Using the yt Cosmology Calculator

The cosmology calculator is a super useful tool for helping plan and analyze your cosmological simulations. In this exercise, we're going to load up a dataset, then access things like the comoving radial distance and angular scales for the cosmology used for the simulation.

In [1]:
```
import yt
import numpy as np
```

In [2]:
```
ds = yt.load("../../sample_data/enzo_cosmology_plus/DD0046/DD0046")
```

After a Dataset is loaded, we can access the cosmology tool with ds.cosmology. From it, we can calculate things like the comoving radial distance from z=0 to z=2 in comoving Mpc/h:

In [4]:
```
ds.cosmology
```

Out[4]:
```
<yt.utilities.cosmology.Cosmology at 0x7fb36114b990>
```

In [5]:
```
ds.cosmology.comoving_radial_distance(0.0, 2.0).in_units("Mpccm/h")
```

Out[5]:
```
3730.68924253 Mpccm/h
```

Let's see how a few cosmological distances vary with starting and ending redshifts.

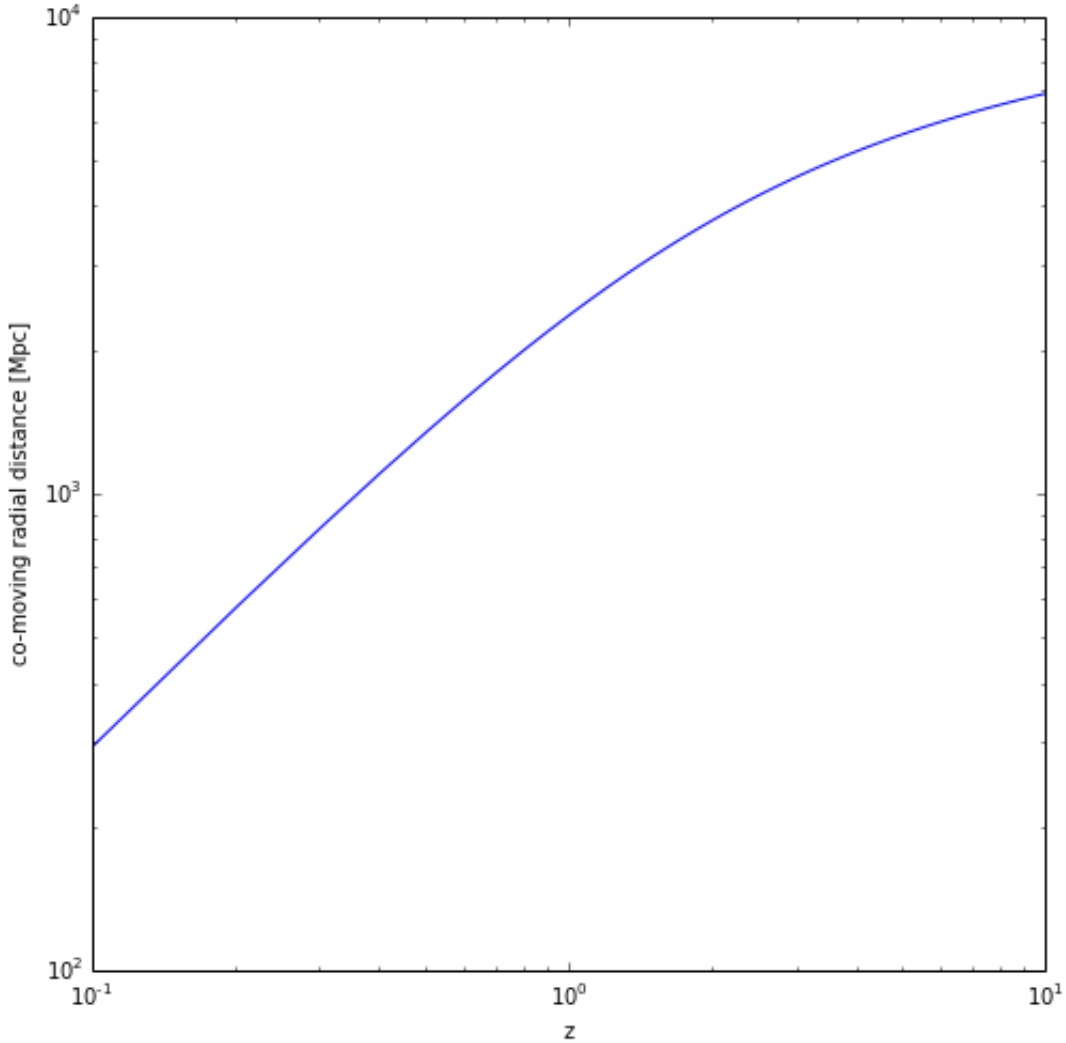
In [6]:
```
zrange = np.linspace(0.0, 10.0, 100)
```

In [7]:
```
cm_radial = [ds.cosmology.comoving_radial_distance(0.0, z).in_units("Mpc/h") for z in zrange]
lookback = [ds.cosmology.lookback_time(0.0, z).in_units("Gyr") for z in zrange]
angular = [ds.cosmology.angular_scale(0.0, z).in_units("kpc/arcsec") for z in zrange]
```

In [8]:
```
%matplotlib inline
import matplotlib.pylab as pl
```

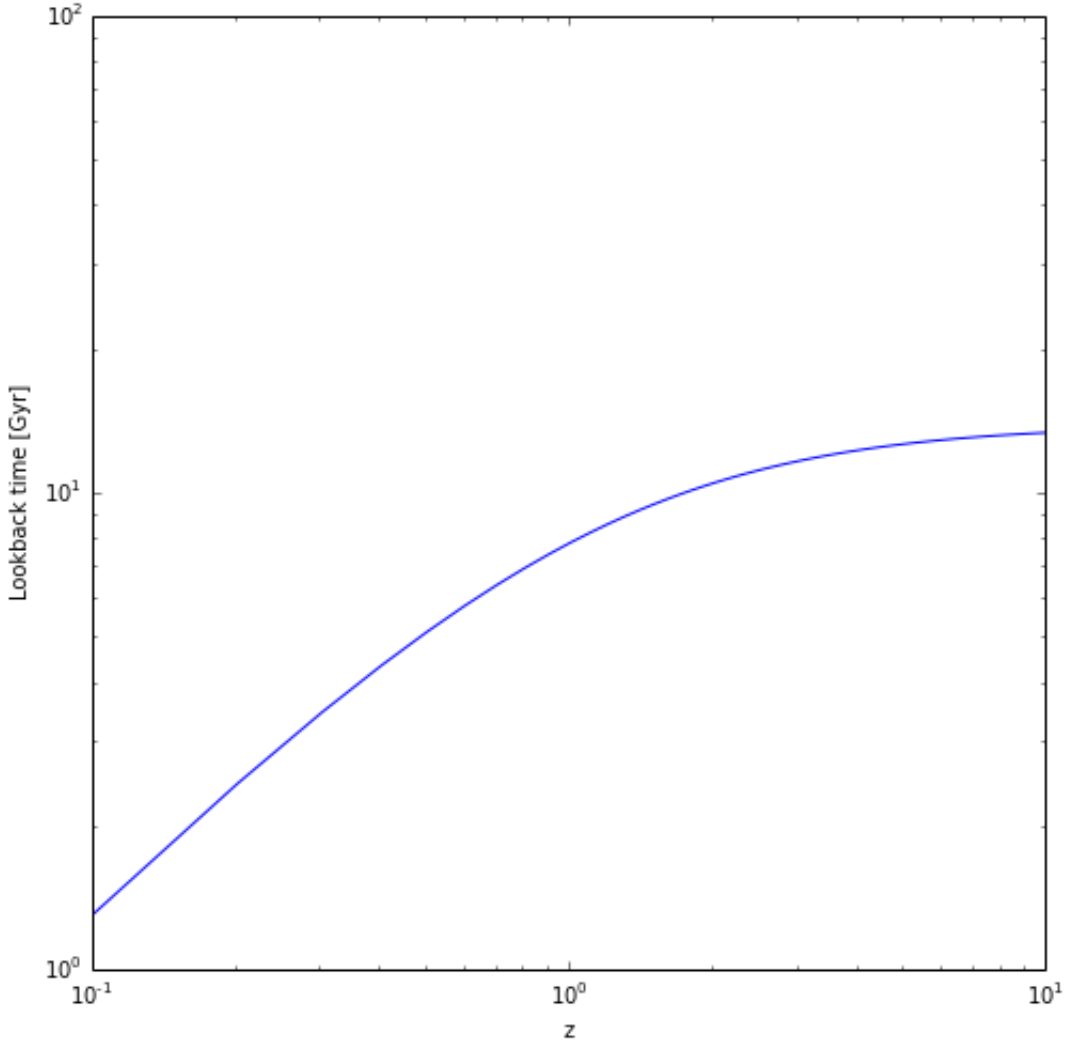
In [9]:
```
pl.figure(figsize=[8,8])
pl.loglog(zrange, cm_radial)
pl.xlabel("z")
pl.ylabel("co-moving radial distance [Mpc]")
```

Out[9]: <matplotlib.text.Text at 0x7fb35e7a5f50>



In [10]:
```
pl.figure(figsize=[8,8])
pl.loglog(zrange, lookback)
pl.xlabel("z")
pl.ylabel("Lookback time [Gyr]")
```

Out[10]: <matplotlib.text.Text at 0x7fb35e5a4190>



In []:
```
pl.figure(figsize=[8,8])
pl.loglog(zrange, angular)
pl.xlabel("z")
pl.ylabel("Angular Scale [kpc/arcsec]")
```

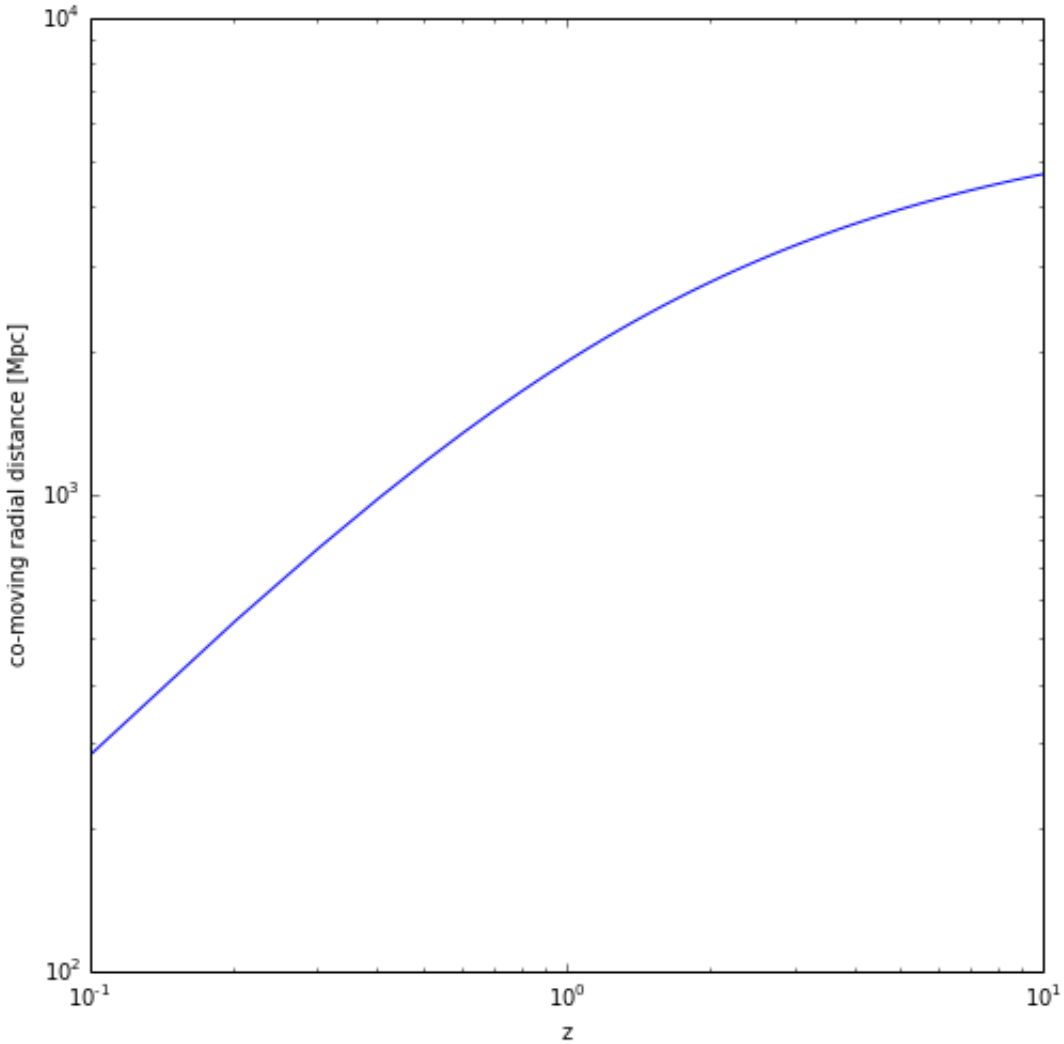
Next let's play a game where we guess what the Planck2018 cosmology will be, and see how things change. We can initalize a new cosmology using yt.utitlities.cosmology.Cosmology(). Check out the online docs or the docstrings associated with that object for information on initializing your favorite cosmology.

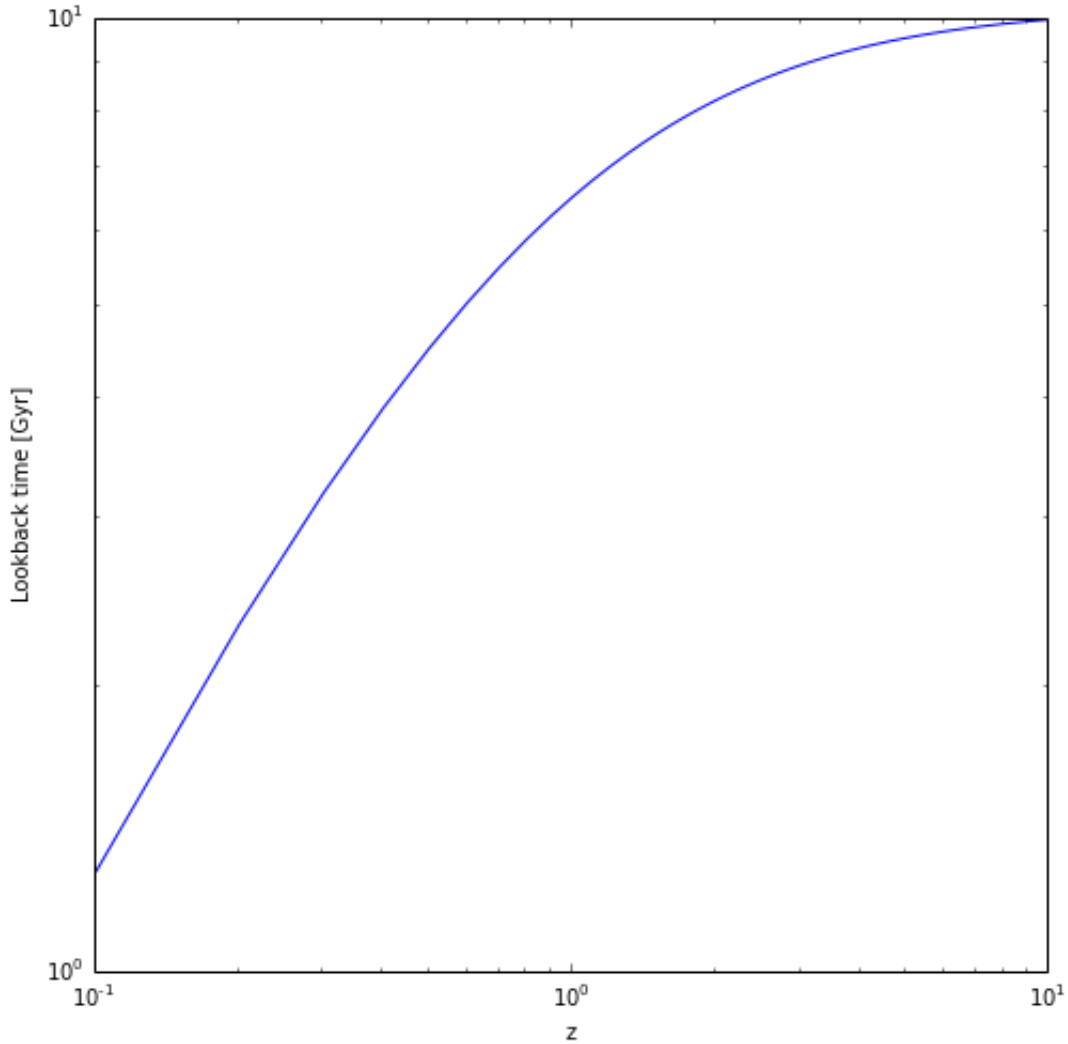
In [12]:
```
planck2018 = yt.utilities.cosmology.Cosmology(hubble_constant=0.7,
                                              omega_matter=0.75,
                                              omega_lambda=0.25,
                                              omega_curvature=0.0)
```

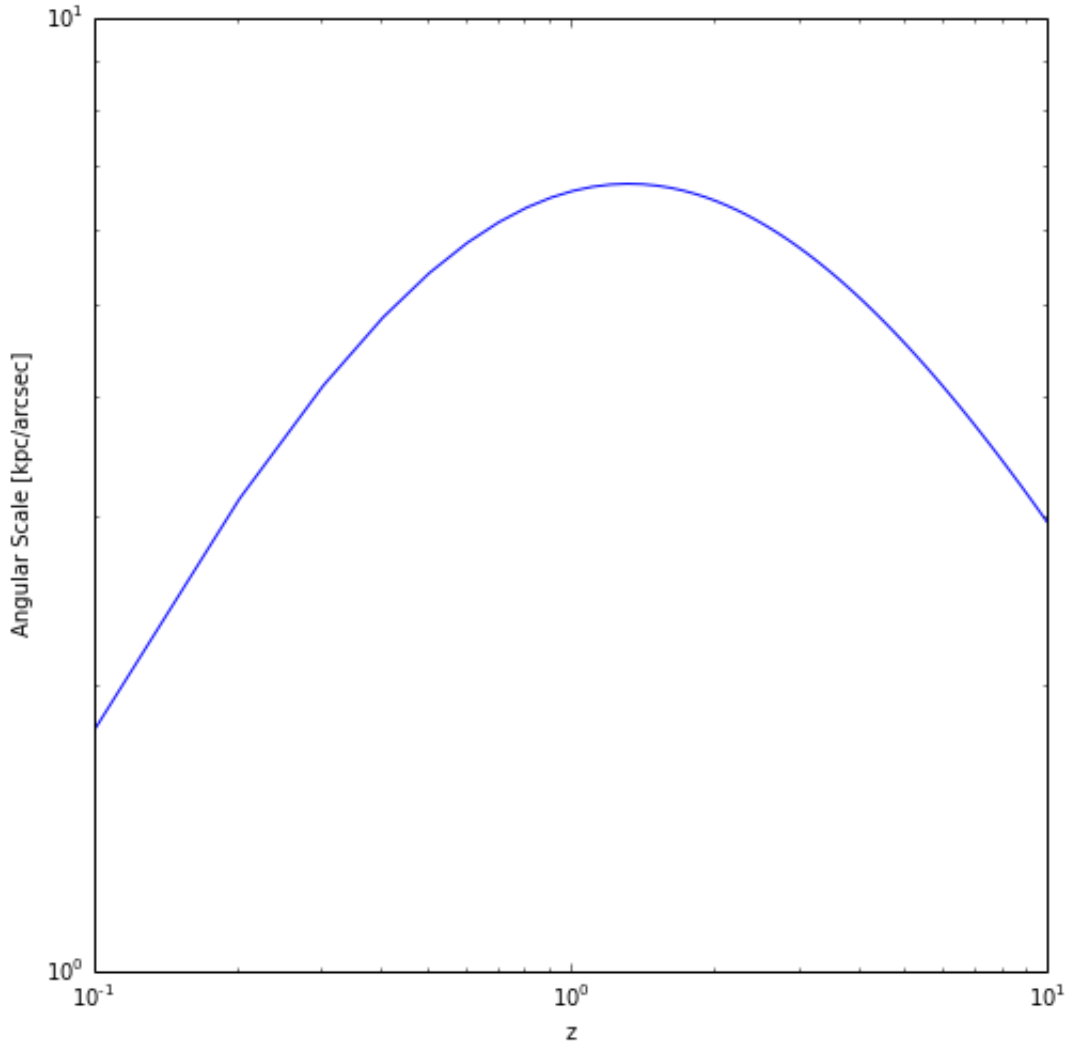
In [13]:
```
cm_radial = [planck2018.comoving_radial_distance(0.0, z).in_units("Mpc/h") for z in zrange]
lookback = [planck2018.lookback_time(0.0, z).in_units("Gyr") for z in zrange]
angular = [planck2018.angular_scale(0.0, z).in_units("kpc/arcsec") for z in zrange]
```

In [14]:
```
pl.figure(figsize=[8,8])
pl.loglog(zrange, cm_radial)
pl.xlabel("z")
pl.ylabel("co-moving radial distance [Mpc]")

pl.figure(figsize=[8,8])
pl.loglog(zrange, lookback)
pl.xlabel("z")
pl.ylabel("Lookback time [Gyr]")

pl.figure(figsize=[8,8])
pl.loglog(zrange, angular)
pl.xlabel("z")
pl.ylabel("Angular Scale [kpc/arcsec]")
```

Out[14]:
```
<matplotlib.text.Text at 0x7fb35e1c84d0>
```


co-moving radial distance [Mpc]
z
10^4
10^3
10^2
10^{-1}
10^0
10^1


10^1
10^0
Lookback time [Gyr]
10^{-1}
10^0
10^1
z



In []: