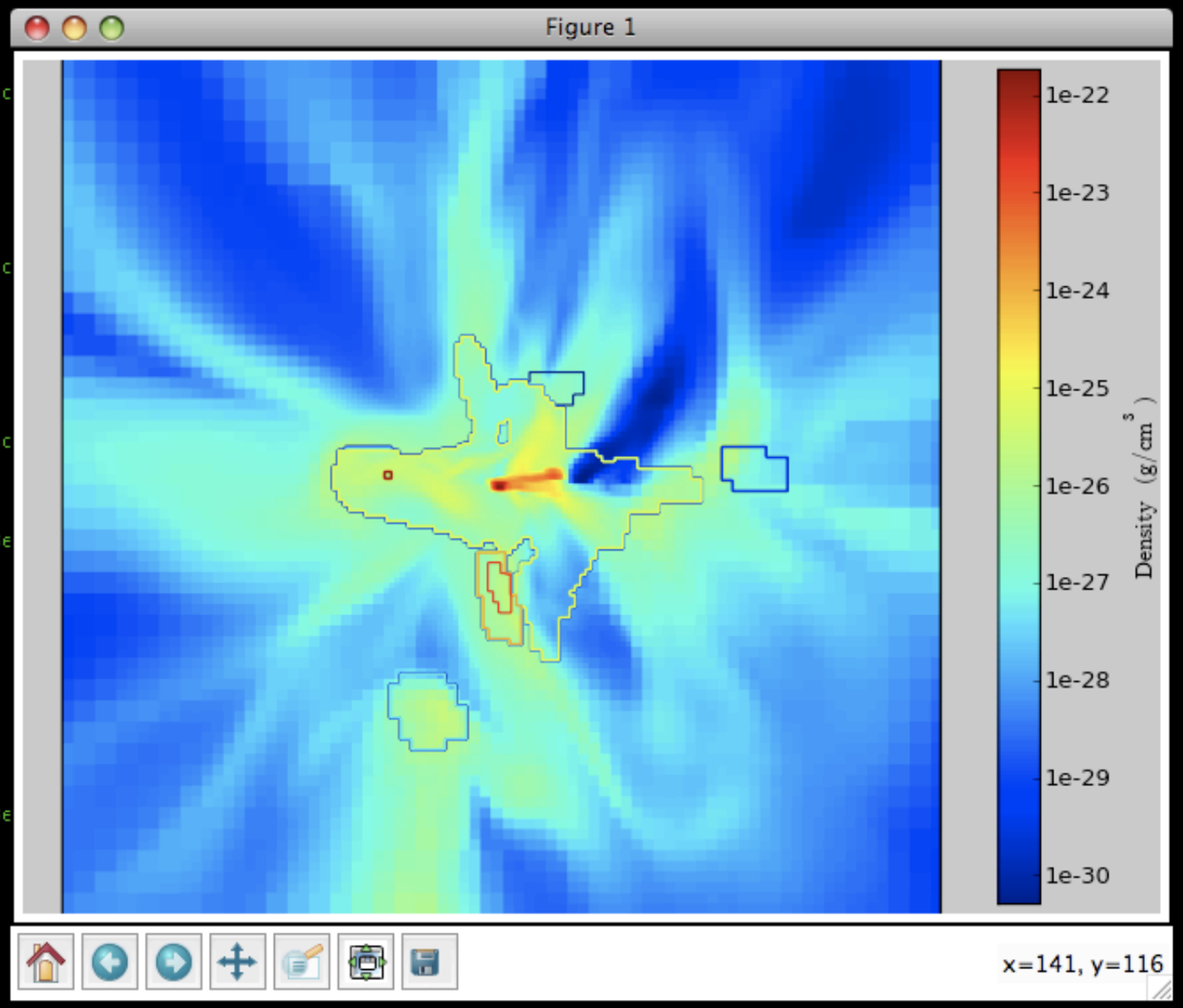
Interfaces

Ets Marcel Lutz S.
03.88.89.48.50
03.88.89.51.04
0
CREDIT
TIME
ELAUT

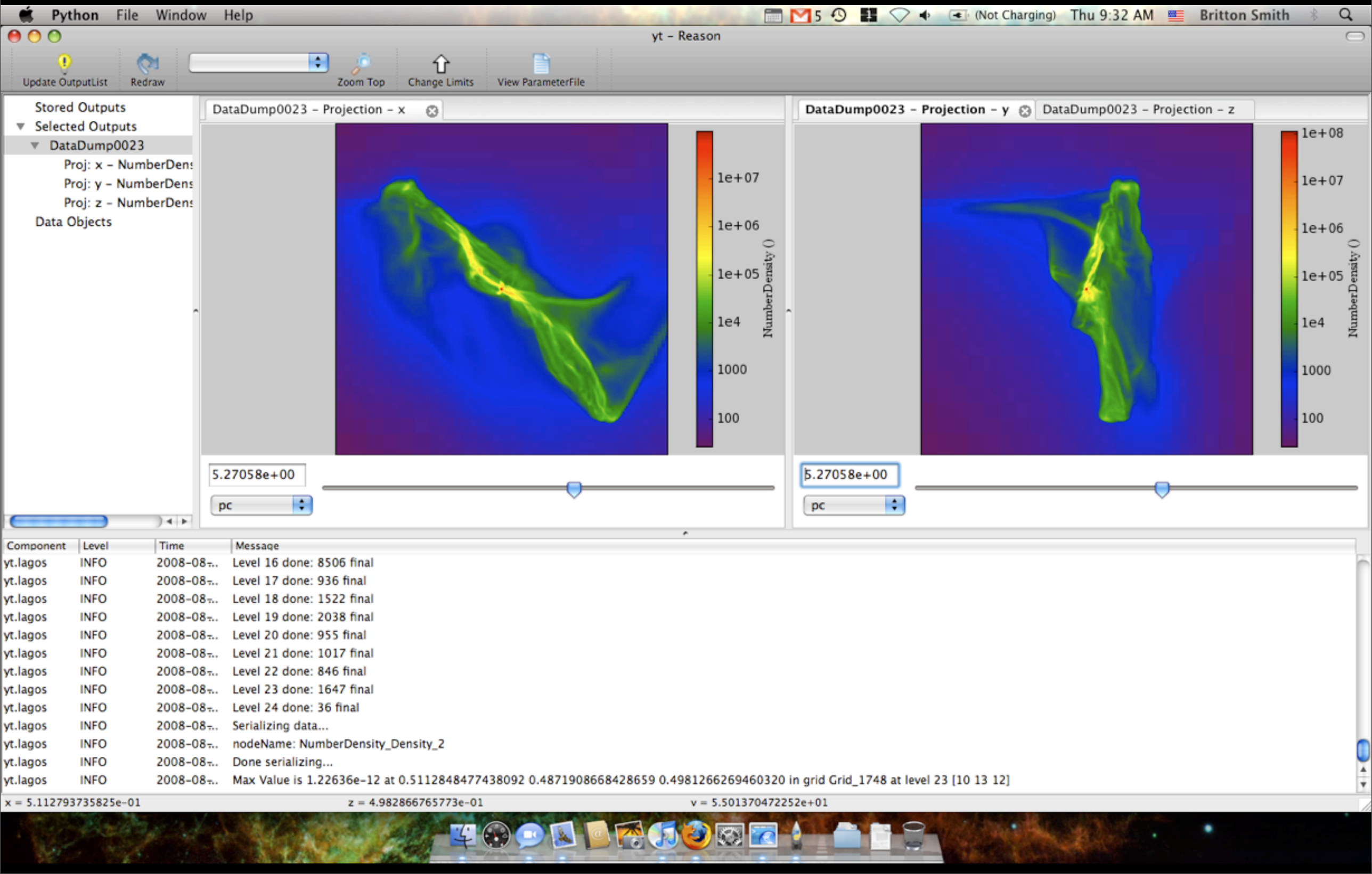
IPython

Terminal — screen — 179×49

```
yt.lagos   INFO      2008-08-22 00:48:38,641 Getting field z from 203
yt.lagos   DEBUG     2008-08-22 00:48:38,655 Going to obtain ['dx'] (['y', 'z', 'dx'])
yt.lagos   INFO      2008-08-22 00:48:38,656 Getting field dx from 203
yt.raven   DEBUG     2008-08-22 00:48:38,687 Pixelizing contour 5
yt.lagos   DEBUG     2008-08-22 00:48:38,688 Going to obtain ['y'] (['y'])
yt.lagos   INFO      2008-08-22 00:48:38,688 Getting field y from 203
yt.lagos   DEBUG     2008-08-22 00:48:38,692 Going to obtain ['z'] (['y', 'z'])
yt.lagos   INFO      2008-08-22 00:48:38,693 Getting field z from 203
yt.lagos   DEBUG     2008-08-22 00:48:38,699 Going to obtain ['dx'] (['y', 'z', 'c
yt.lagos   INFO      2008-08-22 00:48:38,699 Getting field dx from 203
yt.raven   DEBUG     2008-08-22 00:48:38,717 Pixelizing contour 6
yt.lagos   DEBUG     2008-08-22 00:48:38,718 Going to obtain ['y'] (['y'])
yt.lagos   INFO      2008-08-22 00:48:38,718 Getting field y from 203
yt.lagos   DEBUG     2008-08-22 00:48:38,722 Going to obtain ['z'] (['y', 'z'])
yt.lagos   INFO      2008-08-22 00:48:38,722 Getting field z from 203
yt.lagos   DEBUG     2008-08-22 00:48:38,727 Going to obtain ['dx'] (['y', 'z', 'c
yt.lagos   INFO      2008-08-22 00:48:38,727 Getting field dx from 203
yt.raven   DEBUG     2008-08-22 00:48:38,743 Pixelizing contour 7
yt.lagos   DEBUG     2008-08-22 00:48:38,743 Going to obtain ['y'] (['y'])
yt.lagos   INFO      2008-08-22 00:48:38,743 Getting field y from 203
yt.lagos   DEBUG     2008-08-22 00:48:38,749 Going to obtain ['z'] (['y', 'z'])
yt.lagos   INFO      2008-08-22 00:48:38,749 Getting field z from 203
yt.lagos   DEBUG     2008-08-22 00:48:38,755 Going to obtain ['dx'] (['y', 'z', 'c
yt.lagos   INFO      2008-08-22 00:48:38,755 Getting field dx from 203

In [18]: pc.set_width(0.05,'1')
yt.raven   DEBUG     2008-08-22 00:48:48,815 Received buffer of min 4.34971759893e
yt.raven   DEBUG     2008-08-22 00:48:48,955 Pixelizing contour 0
yt.raven   DEBUG     2008-08-22 00:48:48,966 Pixelizing contour 1
yt.raven   DEBUG     2008-08-22 00:48:48,977 Pixelizing contour 2
yt.raven   DEBUG     2008-08-22 00:48:48,988 Pixelizing contour 3
yt.raven   DEBUG     2008-08-22 00:48:49,001 Pixelizing contour 4
yt.raven   DEBUG     2008-08-22 00:48:49,021 Pixelizing contour 5
yt.raven   DEBUG     2008-08-22 00:48:49,033 Pixelizing contour 6
yt.raven   DEBUG     2008-08-22 00:48:49,047 Pixelizing contour 7

In [19]: pc.set_width(0.005,'1')
yt.raven   DEBUG     2008-08-22 00:48:55,261 Received buffer of min 5.17851764088e
yt.raven   DEBUG     2008-08-22 00:48:55,403 Pixelizing contour 0
yt.raven   DEBUG     2008-08-22 00:48:55,415 Pixelizing contour 1
yt.raven   DEBUG     2008-08-22 00:48:55,426 Pixelizing contour 2
yt.raven   DEBUG     2008-08-22 00:48:55,437 Pixelizing contour 3
yt.raven   DEBUG     2008-08-22 00:48:55,452 Pixelizing contour 4
yt.raven   DEBUG     2008-08-22 00:48:55,487 Pixelizing contour 5
yt.raven   DEBUG     2008-08-22 00:48:55,499 Pixelizing contour 6
yt.raven   DEBUG     2008-08-22 00:48:55,511 Pixelizing contour 7

In [20]:
Macintosh-96 0.70 0.72 0.67 |  0* yt  1 vim  2- crm-114
```



wxPython


Python File Window Help
(Not Charging) Thu 9:32 AM Britton Smith
yt – Reason
Update OutputList
Redraw
Zoom Top
Change Limits
View ParameterFile
Stored Outputs
Selected Outputs
DataDump0023
Proj: x – NumberDens
Proj: y – NumberDens
Proj: z – NumberDens
Data Objects
DataDump0023 – Projection – x
DataDump0023 – Projection – y
DataDump0023 – Projection – z
1e+07
1e+06
1e+05
1e4
1000
100
NumberDensity ()
1e+08
5.27058e+00
pc
Component Level Time Message
yt.lagos INFO 2008-08-.. Level 16 done: 8506 final
yt.lagos INFO 2008-08-.. Level 17 done: 936 final
yt.lagos INFO 2008-08-.. Level 18 done: 1522 final
yt.lagos INFO 2008-08-.. Level 19 done: 2038 final
yt.lagos INFO 2008-08-.. Level 20 done: 955 final
yt.lagos INFO 2008-08-.. Level 21 done: 1017 final
yt.lagos INFO 2008-08-.. Level 22 done: 846 final
yt.lagos INFO 2008-08-.. Level 23 done: 1647 final
yt.lagos INFO 2008-08-.. Level 24 done: 36 final
yt.lagos INFO 2008-08-.. Serializing data...
yt.lagos INFO 2008-08-.. nodeName: NumberDensity_Density_2
yt.lagos INFO 2008-08-.. Done serializing...
yt.lagos INFO 2008-08-.. Max Value is 1.22636e-12 at 0.5112848477438092 0.4871908668428659 0.4981266269460320 in grid Grid_1748 at level 23 [10 13 12]
x = 5.112793735825e-01
z = 4.982866765773e-01
v = 5.501370472252e+01

Knoboo

python

mturk | Settings | Help | Logout

Untitled

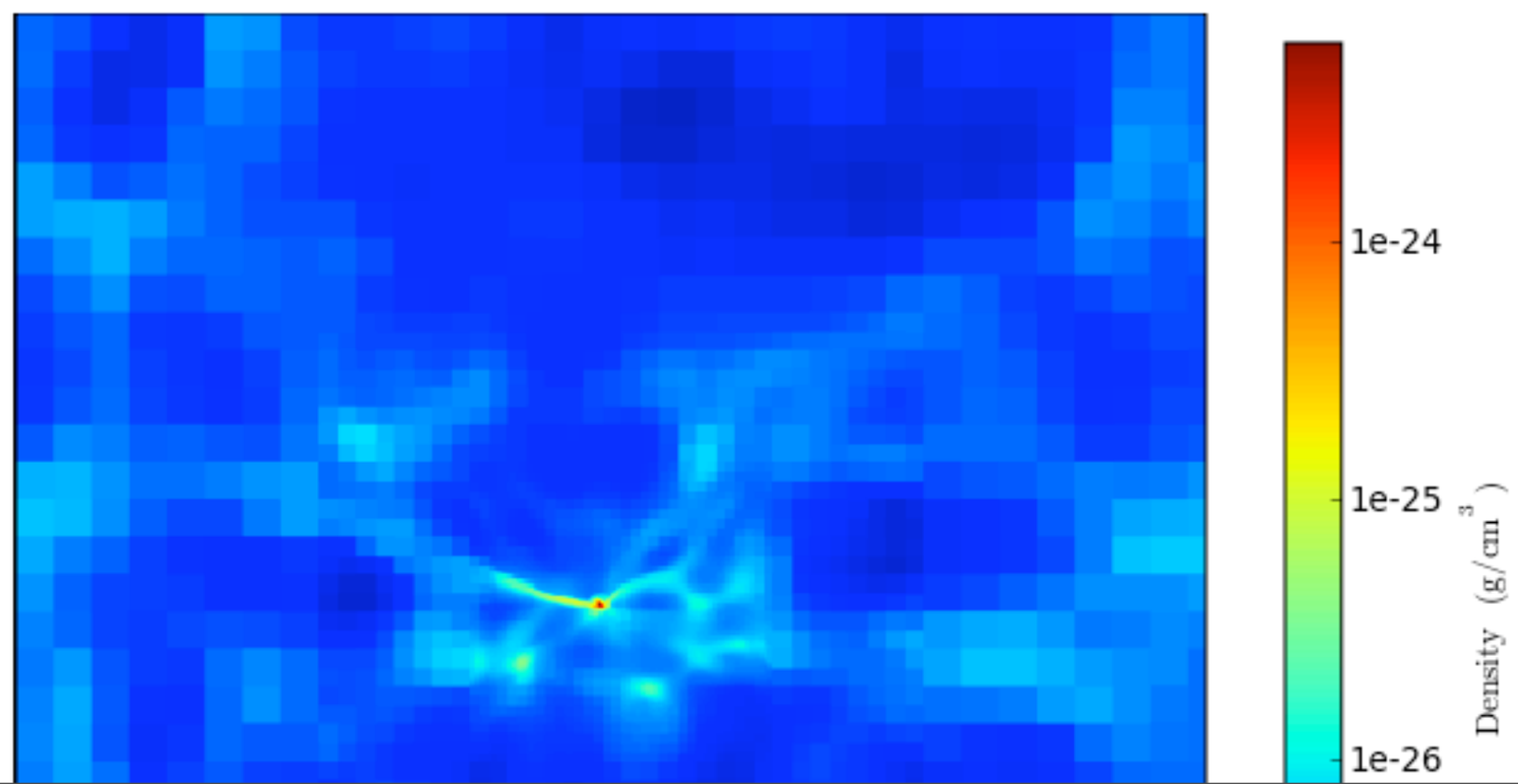
Save | Save & Close | Print | Interrupt | Kill

In[!]:
```
import sys
sys.path.insert(0,"/home/mturk/Development/yt/trunk/")
from yt.mods import *
pf = EnzoStaticOutput("/home/mturk/Research/data/RD0014/RedshiftOutput0014")
pc = PlotCollectionInteractive(pf)
pc.pylab.show = show
pc.add_slice("Density",0)
```

Out[!]:
```
Kernel Error: The interpreter might still be starting up.
```

In[2]:
```
pc.pylab.show()
```

Out[2]:



Web Gallery

THE DELIVERATOR

Stuff to do

- Home
- Select a Run
- Login

TURBOGEARS
UNDER THE HOOD

yt and the Deliverator by Matt Turk

Select your criteria

Parameterfile
DataDump0039
DataDump0040
DataDump0041
DataDump0042
DataDump0043
All
None

Type
Projection
Slice
ThreePhase
All
None

Width
1.0
10.0
100.0
1000.0
All
None

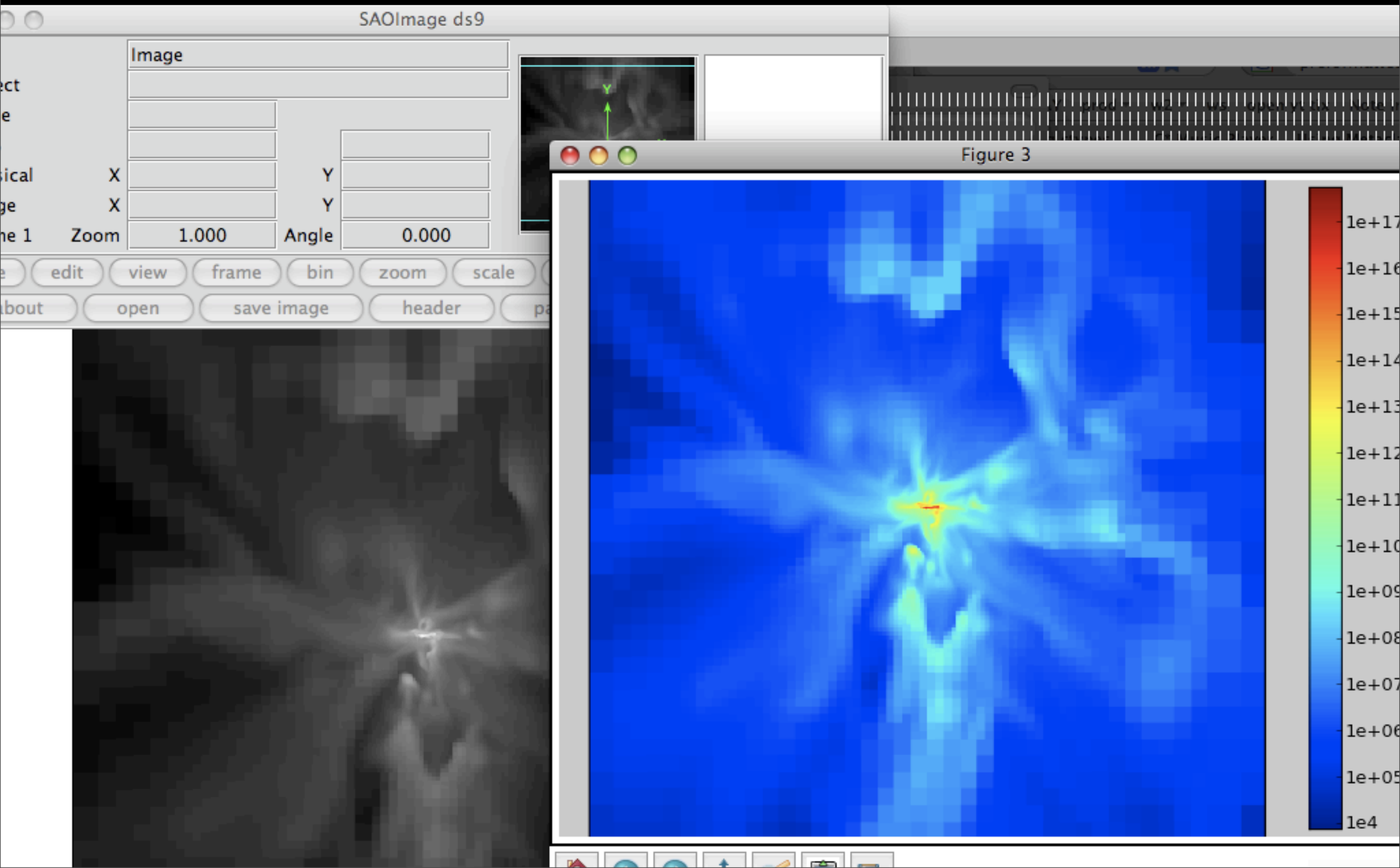
Unit
au
kpc
pc
rsun
All
None

Density

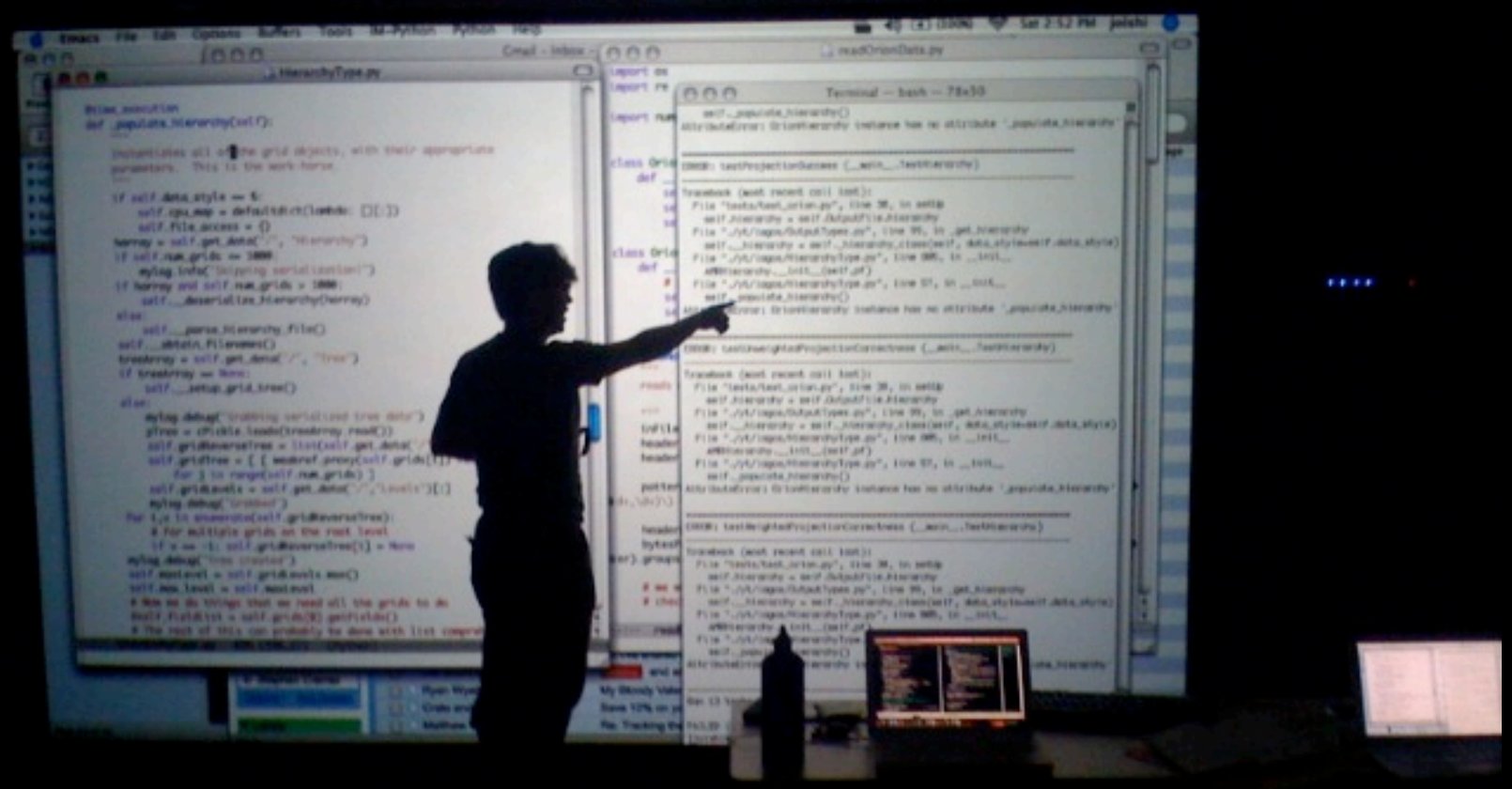
192 images will be returned

List Parameter Files

DS9


SAOImage ds9
Image
X
Y
X
Y
Zoom
1.000
Angle
0.000
edit
view
frame
bin
zoom
scale
open
save image
header
Figure 3
1e+17
1e+16
1e+15
1e+14
1e+13
1e+12
1e+11
1e+10
1e+09
1e+08
1e+07
1e+06
1e+05
1e4

Generalization

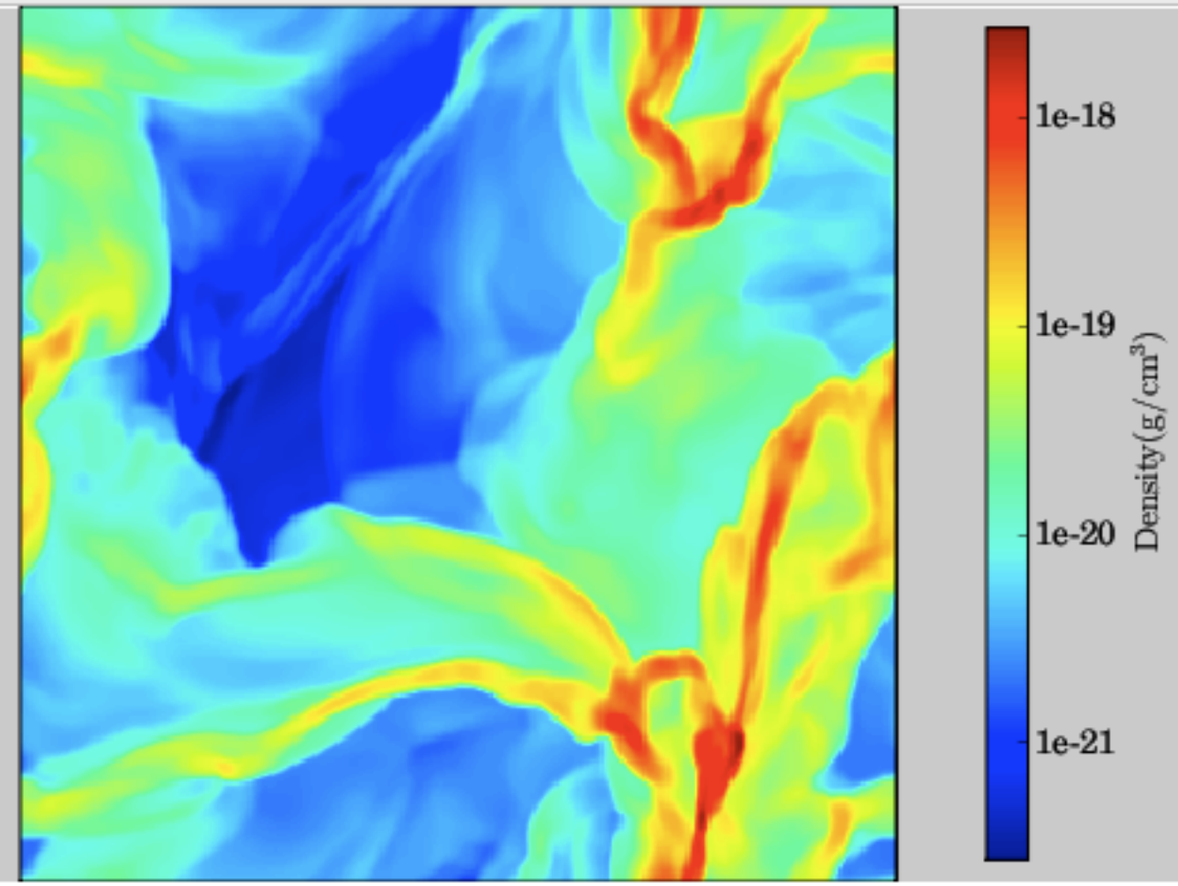


yt – Reason

Update OutputList | Redraw | Zoom Top | Change Limits | View ParameterFile

- Stored Outputs
- Selected Outputs
  - pltgmlcs5600
    - Slice: x
    - Slice: y
    - Slice: z
- Data Objects

pltgmlcs5600 – Slice – z



2.008e+18

1

Component	Level	Time	Message
yt.raven	ERROR	2008-06-..	Setting limits in y: -1.00400e+18 1.00400e+18
yt.raven	ERROR	2008-06-..	Setting limits in x: -1.00400e+18 1.00400e+18
yt.raven	ERROR	2008-06-..	Setting limits in y: -1.00400e+18 1.00400e+18
yt.raven	ERROR	2008-06-..	Setting limits in x: -1.00400e+18 1.00400e+18
yt.raven	ERROR	2008-06-..	Setting limits in y: -1.00400e+18 1.00400e+18

x = 3.198666965219e+17 y = 6.372230183609e+17 v = 1.047073817519e-20

Things yt Can Do

Slices
Projections
1-, 2- and 3-D Profiles
Halo finding
Fixed Resolution Extraction
Generic 3D objects
Extracted Sets
Wrap Enzo Fortran modules
Topologically Connected Sets
Derived Quantities
Particle support
Movies
Time series analysis
Galaxy merger trees
Simple data management
Web gallery
Parallel Analysis
Primitive problem generation support
X-Ray spectrum (from 'Cloudy')

What next?

Committed to Open Science

Committed to Open Science

No Private Branches

Committed to Open Science

No Private Branches

No Secret Features

Committed to Open Science

No Private Branches

No Secret Features

Free Repository Space

Acknowledgments

Enzo: Greg Bryan, Mike Norman, Tom Abel,
Brian O'Shea (and on and on)
YT: Jeff Oishi, Britton Smith, David Collins

Sponsored by: KIPAC, Stanford, and LANL

Thank you.

mturk@stanford.edu
http://yt.enzotools.org/