Quantitative Analysis for Dark Matter Density

Introduction Summary:

Dark matter is a mysterious form of matter which holds galaxies together. It is unobservable as it is non-interactive with electromagnetic radiation. Thus researching on dark matter is essential to know more on the unexplored part of science. Studying on dark matter helps us to understand the expansion of the universe and the formation of galaxies. In this study we will examine dark matter using the observed radial velocity and acceleration of a star. There are many methods to detect dark matter density, here we would used RV method used to detect presence of exoplanets to find DM density. Here the task is to determine local DM density by studying a star.

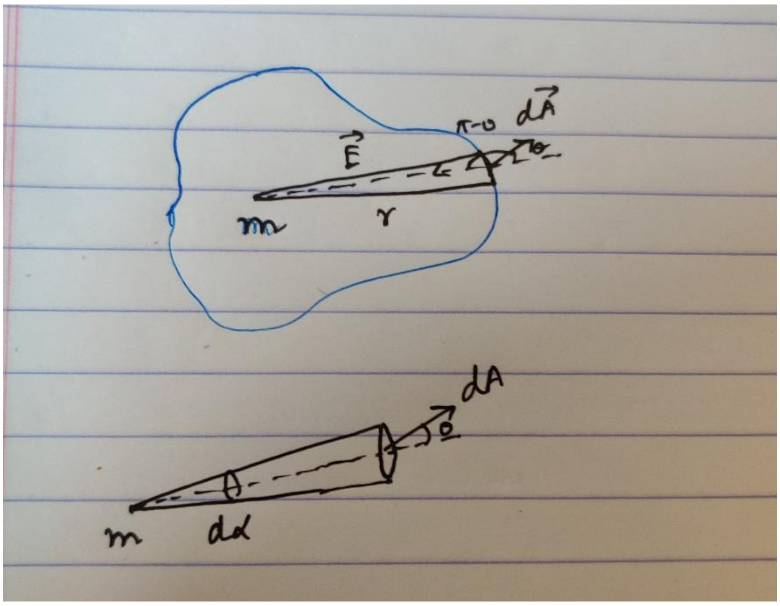
1. **The Gauss law of gravitation** states that the net gravitational flux passing through a closed surface is equal to -$4\pi G$ times the mass inside the gaussian surface.

$$\oint \vec{\boldsymbol{E}} \cdot \boldsymbol{d}\vec{\boldsymbol{A}} = -\boldsymbol{4\Pi G M_{inside}}$$

$\vec{\mathrm{E}} =$ Gravitational field due to all masses inside or outside the surface

M = Mass inside the surface

Derivation:



$$\oint \vec{E} \cdot d\vec{A} = \oint \vec{E}\, d\vec{A} \cos(\pi - \theta)$$

$$= -\oint \vec{E}\, d\vec{A} \cos\theta$$

Here E = $\frac{Gm}{r^2}$,

$$= -\oint \frac{Gm}{r^2} d\vec{A} \cos\theta$$

$$= -Gm \oint \frac{d\vec{A}\cos\theta}{r^2}$$

Here $\frac{d\vec{A}\cos\theta}{r^2} = d\alpha$,

$$= -Gm \oint d\alpha$$

Where $\oint d\alpha = 4\pi$ *since it is a closed surface integral*

$$\boxed{\oint \vec{E} \cdot d\vec{A} = -4\pi mG}$$

2. Wkt, A – B = $\frac{v_o}{r_0}$ where v_0 is circular velocity, by considering the equation, here $R_0 = r_0$

$$\frac{m{v_0}^2}{r_0} = \frac{GMm}{{r_0}^2}$$

$${v_0}^2 = \frac{GM}{r_0}$$

Now $(A - B)^2 = \frac{{v_0}^2}{{r_0}^2}$,

$$\boldsymbol{(A - B)^2 = \frac{GM}{{r_0}^3}} \qquad \textbf{1}$$

Also, Dark matter density,

$\rho_{dm} = \frac{M\prime}{V}$ where V is the volume.

$$\rho_{dm} = \frac{M\prime}{4\pi r^2}$$

$$\boldsymbol{M' = 4\pi {r_0}^2 \rho_{dm}} \qquad \textbf{2}$$

Similarly, the contribution to acceleration $a_r(r)$ is,

$$a_r(r) = \frac{-GM}{{r_0}^2}$$

Here M is also a function of r, hence,

$$\frac{\partial a_r}{\partial r} = 2G\frac{M(r)}{{r_0}^3} - G\frac{M'}{{r_0}^2}$$

Substituting equation 1 and 2,

$$\frac{\partial a_r}{\partial r} = 2(A-B)^2 - 4\pi G\rho_{dm}$$

By rearranging we get,

$$\rho_{dm} = \frac{1}{4(\pi)G}\left[2(A-B)^2 - \frac{\partial a_r}{\partial r}\right]$$

Significance of $(\boldsymbol{A}-\boldsymbol{B})$ term: Oort constants A, B are local description of differential rotation.

- The above term is nothing but the angular speed of Local standard of Rest (sun's motion).
- The Oort constants are used to calibrate the galactic rotation curve, which is one of their most important applications. Although examining the motions of gas clouds in the Milky Way can yield a relative curve, knowing V_0 is required to calibrate the true absolute speeds involved.
- As a result, the Oort constants can provide information about the mass density at a particular radius of the disc.

They're also useful for constraining Galaxy mass distribution models.

3. Consider our MW to be a thin disk then the contribution from the disk would be,

$$M'_{disk} = 2\pi r \, \epsilon_{disk}$$

Where $\epsilon_{disk} = The\ surface\ density\ of\ the\ disk$.

This disk contributes to ρ_{dm} and let the contribution be (δ_{new_dm}) given by,

$$\delta_{new\_dm} = \frac{\epsilon_{disk}}{2r}.$$

From [1] it says ϵ_{disk} value is 50 times solar mass/pc^2 and sun is approximately 8kpc away from GC.

Hence,

$$\delta_{new\_dm} = \frac{50 * Mass\ of\ sun}{2 * 8} = 3.1 * \ 10^{-3} \ M_{\emptyset}/pc^3$$

But, according to [2] the dark matter density is just 0.3 GeV/cm or 7.7 * 10^{-25} g/cm^3 or 0.011362 $M_{\emptyset}/pc^2$.

Hence as we can see the error is $\frac{\delta_{new\_dm}}{\rho_{dm}}$ is approx. 0.3 (ie) there is an error of 30%.

$$(Here\ M_{\emptyset}\ is\ Mass\ of\ Sun\ )$$

4. The hurdles are:

a. Extremely high precision instruments are required to study the change in velocities of stars.
   i. Future experiments must be focused in creating instruments that can record precisely the values.

b. There is wide chance of error from neighbouring stars, exoplanets and other stellar noise from background that would affect the measurements.
   i. Either by improving the instrument or by implementing goof data analytics the noise can be removed. As far as the noise from exoplanets are concerned the data obtain from other instruments can be used to train the model to detect the presence of the noise in the data.

c. It is difficult to obtain precise measurements and require other complex analysis to obtain the corrected data.
   i. Trying out newer algorithms for data analysis is required and improving techniques to clean the data is very much required in order to get good results.

d. In this method there is an assumption that the dynamics has attained steady state (ie) has attained equilibrium but it may not be the case and may lead to wrong measurements.

e. The motion of the star in the plane of the sky might also contribute to the observed velocities of the stars which results in not very reliable data.
   i. This can be prevented by selecting stars with velocities approx. 55km/s or below from very good higher precision instruments.

5. Reading Data:

```
In [1764]: df = pd.read_csv('vel_data.csv')
```

```
In [1875]: df
```

Out[1875]:

	Time (years)	Observed Velocity (cm/s)
0	0.000	6.608006
1	0.001	6.026303
2	0.002	5.567924
3	0.003	5.034716
4	0.004	4.612539
...	...	...
9995	9.996	7.408192
9996	9.997	6.842826
9997	9.998	6.599023
9998	9.999	6.730091
9999	10.000	6.122150

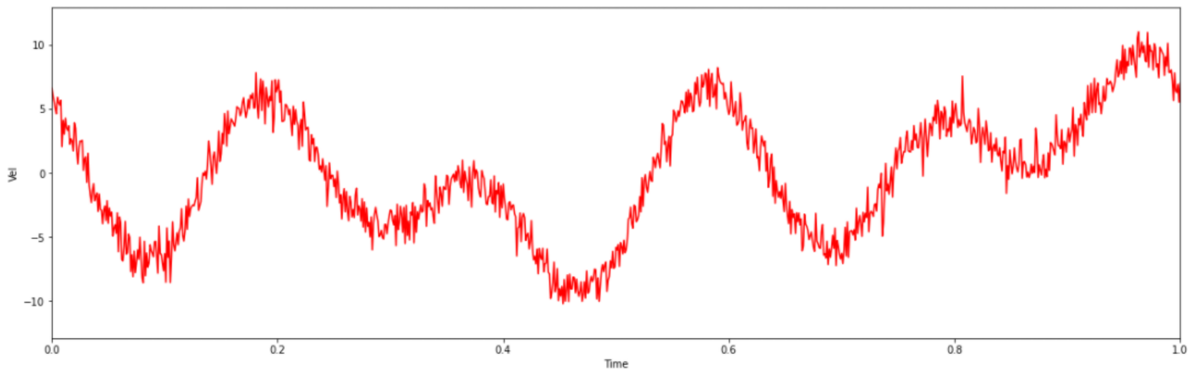
10000 rows × 2 columns

Plotting the data:

```
In [1877]: X = df['Time (years)']
           Y = df[' Observed Velocity (cm/s)']
```

```
In [1916]: def plot_it(X,Y):
               plt.figure(figsize=(20,6))
               plt.plot(X,Y,color = 'red')
               plt.xlabel('Time')
               plt.ylabel('Vel')
               plt.xlim(0,1)
           # plt.xlim(0,4)
```

```
In [1917]: plot_it(X,Y)
```



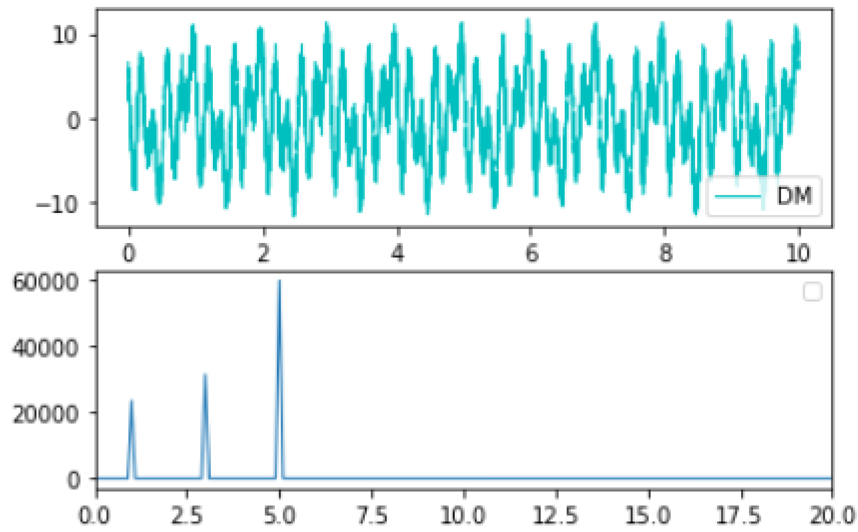
Analysing the data in frequency domain:

Denoise - FFT

In [1907]:

```
n = 10000
delta = 0.001
fhat = np.fft.fft(Y,n)
PSD = fhat * np.conj(fhat) / n
freq = (1/(delta*n)) * np.arange(n)
L = np.arange(1, np.floor(n/2), dtype=np.int32)

fig, axs = plt.subplots(2,1)
plt.sca(axs[0])
plt.plot(X,Y,color='c', linewidth=1, label='DM')
plt.legend()

plt.sca(axs[1])
plt.plot(freq[L],PSD[L], linewidth=1)
plt.xlim(0,20)
plt.legend()
```

```
<matplotlib.legend.Legend at 0x1f78c6d5e50>
```

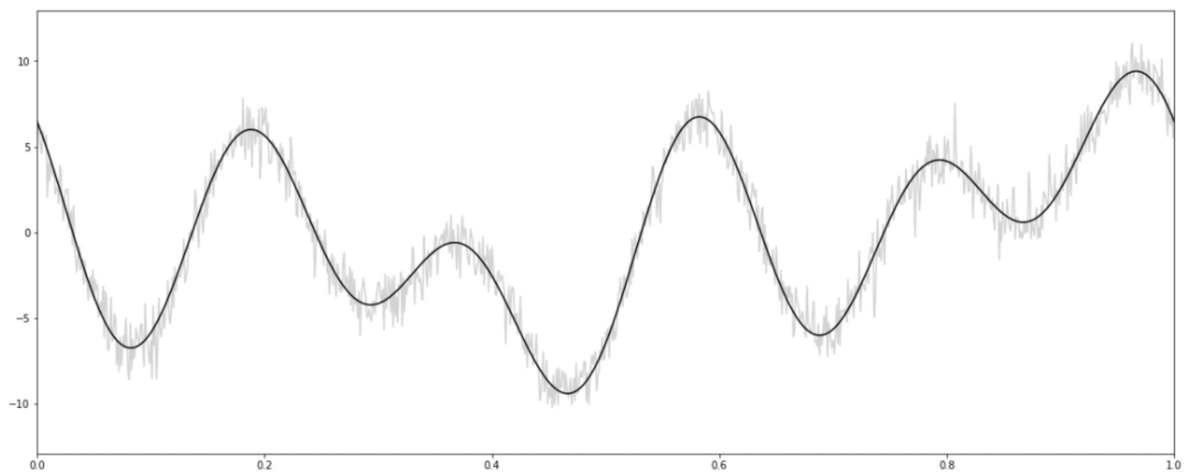


Denoising the data and plotting:

In [1911]:
```
psd_real=[]
def denoise(arr):
    n = 10000
    fhat = np.fft.fft(arr, n) #computes the fft
    psd = fhat * np.conj(fhat)/n
    freq = (1/(delta*n)) * np.arange(n) #frequency array
    idxs_half = np.arange(1, np.floor(n/2), dtype=np.int32) #first half index
    psd_real = np.abs(psd[idxs_half]) #amplitude for first half


    ## Filter out noise
    sort_psd = np.sort(psd_real)[::-1]
    # print(len(sort_psd))
    threshold = 15000
    psd_idxs = psd > threshold #array of 0 and 1
    psd_clean = psd * psd_idxs #zero out all the unnecessary powers
    fhat_clean = psd_idxs * fhat #used to retrieve the signal

    signal_filtered = np.fft.ifft(fhat_clean) #inverse fourier transform
    return signal_filtered
```

In [1912]:
```
y_newest = denoise(Y)
```

In [1918]:
```
plt.figure(figsize=(20,8))

plt.plot(X,Y,color='lightgray')
#pd.Series(y_newest).plot(color='black', ax=ax, figsize=(20, 8))
plt.plot(X,y_newest,color="black")
plt.xlim(0,1)
plt.show()
```



Now the obtained data is free of noise as it was removed using fast Fourier transform module.

Finding the radial velocity function w.r.t time:

The idea is to slice the given data into 2-year interval and do a linear regression on the sliced data. The obtained coefficients and intercepts are stored in an array and mean value is calculated to obtain the final line that fits the data. Since the function must be linear, linear regression was adopted to obtain the parameters of the line.

```
i=0
j=0
```

```
lr.fit(np.array(X[0:2000]).reshape(-1,1), np.real(y_newest)[0:2000])
```

```
LinearRegression()
```

```
plt.figure(figsize=(20,8))
# plt.plot(np.array(X).reshape(-1,1), lr.predict(np.array(X).reshape(-1,1)), color = "green")
# plt.plot(X,Y,color='lightgray')
#pd.Series(y_newest).plot(color='black', ax=ax, figsize=(20, 8))
plt.plot(X[:2000],y_newest[:2000],color="black")
plt.plot(np.array(X[0:2000]).reshape(-1,1) , lr.predict(np.array(X[:2000]).reshape(-1,1)), color = "green")
plt.title("Salary vs Experience (Training set)")
plt.xlabel("Time")
plt.ylabel("Velocity")
plt.show()
```

[One of the graphs that was obtained]

The final function obtained is:

```
In [2032]: coeffs=[]
           intercepts=[]
           for _ in range(5):
               j+=2000
               lr.fit(np.array(X[i:j]).reshape(-1,1), np.real(y_newest)[i:j])
               i+=2000
               coeffs.append(lr.coef_)
               intercepts.append(lr.intercept_)

In [2034]: np.mean(coeffs)

Out[2034]: 1.5057426767186555

In [2035]: np.mean(intercepts)

Out[2035]: -7.528713383593276
```

Here we can see that the coefficient is 1.51 and intercept is -7.53. Hence the radial velocity w.r.t time is,

$$v_{radial} = 1.51 * t - 7.53$$

We have finally **Filtered the periodic signal from noise using FFT** and **have obtained a linear function that relates radial velocity due to DM and GC and time using Linear Regression.**

From the obtained function radial acceleration can be found by taking a derivative of v_{radial}. Hence the radical acceleration is,

$$a_{radial} = \frac{dv_{radial}}{dt} = 1.51$$

The reason for radial velocity due to planets is sinusoidal is that, the planet revolves around the star periodically in an approximated circle (ellipse) and hence the radial velocity induced by that motion will be sinusoidal in nature.

6.

Reading the data:

```
df = pd.read_csv('acc_data.csv')
```

```
df
```

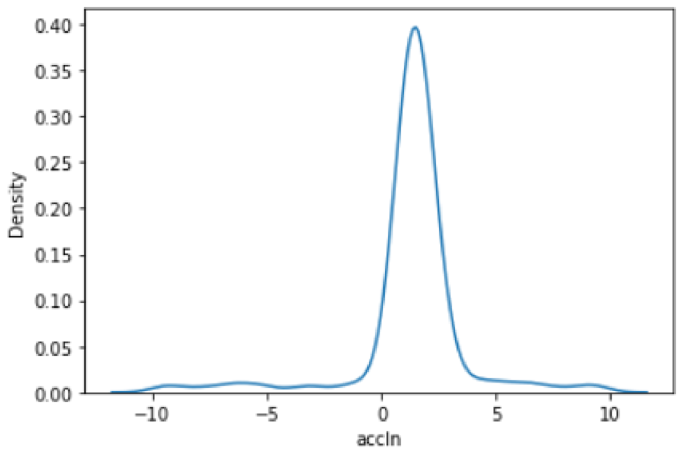
	Stars	accln
0	0	2.567932
1	1	2.345989
2	2	0.015347
3	3	0.646854
4	4	1.932717
...	...	...
1195	1195	6.175684
1196	1196	1.196303
1197	1197	-8.301368
1198	1198	8.903354
1199	1199	-0.862692

1200 rows × 2 columns

```
X = df['Stars']
Y = df['accln']
```

Plot and average acceleration:

```
s = sns.kdeplot(Y,label = 'accln')
```




```
np.mean(Y)
```

```
1.3242907586933332
```

Using gaussian fit over the data with mean and variance:

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} * e^{-(\frac{(x-\mu)^2}{2\sigma^2})}$$

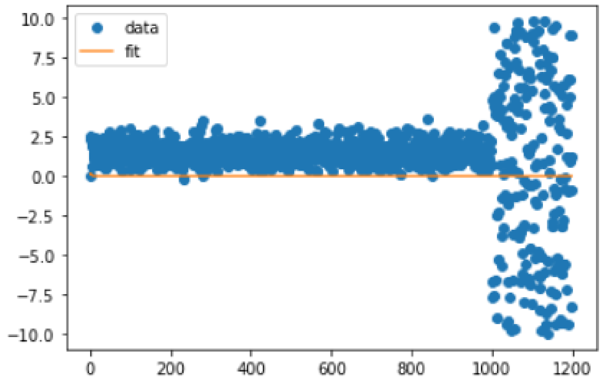
Here,

$$\sigma = Std\ deviation$$

$$\mu = mean$$

$$\sigma^2 = Variance$$

```
In [19]: def Gauss(x, A, B):
             y = (1/np.sqrt(2 * np.pi *np.var(Y))) * np.exp(-1*((x-np.mean(Y)) / np.sqrt(2 * np.var(Y)))**2)
             return y
         # parameters, covariance = curve_fit(Gauss, X[:1000], Y[:1000])

         # fit_A = parameters[0]
         # fit_B = parameters[1]

         fit_y = Gauss(X, fit_A, fit_B)
         plt.plot(X, Y, 'o', label='data')
         plt.plot(X, fit_y, '-', label='fit')
         plt.legend()

Out[19]: <matplotlib.legend.Legend at 0x1fc3877fcd0>
```



7. The calculations are as follows,

A = 15.3 ± 0.4 km s−1 kpc−1

B = −11.9 ± 0.4 km s−1 kpc−1

A – B = 27.2 ± 0.8 km s−1 kpc−1

$$\rho_{dm=\frac{1}{4(\pi)G}}\left[2(A-B)^2 \; - \; \frac{\partial a_r}{\partial r}\right]$$

$(A-B) = 8.5566 \; * \; 10^{-16}\, s^{-1}$

$\frac{\partial a_r}{\partial r} = 4.45256 * \; 10^{-30}\, s^{-2}$

Plugging in the values of π, G, A-B, and $\frac{\partial a_r}{\partial r}$,

$$\rho_{dm} \; = \; 3.348 * 10^{-21} \; kg/m^3$$

Or

$$\rho_{dm} \; = \; 3.348 * 10^{-3} \, M_{\emptyset}/pc^3$$

Conclusion:

1. The Gauss Law of Gravitation is stated and derived using vector calculus.
2. By using Poisson equation the relationship between dark matter density, Oort constants and gradient of acceleration (contribution of dark matter) is established. The significance of Oort constants is also discussed.
3. The above mentioned dark matter density formula is numerically analysed with a real observational value the relative uncertainty is predicted to be 30%.
4. The hurdles that would be faced upon making observations of radial velocities of stars is discussed and the possible solutions are pointed out.
5. From the given data, the radial velocity is obtained as a function of time by filtering the periodic signal and noise using fast fourier transform and linear regression. The radial acceleration is also determined by taking the derivative of radial velocity.
6. The average acceleration is found by plotting the data given in a probability distribution curve and the data is fitted with Gaussian function.
7. By using the values obtained from the previous exercises, the dark matter density is calculated by using the formula mentioned before.

   The required codes and plots are inserted in necessary places.

References :

[1] https://academic.oup.com/mnras/article/366/3/899/993295

[2] https://arxiv.org/abs/1212.3670