The determination of the Density parameters

Manu Upadhyay

October 10, 2021

Abstract

In this project, I have obtained the density parameters for mass and Λ assuming a Benchmark comsological model which assumes a universe governed roughly by the FLRW metric at large scales where the curvature is generally non-existent $\Omega_k = 1$ throught numerical methods for theoretical curve fitting over the observational data from the *Pantheon library*

1 Theoretical Background

The theoretical background is explained bfore proceeding to the actual project for both of the parts in relevant details.

1.1 FLRW Cosmology and Luminosity Distances

My main assumption in this project is the the fact that the universe in which we live in is, (i) described very well by the so called *Big Bang Theory* as opposed to the supposed *Steady State Theory* in which the universe has been in existence since eternity without any arbitrary starting point and (ii) our universe is very well described by the fact that on very large scale (300 Mpc) [3] it is *Isotropic and Homogeneous* at every point in space. The latter implying that the metric that governs the space is the Robertson-Walker metric, written as:

$$\mathrm{d}s^2 = c^2 dt^2 - \frac{S(t)^2 \mathrm{d}r^2}{1 - kr^2} - r^2 S(t)^2 (\mathrm{d}\theta^2 + \sin^2\theta \, \mathrm{d}\phi^2) \tag{1}$$

in the system of spherical polar coordinates with the metric sign convention as $(+,-,-,-)$.

Now ofcourse, the solution of the Einstein Field equations in the presence of this metric would give rise to the two independent *Friedmann Equations* (2 instead of 4 since, 3 are identical in the case of perfect fluid):

$$\left(\frac{\dot{S}}{S}\right)^2 + \frac{kc^2}{S^2} - \frac{\Lambda c^2}{3} = \frac{8\pi G}{3}\rho \tag{2}$$

$$2\frac{\ddot{S}}{S} + \left(\frac{\dot{S}}{S}\right)^2 + \frac{kc^2}{S^2} - \Lambda c^2 = -\frac{8\pi G}{c^2}p. \tag{3}$$

where the einstein eqaution is:

$$G_{\mu\nu} + \Lambda g_{\mu\nu} = \frac{8\pi G}{c^4}T_{\mu\nu} \tag{4}$$

From these relations and taking in into account the apparent time at which light from these galaxies reach us, we calculate the expression for the Luminosity distance as function of redshift for a given cosmolgy.

For instance, we know that:

$$d_L = \sqrt{\frac{L}{4\pi F}}$$

where, we are considering the spatial space that we usually see around us

$$d_L = f(r)(1+z)$$

for a general case of a curved space-time where f(r) is the curvature.

These equation comes from the fact that, we are assuming a FLRW Cosmology and hence the generally Euclidean metric doesn't suffices to get any relations in these cases!

The $(1+z)$ factors comes from the fact that, $S(t)$ can be written as a function of z as well, in which the relation between a *type* of universe's radiation that was left by the Galaxy in consideration at time t_0 reaches us (in the present universe where the Scale factor and things under go change w.r.t. to the redshift of the Galaxy) is given by $E_{obs} = E_0/(1+z)$ and this is implied due to the scale factor following the relation $S(z) = 1/1+z$

After further calculations using the mteric and above relations between the various varibales in play, we get the expression for the Luminosity distance (d_L) as:

$$d_L(z) = \frac{c}{H_0}\int_0^z \frac{dz'}{E(z')} \tag{5}$$

where, H_0 is the Hubble's Paramter in the present day &

$$ \tag{6} $$

$$E(z) = \frac{H(z)}{H_0} = \sqrt{\Omega_r(1+z)^4 + \Omega_m(1+z)^3 + \Omega_k(1+z)^2 + \Omega_\Lambda} \tag{7}$$

where, Ω_m, Ω_k, Ω_r and Ω_{Lambda} are the different Density parameters related to the mass, curvature, radiation and dark energy cimonents in the universe.

$$ \tag{8} $$

For a Benchmark model of the FLRW cosmology, we will shortlist these zoo of variables to only 2 independent parameters (as also invertigated in this project) and solve the simplfied intgral (numerically) to obtan for the parameters' values. Hence, $\Omega_r = 0$ and $\Omega_k = 1$ (for a flat-universe). The integral simplifies to:

$$d_L = \frac{c}{H_0} \int_0^z \frac{z+1}{\sqrt{\Omega_m[(1+z)^3 - 1] + 1}} \tag{9}$$

where, we know that.

$$\Omega_k = \Omega_m + \Omega_\Lambda \Longrightarrow \Omega_m + \Omega_\Lambda = 1 \tag{10}$$

The above exolanation encompassses the reasons for the obtaining the luminosity distances in the FLRW universe.

1.2 Observational Data and Luminosity distances

The distances obtained from the very bright objects in the galaxies are the observed luminosty distances that we can obtain through space surveys. These daat have to analysed to obtain the required plots that gives the required understanding of the various universal parameters available to us.

We know that the distance modulus is related to the absolute and apparent magnitude of the observed object and that relation goes as:

$$m_{app.mag.} - M_{abs.mag.} = 5logd_L - 5 \tag{11}$$

where d_L is the Luminosity distance of the objects from the observer moving at redshift z.

Hence, we get the expression for the d_L in terms of the distance modulus $m - M$ and hence:

$$d_L = 10^{\frac{m-M+5}{5}} \tag{12}$$

This way we can make use of the observed data nd compare the observed values for the Luminosity distance to that of the theoretical ones and describe teh cosmolgy of the observable universe!

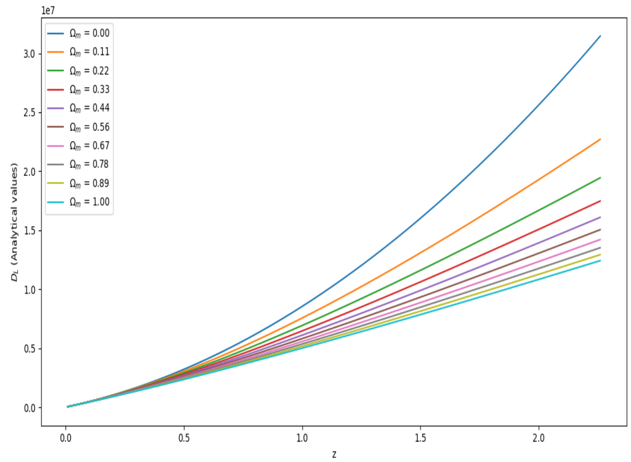

Figure 1: Ideal plots that can be obtained for the Benchmark model's Luminosity distance (d_L) considering various values of the parameter Ω_m (the independent parameter). In this program, I had chosen the H_0 value as $70 \pm 7 kms^{-1} Mpc^{-1}$ for simplicity.

2 Statement of the Problem

To solve for the density parameters of the universe assuming a FLRW cosmology by taking the data of SN1a distance modulus obtained from the *Pantheon* data [1] and further fit the data to linear and non-linear curves using least sqaure regression from the *SciPy* library's `curve_fit` function and compare the values to the standard cosmological data that is obtained and analyzed in the WMAP program for verifcation.

3 Details of the methods of study

The SN1a are generally used as the Distance measurers of the parts of the universe which are usually very *dim*. Since the absolute luminosity of a SN1a is standardised [5], we can obtain the Luminosity distance-redshift data and extract the observational data out of it.

I have used the *Pantheon* [1] to obtain the Distance-Modulus($m - M$) relation vs the redshift(z) relation of SN1a and then further obtain the denisty parameters of matter(Ω_m) and the radiation(Ω_Λ) taking into account that $\Omega_r = 0$ (considering a Minimalistic 6 paramter model [2]) by fitting the obtained curve to a theoretical curve:

- loaded by the *astropy* modeule of the ΛCDM model (where the data is taken as the estimate from the Luminosity distance-redshift relation ($D_L - z$)) 12
- fitted on the *SciPy* module using the package of the `curve_fit` to find the unknown paramters (H_0 and Ω_m)

3.1 Opening and formating the Pantheon csv files

The Pantheon data is available on [1], which is in the form of '.txt' form and hence can be converted o *Pandas Dataframe* and the corressponding columns can be modified edited and worked with in the form of 'np.arrays'. These process is common in all of the porgrams following!

```
file_pth = 'save.txt'

df = pd.read_csv(file_pth, sep='\s{1,}')
df.columns = [ 'Name of the SN', 'z_cmb', 'z_hel', 'dz', 'm', 'dm' , 'x1' , 'dx1


df['M'] = pd.Series([-19.6 for x in range(len(df.index))]) ########Absolute magr
df['dM'] = pd.Series([0.5 for x in range(len(df.index))]) ########Absolute magni
df['5'] = pd.Series([5 for x in range(len(df.index))]) ########Absolute magnitud
df['10'] = pd.Series([10 for x in range(len(df.index))])
M = df['M']
z_hel = df['z_hel']
m = df['m']
df['m-M'] = df['m'] - df['M']
DM = df['m-M']

df['factor'] =( DM + df['5'] )/df['5']
g = df['factor']
dM = df['dM']
df['d_z'] = df['10'] ** g
d_z = df['d_z']

import matplotlib.pyplot as plt
plt.scatter(z_hel, d_z, label ='fn of d(z)')

c = 10**(df['dm'] +dM)

plt.errorbar(z_hel,d_z, yerr=c , fmt="o")
plt.title('$d_L vs. z relation')
####Plot a scatter plot###########
plt.ylabel('$d_L(z)$')
plt.xlabel('z')
plt.ylim(-2*10**10,2*10**10)
```

```
plt.legend()
#plt.grid()
#plt.show()
```

3.2 SciPy curve_fit

Fitting a curve (or in this case a linear curve - straight line) comes with the challenge of choosing the optimal algorithm so that the fitted curve demonstrates the most accurate trend of the obtained data. The data's *noise* can be eradcated byt choosing appropriate methods but with a loss of the actual data that has been obsereved. [[4]].

This package does the Linear regression of data following the algorithm of the Linear least squares method wehre the parameter of χ^2 is minimised and the curve is fitted on the *observational* data.

Code:

```
pth = 'made.csv'


mn= genfromtxt(pth, delimiter=',')
print(np.shape(mn))
x = mn[1:,2]
y = mn[1:,24]

#plt.scatter(x, y)
m = (mn[1:,5])*(mn[1:,24])

plt.errorbar(x,y, yerr=m, fmt="o", label = "Original Observation data")
'''plt.grid()
plt.xlabel('z')
plt.ylabel('$d_L$')
plt.title('Graph of $d_L$ vs. z')
plt.show()'''

def linearfunction(x,a,b):
      return a*x + b
constants = cf(linearfunction,x,y)
a_fit = constants[0][0]
b_fit = constants[0][1]

fit = []
for i in x:
  fit.append(linearfunction(i,a_fit,b_fit))
```

```
plt.plot(x,fit ,c='k',label="Fit")
plt.legend()
plt.xlabel("z")
plt.ylabel("$d_L$")
plt.title("Fitting the data using SciPy")
plt.show()
print(constants)
```

The graphical output of this file is included int he section 4 of this report. The numerical output in the forms of the straight line parameters was given out in the form of the array after the execution of the program.

$$\begin{pmatrix} 7.61761377 \times 10^{+09} & -3.72784249 \times 10^{+08} \end{pmatrix}$$

and

$$Covariance = \begin{pmatrix} 1.10302503 \times 10^{+15} & -3.55742525 \times 10^{+14} \\ -3.55742525 \times 10^{+14} & 2.04924840 \times 10^{+14} \end{pmatrix}$$

These values of the parameter of the slope m and the y-intercept is obtained from the fitting as the first sarray elements along with the error in the values as given by the square root of the diagonal values of the Covariance matrix (2nd matrix). Hence,

$$d_L = \frac{cz}{H_0}(1+z)(1 - \frac{1+q_0z}{2}) \implies \frac{cz}{H_0}(1 + \frac{1-q_0z}{2})$$

where q_0 is the parameter which is dependent upon the cosmology (in this case $q_0 = -0.55$ [5])

Since, we know that,

$$q_0 = \frac{\Omega_m}{2} - \Omega_{Lambda} \implies q_0 = \frac{3}{2}\Omega_m - 1 \tag{13}$$

Equating 13 to the equation of a line and uing the parameters obtained after the curve fitting, we determine the Density parameters and the Hubble's constant along with the error.

3.3 Failure of linear model and reformation

Since the equation of the theoretical model of the d_L is of second order instead of a linear function , we try to fit the observed data, using the same method, with a curve of the type $ax^2 + b$ and directly equate and obtain the parameters q_0 and H_0.

$$\frac{cz}{H_0} + \frac{c}{H_0}(\frac{1-q_0}{2})z^2 = ax^2 + b \tag{14}$$

Now, we can directly see that from this type of eqaution we can directly obtain the values of the parameters unknown to us!

Code:

```
def SQfunction(x,a,b):
        return a*x**2+b*x
constants = cf(SQfunction,x,y)
a_fit = constants[0][0]
b_fit = constants[0][1]

fit = []
for i in x:
   fit.append(SQfunction(i, a_fit, b_fit))



plt.plot(x,fit,c='k',label="Fit")
plt.legend()
plt.xlabel("z")
plt.ylabel("$d_L$")
plt.title("Fitting the data using SciPy for $ax^2 + b$")
plt.show()
print(constants)
```

$$a\&b = \begin{pmatrix} 1.55130661 \times 10^{09} & 5.68359084 \times 10^{09} \end{pmatrix}$$

$$Covariance_{NL} = \begin{pmatrix} 1.42561153 \times 10^{15} & -1.18261218 \times 10^{15} \\ -1.18261218 \times 10^{15} & 1.28710835 \times 10^{15} \end{pmatrix}$$

3.4 Astropy LambdaCDM module

Using the Astropy's `LambdaCDM` module we can construct the $d_L - z$ graph by taking the standard values that shall be obtained by the curve fitting process of SciPy curve_fit or any other similar methods.

4 Results and Discussions

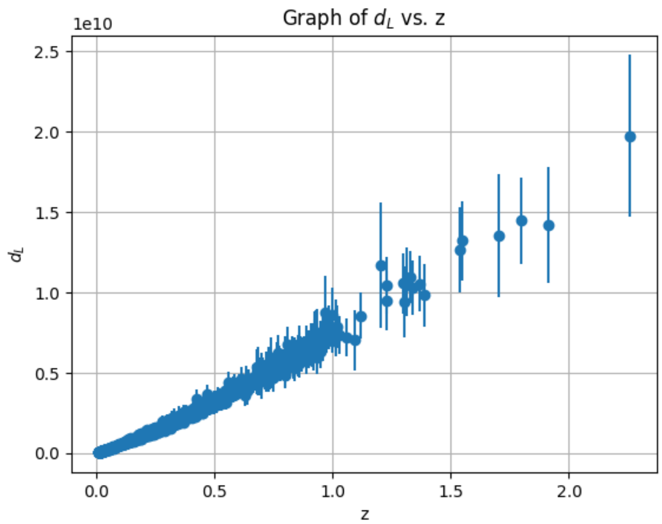

Figure 2: The plot obtained from the Pantheon data [1] using the program written in the section 3.1. The error bars in the d_L are plotted to get the optimal fit for the curve obtained and this graph is further analysed for the further processing until the required data is obtained from it.

Through this plot we applied the method of least square linear regression and thus obtained the following plot after the fitting of the parameters.

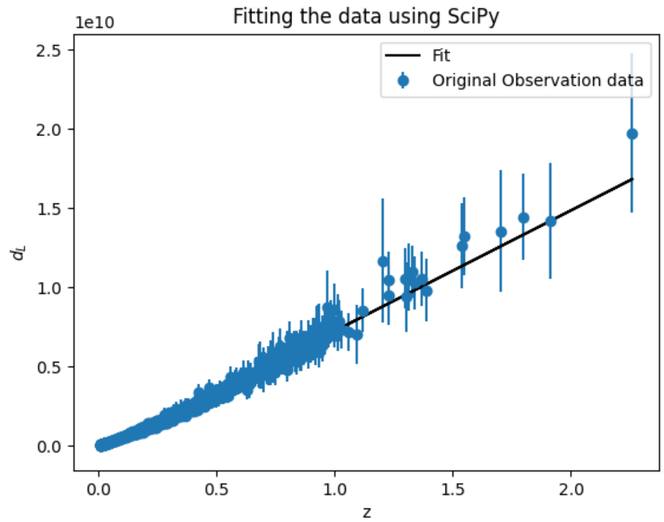

Figure 3: We see that the linear fit doesn't cover the graph well and in our discussion of the polynomial used to fit this observed curve we saw that instead of a linear curve, we shall need a 2nd order curve to fit this and obtain the parameters 14

The second order curve is of the form $ax^2 + bx$ wit two unkown parameters to find:

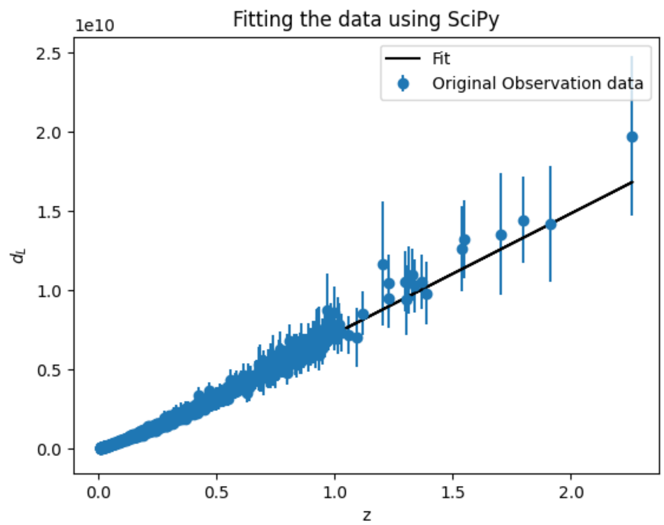

Figure 4: The curve follows a second order relation and it, also doesn't fit that well but is a better fit than the linear fit one.

From this second order curve we can deduce the H_0 and q_0 parameters easily.

$$\frac{cz}{H_0} + \frac{c}{H_0}(\frac{1-q_0}{2})z^2 = az^2 + bz \implies \frac{c}{H_0} = a, \frac{c}{H_0}\frac{1-q_0}{2} = b \tag{15}$$

After doing these calculations we obtain the value of the three dependent and two independent parameters as:

$$H_0 = 52.78 \tag{16}$$

from the covariance matrixc, we can calculate the error in this distribution

$$\tag{17}$$

$$q_o = -0.462 \implies \Omega_m = 0.37 \implies \Omega_\Lambda \approx 0.7 \tag{18}$$

5 Relevance and Limitations

Although we were successful in gauging the parameters through the linear least sqaure fitting method, in the ned the parameters weren't perfectly amnmping tho the WMAP data that comments about the cosmology of the universe after taking into account the other parametrs as well including the 2-03 that we worked with over here and with very high relativce precision. [7]

Using the data obtained from this project when we impose the theoretical graph on the obtained one , the results aren't so accurate is what we observe.

Code:

```
def func_01(z, H0, Om): #Defining the function
   #Dl =  (300000/H0)*(1+z)*((1/(Om*(((1+z)**3)-1)+1)))**0.5
   #Dl =  (1/H0)*((1/(Om*(((1+z)**3)-1)+1)))**0.5
   Dl=(1/H0)*((1+z)/((Om*((1+z)**3-1))+1)**0.5)
   return Dl

a = 0 #Entering the limits of integration dl
b = 2
n = 1000000 #No. of iterations
func_02 = 0
dz = (b-a)/n
res = np.zeros(n)
z = np.linspace(a, b , n) #Vectorisation

for i in range(n):
    func_02 = func_02 + (dz*func_01(z[i]+dz/2, 70./10**13*3.086, 0.3))
    res[i] = func_02


plt.plot(z, res/10, label = "dispalcement")
plt.xlabel("time")
plt.ylabel("y")
plt.legend()
plt.show()
```

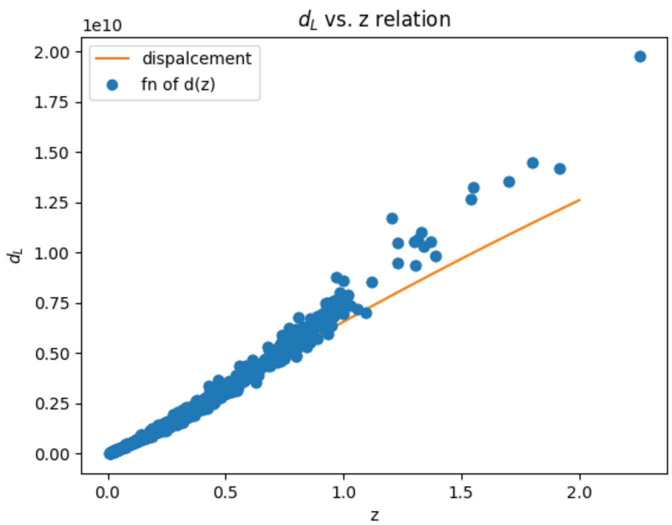

Figure 5: As we can see the fit is perfect for the vlaues for which the graph is plotted even though the resutls don't match with the analytical results of [7], it fits good to the observed data.

References

[1] `https://github.com/dscolnic/Pantheon`

[2] `https://en.wikipedia.org/wiki/Lambda-CDM_model#Cosmic%20expansion%20history`

[3] Extragalactic astronomy and Cosmology - Peter Schneider

[4] `https://arxiv.org/pdf/1008.4686.pdf`

[5] `https://arxiv.org/pdf/0908.4277.pdf`

[6] *Introduction to Cosmology* - Barbara Ryden

[7] `https://arxiv.org/pdf/1912.11879.pdf`