TO DETERMINE THE ECCENTRICITY OF MOON's ORBIT

Swayamjit K., Anshika S., Subhajit B., Aditya Raj, Samarth M. N.
Suggested by: Rabi N. M. and Pratik from batch 22
National Institute Of Science Education and Research, Bhubaneswar

Abstract

The aim of this project is to calculate the eccentricity of the moon's orbit by measuring the apparent radius of the moon over a complete revolution cycle of moon around earth. Understanding the eccentricity of the moon's orbit helps in studying the dynamics of the Earth moon system and variation in tidal effects on Earth. Best of the moon images captured, have been processed to get the apparent radius of the moon in the data collection period. The plot of apparent radius v/s time is fitted with a sinusoidal fit and a Kepler orbital model fit to find eccentricity and other fitting parameters corresponding to each fit. Though experimental eccentricity values from both the fitting plots are found to be greater than its literature value, Kepler fit is found to give a more accurate value.

Contents

1 Introduction

The study of celestial mechanics forms the backbone of our understanding of the motion of astronomical objects under the influence of gravity. One of its cornerstone principles is the **two-body problem**, which examines the motion of two masses interacting solely through their mutual gravitational force. The two-body problem assumes no external forces act on the system and simplifies the complex interactions of celestial mechanics into a solvable model.

For such a system, Newton's law of gravitation and second law of motion lead to the equations describing the relative motion of the two

bodies. The resulting trajectory of the bodies is found to be a conic section (circle, ellipse, parabola, or hyperbola) depending on the total energy of the system. For bound systems, such as the Earth-Moon system, the relative motion is **elliptical**.

The **eccentricity** (e) is a dimensionless parameter that quantifies the deviation of the orbit from a perfect circle. For circular orbits, $e = \mathbf{0}$, while for elliptical orbits $0 < e < 1$.

The Moon's orbit around the Earth is a classic example of an elliptical motion described by the two-body problem. While the Moon's orbit is close to circular, its slight eccentricity (**Literature value of $e = 0.0549$**) results in observable variations in its distance from the Earth over time, ranging from perigee (closest point) to apogee (farthest point).

In this experiment, we aim to calculate the eccentricity of the Moon's orbit by **analyzing the change in its apparent radius**. Observing the Moon's angular size or apparent size at various points in its orbit allows us to calculate the necessary parameters. This calculation enables us to quantify the elliptical nature of the Moon's orbit and compare it with theoretical predictions.

1.1 Astrophysical Motivation

The astrophysical motivation for studying the Moon's orbital eccentricity lies in understanding the complex dynamics of the Earth-Moon system, testing gravitational theories, and exploring tidal interactions. The Moon's orbit is slightly elliptical, meaning its distance from Earth varies over time. By precisely measuring the eccentricity, scientists can refine models of how the Earth-Moon system has evolved and how it continues to change. This helps in understanding the long-term behavior of the Moon's orbit, including its gradual increase in distance from Earth and its impact on Earth's rotation.

Studying the eccentricity also provides valuable insights into the formation of the Moon and the early solar system. Variations in the Moon's orbit offer clues about the forces that shaped the Earth-Moon system after the Moon's formation, such as gravitational interactions and tidal forces between Earth, the Moon, and the Sun.

Additionally, the eccentricity of the Moon's orbit plays a significant role in the tidal forces between Earth and the Moon. These forces influence Earth's rotation, the length of days, and the stability of Earth's axial tilt. This research is crucial for predicting future changes in Earth's rotation and orbit. Studying these effects also improves our knowledge of the dynamics of other planetary systems and their moons.

1.2 Background theory

1.2.1 Two Body Problem

To study the earth and moon system we need to study the motion of a system consisting two bodies affected by a force directed along the line joining the centers of two bodies without subject to any other, "external," forces.

Let us consider two objects, with masses m_1 and m_2 whose center of mass is at a distance R from the origin. And the distance between them is r. Figure- 1

Let's move the origin on the center of mass of the two body system for the shake of convenience. That makes our new $R = 0$. Figure- 2

We Define,

$$r = r_2 - r_1 \tag{1}$$

$$\frac{m_1 r_1 + m_2 r_2}{m_1 + m_2} = 0 \tag{2}$$

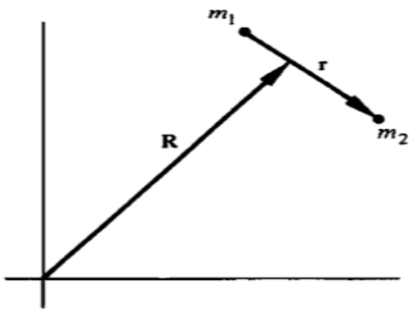

Figure 1: coordinates of the two body problem

From (1) & (2) we get :-

$$r_1 = -\frac{m_2}{m_1 + m_2} r \tag{3}$$

$$r_2 = \frac{m_1}{m_1 + m_2} r \tag{4}$$

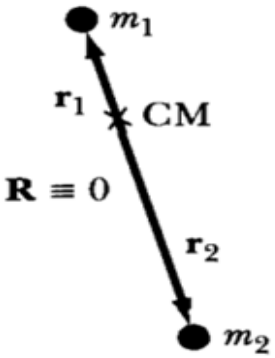

Figure 2: Centre of mass frame

Now writing the kinetic energy term for the system -

$$E_k = \frac{1}{2} m_1 \dot{r}_1^2 + \frac{1}{2} m_2 \dot{r}_2^2 \tag{5}$$

Replacing r_1 and r_2 with above mentioned relation with r we get -

$$E_k = \frac{1}{2} m \dot{r}^2 \tag{6}$$

Where m represents reduced mass, $m = \frac{m_1 m_2}{m_1 + m_2}$.

The potential energy is given by –

$$E_G = -\frac{G m_1 m_2}{|r_2 - r_1|} \tag{7}$$

Substituting for r & m in the equation, we get,

$$E_G = -\frac{G(m_1 + m_2)m}{r} \tag{8}$$

For the Earth & moon system, the combined mass of Earth and moon is nearly equal to Earth. So taking $M = m_1 + m_2$ will also give a fairly accurate system representation.

$$E_G = -\frac{GMm}{r} \tag{9}$$

With these changes now the two-body problem is the same as a one-body problem with mass m orbiting mass M.

Angular momentum is given by:

$$L = mr\hat{\mathbf{r}} \times \dot{\mathbf{r}}$$

$$= mr\hat{\mathbf{r}} \times (\dot{r}\hat{\mathbf{r}} + r\dot{\hat{\mathbf{r}}})$$

$$= mr^2 \hat{\mathbf{r}} \times \dot{\hat{\mathbf{r}}}$$

Now $F = -\frac{GMm}{r^2}\hat{\mathbf{r}}$,

Cross-multiplying $\ddot{\mathbf{r}}$ in the above angular momentum equation:

$$\ddot{\mathbf{r}} \times L = -\frac{GM}{r^2}\hat{\mathbf{r}} \times (mr^2 \hat{\mathbf{r}} \times \dot{\hat{r}})$$

Using the relation-

$$\mathbf{A} \times (\mathbf{B} \times \mathbf{C}) = (\mathbf{A} \cdot \mathbf{C})\mathbf{B} - (\mathbf{A} \cdot \mathbf{B})\mathbf{C}$$

And further simplifying:

$$\ddot{\mathbf{r}} \times L = GMm\dot{\hat{r}}$$

This can also be written as:

$$\frac{d}{dt}(\ddot{\mathbf{r}} \times L) - \dot{\mathbf{r}} \times \dot{L} = \frac{d}{dt}(GMm\hat{\mathbf{r}})$$

Integrating both sides gives us:

$$\ddot{\mathbf{r}} \times L = GMm\hat{\mathbf{r}} + D \quad \text{as } \dot{L} = 0$$

Taking the dot product with $\mathbf{r}$ on both sides gives:

$$\mathbf{r} \cdot (\ddot{\mathbf{r}} \times L) = GMm\mathbf{r} \cdot \hat{\mathbf{r}} + \mathbf{r} \cdot D$$

Rearranging LHS by:

$$\mathbf{A} \cdot (\mathbf{B} \times \mathbf{C}) = (\mathbf{A} \cdot \mathbf{B}) \cdot \mathbf{C}$$

$$(\mathbf{r} \times \dot{\mathbf{r}}) \cdot L = GMmr + rD\cos\alpha$$

Rewriting:

$$\frac{L^2}{m} = GMmr\left(1 + \frac{D\cos\alpha}{GMm}\right)$$

Substituting $D = GMme$ and solving for r gives:

$$r = \frac{\frac{L^2}{m^2}}{GM(1 + e\cos\alpha)}$$

$$r = \frac{c}{1 + e\cos\alpha} \tag{10}$$

where $c = \frac{L^2}{m^2 GM}$, This is our solution for r as a function of α, in terms of the positive constant e and c, is an equation of conic section. The behavior of the orbit r (α) in eq.(10) is controlled by the positive constant e. This equation shows different behavior according to $e < 1$ or $e > 1$. If $e < 1$, the denominator of never vanishes, and r (α) remains bounded for all α and unbounded for all other cases. As our earth moon system is bounded we will take the range $0 \leq e < 1$, Therefore r oscillates between the maximum and minimum values given by:

$$r_p = \frac{c}{1 + e} \quad \text{and} \quad r_a = \frac{c}{1 - e}$$

Where,

$r_p = r\min$ at perihelion when $\alpha = 0$,

$r_a = r\max$ at aphelion when $\alpha = \pi$.

Solving for e from the equations of r_p and r_a gives:

$$e = \frac{r_a - r_p}{r_a + r_p} \tag{11}$$

1.2.2 Relation of eccentricity with apparent radius

We know, any object's apparent size decreases with increasing distance between the object and the observer. Hence, apparent size and distance are inversely proportional to each other. If we do imaging of the moon for 30 consecutive days, then the moon's apparent size would complete one cycle of oscillation. This happens because with each cycle of revolution, the radial distance oscillates with its periodicity same as the revolution period and hence, the apparent size also oscillates with same periodicity.

At **APOGEE,** moon is farthest from earth, hence, its apparent radius is the smallest.

At **PERIGEE,** moon is closest to earth, hence, its apparent radius is the largest.

The plot of apparent diameter v/s time would produce a periodically oscillating curve as shown in Fig.4. We can get the eccentricity by equation (11).

Since radial distance is inversely proportional to apparent diameter, we can write,

$$r_a = \frac{k}{A} \tag{12}$$

$$r_p = \frac{k}{P} \tag{13}$$

Where, k is proportionality constant, and A and P are the appent diameters of moon at apogee and perigee respectively. Putting equations (12) and (13) in eqn (11), we get:

$$e = \frac{P - A}{P + A} \tag{14}$$

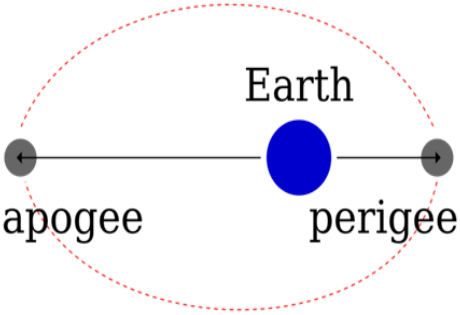

Figure 3: Elliptical orbit of moon

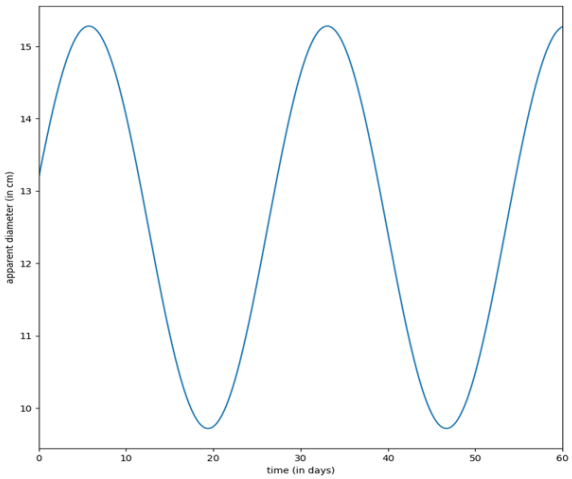

Figure 4: Sample graph showing periodic oscillation of apparent diameter with time

2 Description of the experiment

2.1 Experimental setup

APPARATUS REQUIRED: 6-inch Newtonian telescope, mobile phone, mobile phone holder, a PC (laptop or desktop)

Detailed Description:

To perform this experiment, a simple phone camera with an ISOCELL JN1 sensor alone cannot produce precise enough images of the Moon for analysis. Therefore, we used a 6-inch Newtonian telescope, a popular choice among amateur astronomers for its excellent balance of portability, affordability, and performance. It allows observation of a wide range of celestial objects, including the Moon, planets like Jupiter and Saturn (along with their moons and rings), star clusters, nebulae, and even some distant galaxies under dark skies.

The 6-inch diameter collects about 450 times more light than the human eye, enabling clear views of faint objects. Overall, a 6-inch telescope is an excellent tool for beginners and intermediate astronomers, providing a gateway to the wonders of the night sky.

In our experiment, we used a 6-inch telescope with a focal length of 750mm and a 25mm eyepiece, which provides a magnification of 30x. The true field of view (TFOV) is 1.73°, assuming the aperture is not being vignetted by dew shields, internal baffles, or the focuser tube itself.

The images were captured using a smartphone with an aperture of f/1.8, a sensor area of 1/2.76”, and a 50 MP camera.

Links:

For mobile phone Holder, tap here.

For a 6-inch telescope, tap here.

2.2 Experimental procedure and data acquisition

2.2.1 Procedure

1) Setup and align the telescope and make sure to level the base horizontally. refer sec-2.2.2

2) A mobile phone camera was mounted using a dedicated mobile holder attached to the telescope. This arrangement allowed for seamless tracking of the Moon as it moved across the sky. The smartphone camera was positioned in front of the eyepiece such that the entire field of view was visible on the phone's screen. Focus adjustments were made to achieve a sharp and well-defined image of the Moon.

3) Click few pictures of the moon ensuring that the edges of the lunar disk is clearly visible. Adjustments to the camera settings, such as shutter speed and ISO, can be made using the phone's pro mode to account for variations in sky conditions and the Moon's phase cycle. Obtaining sharp and distinct edges of the Moon should be prioritized.

4) To minimize atmospheric disturbances, imaging was conducted when the Moon was near the zenith, where the path of light through the atmosphere is shortest. This ensured reduced distortion and enhanced image clarity.

5) To capture a full sinusoidal variation in the Moon's apparent radius, images were taken over at least one complete lunar orbital period ($\tilde{3}0$ consecutive days) with the same mobile camera, starting from the first day of observation.

6) A computer algorithm(code) was developed, which measures the apparent radius of the moon in pixel widths. Code is provided in the Appendix section.

7) The apparent diameter of the Moon, expressed in pixel values, was plotted as a function of time. A sinusoidal and Kepler curve fitting was applied to the data.

8) The required P and A values were taken from the fitted plots and put into eq(14) to find the value of eccentricity.

2.2.2 Steps to setup a 6" telescope

- Open up the tripod. Make sure the legs are at the maximum separation.
- Make one of the legs face the South with the help of a compass.
- Level the top of the mount with the help of spirit level. Just Google search 'Bubble level'. Adjust the lengths of the legs using the adjustment knobs so as to make the reading 0.
- Set the mount on the top of the tripod using a screw and a washer. The mount should be so oriented that a long screw coming out of it near the bottom is aligned with the South-facing leg. Tighten the screw properly.
- Attach the weight to the mount with the help of the rod. Tighten the weight and the screws properly, otherwise the weight can slip on the rod.
- Fit the tube rings with help of allen keys. Make the two rings parallel or antiparallel to each other.
- Carefully take out the telescope tube out of the bag. Remove the plastic cover and the cap. Two people should always be holding the tube.
- Loosen both the axes of the mount. Put the tube into the groove created by the two rings and tighten the screws properly so that the tube does not slip.
- Attach the view-finder/ finderscope and the eyepiece.
- Balance each of the two axes by locking the other. At near-equilibrium situations, lock the axis to be balanced and rotate the ne adjustment knob for that axis. You will feel more resistance while rotating in one direction than the other if the axis is not perfectly balanced.
- When perfectly balanced, you can put the telescope in any random orientation (except at extreme angles) and it should stay that way forever given there is no wind.

- After the balancing is over, alignment has to be done. Choose a terrestrial source of light far away from the telescope, generally the red lights on the top of the towers. Bring the source at the center of the eyepiece.
- Adjust the three screws on the finderscope so as to bring the same source at the center of the cross-wire in the finderscope. Crosscheck the alignment with another source.
- The telescope is now set for observations.

3 Analyses

3.1 Determining the radius from the images

We wrote a Python script to perform this task. The code for detecting the moon's edge and finding the apparent radius with its explanation is given in the appendix. It detects the edge by using the concept of sudden drastic change of intensity of pixels while passing across the edge. The intensity values used are taken from gray-scale image of the images taken. The edge points are then fits to a circle to determine the radius.

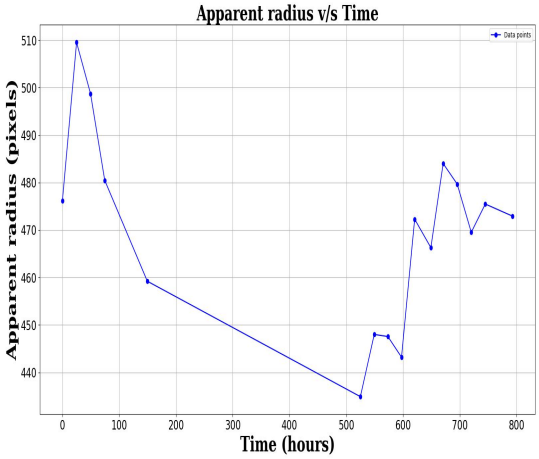

Figure 5: observed data points

3.2 Observation tables

The top three images from each day were selected(refer to Appendix 6.4 to know what is a good image and what is a bad image), and their radius was determined using the code in Appendix 6.1, and the corresponding table 1 was made.

3.3 Data Analysis

We determined the avarage radius of the moon from the best three clicked pictures of dimentions $4080 * 3060$ (given in Table 1) and plotted it against total mumber of hours to get a proper scalled graph. Figure-5

We have done a sinosudal fiting of the above plot. The motivation for this approach is that the variation in the apparent size of the Moon is periodic in nature, and the simplest periodic function that can describe such a variation is a sine function.

The sinusoidal function used for fitting is given by:

$$y(t) = A\sin(wt + c) + C$$

where:

- A: Amplitude of the sine wave, representing half the range of the variation in apparent radius.
- w: Angular frequency, related to the time period (T) of the variation by $T =$

Table 1: Determining the radius of the moon from the best three images per night

Date and Time of Data Taking	**Sl. no.**	**Measured Apparent Radius (pixel)**	**Mean Apparent Radius (pixel)**	**Time (hrs)**
15.10.2024 22:43	1	474.290	476.122	0.000
	2	475.707		
	3	478.370		
16.10.2024 23:43	1	491.362	509.519	25.000
	2	512.792		
	3	524.404		
18.10.2024 00:20	1	485.608	498.704	49.616
	2	504.610		
	3	505.894		
19.10.2024 01:19	1	480.634	480.466	74.599
	2	484.473		
	3	476.290		
22.10.2024 04:18	1	461.480	459.209	149.599
	2	454.154		
	3	461.993		
06.11.2024 19:40	1	433.297	434.900	524.832
	2	435.481		
	3	435.737		
07.11.2024 19:58	1	429.272	448.004	549.265
	2	477.928		
	3	436.939		
08.11.2024 19:56	1	440.142	447.563	573.265
	2	443.007		
	3	459.543		
09.11.2024 19:55	1	443.714	443.212	597.265
	2	442.925		
	3	442.998		
10.11.2024 19:07	1	472.290	472.290	620.449
	2	472.290		
	3	472.290		
11.11.2024 23:29	1	461.739	466.288	648.982
	2	461.351		
	3	475.713		
12.11.2024 21:17	1	496.381	484.007	670.632
	2	477.925		
	3	477.716		
13.11.2024 22:02	1	498.619	479.676	695.365
	2	467.266		
	3	473.144		
14.11.2024 22:49	1	473.020	469.520	720.148
	2	466.020		
	3	0.000		
15.11.2024 23:32	1	473.432	475.473	744.865
	2	475.553		
	3	477.435		
17.11.2024 23:35	1	482.037	472.919	792.915
	2	467.632		
	3	469.088		

$\frac{2\pi}{w}$.

- c: Phase offset, accounting for the shift in the sine wave along the time axis.
- C: Vertical offset, representing the mean apparent radius.

For more details refer to sec-6.3. The r_a corresponds to maximum of the sine function, and the r_p corresponds to minimum of the sine function up to a proportionality constant. The eccentricity is calculated using the formula(11), mentioned earlier.

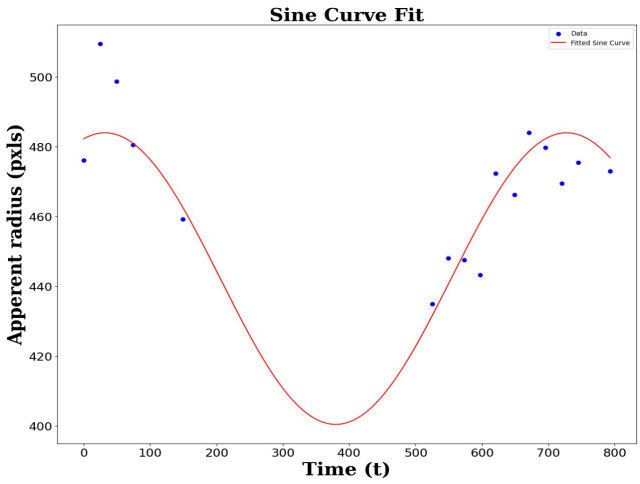

Figure 6: Sinusoidal curve fit

Fitted parameters for sinusoidal curve fit(fig.8) with corresponding uncertainties:

A = 41.786 ± 9.973,

w = 0.009 ± 0.001,

c = 1.281 ± 0.343,

C = 442.187 ± 8.946,

time period (in hrs)= 695.359,

time period (in days) =28.973

Keplerian Orbital Model The Keplerian model describes the apparent radius of an orbiting body based on its elliptical orbit, using:

- **Mean Anomaly** (M): Defined as $M = 2\pi\frac{(t-t_0)}{P}$, where t_0 is the time of pericenter passage and P is the orbital period.
- **Eccentric Anomaly** (E): An auxiliary angle that is iteratively solved using the relationship $E = M + e\sin(E)$, where e is the orbital eccentricity. This iterative method ensures numerical accuracy.
- **True Anomaly** (θ): Computed from E using $\theta = 2\arctan\left(\sqrt{\frac{1+e}{1-e}}\cdot\tan\frac{E}{2}\right)$. The true anomaly describes the position of the orbiting body along its orbit.
- **Radius** (r): Determined as $r = \frac{a(1-e^2)}{1+e\cos(\theta)}$, where a is the semi-major axis of the orbit.
- **Apparent Radius** (r_{apparent}): Simulated as the inverse of the distance, scaled by a diameter-like factor D, i.e., $r_{\text{apparent}} = \frac{D}{r}$.

For Detailed Model refer sec-6.2

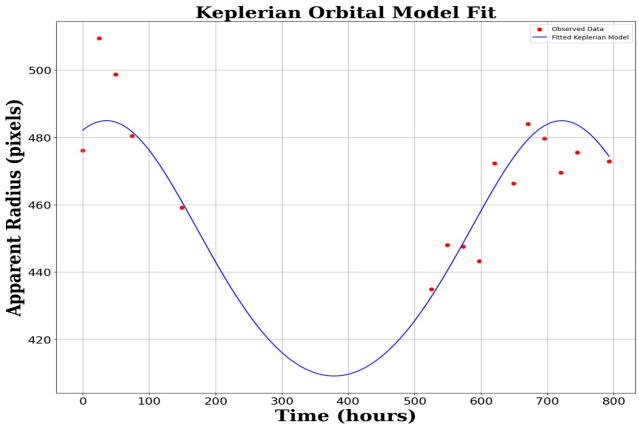

Figure 7: Keplerian orbital model fit

Fitted Parameters for Keplerian model fit(fig.9) with corresponding Uncertainties:

Semi-Major Axis (a): 0.00225 ± 0.00003,

Eccentricity (e): 0.084 ± 0.016,

Orbital Period (P): 685.885 ± 42.184,

Time Offset (t_0): 35.695 ± 29.882

Diameter (D): 699.676 ± 817364528.563 pixels

To investigate the potential impact of atmospheric noise and the viewing angle (zenith being 90 degrees and horizon being 0 degrees) on our observations, we captured images the Moon at successive intervals of 20-25 minutes overnight on November 6 and November 9. The Imaging sequence spanned from the Moon's zenith position to its descent below the horizon. This approach allowed us to analyze whether moving away from the zenith significantly affected the quality of the data.

Sl. no.	Time (hrs)	Measured Apparent Radius (pxls)
1	0.000	409.384
2	0.229	430.858
3	0.525	464.771
4	0.877	455.539
5	1.102	439.355
6	1.125	438.130

Table 2: Measured Apparent Radius for 6th November

Sl. no.	Time (hrs)	Measured Apparent Radius (pxls)
1	0.000	439.18
2	0.433	402.34
3	0.816	480.48
4	1.167	429.76
5	1.517	450.22
6	1.850	440.22
7	2.233	447.92
8	2.583	433.50
9	3.000	447.88
10	3.583	447.18

Table 3: Measured Apparent Radius for 9th November

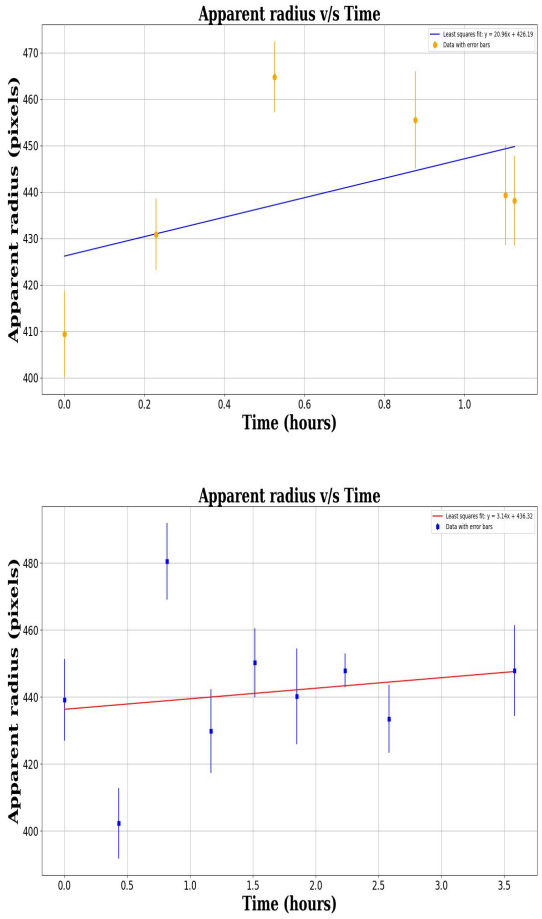

Figure 8: observing the change in size of the moon as it decends on 6th and 9th November data

We fit it with a straight line so that we could, capture the trend of the variation. This was performed because it is nearly impossible to capture the images of the moon every day at the same point in the sky.

3.4 Error Analysis

The uncertainties in the fitted parameters are derived from the covariance matrix calculated during the optimization process in `scipy.optimize.curve_fit`. The covariance matrix, $\mathbf{V}$, is related to the inverse of the Hessian matrix of the least-squares cost function, $\mathbf{V} = (\mathbf{J}^T\mathbf{J})^{-1}\sigma^2$, where $\mathbf{J}$ is the Jacobian matrix of the residuals and σ^2 is the variance of the data points. The diagonal elements of the

covariance matrix represent the variances of the parameters, and their square roots provide the standard errors (uncertainties). This process assumes that the residuals are normally distributed, the model accurately represents the data, and the data points have uncorrelated errors. If these assumptions hold, the uncertainties offer a reliable measure of the confidence in the fitted parameters. In Python, the uncertainties are computed as `np.sqrt(np.diag(pcov))`, where `pcov` is the covariance matrix output by the fitting process.

- Covariance matrix for the fitted sine model(parameters: A, w, c and C):

$$\begin{bmatrix} 9.94 \times 10^{1} & 2.36 \times 10^{-3} & -1.88 \times 10^{-1} & -8.19 \times 10^{1} \\ 2.36 \times 10^{-3} & 4.86 \times 10^{-7} & -2.14 \times 10^{-4} & -2.84 \times 10^{-3} \\ -1.88 \times 10^{-1} & -2.14 \times 10^{-4} & 1.17 \times 10^{-1} & 4.21 \times 10^{-1} \\ -8.19 \times 10^{1} & -2.84 \times 10^{-3} & 4.21 \times 10^{-1} & 8.07 \times 10^{1} \end{bmatrix}$$

  The covariance matrix has relatively larger values, indicating less precise parameter constraints.These are larger, suggesting stronger correlations between parameters, which may lead to ambiguity in parameter estimation.Variances are larger, reflecting higher uncertainty in the fitted parameters.

- Covariance matrix for the fitted Keplerian model(parameters: a, e, P and t0):

$$\begin{bmatrix} 1.12 \times 10^{-9} & 4.82 \times 10^{-7} & -5.48 \times 10^{-4} & 7.12 \times 10^{-5} \\ 4.82 \times 10^{-7} & 2.62 \times 10^{-4} & -2.06 \times 10^{-1} & 1.54 \times 10^{-2} \\ -5.48 \times 10^{-4} & -2.06 \times 10^{-1} & 1.77 \times 10^{3} & -1.10 \times 10^{3} \\ 7.19 \times 10^{-5} & 1.54 \times 10^{-2} & -1.10 \times 10^{3} & 8.92 \times 10^{2} \end{bmatrix}$$

  The covariance matrix has smaller values, indicating better constraints on the parameters.These are relatively small, showing minimal correlation between parameters. Variances (uncertainties) are small, reflecting higher precision in parameter estimation.

The error in eccentricity determined from the sine curve fit is calculated by propagating the error using formula 11. First we need to determine the error in the maximum and minimum values of radius found by the sine curve fitting, these values will have an error because of the error in the fitting parameters. This is given by:

$$\sigma_y^2 = (\sin(wt+c)\sigma_A)^2 + (At\cos(wt+c)\sigma_w)^2 + (-A\cos(wt+c)\sigma_c)^2 + (\sigma_C)^2$$

where σ_A, σ_A, σ_A, and σ_A are determined from the fit in figure 6. The calculated error turns out to be for r_max=13.42 and r_min=21.79.

the error in eccentricity is then found to be 0.0403.

4 Drawing results from the analyses

4.1 Results

1. From a general calculation using formula without any Fitting, eccentricity is 0.079.

2. After Sinusoidal fitting as shown in Fig. 6, eccentricity is 0.094±0.040 .

3. After Keplerian fitting as shown in Fig.7, eccentricity is 0.084 ±0.016 .

4.2 Inferences from results

The Keplerian fit is better because it shows smaller covariance values, indicating higher precision in parameter estimation. The off-diagonal elements of its covariance matrix are significantly smaller, suggesting reduced correlations between parameters, which reflects a more stable and independent estimation process. The diagonal elements also indicate lower uncertainties in the parameters, making the fit more reliable. In contrast, the sine fit has larger covariance values and higher

correlations between parameters, resulting in higher uncertainties and less precise parameter estimates. Thus, the Keplerian model provides a more accurate and robust fit to the data.

v

5 Conclusions

Our calculated value of eccentricity is slightly higher than the literature value. Possible reasons for deviations are the following:

Data collection was hampered due to the Moon's cyclic pattern. Its rising time shifted to the early morning, making it visible only during the day for a few days, so during this period capturing sharp-edged images was almost impossible.Hence, the number of data points was reduced.

In addition, haze and light pollution in the humid local weather were two major challenges during the night. The haze limits the sharpness of the images we can get using a phone camera by causing light to disperse and dilute the pixels adjacent to the edges, which could be misread by the code as edge coordinates.

Since the phone sensor collects light from the eyepiece, a larger sensor would produce higher-resolution images. The greater the number of pixels, the better the approximation of the edges can be achieved.

Finally, the graphs from 6 November and 9 November demonstrate no significant deviation from the value at zenith. However, it shows a positive slope after linear fitting, indicating an apparent increase in size near the horizon. This variation is suspected to be due to an increase in air mass (i.e., the amount of atmosphere that the Moon's light passes through as it moves lower in the sky) and atmospheric turbulence.

Percentage deviations from literature value which is 0.0549[1]:

- For sinusoidal fit, Percentage deviation= 70.9 %
- For Keplerian fit, Percentage deviation= 52.7 %

All the three values of eccentricity that have been determined point out to the fact the the moons orbit is indeed an ellipse.

6 Appendix

6.1 Python script to calculate the radius of moon

1: Script to calculate the radius of moon

```
import cv2
import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import
    minimize

moon=cv2.imread('moon_image.jpg')
moon_gs=cv2.cvtColor(moon_d1,cv2.
    COLOR_BGR2GRAY)

L1=[] #(row,column)
Lnew1=[]
a1,b1,c1,d1=(,,,) #coordinates of
    the 40x40 portion
# (ymin,ymax,xmin,xmax)
for row in range(40):
    for col in range(40):
        if moon_d1_gs[a1:b1,c1:d1
            ][row][col]<0.8*(
            moon_d1_gs[b1,d1]-
            moon_d1_gs[a1,c1]):
            pass
        else:
            L1.append((a1+row,c1+
                col))
            break
for i in L1:
    if i[1]!=c1:
        Lnew1.append(i)

L2=[] #(row,column)
Lnew2=[]
a2,b2,c2,d2=(,,,)
for row in range(40):
    for col in range(40):
        if moon_d1_gs[a2:b2,c2:d2
            ][row][col]<0.8*(
            moon_d1_gs[a2,d2]-
            moon_d1_gs[b2,c2]):
            pass
        else:
            L2.append((a2+row,c2+
                col))
            break
for i in L2:
    if i[1]!=c2:
        Lnew2.append(i)

L3=[] #(row,column)
Lnew3=[]
a3,b3,c3,d3=(,,,)
for row in range(40):
    for col in range(40):
        if moon_d1_gs[a3:b3,c3:d3
            ][row][col]>0.8*(
            moon_d1_gs[b3,c3]-
            moon_d1_gs[a3,d3]):
            pass
        else:
            L3.append((a3+row,c3+
                col))
            break
for i in L3:
    if i[1]!=c3:
        Lnew3.append(i)

L4=[] #(row,column)
Lnew4=[]
a4,b4,c4,d4=(,,,)
for row in range(40):
    for col in range(40):
        if moon_d1_gs[a4:b4,c4:d4
            ][row][col]>0.8*(
            moon_d1_gs[a4,c4]-
            moon_d1_gs[b4,d4]):
            pass
        else:
            L4.append((a4+row,c4+
                col))
            break
```

```
for i in L4:
    if i[1]!=c4:
        Lnew4.append(i)

def residuals(params, points):
    """
    Compute the sum of squared
       residuals for the given
       parameters (h, k, r).

    params: (h, k, r) -
       Parameters for the circle:
        center (h, k) and radius
       r.
    points: List of tuples (x, y)
        representing the
       coordinates of the points.

    Returns: Sum of squared
       residuals.
    """
    h, k, r = params
    points = np.array(points)
    distances = np.sqrt((points
       [:, 0] - h)**2 + (points
       [:, 1] - k)**2)
    return np.sum((distances - r)
       **2)

def best_fit_circle(points):
    """
    Find the best fit circle
       using least squares
       fitting.

    points: List of tuples (x, y)
        representing the
       coordinates of the points.

    Returns: The center (h, k)
       and radius r of the best
       fit circle.
    """
    initial_guess = [1939, 1491,
       np.mean(np.linalg.norm(np.
       array(points), axis=1))]
    # here,1939 and 1491 are the
       initial guess of
       coordinates of the centre

    # Use the minimize function
       to find the best
       parameters (h, k, r)
    result = minimize(residuals,
       initial_guess, args=(
       points,), method='Nelder-
       Mead')

    h, k, r = result.x  # Extract
        the center and radius
    return h, k, r

points = Lnew1 + Lnew2 + Lnew3 +
   Lnew4
h, k, radius = best_fit_circle(
   points)
print("Center of the best fit
   circle: ({}, {})".format(h, k)
   )
print("Radius of the best fit
   circle:", radius)
```

The above code takes the jpg file of the moon image and converts it into grayscale image. Then, we have to manually zoom the image and select a region of dimension (40 pixel x 40 pixel). The selected region should contain a portion of the edge. Based on the location of the selected region, we define 4 lists (L1 , L2, L3 and L4) in order to store the coordinates of the detected edge points. Lnew1 , Lnew2 , Lnew3 and Lnew4 store the edge points after removing internal/external points detected at the left margin of the region. L1 refers to the left upper portion of the moon. Similarly, L2, L3 and L4 refers to the left lower, the right upper and the right lower portion of the moon respectively. The coordinates of the selected 40x40 region is given as input in the variables a1, b1,c1 and d1 in the form (ymin, ymax, xmin, xmax). The edge points are detected by the code where the intensity changes drastically (here, 80% of the intensity difference of an internal point on moon and an external point is given to be the

edge detection condition).

Once the coordinates of the edge points are stored, now we need to find out the radius of the best fit circle passing through the given edge coordinates. The functions residuals(params, points)and best_fit_circle(points) perform the task of finding the centre and radius of the best fit circle by using minimize function of scipy.optimize module. Initial guess of the centre of the moon is provided as input.

Result of the code provides the centre and radius of the best fit circle.

Figure 9: example of a best fit circle on a half disk illuminated iamge from 8th Nov

6.2 Python script to do the keplerian fit

2: Script to do the keplerian model fit of the data

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import
    curve_fit

# Data
t_data = np.array([0, 25, 49.616,
     74.5994, 149.5994, 524.8328,
    549.2658, 573.2658, 597.2658,
    620.4491, 648.9821, 670.6321,
    695.3655, 720.1488, 744.8654,
    792.9154])
y_data = np.array([476.122,
    509.519, 498.704, 480.466,
    459.209, 434.900, 448.004,
    447.563, 443.212, 472.29,
    466.288, 484.007, 479.676,
    469.52, 475.473, 472.919])

# the Keplerian orbital model
def keplerian_model_apparent(t, a
    , e, P, t0):
    M = 2 * np.pi * (t - t0) / P
    E = M
    for _ in range(10): #
        Iterative solver for
        eccentric anomaly
        E = M + e * np.sin(E)
    theta = 2 * np.arctan2(np.
        sqrt(1 + e) * np.sin(E /
        2), np.sqrt(1 - e) * np.
        cos(E / 2))
    r = a * (1 - e**2) / (1 + e *
        np.cos(theta))
    r_apparent = 1 / r
    return r_apparent

# initial guesses
r_apparent_max = np.max(y_data)
r_apparent_min = np.min(y_data)
P_guess = t_data[-1] - t_data[0]
e_guess = 0.05
t0_guess = 0
a_guess = (1 / r_apparent_min + 1
    / r_apparent_max )/ 2
initial_guesses = [a_guess,
    e_guess, P_guess, t0_guess]

# Bounds
bounds = (
    [0, 0, P_guess * 0.5, min(
        t_data)],
    [np.inf, 1, P_guess * 2, max(
        t_data)]
)

try:
    params, covariance =
        curve_fit(
```

```
        keplerian_model_apparent,
            t_data, y_data, p0=
            initial_guesses,
            bounds=bounds
    )
    a, e, P, t0 = params
    errors = np.sqrt(np.diag(
        covariance)) if covariance
        is not None else [None] *
        4
    a_err, e_err, P_err, t0_err =
        errors

    x_fit = np.linspace(min(
        t_data), max(t_data),
        1000)
    y_fit =
        keplerian_model_apparent(
        x_fit, a, e, P, t0)

    plt.figure(figsize=(10, 6))
    plt.scatter(t_data, y_data,
        color='red', label='
        Observed Data')
    plt.plot(x_fit, y_fit, color=
        'blue', label='Fitted
        Keplerian Model')
    plt.xlabel('Time (hours)')
    plt.ylabel('Apparent Radius (
        pixels)')
    plt.title('Keplerian Orbital
        Model Fit')
    plt.legend()
    plt.grid(True)
    plt.show()


    print("Fitted Parameters with
        Uncertainties:")
    print(f"- Semi-Major Axis (a)
        : {a:.5f}  {a_err:.5f}"
        if a_err else f"{a:.5f}
        N/A")
    print(f"- Eccentricity (e): {
        e:.5f}  {e_err:.5f}" if
        e_err else f"{e:.5f}  N/
        A")
    print(f"- Orbital Period (P):
        {P:.5f}  {P_err:.5f}"
        if P_err else f"{P:.5f}
        N/A")
    print(f"- Time Offset (t0): {
        t0:.5f}  {t0_err:.5f}"
        if t0_err else f"{t0:.5f}
        N/A")
    print(covariance)

except RuntimeError as err:
    print(f"Fit failed: {err}")
```

This Python code fits a Keplerian orbital model to observational data of apparent radii at specific time intervals, using the `numpy`, `matplotlib`, and `scipy` libraries. Below is an explanation of its components:

1. Purpose of the Code The purpose of the code is to fit a Keplerian orbital model to the observed data of apparent radii (y) as a function of time (t). The fitted model provides insights into orbital parameters, such as the semi-major axis, eccentricity, orbital period, and the time of pericenter passage. The apparent radius is modeled as an inverse function of distance, simulating observational data as would be perceived from Earth.

2. **Fitting the Model to Data** The observed data of time and apparent radius are provided as input arrays `t_data` and `y_data`, respectively. The `scipy.optimize.curve_fit` function is used to optimize the model parameters:

   - `a`: scaled Semi-major axis of the orbit.
   - `e`: Orbital eccentricity.
   - `P`: Orbital period.
   - t_0: Time offset (time of pericenter passage).

3. **Initial Guesses and Bounds** To ensure effective fitting, initial guesses for the parameters are calculated:

- P_{guess}: Estimated as the difference between the first and last time points in the dataset.
- e_{guess}: Set to a small value (0.05) as orbits are typically near circular.
- $t0_{\text{guess}}$: Set to 0 as an initial estimate.
- a_{guess}: Derived as an intermediate value based on the apparent radius range.

The bounds for the parameters ensure physically meaningful values during the optimization process.

4. **Output and Visualization** The fitted parameters and their uncertainties are calculated from the covariance matrix returned by `curve_fit`. A plot is generated to compare the observed data points with the fitted Keplerian model:
   - The observed data (`t_data` vs. `y_data`) are plotted as red scatter points.
   - The fitted model is plotted as a blue curve over a finer time grid.
5. **Key Libraries Used**
   - `numpy`: For efficient numerical computations and data handling.
   - `matplotlib`: For plotting the data and the fitted model.
   - `scipy.optimize.curve_fit`: For non-linear least squares optimization of the model to fit the data.

The output provides the best-fit parameters with their uncertainties, enabling a deeper understanding of the orbital characteristics of the system under investigation.

6.3 Python script to do the sine fit

3: Script to do the sine function model fit of the data

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import
    curve_fit

# the sine function
def sine_function(t, A, w, c, C):
    return A * np.sin(w * t + c)+
        C

#Data
t_data = np.array([0, 25, 49.616,
    74.5994, 149.5994, 524.8328,
    549.2658, 573.2658, 597.2658,
    620.4491, 648.9821, 670.6321,
    695.3655, 720.1488, 744.8654,
    792.9154])
y_data = np.array([476.122,
    509.519, 498.704, 480.466,
    459.209, 434.900, 448.004,
    447.563, 443.212,
    472.29,466.288, 484.007,
    479.676, 469.52, 475.473,
    472.919])


# Initial guesses
A_guess = (max(y_data) - min(
    y_data)) / 2
w_guess = 2 * np.pi / (t_data[6]
    - t_data[0]) # Estimate from
    range
c_guess = 0
C_guess = 470
p0 = [A_guess, w_guess, c_guess,
    C_guess]
popt, pcov = curve_fit(
    sine_function, t_data, y_data,
     p0=p0)

#parameters and errors
A, w, c, C = popt
A_err, w_err, c_err, C_err = np.
    sqrt(np.diag(pcov))
```

```
print(f"Fitted parameters:\nA = {A:.4f} ± {A_err:.4f}\nw = {w:.4f} ± {w_err:.4f}\nc = {c:.4f} ± {c_err:.4f}\nC = {C:.4f} ± {C_err:.4f}")
print("time period in hrs",2 * np.pi /w, "time period in days", 2 * np.pi /(24*w) )

t_fit = np.linspace(min(t_data), max(t_data), 1000)
y_fit = sine_function(t_fit , *popt)

plt.figure(figsize=(10, 10))
plt.scatter(t_data, y_data, label="Data", color="blue")
plt.plot(t_fit , y_fit , label="Fitted Sine Curve", color="red")
plt.title("Sine Curve Fit")
plt.xlabel("Time (t)")
plt.ylabel("Value (y)")
plt.legend()
plt.show()

print("the eccentricity is :", (np.max(y_fit)-np.min(y_fit))/(np.max(y_fit)+np.min(y_fit)),"rmax",np.argmax(y_fit),np.max(y_fit),"rmin",np.argmin(y_fit), np.min(y_fit))
```

This Python code uses a sinusoidal model to fit the apparent radii data as a function of time. The explanation of common details such as data handling, libraries, and the purpose of fitting a model to observational data is provided in the description of the Keplerian model above.

1. **Initial Guesses for Parameters:** The initial guesses for the parameters are chosen based on the data:
   - A_{guess}: Half the difference between the maximum and minimum apparent radii in the dataset.
   - w_{guess}: Estimated angular frequency, calculated using the time range of the dataset.
   - c_{guess}: Set to zero as an initial estimate.
   - C_{guess}: Set to an approximate mean value of the apparent radii.
2. **Fitting the Model**

   The `scipy.optimize.curve_fit` function is used to fit the sinusoidal model to the data. The function returns:
   - Optimized values of the parameters A, w, c, and C.
   - Uncertainties in the fitted parameters, derived from the covariance matrix.
3. **Calculation of Eccentricity** The eccentricity of the Moon's orbit is determined using the maximum and minimum values of the fitted curve:

$$e = \frac{r_{\max} - r_{\min}}{r_{\max} + r_{\min}}$$

   where $r_{\max}$ and $r_{\min}$ represent the maximum and minimum apparent radii, respectively.
4. **Output and Visualization**
   - The fitted parameters (A, w, c, C) and their uncertainties are printed.
   - The time period (T) of the periodic variation is computed and displayed in both hours and days using $T = \frac{2\pi}{w}$.
   - A plot is generated showing the observed data points and the fitted sine curve.
   - The calculated eccentricity, along with the indices and values of $r_{\max}$ and $r_{\min}$, is displayed.
5. **Key Difference from the Keplerian Model** Unlike the Keplerian model,

which explicitly incorporates the elliptical orbital mechanics of the Moon, the sinusoidal model provides a simpler, periodic approximation. This approach is suitable for identifying the periodic behavior and estimating orbital eccentricity through the maxima and minima of the fitted curve.

For additional details on the data and libraries used, refer to the description of the Keplerian orbital model above.

6.4 Judging image quality

Here, a good image mainly refers to an image with sharp moon edges(with less spread of intensity at the edges).

Figure 10: Good quality image(with sharp edges)

Figure 11: Bad quality image(with much spreaded intensity at the edges)

References

[1]https://nssdc.gsfc.nasa.gov/planetary/factsheet/moonfact.html