ASTR232, Fall 2019

Gil Garcia[1]

Homework # 7

Problem 1a

You have received a file containing information about brightness and spatial distribution of galaxies in the local universe. The galaxies have a distance between 18 Mpc and 80 Mpc and are located within a wedge of sky that spans 90° in right ascension and has a thickness of 2.1 Mpc. Assuming $H_0 = 73$ km s^{-1} Mpc^{-1}, use the redshift information to calculate the distance d to each galaxy. Then, make a plot showing the distribution of galaxies in RA and d. Discuss the distribution of galaxies in your slice in the context of the Cosmological Principle.

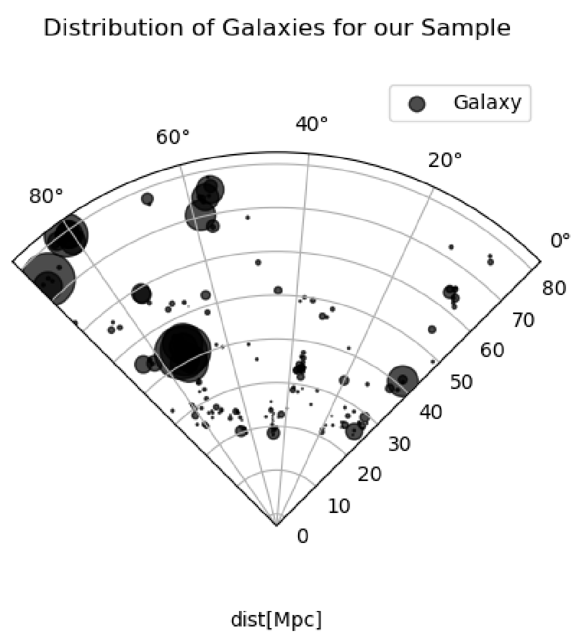
Solution

To calculate the distances to each galaxy, we use the following formula:

$$d = \frac{zc}{H_o} \tag{1}$$

where $H_0 = 73$ km s^{-1} Mpc^{-1} and $c = 299{,}792$ km s^{-1}.

We plug in our z values and using the distances and RA, we produce the following plot (made after computing masses so that size of point reflects relative mass of galaxy):



From our plot above, we see that our sample of galaxies does not follow the Cosmological Principle. There is not an even spread of galaxies throughout our wedge as some boxes contain many galaxies while other boxes in the grid contain few or none. This violated the isotropic law. Additionally, the make up would not look the same at all locations in this wedge. If we pick any two galaxies in the plot, the arrangement of nearby galaxies will likely not look very similar. This violates the homogeneous law. Thus, our distribution of galaxies does not follow the Cosmological Principle. We can argue that this is maybe not a large enough spatial scale, but even when considering cosmic structures like the Laniakea super cluster, we do not see the CP being followed very well.

---

[1] Do you wanna build a snowman?

Problem 1b

Computer the stellar mass of each galaxy an determine the combined stellar mass of the galaxies within you slice. Then, assuming stellar mass is $\sim$15% of total mass (the rest being dark matter), calculate combined *total* mass. Use this to estimate the density of matter $\rho_{m.0}$ in the local universe.

Solution

To compute the stellar mass of each galaxy, we use the following equation from the Bell et. al. paper:

$$\log(\frac{M_*}{M_\odot})=0.4\Big(2.5(a\lambda+b\lambda\times\text{color})+5\log(d)-5+M_\odot-m_*\Big) \tag{2}$$

In this equation, we use $M_\odot=4.75$, $a_i=-0.222$, $b_i=0.864$, color$=g_{mag}-r_{mag}$, d=dist in pc,and $m_*=$total i_{mag}. We have all these values for each galaxy. Summing up all the total masses, we get that

$$\boxed{\text{combined stellar mass}=1.57\times10^{12}\ \text{M}_\odot}$$

Assuming that stellar mass is $\sim$15% of the total mass, we calculate the total combined mass to be:

$$\text{total combined mass}=\frac{\text{combined stellar mass}}{0.15}=\frac{1.57\times10^{12}}{0.15}$$

$$\boxed{\text{total combined mass}=1.04\times10^{13}\ \text{M}_\odot}$$

Now, using the total combined mass, we want to estimate the density of matter $\rho_{m,o}$ for our slice of the local universe. We calculate this value using the following equation:

$$\rho=\frac{\text{mass}}{\text{volume}} \tag{3}$$

We calculate our the volume for our wedge to be

$$\text{v}=(\text{width})(\pi)(\text{max distance}^2-\text{min distance}^2)(\frac{\text{angle}}{360})$$

$$\text{v}=(2.1)(\pi)(80^2-18^2)(\frac{90}{360})=1\times10^4\ \text{Mpc}^3.$$

Plugging our values into equation 3, we get our final answer:

$$\boxed{\rho_{m,0}=1.05\times10^9\ \frac{\text{M}_\odot}{\text{Mpc}^3}\ =\ 7.07\times10^{-29}\ \frac{\text{kg}}{\text{m}^3}}$$

Problem 1c

Using the above H_0, compute critical density of the universe, $\rho_{c,0}$. Compare to $\rho_{m.0}$. Based on the assumption that energy density of matter dominates that of radiation, what would you conclude of (i) the expansion law and (ii) the age of the universe? Explain.

Solution

To solve for the critical density, we use the following equation:

$$\rho_{c,0}=\frac{\epsilon_{c,0}}{c^2}=\frac{3H_0^2}{8\pi G} \tag{4}$$

We plug in $H_0=73$ km s^{-1} Mpc$^{-1}=2.37\times10^{-18}$ s^{-1} and G=6.67×10^{-11} m^3 kg^{-1} s^{-2}. We get a critical mass density:

$$\boxed{\rho_{c,0}=\ 1.001\times10^{-26}\ \frac{\text{kg}}{\text{m}^3}}$$

Comparing our values, $\Omega=\frac{\text{mass density}}{\text{critical density}}=7.06\times10^{-3}$. Our value for critical density is 3 orders of magnitude larger than the computed mass density of the local universe.

We get that, although the mass density is very small, it is not 0, so we are not in an empty universe. But, given our Ω, we are also do not live in a matter only universe, but are much closer to it that empty. Thus, we get that our expansion law is $a_m(t)=(\frac{t}{t_0})^{2/3}$ and our age of the universe is $t_0=\frac{2}{3}H_0^{-1}=2.81\times10^{17}$secs $=8.92\times10^9$ years. We know that this is a low limit since we are not accounting for galaxy clusters, which bring in a lot of mass and would increase our Ω value.

Problem 2a

Suppose dark energy is not related to the Cosmological Constant, but is associated instead with a phantom component that has an equation of state parameter $w=-0.5$.

> For a spatially flat universe in which this phantom dark energy component dominates over all other components, derive the expansion law $a(t)$ vs. t. Then plot $a(t)$ vs. $H_0(t-t_0)$ for this universe, along with the expansion laws appropriate for (i) an empty, $\kappa=-1$ universe and (ii) a flat, matter-dominated universe. (Be sensible about the range of values on the x-axis!)

Solution

The described universe is a flat, single component universe with a $\omega=-0.5$. Thus, we derive the following expansion law for this universe:

$$a(t)=(\frac{t}{t_0})^{\frac{2}{3(1+\omega)}}\rightarrow a(t)=(\frac{t}{t_0})^{\frac{2}{3(1-0.5)}}\rightarrow \boxed{a(t)=(\frac{t}{t_0})^{\frac{2}{3(0.5)}}} \tag{5}$$

where

$$t_0=(\frac{2}{3})(\frac{1}{1+\omega})H_0^{-1} \tag{6}$$

We plot for the universe above, but also the expansion law for a flat, matter dominant universe and a $\kappa=-1$, empty universe. These have the following expansion laws, respectively:

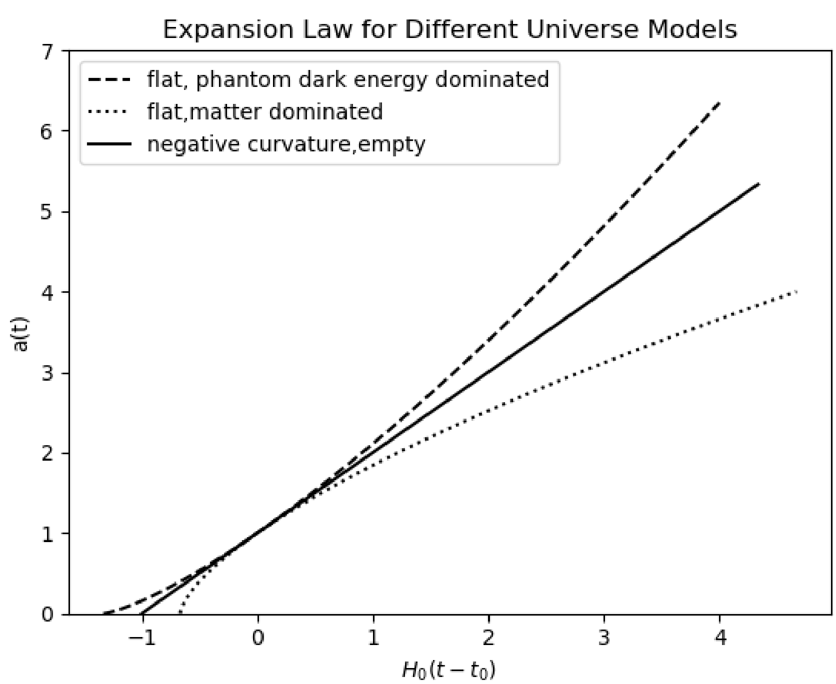
$$a_m(t)=(\frac{t}{t_0})^{2/3}$$

where $t_o=\frac{2}{3}H_0^{-1}$ and

$$a_{\text{empty}}=(\frac{t}{t_0})$$

where $t_0=H_0^{-1}$.

Plotting all expansion laws, we get the following plot:



Problem 2b

Assuming $H_0 = 73$ km s^{-1} Mpc^{-1}, compute the current age of this universe.

Solution

To compute the age of the flat, phantom dark energy dominated universe, we use equation 6, with $H_0 = 2.37 \times 10^{-18}$ s^{-1} and $\omega = -0.5$. Plugging in, we get that the age of this universe is

$$\boxed{t_0 = 5.64 \times 10^{17} \text{ secs} = 1.79 \times 10^{10} \text{ years}}$$

Problem 2c

What is the lookback time for a $z = 4.3$ galaxy of this universe?

Solution

To solve for lookback time, we use:

$$t_{\text{lookback}} = t_0 - t_e \tag{7}$$

We have t_0 from 2b. To solve for t_e we need another equation:

$$t_e = \frac{t_0}{(1+z)^{\frac{3}{2}(1+w)}} \tag{8}$$

Plugging in t_0 from 2b, $\omega = -0.5$, and $z = 4.3$ into equation 8 and then our result into equation 7, we get that the lookback time is

$$\boxed{t_{lookback} = 4.02 \times 10^{17} \text{ secs} = 1.28 \times 10^{10} \text{ years}}$$

Problem 3a

First, write a Python program to calculate the proper distance d_p to an object at an arbitrary redshift z based on the formula derived in class. The output should be given in terms of the Hubble distance $d_{\rm H} = c/H_0$, and you should assume $H_0 = 73$ km s^{-1} Mpc^{-1}, $\Omega_m = 0.27$, $\Omega_\Lambda = 0.73$, and $\Omega_0 = 1$. Then, compute the corresponding luminosity distance of the object (again, normalized by $d_{\rm H}$). Recall that $d_L = (1+z)\, d_p$.

Solution

Proper distance is given by the following equation:

$$\text{dp} = \text{dh} \int_0^z \frac{dz}{\Big(\Omega_{r,0} a^{-4} + \Omega_{m,0} a^{-3} + \Omega_{\Lambda,0} + (1-\Omega_0) a^{-2}\Big)} \tag{9}$$

and the luminosity distance equation is

$$d_L = (1+z) d_p \tag{10}$$

We use the values given in the question and integrate numerically using a Simpson integrator. Once we integrate, we plug in our dp and z values into equation 10 to obtain a luminosity distance value. Code for this question and other questions is attached at the end.

Problem 3b

Next, modify your code to compute d_L at small steps in redshift (i.e., $\Delta z \leq 0.01$) over the range $0 \leq z \leq 7$. Bear in mind that an integration over redshift must be performed at each redshift step in this range. Plot your results as $d_L/d_{\rm H}$ vs. z and compare your curve to that shown for the "Benchmark" model on one of the lecture slides — they should be very similar.

Solution

We use our code from 3a to integrate our proper distance function for $z:[0 \to 7]$ at very small intervals. We plot our results, which match the benchmark model.

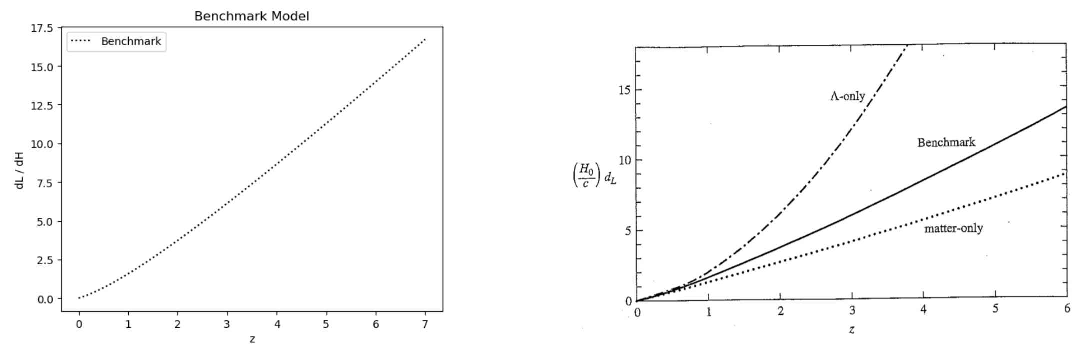


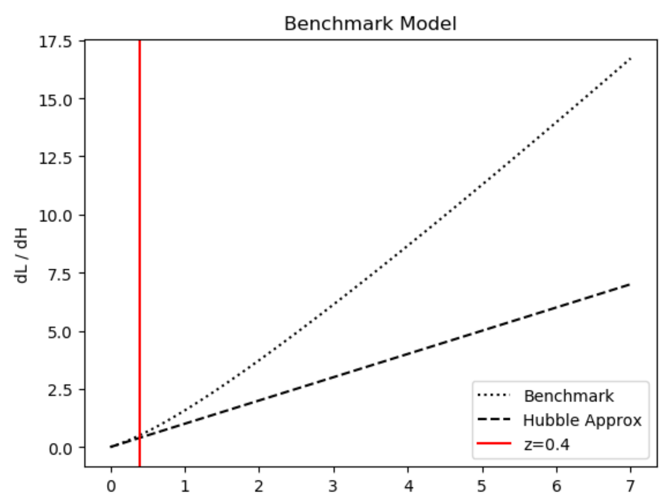
Figure 1: My benchmark plot (left) compared to the standard benchmark (right).

Problem 3c

> At low redshift, all types of distances have about the same value, i.e., the one given by the Hubble Law: $d = v_r/H_0 = cz/H_0$. Add to your plot a curve that represents distances calculated this way. At about what redshift does this distance begin to diverge significantly (e.g., by more than $\sim 10\%$) from the cosmologically accurate one? You might need to zoom in on the low-z portion of your plot to see...

Solution

We add the hubble law approximation $(d = \frac{cz}{H_0})$ to the plot and then find the z value for which the two functions begin to deviate:



We see that the two models deviate at very early redshift values.

Problem 4a

> As part of my doctoral thesis, I discovered what was for a brief time the most distant X-ray source known — a quasar at a redshift of 4.30. In the observed 0.16–3.5 keV band, the source has a flux of 6.4×10^{-13} erg cm^{-2} s^{-1}. A discovery such as this naturally leads to the following practical quesitons:

In the rest frame of the quasar, what X-ray energy band was observed?

Solution

To compute observed wavelength from the given energy, we use the following equation:

$$\lambda_{\text{obs}} = \frac{\text{ch} \times 10^9}{\text{energy}} \tag{11}$$

to get a lambda in nm, where c=2.998×10^8 m/s and h=4.135×10^{-15} eV s and energy is the given energies converted to eV. When we find the observed wavelength, we solve for the intrinsic lambda with the following:

$$\lambda_{\text{int}} = \frac{\lambda_{\text{obs}}}{(1+z)} \tag{12}$$

Plugging in 160 eV and 3500 eV, we get a range of wavelengths of $\boxed{(0.067\text{nm}, 1.46\text{nm})}$. Thus, our values tells us that it was observed in the hard x-ray band.

Problem 4b

What is the X-ray luminosity of the quasar in this band? (As in problem (3), you should assume $H_0 = 73$ km s^{-1} Mpc^{-1}, $\Omega_m = 0.27$, and $\Omega_\Lambda = 0.73$.)

Solution

The luminosity of an object is give by

$$L = 4\pi d^2 F \tag{13}$$

To get our distance (d) value, we integrate equation 9 using an upper bound of $z = 4.3$. This gives us a luminosity distance of $d_L = 3.857 \times 10^4$ Mpc $= 1.96 \times 10^{29}$ cm.

Plugging in the calculated d_L and $F = 6.4 \times 10^{-13}$ erg cm^{-2} s^{-1}, we get that the luminosity of the quasar is

$$\boxed{L = 1.15 \times 10^{47} \text{ erg / s}}$$

This is a luminosity that is very much in quasar range.

Problem 4c

The cosmic time at which the observed X-rays were emitted corresponds to what fraction of the universe's current age?

Solution

To compute the time when the light was emitted, we use the following function:

$$t = \frac{1}{H_0} \int_0^z \frac{dz}{\left(1+z\right)\left(\Omega_{m,0}(1+z)^3 + \Omega_{\Lambda,0}\right)^{1/2}} \tag{14}$$

Using our simpson integrator, we integrate equation 14 with an upper bound of $z = 4.3$. We get that the time of emission was

$$\boxed{t_e = 4.43 \times 10^{16} \text{ secs} = 1.4 \times 10^9 \text{ years}}$$

This is about about 10% the current age of the universe.

Code:

```
'''
ASTR232
HW7 - questions 1 and 2
date: 12-11-19
garcia, gil
'''



import pandas as pd
import matplotlib.pyplot as plt
import numpy as np


######################
######################
######## #1a #########
######################
######################


file = '40lss.txt'

#importing our file and adding headers to each row
columns = ['ra','z','g_mag','r_mag','i_mag']
df = pd.read_csv(file,delimiter=' ',index_col=False, header=None,names=columns)
print('file: ',file)
print('#1-----------------------------------------------')

    #calculating distance: d = z*c/h_o
c = 299792 #speed of light in km/s
h_o = 73 #hubble constant in km/s/Mpc
df['dist(Mpc)'] = (df['z'] * c) / h_o #calculating the distance in pc

#plotting the distribution of the galaxies in a polar plot
fig = plt.figure()
ax = fig.add_subplot(111, polar=True)
c = ax.scatter(df['ra']*(np.pi/180),df['dist(Mpc)'], s=10,color='k',alpha=0.75,label='Galaxy')
ax.set_theta_offset(np.pi/4)
ax.set_thetamin(0)
ax.set_thetamax(90)
plt.title('Distribution of Galaxies our Sample')
plt.xlabel('dist[Mpc]')
plt.legend(markerscale=0.4)
plt.close()

#some squares more filled in than others. distribution of galaxies is not isotropic & homogeneous.
#this goes against the cosmological principle.


######################
######################
######## #1b #########
######################
######################

#setting up all the components required for the Bell equation

df['g-r'] = df['g_mag'] - df['r_mag'] #color
```

```
df['dist(pc)'] = df['dist(Mpc)'] * 10**6 #converting Mpc to ps
M_o = 4.75 #abs mag of sun
a_i = -0.222 #coefficient from table 7, bell et al, 2013
b_i = 0.864 #coefficient from table 7, bell et al, 2013
# i tot is the apparent magnitude of the galaxy

#computing the stellar mass using the Bell equation
df['stellar_mass(Msun)']
    = 10**(0.4*( M_o - df['i_mag'] + (5*np.log10(df['dist(pc)'])) -5 + 2.5*(a_i+(b_i*df['g-r']))))
#computing total mass:
df['total_mass(Msun)'] = df['stellar_mass(Msun)'] / 0.15

#finding combined stellar mass of our wedge:
combined_stellar_mass = df['stellar_mass(Msun)'].sum()
print('combined stellar mass: \t', format(combined_stellar_mass,'E'))
#finding the combined total mass of our wedge:
total_combined_mass = df['total_mass(Msun)'].sum()
print('total combined (stellar+dark) mass: \t', format(total_combined_mass,'E'))
#calculating the volume of our wedge:
volume = 2.1*np.pi*(90/360)*(80**2 - 18**2)
print('volume(Mpc^3): \t',format(volume,'E') )
#finding the mass density of our wedge
mass_density = total_combined_mass / volume
print('mass density of local universe now (solar masses/Mpc^3): \t ',format(mass_density,'E'))
#converting the mass density to units of kg/m^3
mass_of_sun = 1.988e30 #kg
cubic_mpc_to_cubic_m = 2.938e67
mass_density_kg_per_m_cubed = mass_density * mass_of_sun / cubic_mpc_to_cubic_m
print('mass density of local universe (kg/m^3) \t', format(mass_density_kg_per_m_cubed,'E'))

######################
######################
######## #1c #########
######################
######################

#the critical density fxn
def critical_density(H):
    G = 6.67e-11 #m^3 kg^-1 s^-2
    #g = 4.3e-3 #pc solar mass (km/s)^2
    return (3. * H**2) / (8.*np.pi*G)

#converting H0 to units of 1 / secs
Ho_in_one_over_secs = 73 / 3.086e19
#plugging in Ho to find critical density of the universe
critical_density = critical_density(Ho_in_one_over_secs)
print('critical density (kg/m^3): \t ',format(critical_density,'E'))
#fraction of calculated mass density over critical_density
print('mass density / critical density:\t ', format(mass_density_kg_per_m_cubed / critical_density,'E') )


##########
#dif plot for 1a - size corresponds to relative size
##########

fig = plt.figure()
ax = fig.add_subplot(111, polar=True)
c = ax.scatter(df['ra']*(np.pi/180), df['dist(Mpc)'],
    color='k', s=100*df['total_mass(Msun)']/df['total_mass(Msun)'].max(), alpha=0.4,label='Galaxy')
ax.set_thetamin(0)
ax.set_thetamax(90)
```

```
ax.set_theta_offset(np.pi/4)
plt.title('Distribution of Galaxies for our Sample')
plt.xlabel('dist[Mpc]')
plt.legend(markerscale=0.4)
plt.show()

######################
######################
######## #2a #########
######################
######################

print('#2a----------------------------------------------')
#one over Ho is hubble time
hubble_time_sec = (1/(73/3.086e19))

'''
we are in a flat, single component (phantom dark energy) universe.so,
'''

# age of universe, eqn 1
def age_of_universe_flat_phantom_dark_energy(w,hubble_parameter=73):
    term1 = 1/(1+w)
    term2 = 1/(hubble_parameter / 3.086e19) #gives hubble time in secs
    return (2/3)*term1*term2 #returns to in seconds

w_phantom = -0.5

#calculating age of this universe
t_o_flat_phantom = age_of_universe_flat_phantom_dark_energy(w_phantom)
t_o_flat_phantom_yr = t_o_flat_phantom * 3.171e-8


#expansion law 1
def scale_factor_flat_phantom(t,w=w_phantom, t_o = t_o_flat_phantom):
    expo = 2/ (3*(1+w))
    return (t/t_o)**(expo)

#reporting age of this unvierse
print('t_o(secs) where universe is flat, phantom dark energy dominates:\t',format(t_o_flat_phantom,'E'))
print('t_o(yrs)
     where universe is flat, phantom dark energy dominates:\t',format(t_o_flat_phantom_yr,'E'))
print()

'''
in a flat, matter dominated universe:
'''

# age of universe, eqn 2
def age_of_universe_flat_matter(w,hubble_parameter=73):
    return (2/3) * 1/(hubble_parameter/3.086e19)

w_matter = 0

#calculating age of this universe:
t_o_flat_matter = age_of_universe_flat_matter(w_matter)
t_o_flat_matter_yr = t_o_flat_matter * 3.171e-8

#expansion law 2
def scale_factor_flat_matter(t,w=w_matter, t_o = t_o_flat_matter):
    return (t/t_o)**(2/3)
```

```python

#reporting age of this universe:
print('t_o(secs) where universe is flat, matter dominates:\t', format(t_o_flat_matter,'E'))
print('t_o(yrs) where universe is flat, matter dominates:\t', format(t_o_flat_matter_yr,'E'))
print()

'''
in an empty, k = -1 (negative curvature)
'''

#age of this universe, eqn 3
def age_of_universe_neg_curve_empty(hubble_parameter=73):
    return 1/(hubble_parameter/3.086e19)

#calculating age of this universe:
t_o_neg_curve_empty = age_of_universe_neg_curve_empty()
t_o_neg_curve_empty_yr = t_o_neg_curve_empty * 3.171e-8

#expansion law 3
def scale_factor_neg_curve_empty(t,t_o= t_o_neg_curve_empty):
    return t/t_o

#reporting age of this universe:
print('t_o(secs) where universe is neg. curvature, empty :\t',format(t_o_neg_curve_empty,'E'))
print('t_o(yrs) where universe is neg. curvature, empty :\t',format(t_o_neg_curve_empty_yr,'E'))


#plotting a(t) vs Ho(t-to) for each universe model
plt.title('Expansion Law for Different Universe Models')
t = np.linspace(-1,4*t_o_flat_phantom,100000) #the time values we want to plug in
plt.ylim(0,7)
#phantom universe, using expansion law 1
plt.plot((1/hubble_time_sec)*(t-t_o_flat_phantom), scale_factor_flat_phantom(t), color='k',linestyle='--', label='flat, phantom dark energy dominated' )
#flat, matter dominanted universe, using expansion law 2
plt.plot((1/hubble_time_sec)*(t-t_o_flat_matter), scale_factor_flat_matter(t), color='k',linestyle=':', label='flat,matter dominated' )
#negative curvature, empty universe, using expansion law 3
plt.plot((1/hubble_time_sec)*(t-t_o_neg_curve_empty), scale_factor_neg_curve_empty(t), color='k', label='negative curvature,empty' )
plt.legend()
plt.ylabel('a(t)')
plt.xlabel(r'$H_0 (t-t_0)$')
plt.show()

######################
######################
######## #2b #########
######################
######################
print('#2b---------------------------------------------')
#we calculated this in 2a, we report them again here:
print('t_o(secs) where universe is flat, phantom dark energy dominates:\t',format(t_o_flat_phantom,'E'))
print('t_o(yrs) where universe is flat, phantom dark energy dominates:\t',format(t_o_flat_phantom_yr,'E'))
print()


######################
######################
######## #2c #########
```

```
######################
######################
print('#2c----------------------------------------------')

#eqn for time emmited:
def time_emmited(z,w,t_o=t_o_flat_phantom):
    expon = 3/2 * (1+w)
    denom = (1+z)**expon
    return t_o / (denom)

#t_lookback = t_0 - t_e (using all values for phantom universe)
lookback = t_o_flat_phantom - time_emmited(4.3,-0.5)
#reporting our values:
print('lookback time (secs): \t', lookback)
print('lookback time (yrs):\t', format(lookback * 3.171e-8,'E'))
```

```
'''
ASTR232
HW7 - questions 3 and 4
date: 12-11-19
garcia, gil
'''

import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

#creating our simpson integrator
def simp_intergration(function,a,b,steps):
    h = (b-a)/steps
    x = np.linspace(a,b,steps+1)
    f_x = function(x)
    integral_evaluation = h/3 * np.sum(f_x[0:-1:2] + 4*f_x[1::2] + f_x[2::2])
    return integral_evaluation


######################
######################
######## #3a #########
######################
######################

#defining some constants
Ho = 73 #hubble const in km/s/Mpc
c = 3e5 # speed of light in km/s
dh = c/Ho
omega_r,omega_m,omega_lam,omega_0 = 0,0.27,0.73,1

#proper distance integral
def func(z):
    dh = c/Ho
    a = (1+z)**(-1)
    denom = omega_r *a**-4 + omega_m*a**-3 + omega_lam + (1-omega_0)*a**-2
    return dh/np.sqrt(denom)

# using simpson on our proper distance integral to get proper distance
def dp(z,steps=2000,func=func):
    a=0
    return  simp_intergration(func,a,z,steps)

```

```
#using proper distance to compute luminosity distance
def lum_dist(z,steps=2000,func=func):
    return (1+z) * dp(z,steps,func)


#testing integrator on x^2
def x_squared(x):
    return x*x

value = simp_intergration(x_squared,0,7,700)
#print('value',value)
# it works!!



######################
######################
######## #3b #########
######################
######################

print('#3b-------------------------------------------')

#creating empty lists to append our values, to be plotted
lum_dist_lst =[]
lum_dist_lst_hubble =[]
z_lst = np.linspace(0,7,5000) #range of z values to plug in

counter = 0
for z in z_lst: #calculating luminosity distance for 5,000 values between 0 and 7
    lum_dists = lum_dist(z) #lum dist calculation
    lum_dist_hubble = (c*z)/Ho #hubble approximation calculation
    lum_dist_lst += [lum_dists] #append lum dist to lum dist lst
    lum_dist_lst_hubble += [lum_dist_hubble] #append hubble approx to hubble approx lst
    ratio = (lum_dists - lum_dist_hubble) / lum_dist_hubble #finding ratio of distances
    if ratio > 0.1 and counter == 0: #prints out when it crosses 10% difference
        print('z at which they differ by more than 10%: ',z,'\t ratio: ',ratio)
        counter +=1

#converting lists to array - dividing by dh
lum_dist_array = np.array(lum_dist_lst)
lum_dist_array = lum_dist_array / dh


lum_dist_array_hubble = np.array(lum_dist_lst_hubble)
lum_dist_array_hubble = lum_dist_array_hubble / dh



######################
######################
######## #3c #########
######################
######################


#plotting the benchmark + hubble approx models
plt.title('Benchmark Model')
plt.plot(z_lst,lum_dist_array,c='k',linestyle=':',label='Benchmark')
plt.plot(z_lst,lum_dist_array_hubble,c='k',linestyle='--',label='Hubble Approx')
plt.axvline(0.13,color='red',label='z=0.13')
```

```
plt.xlabel('z')
plt.ylabel('dL / dH')
plt.legend()
plt.close()


######################
######################
######## #4a #########
######################
######################


print('#4a------------------------------------------')

#constants
energy = [0.16e3,3.5e3] #eV
h = 4.135e-15 #plancks constant in eV * s
c = 2.998e8 #speed of light in m/s

#fxn for converting energy to wavelength
def energy_to_wavelength(energy): #enrgy must be in eV. returns wavelength in nm
    h = 4.135e-15 #plancks constant in eV * s
    c = 2.998e8 #speed of light in m/s
    return (10**9*c*h) / energy

#fxn for converting observed wavelength to intrinsic (emmited) wavelength
def intrinsic_lambda(z_observed,lambda_observed):
    return lambda_observed / (1+z_observed)



#computing the intrinsic wavelength for our 2 energy values
for val in energy:
    wav = energy_to_wavelength(val)
    print('intrinsitc wavelength for energy: ',val,'is \t',intrinsic_lambda(4.3,wav))
# our values give us hard x-ray band


######################
######################
######## #4b #########
######################
######################
print('#4b------------------------------------------')

#luminisisty fxn
def luminosity(flux,distance): #distance in cm, flux in erg * cm^-2 * s^-1
    return 4*np.pi*distance**2*flux

#constants
Ho = 73 #hubble const in km/s/Mpc
c = 3e5 # speed of light in km/s
dh = c/Ho
flux = 6.4e-13 # erg cm^-2 s^-1
omega_r,omega_m,omega_lam,omega_0 = 0,0.27,0.73,1
z = 4.3

#calculating lum dist at a z of 4.3
lum_dist_4b = lum_dist(z,steps=2000,func=func)
print('z:',z,'\t lum dist(Mpc): \t',format(lum_dist_4b,'E'))
```

```
#converting to lum dist to cm
lum_dist_4b_cm = lum_dist_4b * 3.086e24
print('z:',z,'\t lum dist(cm): \t',lum_dist_4b_cm)

#computing luminosity of our object
print('luminosity(erg/s):\t',luminosity(flux,lum_dist_4b_cm))

######################
######################
######## #4c #########
######################
######################

print('#4c-----------------------------------------')

#integral fxn for finding time since light was emmited
def func_4c(z):
    Ho = 73/3.086e19
    denom = (1+z)*np.sqrt(omega_m*(1+z)**3 + omega_lam)
    return (1/Ho)/(denom)

#integrating fxn to find time for our object
#in theory, we would use an upper bound of infinity but we cant do that
#we use an upper bound of z = 20,000
time_emit = simp_intergration(func_4c,4.3,20000,200000)
time_emit_yr = time_emit * 3.171e-8
print('time emit(s):\t',format(time_emit,'E'))
print('time emit(yr):\t',format(time_emit_yr,'E'))
#comparing our answer to hubble time
hubble_time = 1/(2.37e-18)
print('hubble time: \t',hubble_time)
print('fraction of the universe current age: \t',time_emit/hubble_time)
```