ASTR 119: Session 20

Simulation Visualization

Outline

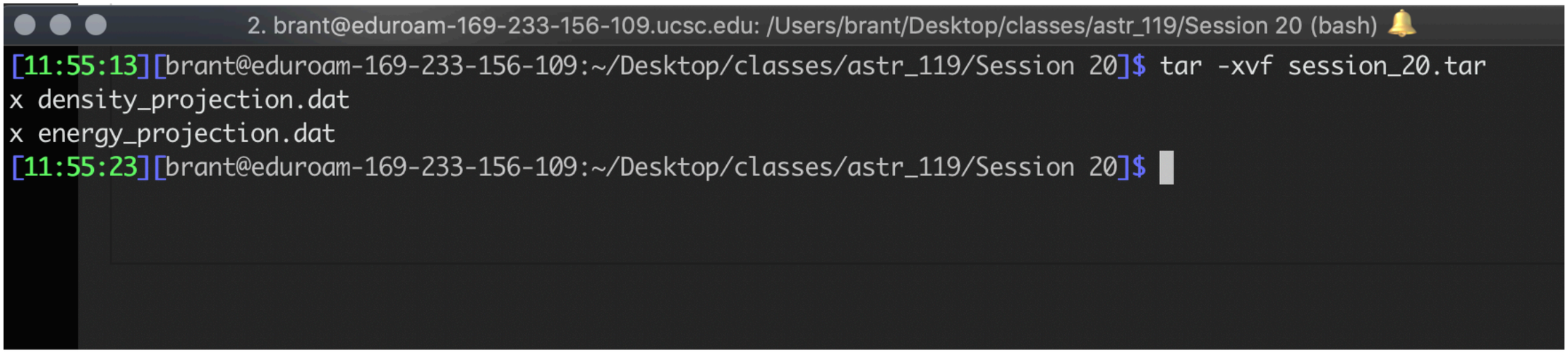
1) Final projects: **Due Tuesday, December 15, 2020 at 3pm.**
2) Visualization of the Day: Let's make our own!
3) Final project organization time
4) Save your work to GitHub

Policy about Final Project Help

1) We (prof + TAs) *want* to help you!
2) We are close to the deadline, and we can't help at the last hour.
3) Emails asking for help before midnight Monday we will try to respond to, but we need 24hrs to respond(!).

Simulation Visualization

Please download the tarball “session_20.tar” from Canvas, save it to your desktop, and then expand it:



On Windows, just download the density_projection.dat and energy_projection.dat files directly.

Simulation Visualization

A visualization of a astrophysical simulation

There are two files, "density_projection.dat" and "energy_projection.dat", that contain the gas density and internal energy from a slice through a cosmological simulation. Let's read them in and make a three color image.

In [12]:
```
%matplotlib inline
import matplotlib.pyplot as plt
import matplotlib.colors as colors
import numpy as np
```

Simulation Visualization

Read in the density file.

The file is a 512x512 projection of a 512^3 dataset. Wec can read it in using numpy's fromfile() function and reshape it into 512x512 using numpy's reshape().

In [5]:
```
fname = "density_projection.dat"
d = np.fromfile(fname,dtype=np.float32)
d = np.reshape(d,(512,512))
```

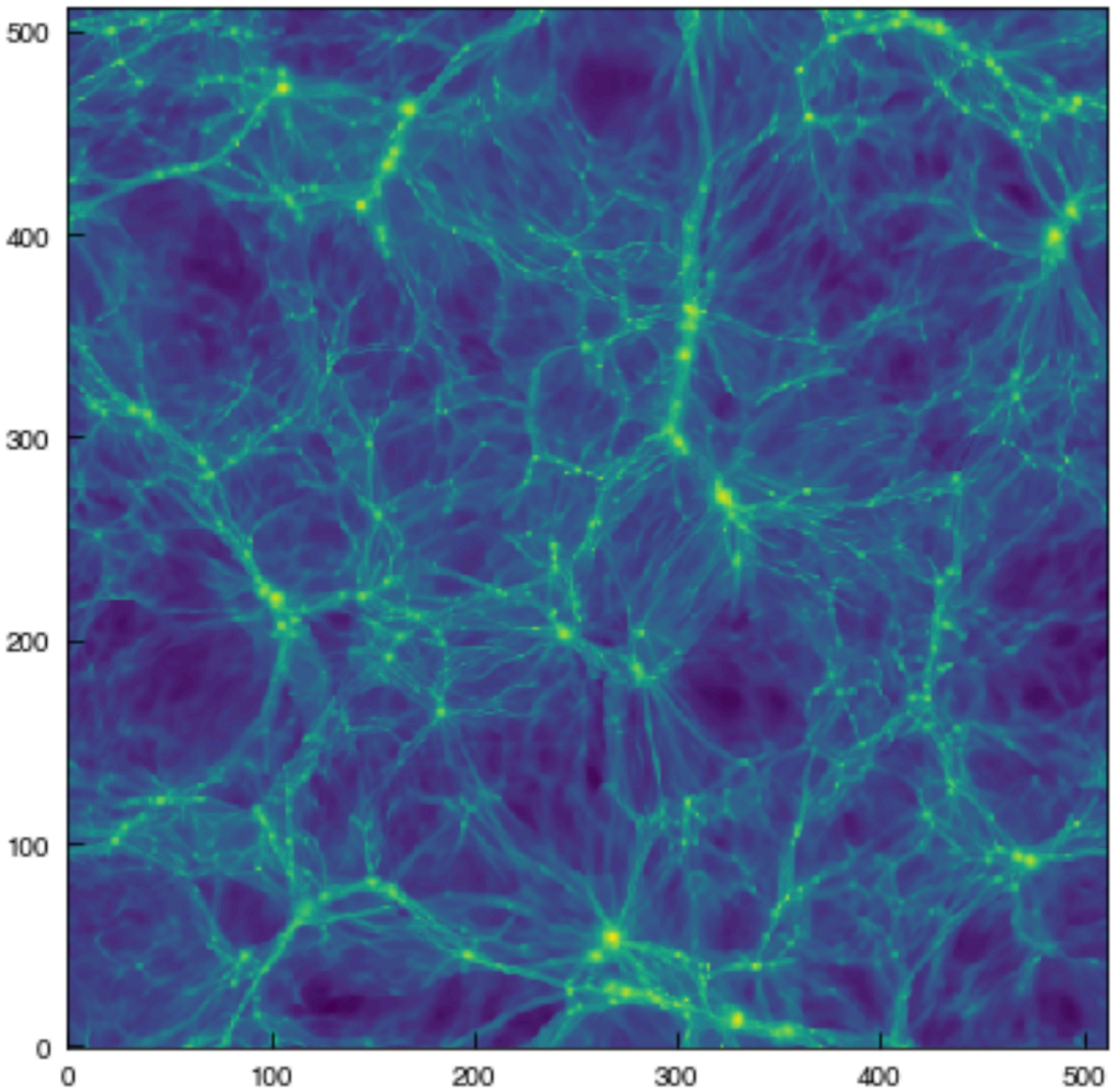
Repeat for the energy file.

In [6]:
```
fname = "energy_projection.dat"
energy = np.fromfile(fname,dtype=np.float32)
energy = np.reshape(energy,(512,512))
```

Simulation Visualization

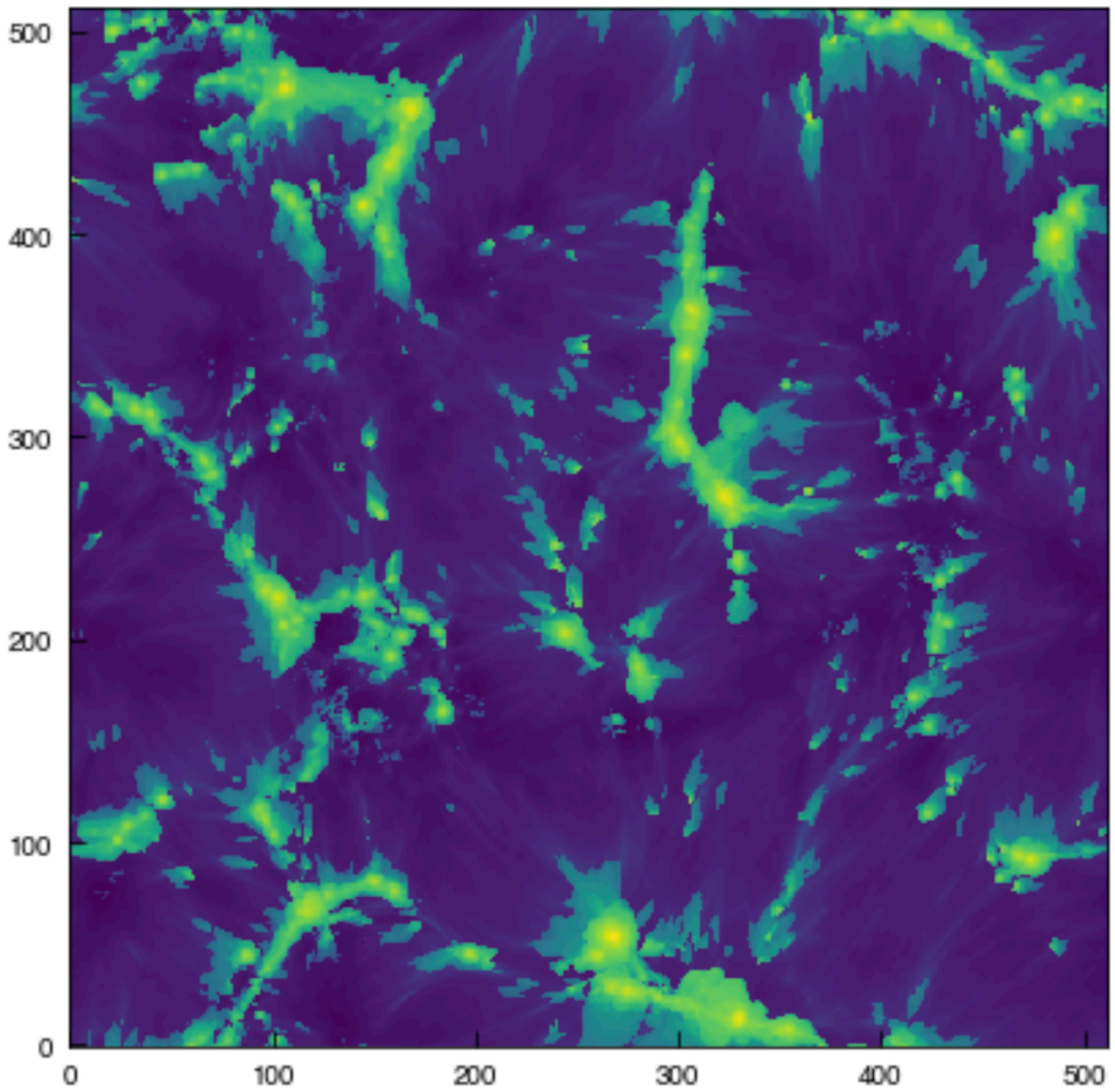
Let's plot them using the default color map.

```
d_log = np.log10(d)
f = plt.figure(figsize=(7,7))
plt.imshow(d_log)
```



Simulation Visualization

In [8]:
```
e_log = np.log10(energy)
f = plt.figure(figsize=(7,7))
plt.imshow(e_log)
```



Simulation Visualization

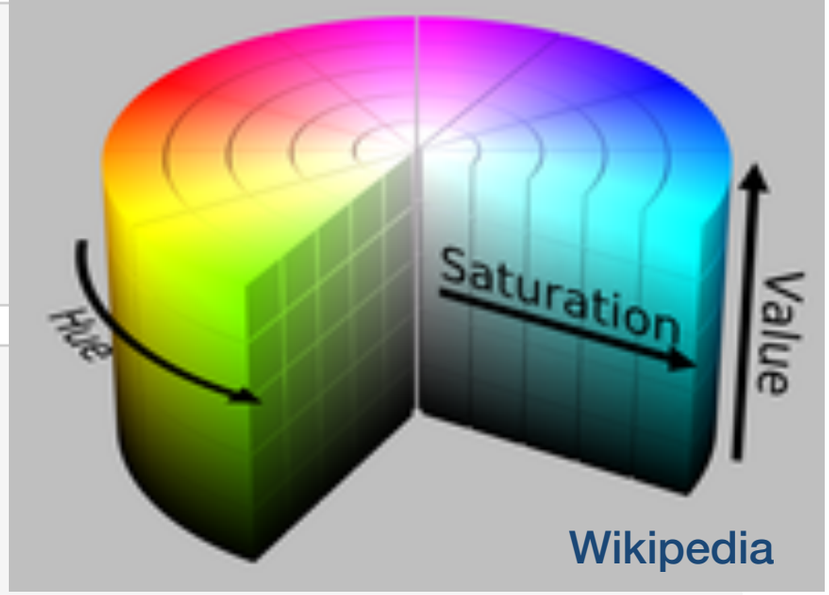
Making a 3-color image

We can combine the density and energy maps into a three color image using the HSV color space. Here H=[0,1] corresponds to the color wheel going from red->red through yellow, green, blue, and purple. V=[0,1] is the intensity of the image. S=[0,1] is the saturation of the color, with 0 being white (for V=1) or black (for V=0) and 1 being a deep color.

We will map density to V and S, and energy to H.

In [9]:
```
d_min = d_log.min()
d_max = d_log.max()

v = (d_log - d_min)/(d_max-d_min)
s = 1.0 - v
```

In [10]:
```
e_min = e_log.min()
e_max = e_log.max()

h = 0.8 - 0.2*(e_log - e_min)/(e_max-e_min)
```



Simulation Visualization

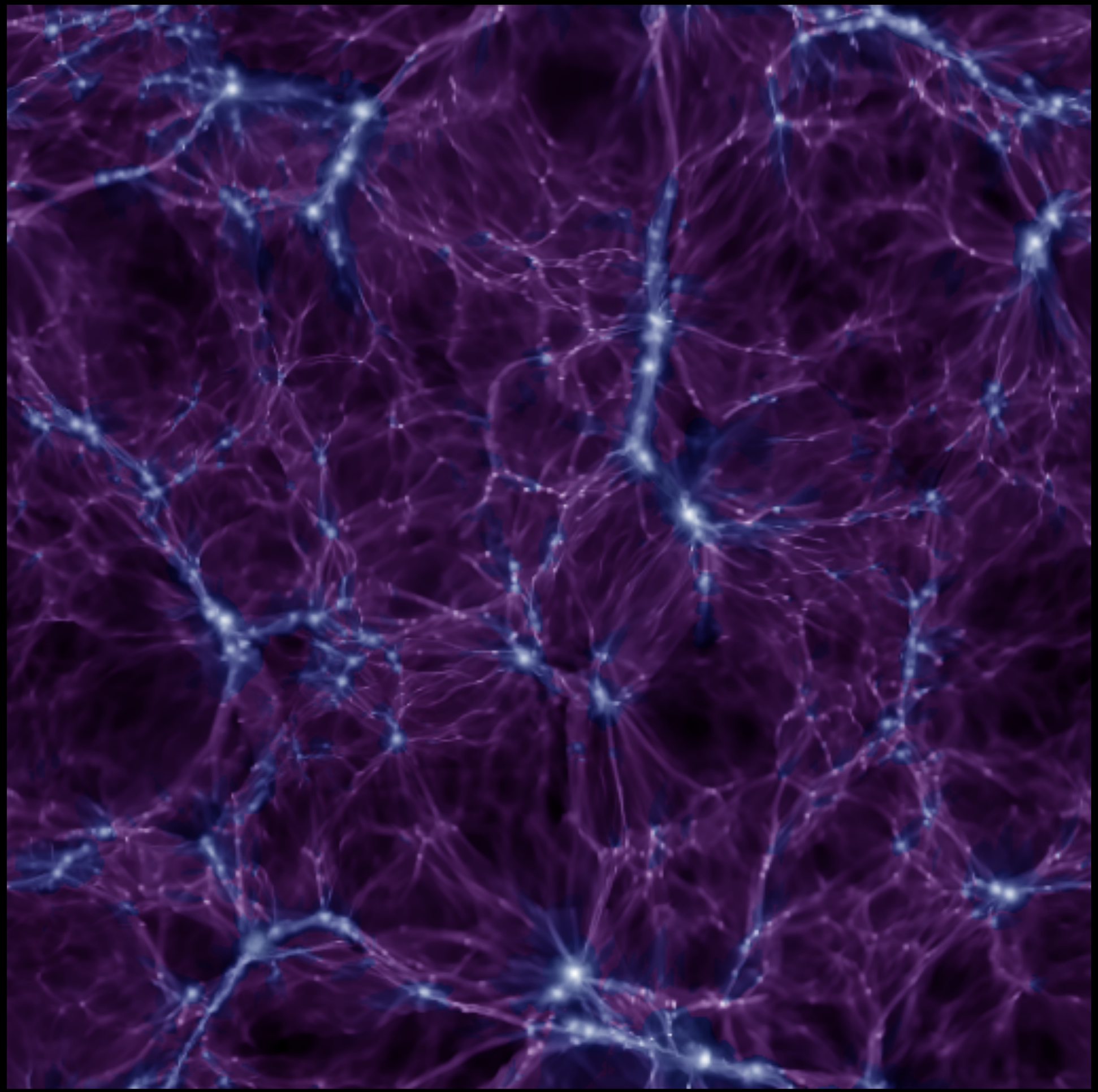
Now we have to make a HSV image, and then convert to RGB

In [14]:
```
hsv_image = np.zeros((512,512,3))

hsv_image[:,:,0] = h
hsv_image[:,:,1] = s
hsv_image[:,:,2] = v

rgb_image = colors.hsv_to_rgb(hsv_image)
```

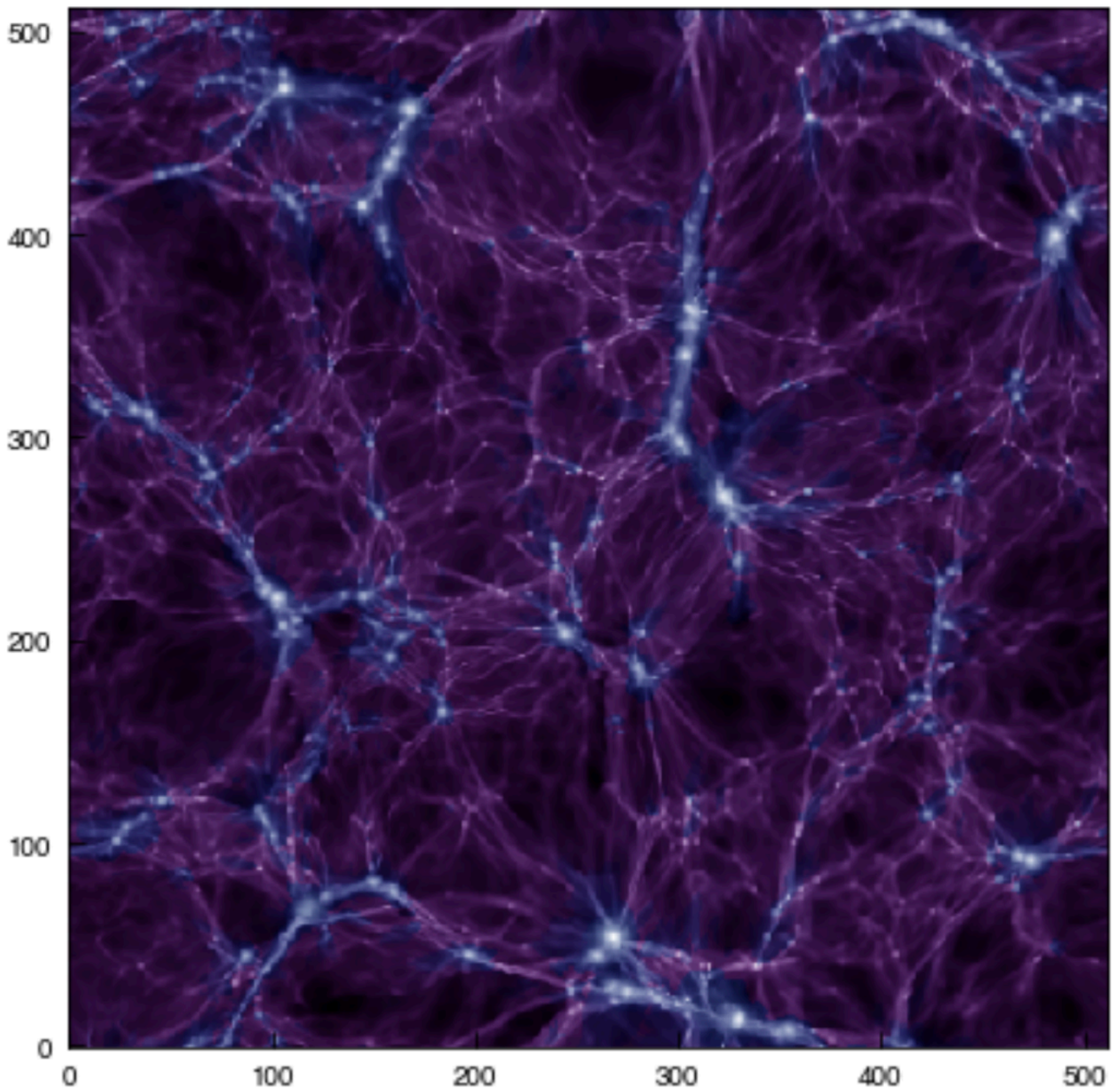
Now let's see the 3-color image

In [16]:
```
f = plt.figure(figsize=(7,7))
plt.imshow(rgb_image)
```

Out[16]:
```
<matplotlib.image.AxesImage at 0x11087a9b0>
```



Simulation Visualization



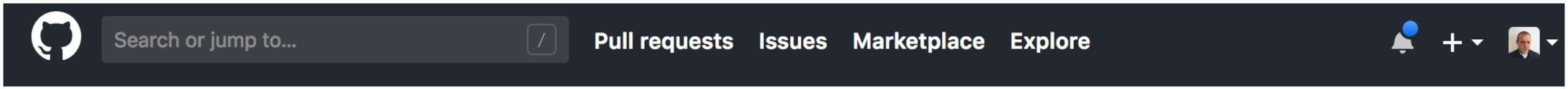
Lastly, let's save the figure

19]:
```
plt.imsave("test.png", rgb_image)
```

[ ]:

Save Your Work

Make a GitHub project “astr-119-session-20”, and commit the programs you made today.



Create a new repository

A repository contains all the files for your project, including the revision history.

Owner **Repository name**

brantr / astr-119-session-5

Great repository names are short and memorable. Need inspiration? How about **fantastic-spork**.

Description (optional)

We learned a new trick! -- Jupyter notebooks.

Public
Anyone can see this repository. You choose who can commit.

Private
You choose who can see and commit to this repository.