Assignment 1

Student 6463360:

Department of Physics, University of Surrey, Guildford GU2 7XH, United Kingdom

Three tasks have been presented for solving using programmatical techniques in Python and image analysis using DS9. The first of these tasks was an examination of the spherical Bessel functions. Included is a discussion of their use and importance in physics, followed by the first 3 spherical Bessel functions being solved and plotted in Python. The issues encountered in writing the programme to preform these calculations have been discussed and resolved. The second task was to evaluate the mass and density of the Andromeda galaxy. This was done by making assumptions of the nature of the universe and what this means for the shape of dark matter halos surrounding galaxies. A comparison of this density and the critical density of the universe based on the assumptions has been made. An RGB image of Andromeda has been evaluated and 2 satellite galaxies were located for further discussion. The final task used data that was provided from the Yale La Silla-QUEST Survey. This data went was reduced using correction techniques in Python and DS9. The resulting corrected images were saved to fits files by the programme. These files were then analysed using DS9 to determine the locations of any moving objects. Neptune was the main focus in the images and three moons were found and highlighted. Python was an effective programming tool to reduce the fits images. This has been discussed. DS9 was a capable software to use in the analysis of the produced fits images. The ability to adjust the scaling of the images and blink between them was invaluable.

Visualisation of Spherical Bessel Functions

Introduction

In 1817 Friedrich Wilhelm Bessel published his work on the problem of determining the motion of three objects caught in each other's gravity.[1] In this paper he used Bessel functions in the calculation of the motion of 3 bodies caught in each other's gravity. In 1824 these functions were refined in his study of planetary perturbations.[2] By considering the direct and the indirect perturbations separately, the Bessel functions arose as coefficients.

Bessel functions can be used in the calculation of perturbations of planetary orbits in astronomy. Perturbation is the movement or misshapen of a star by the gravitational effects of its satellites. This can be observed and used as an indicator that a planet is present.

In this section the task was to solve the first 3 spherical Bessel functions in a Python programme and create a visual representation of the results.

Methods

A programme was written in Python that solved the first 3 spherical Bessel functions. These functions are:

$$J_0(x) = \frac{\sin(x)}{x} \qquad (eq. 1)$$

$$J_1(x) = \frac{\sin(x)}{x^2} - \frac{\cos(x)}{x} \qquad (eq. 2)$$

$$J_2(x) = \left(\frac{3}{x^2} - 1\right)\frac{\sin(x)}{x} - \frac{3\cos(x)}{x^2} \qquad (eq. 3)$$

These were defined in a Python function which took for its arguments; a numpy array of x values over which to calculate the Bessel functions, and an integer which defined the order of Bessel function the user wished to calculate (0, 1 or 2 in this case).

As each Bessel function includes a division by x, it was important to programme in the behaviour of each as x tends to 0. Otherwise, Python would have misbehaved while preforming the calculation. For the first spherical Bessel function, as x tends to 0, the function tends to 1. For the next 2 orders of Bessel functions, as x tends to 0, so does the function.[3] This was implemented as an if statement within the Python function. For example, if $x = 0$, give the output array the appropriate value (1 or 0 depending on the order of function selected).

A numpy array of x values was created for the calculation. In order to gain a smooth plot of the functions the numpy linspace function was called defining 1000 data points over the x range 0 to 20. The Python function was then called 3 times to create individual arrays of values for the Bessel functions of each order.

Matplotlib.pyplot was then used to create a graph of each Bessel function array. Different colours and line styles were used to distinguish between them clearly. Various other features were used from the plotting library to improve the graph quality such as adding a black line at $J_n(x) = 0$ and labelling the axis. The black line at $J_n(x) = 0$ was plotted to better highlight the oscillation of the functions over the range of x. The graph was saved as a png file.

Results

The results are shown below in figure 1, generated by the Python programme. Each Bessel functions behaviour at $x = 0$ is shown clearly to follow the rules stated in the previous section.[3] This an indicator that the programme ran as intended.

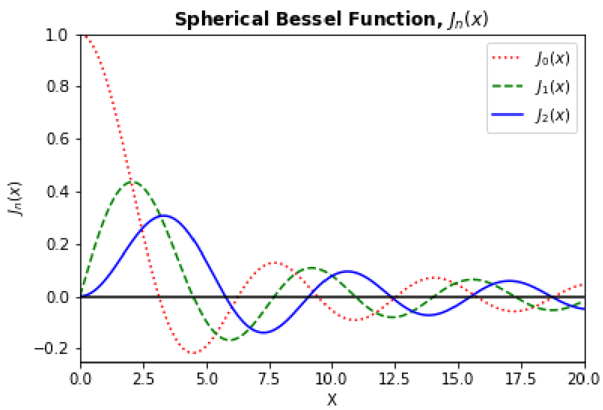

Figure 1 A plot of the first 3 spherical Bessel functions generated by the Python programme.

Discussion

As x increases it is seen that the amplitude of each Bessel function falls to levels comparable to each other. This “synchronisation” of oscillation amplitudes occurs between $x = 5$ and $x = 7.5$. The amplitude of these oscillations are reducing as x increases. This could imply that as x tends towards infinity, the value of $J_n(x)$ would tend towards 0. A larger range of x values would need to be evaluated to determine this without the use of algebraic method, however.

Conclusions

The Bessel functions are a useful tool in astrophysics. By calculating them using programmatical techniques, their nature can be determined and the way they behave under certain conditions can be understood.

The Python programme that was written was able to calculate and plot the first 3 spherical Bessel functions accurately and with ease. The simplicity of the writing of the programme makes this an attractive option for anyone intending to work with Bessel functions.

Evaluation of the Andromeda Galaxy

Introduction

This task was centred around the Andromeda galaxy. Named from Greek mythology, Andromeda is our closest galactical neighbour.[4] The galaxy has 2 satellite galaxies, Messier 110 and Messier 32. These will be discussed later. Andromeda has been used heavily for research into galaxy evolution and as a laboratory to observe theories in astronomy in action.[5] Our own Milky Way and the Andromeda Galaxy appear to be on a collision course. This collision is expected to occur within our Sun’s lifetime.[6]

In this section the mass and density of the Andromeda Galaxy has been calculated using programmatical techniques with Python. Assumptions were made about the nature of the universe. This led to using the spherical model of dark matter halos surrounding galaxies. A comparison of the critical density of the universe - considering the previous assumptions – and a comparison of this density with that of Andromeda was made.

An RGB fits image of the Andromeda galaxy has been evaluated using DS9. Scaling techniques have been utilised to produce a clear image of the galaxy. The locations of the dwarf satellite galaxies, Messier 110 and Messier 32 have been determined.

Methods

Calculations of Andromeda’s properties using Python

When calculating the enclosed mass of the Andromeda galaxy one needs only consider the radius and velocity of a point at the outermost rim of its spiral arm. The outermost rim of the galaxy is at a radius of $R \sim 20kpc$ and orbits the centre at a velocity of $V = 226kms^{-1}$.

To perform these calculations, it was important to make assumptions about the nature of the universe. Working under the assumption that the universe has an inherently flat cosmology, the equation for the velocity of an orbiting body can be used.[7]

$$V = \sqrt{\frac{GM(<R)}{R}} \qquad (eq. 4)$$

This re-arranges to give

$$M(<R) = \frac{V^2R}{G} \qquad (eq. 5)$$

Equation 5 has been implemented in a Python programme to calculate the enclosed mass of the Andromeda galaxy.

The density was calculated by assuming the shape of the total potential of the galaxy to be spherical. This was a sensible assumption. When one considers the universal cosmology to be flat, the shape of dark matter halos that surround galaxies can be modelled as a sphere.[7] It proven that a spherically symmetric body behaves as if all of its mass were concentrated in the centre.[8] Therefor the density can be calculated by using the following equation;

$$\rho_{Andromeda} = M\left(\frac{4}{3}\pi R^3\right)^{-1} \qquad (eq. 6)$$

This was implemented in the Python programme. To calculate the critical density of the universe equation 7 was used.[7]

$$\rho_c = \frac{3H_0^2}{8\pi G} \qquad (eq. 7)$$

The ratio of the density of Andromeda compared to that of the universe was calculated and output to the terminal by the Python programme.

Evaluation of Andromeda Using DS9

Three images of the Andromeda galaxy were provided, offering red, green and blue perspectives of an observation. These fits images were combined into an RGB composite file using DS9 and the scale settings were altered to provide a clear and realistic image. Using this altered image, the locations of Andromeda's satellite galaxies were found and labelled with regions. These regions were saved as a reg file. They were then applied to the RGB image which was then saved as a png file.

Results

The Python programme produced the following results for the enclosed mass and density of the Andromeda Galaxy.

$$M(<R) = 4.723 \cdot 10^{44} kg$$
$$\rho_{Andromeda} = 4.796 \cdot 10^{-25} gcm^{-3}$$

The output for the critical density of the universe was given as,

$$\rho_c = 8.346 \cdot 10^{64} gcm^{-3}$$

The ratio of the densities was given by the Python programme as,

$$\frac{\rho_{Andromeda}}{\rho_c} = 5.747 \cdot 10^{-90}$$

The results of the evaluation of the RGB composite of Andromeda galaxy is shown in figure 2. The dwarf satellite galaxies are pictured towards the north-east of the central bulge (Messier 110) and directly below the central bulge (Messier 32) in figure 2.

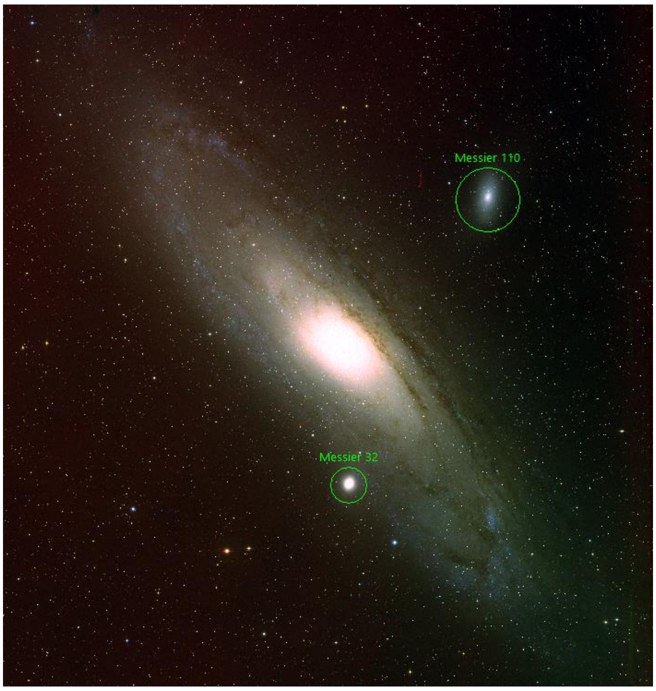

Figure 2 An RGB composite png image of the Andromeda Galaxy with the Dwarf Satellite Galaxies, Messier 110 and Messier 32, labelled.

Discussion

Andromeda

The ratio calculated is very small. This indicates that the density of Andromeda is much less than the critical density of the universe. The objects located in figure 2 are Messier's 110 and 32. These are dwarf satellite galaxies orbiting Andromeda.

Messier 110

Messier 110 orbits the Andromeda galaxy. It is classed as a dwarf elliptical galaxy. The galaxy is of particular interest as there are pockets of dust and blue stars near the centre.[9] This can be seen in figure 2. It is thought that the galaxy was once larger, featuring a halo surrounding the centre. This was then broken up when Messier 110 made a close pass to Andromeda.[10]

Messier 32

Messier 32 is very small. With most of its stars being located close to the centre. It has a radius of 100pc.[11] Messier 32 is possibly much older than Messier 110 as it shows little sign of star formation. With no dust and being mostly home to faint red and yellow stars.[12] These red and yellow star features can be seen in figure 2.

Conclusions

The Andromeda galaxy is a stunning example of a spiral galaxy such as our own. By studying it and its satellite galaxies we can draw new conclusions about possible processes that occur during galaxy formation and the effects it would have on galaxies in close proximity.

The Python programme was able to calculate the enclosed mass and density of Andromeda. The universal critical density task was performed with ease. This makes Python a useful language when making calculations with large numbers. Converting the values into different units was also a simple task using Python.

DS9 was useful for analysing the RGB image of the Andromeda galaxy. The software was capable of sufficiently altering the image scaling for viewing and identifying the satellite galaxies. Some key features of Andromeda such as the spiral and central bulge can be clearly seen. Features of the satellite galaxies such as their colour can be seen too and agree with the way literature implies they should look.[9,12]

Identifying Solar System Objects in Yale La Silla-QUEST Data

Introduction

The La Silla-QUEST Variability Survey is a project with the goal of studying astronomical objects such as planets, stars and supernova. The survey began in 2009 and is expected to cover $\frac{1}{3}$ of the night sky. The main focuses of the survey is the Edgeworth-Kuiper belt.[13]

First postulated by Edgeworth in 1943, the Edgeworth-Kuiper belt is a ring in our solar system that extends from Neptune (at 30AU from the Sun) to approximately 50 AU from the Sun.[14] The belt is populated with approximately $4 - 7 \cdot 10^4$ objects greater than 50km in radius.[15]

For this task, data was provided by Professor Marla Geha from the Yale La Silla-QUEST Survey. Dark, flat and science fits images were given in their raw form. The flat and science images had an overscan region which was used to correct the bias of the images. A reduction of the images was conducted using image correction techniques in Python. This led to three fully corrected science images being saved by the programme. An analysis of the images was completed using DS9 which was used to highlight the movement of three objects that orbit Neptune.

Methods

In order to reduce the raw images, the overscan region was first found. This was done by examining the science and flat images in DS9 and noting the coordinate values where the overscan lay. This was from 600 to 640 along the x axis. Once this was done, a bias correction function was written in Python. The science and flat images were read into the programme along with the information contained in their headers. The function then calculated the average pixel value for each row of overscan and subtracted that value from the corresponding row in the science and flat images. This overscan region was then trimmed from these bias corrected images.

The dark regions were resized to match the newly trimmed science and flat images. The headers were read to determine the exposure times of the dark images. The flat and science exposure times were also. This information made it possible to determine which dark images corresponded to either the flat or science images. Once this was done, the dark images were subtracted from their corresponding science and flat images.

A master flat file was created (a numpy array containing all the flats images data) and normalised by a division by the median of itself. This was saved as a normalised flat fits with the original flats headers. To check the average value of the flats pixels, a histogram was plotted and displayed by the programme. This is shown in figure 3.

The science images were next flattened. This was done by a division by the normalised flats array. The next step for the science images was to conduct a sky correction. This is the process of subtracting the average sky pixel value from the images. The average sky value was calculated by locating the maximum and minimum pixel across all 3 science images. The mode was then found and subtracted from each science image. These final corrected images were then automatically saved by the programme with their original headers.

An analysis of these corrected science images was conducted in DS9 to determine the shift difference between the second and third images relative to the first. This was done be mapping the difference in positions of motionless, unsaturated stars in each frame. Once the shift values had been calculated for each, these were reversed in the Python programme to ensure all 3 of the corrected science images resided on a common pixel frame. The second and third images, once shifted to the common pixel frame, were saved automatically by the programme. These final corrected and shifted images were then combined into one image and saved by the programme.

These 3 final images were evaluated in DS9 to determine the locations of Neptune’s moons. The images were blinked between to determine which objects displayed movement. These moving objects were then highlighted with regions and colour coded. The region files highlighting these positions were then saved and each corrected image was saved with the moons highlighted as png.

Results

The automatically plotted histogram of the normalised flats images pixel value shows the highest frequency at a value of 1.0.

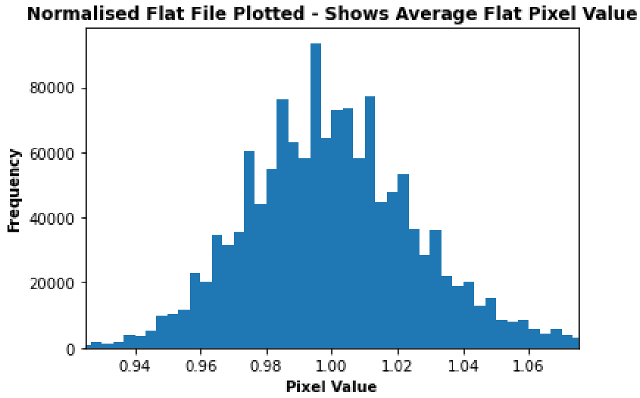

Figure 3 The normalised flats pixel values plotted by the Python programme. The average value is seen to be 1.

The fully reduced science images with the moons highlighted are shown in figures 4-5. They are named by the final 3 numbers of the original fits file. These are 901s, 246s and 543s relating to image 1, 2 and 3 respectively. In this order they progress forwards in time.

A better view of this is from the combined image. This allows the moons positions at each time step to be seen in one image and each moon can be examined separately. This is shown in figure 5 with a cropped version to show each moon in figures 6a, 6b and 6c.

Each image has been altered by adjusting the scale such that the stars are visible for inspection. This results in some images appearing paler than others.

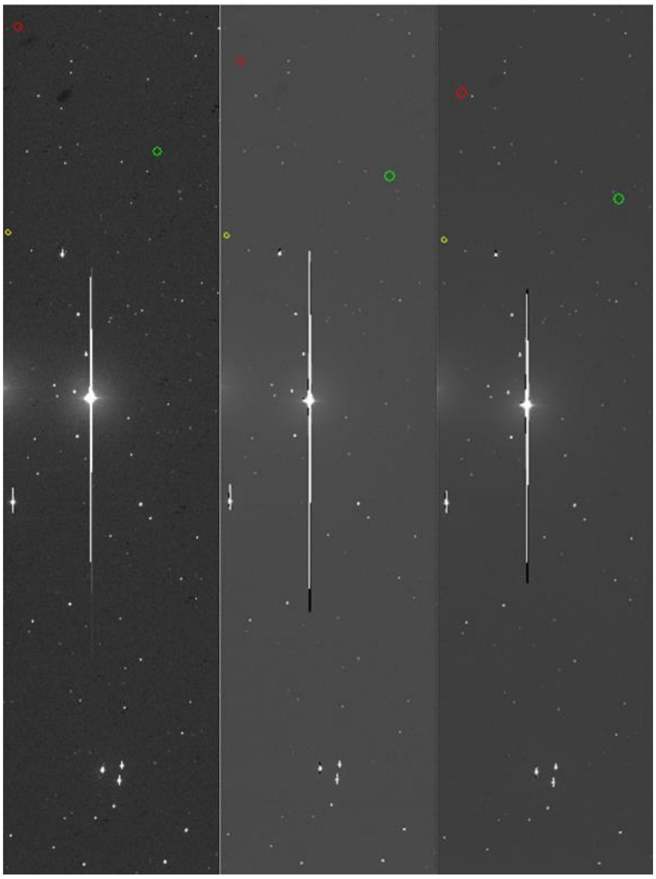

Figure 4 Fully corrected science images 901s, 246s and 543s shifted to a common pixel frame. The moons can be seen highlighted and moving in the progression of time (left to right).

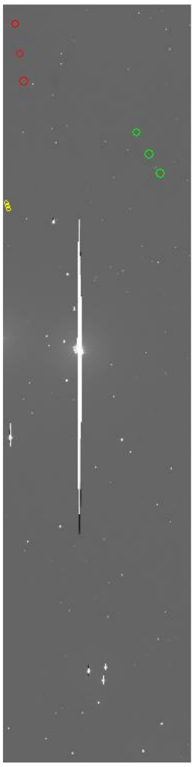

Figure 5 All 3 science images combined to more clearly show the movement of the moons over time. Neptune appears to be blurred as it too is moving over time.

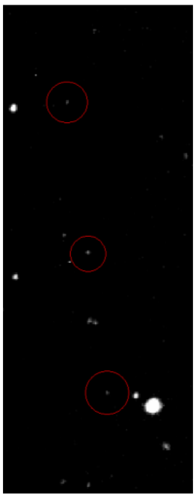

Figure 6a A combination of the 3, sky corrected and shifted images. This shows the movement of the moon at the top left of figure 5.

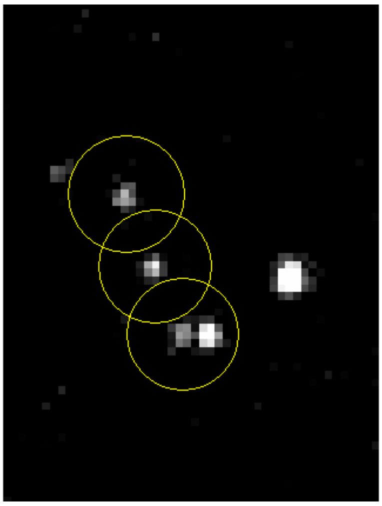

Figure 6b A combination of the 3, sky corrected and shifted images. This shows the movement of the moon to the left of Neptune in figure 5.

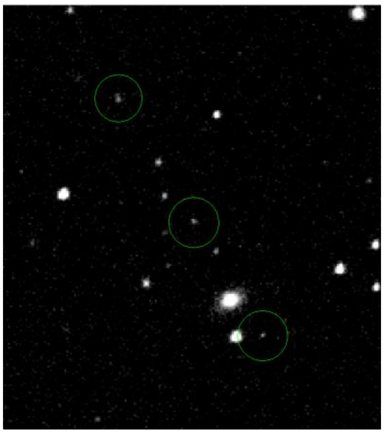

Figure 6c A combination of the 3, sky corrected and shifted images. This shows the movement of the moon to the right of Neptune in figure 5.

Discussion

In figure 4 the progression of time from one image to the next can be seen. The movement of Neptune to the right indicates this. The highlighted regions show each moon from one frame to the next. In figure 4 it is clear that these are the moons as no other objects can be seen moving (other than Neptune of course).

The "red" moon in figure 6a is located towards the north-west of Neptune in figure 4 and 5. Its movement is towards the planet and appears to move large distances over the time period of the frames. This can be an indicator of a few things. It could be a very fast-moving satellite. Orbiting Neptune quickly. It could also be moving north to south (from our perspective). This means that rather than moving towards or away from us as it would when it is above or below Neptune (from our perspective), it is either at the far side of Neptune or the near side. Given that it is so faint, this could be on the far side of the planet.

Figure 6b shows the "yellow" moon. It is located to the left of Neptune in figure 4 and 5. It seems to be brighter than the other moons. This implies that it is closer to us than the "red" or "green" moons. It moves very little from one frame to the other which could be due to it either moving towards or away from us. Examining figure 6b, the moon appears to become fainter over time. This could mean that it is moving away from us, heading on course to go behind Neptune. However, there is another object in the path of this satellite which is much brighter. This could be causing the satellite to appear fainter.

The "green" moon in figure 6c displays large levels of movement from one frame to the next. As was the case with the "red" moon, this could mean that it is moving fast or is on the vertical section of the orbit (from our perspective). As with the "yellow" moon, its light appears to be interfered with by other bright objects in the image.

Conclusions

The Python programme was able to efficiently reduce the fits images for inspection. It produced clear pictures that were able to be used to locate Neptune's moons easily. The programme runs quickly on some computers but very slowly on others. This implies that it is efficient and that some computers are able to process the calculations more efficiently than others. DS9 was a useful tool for flickering between the images to distinguish between stationary and moving objects.

The methods of reduction described above are efficient in producing clear images of celestial objects. These can be used to reduce more than just the data provided. By applying these methods to other astronomical data, one could study many things, such as the Andromeda galaxy.

By using the methods described in this report and more exposures over a longer period of time, the full orbits of Neptune's moons can effectively be mapped. This could lead to the axis and speeds of the orbits being determined.

References

1. **Bessel, Friedrich Wilhelm.** Analytische Auflosung der Kepler'schen Aufgabe. *Abhandlungen der Berliner Akademie.* 1817, pp. 49-55.

2. **Fricke, Walter.** Friedrich Wilhelm Bessel . [Online] Encyclopedia.com, 2008. [Cited: 11 22, 2020.] https://mathshistory.st-andrews.ac.uk/DSB/Bessel.pdf.

3. **University of Waterloo.** Bessel Functions of the First and Second Kind. [Online] [Cited: 11 22, 2020.] http://www.mhtlab.uwaterloo.ca/courses/me755/web_chap4.pdf.

4. **Schultx, David.** *The Andromeda Galaxy and the Rise of Modern Astronomy.* New York : Springer Science and Business Media, 2012.

5. **Hodge, Paul W.** *Atlas of the Andromeda Galaxy.* Seattle and London : University of Washington Press, 1981.

6. **T. J. Cox, Abraham Loeb.** *The Collision Between the Milky Way and Andromeda.* Cambridge, USA : Harvard-Smithsonian Center for Astrophysics, 2008.

7. *A Universak Density Profile from Hierarchial Clustering.* **Narvarro, Julio F., Frenk, Carlos S. and White, Simon D. M.** s.l. : The American Astronimical Society, 1997, The Astrophysical Journal, Vol. 490, pp. 493-508.

8. **Newton, Isaac.** *Philosophiae Naturalis Principia Mathematica.* 1687.

9. *Planetary Nebulae in Local Group Galaxies. I. Identifications in NGC 185, NGC 205, and NGC 221.* **Ford, Holland C., Jenner, David C. and Epps, Harland W.** s.l. : The Amer, 1973, The Astrophysical Journal, Vol. 183, pp. L73-L75.

10. *The Potential role of NGC 205 in Generating Andromeda's Vast Thin Corotating Plane of Satallite Galaxies.* **Angus, Garry W., et al.** 2016, Monthly Notices of the Royal Astronomical Society, Vol. 462, pp. 3221-3242.

11. *Surface Photometry of Six Local Group Galaxies.* **Kent, Stephen M.** 2, 1987, The Astrophysical Journal, Vol. 94, pp. 306-314.

12. **Kepple, George Robert and Sanner, Glen W.** *The Night Sky Observer's Guide. Vol. 1.* s.l. : William-Bell, 1998.

13. **Yale University.** The La Silla-QUEST Vairability Survey. [Online] Yale University. [Cited: 11 22, 2020.] https://hep.yale.edu/research/la-silla-quest-variablity-survey.

14. ***Edgeworth, Kenneth E.*** *54, 1943, British Astronomical Association, p. 181.*

15. Collisional Erosion in the Primordial Edgeorth-Kuiper Belt and the Generation of the 30-50 AU Kuiper Gap. ***Stern, S. Alan and Colwell, Joshua E.*** *1997, The Astrophysical Journal, pp. 879-882.*