yt Lightcone Observations

In this exercise we will construct a lightcone dataset, that uses the Enzo outputs at such intervals that it can construct a synthetic lightcone of data that extends from an initial to final redshift. This is particularly useful for construction mock catalogs of observed galaxies and clusters of galaxies. In this example we'll create two such datasets, one of the Sunyaev-Zel'dovich effect, and one of the X-ray emission. For more information, see the yt online documentation.

In [1]:
```
%matplotlib inline
```

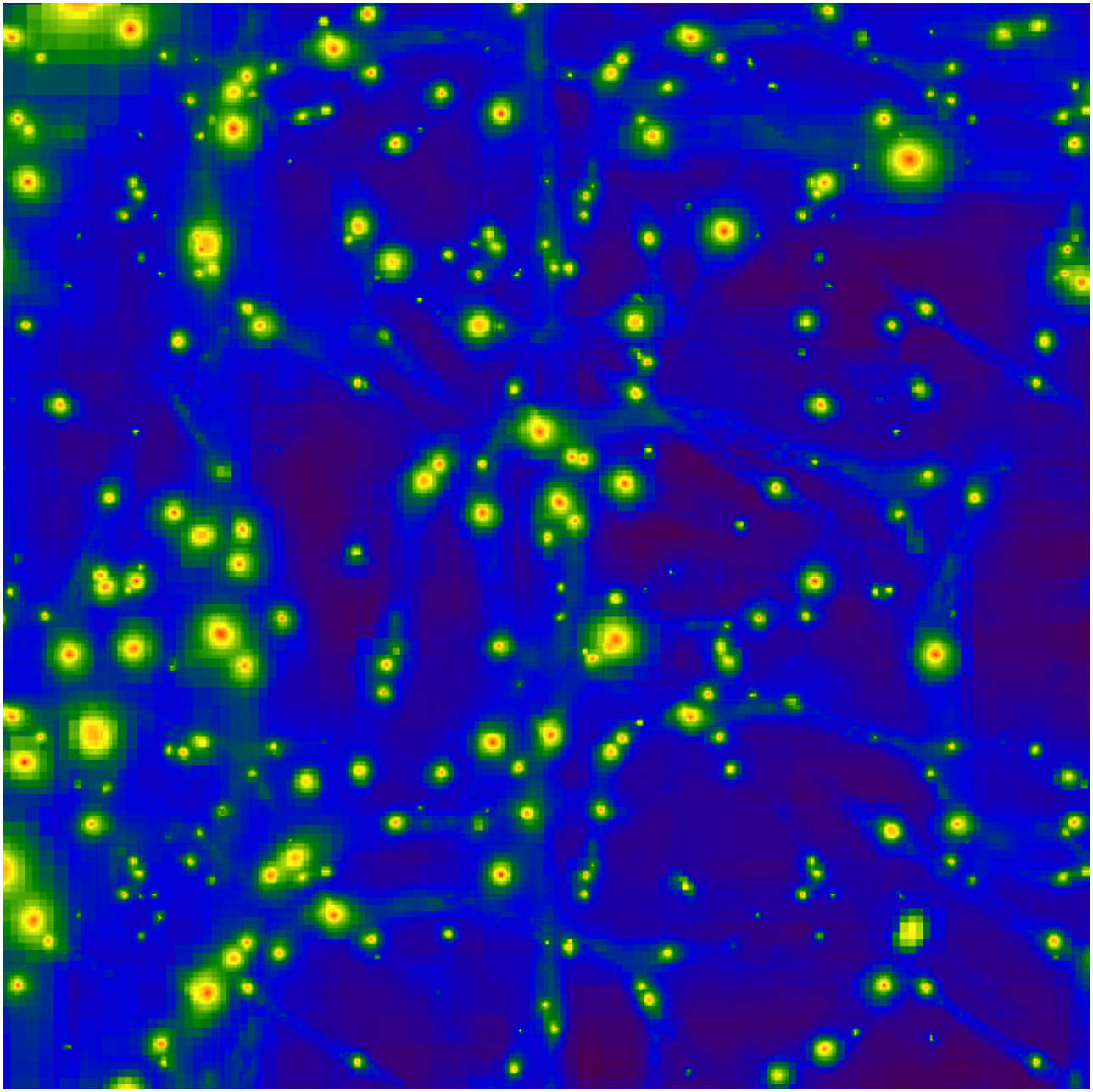
In [2]:
```
import yt
from yt.analysis_modules.cosmological_observation.api import \
     LightCone

# Create a LightCone object extending from z = 0 to z = 0.1.

# We have already set up the redshift dumps to be
# used for this, so we will not use any of the time
# data dumps.
lc = LightCone('../../sample_data/enzo_tiny_cosmology/32Mpc_32.enzo',
               'Enzo', 0., 0.1,
               observer_redshift=0.0,
               time_data=False)
```

In [3]:
```
# Calculate a randomization of the solution.
lc.calculate_light_cone_solution(seed=123456789, filename="LC/solution.txt")

# Choose the field to be projected.
field = 'szy'

# Use the LightCone object to make a projection with a 600 arcminute
# field of view and a resolution of 60 arcseconds.
# Set njobs to -1 to have one core work on each projection
# in parallel.
lc.project_light_cone((600.0, "arcmin"), (60.0, "arcsec"), field,
                      weight_field=None,
                      save_stack=True,
                      save_final_image=True,
                      save_slice_images=True,
                      njobs=-1)
```

```
/home/skillman/local/yt-x86_64/src/yt-3.0/yt/analysis_modules/cosmological_observation/light_cone
/light_cone.py:359: RuntimeWarning: divide by zero encountered in log10
  only_on_root(write_image, np.log10(my_image),
```

In [5]:
```
from IPython.display import Image
Image("LC/LightCone_szy.png")
```

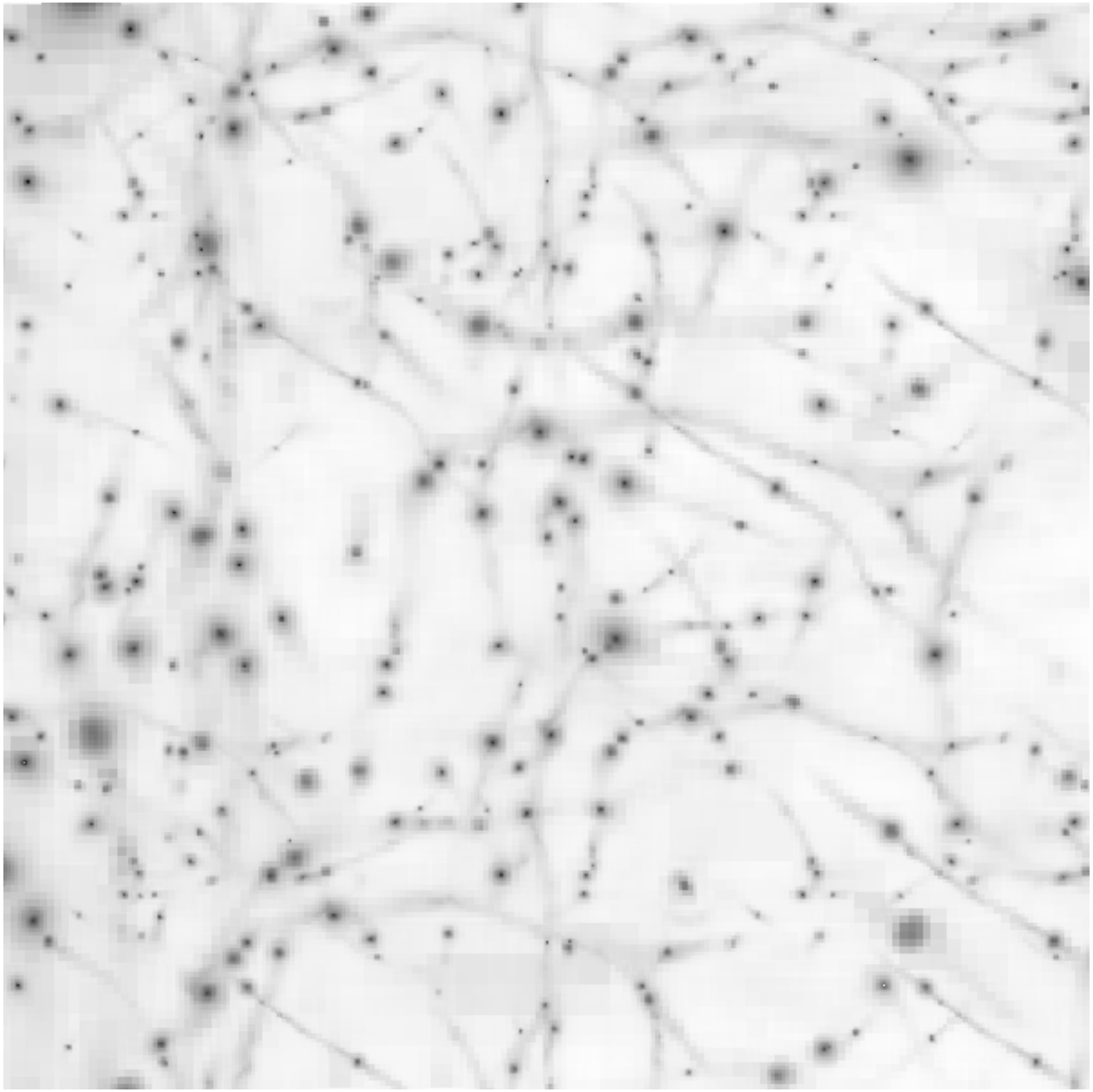
Out[5]:



In [6]:
```
# Calculate a randomization of the solution.
lc.calculate_light_cone_solution(seed=123456789, filename="LC/solution.txt")

# Choose the field to be projected.
field = 'xray_emissivity'

# Use the LightCone object to make a projection with a 600 arcminute
# field of view and a resolution of 60 arcseconds.
# Set njobs to -1 to have one core work on each projection
# in parallel.
lc.project_light_cone((600.0, "arcmin"), (60.0, "arcsec"), field,
                      weight_field=None,
                      save_stack=True,
                      save_final_image=True,
                      save_slice_images=True,
                      njobs=-1, cmap_name='Greys')
```

In [7]:
```
Image("LC/LightCone_xray_emissivity.png")
```

Out[7]:



In []: