Three-body Simulation of the Orbits of the Sun, Earth, and Moon with Variable Relative Masses.

Matthew Parker
Department of Chemistry, University of Bristol.
(Dated: December 12, 2024)

This report explains and analyses a model used to simulate a 3-body simulation of the orbits of the Sun, Earth, and Moon in which the relatives masses of the Sun and Moon could be varied. Under standard conditions, the eccentricity of Earth's orbit around the Sun was calculated as 0.0011 which is significantly less than the accepted value of 0.017. However, the simulation of the Moon around the Earth produced an accurate prediction for the time period of the orbit of 29.7 days - very similar to the true value of 29.5 days. The discrepancies are likely due to the simplicity of the simulation, as it does not account for forces between any other celestial bodies in the Solar System. The report also explores how these values for the eccentricity and time period of the orbits vary as the relative mass of the Sun and Moon are varied from 0.7 to 1.3 solar masses and 1 to 100 lunar masses respectively, with the Earth-Moon system becoming a binary system with the Moon's mass set to 100 lunar masses. These variables allow exploration of the effect of mass on these celestial systems, which can potentially be applied to determining the habitability of exoplanets, in particular in relation to the habitability zone of stars.

INTRODUCTION

N-body simulations are used across science to model a variety of different things. The most common use is in astrophysics for simulations of the formation and evolvement of galaxies and interactions between multiple celestial bodies. [1] However, n-body simulations are also used in chemistry for example to simulate growth of materials at the atomic level, and modelling protein-folding. [2] In these models, the electrostatic and Van der Waal forces between particles are the forces continuously calculated and updated rather than the gravitational forces used in models of celestial bodies. [3] [4]

Small n-body simulations are useful for simulating interactions between stars and planets and determining the paths of their orbits.

As more bodies are introduced to n-body simulations, the models become increasingly complex and computationally intensive. This is because the number of interactions increases exponentially. The all-pairs method is the main method used for low numbers of bodies, as it involves calculating every interaction between bodies, which is required to maintain high accuracy for a simulation. However, this method has order O(n^2) which means it is inefficient and computationally intensive for many-body simulations. [3].

For larger simulations, other algorithms such as the Barnes-Hut algorithm are used to improve the computational efficiency of the model while maintaining accuracy, with this algorithm reducing the computational demands to order O(n log$n$). [5]

This report analyses the results of a corresponding piece of code which can be used to teach a basic understanding of n-body simulations, in particular the movement of a system under gravitational forces. The code produces a simple 3-body simulation, recording the positions of the Sun, Earth, and Moon over a time period of three years. The all-pairs method is required in this situation because of the low number of bodies, as all interactions must be precisely calculated.

The code has the option to change the relative mass of the Sun and Moon. This is used to determine the influence the mass of these celestial bodies have on their orbits, in particular exploring how the relative masses affect the eccentricity of the orbit of the Earth round the Sun and the time period of the Moon's orbit round the Earth.

This can also be used to explore the habitability of exoplanets - a term referring to any planet beyond our solar system. The option within the provided code to change the relative mass of the Sun and the Moon allows exploration of a range of scenarios, enabling analysis of the conditions that influence the orbits of planets.

Some exoplanets garner particular interest in astrophysics due to the possibility they may have suitable conditions for extraterrestrial life. The most important condition for a planet to contain life is the presence of water. This condition directly depends on the pressure and temperature of a planet, with the temperature of a planet having to reside between 288 K and 399 K in order for the water to not freeze or evaporate. [6] In turn, the temperature of a planet partially depends on the distance of the planet from a star, which is why simulating orbits is important. If initial conditions (position, velocity, mass) of each body in a system are determined, it can be seen whether an exoplanet remains within a star's habitable zone – the distance from stars which are neither too hot nor too cold to destroy life.

To calculate the habitable zone of a star, the luminosity of the star must be known. [7] This project does not account for the luminosity of a star, and therefore cannot be used to predict the habitability zone of a star. However, this is a simulation involving the Sun, for which the habitable zone limits are known, with the inner edge at 1.481x10^{11} m and the outer edge at and 2.543x10^{11} m. [8]

The characteristics of the habitable zone are not only dependent on the luminosity of a star but also influenced by the mass of the star. Increasing the mass of a star causes both the inner and outer edges of the habitable zone to move further away from the star, and the width of zone to increase. [9] The mass of a star also affects the eccentricity of the planets orbiting it – as the corresponding code explores.

SIMULATION DETAILS

C++ was used in calculating the positions of each celestial body because it offers significantly faster execution speeds than Python, due it being a compiled language and therefore only needing to be compiled once. Its efficiency makes it better at handling large numbers of calculations, as required in computationally intensive tasks like n-body simulations. When the C++ script is run, the calculated positions are written to a CSV file, which is then saved under a filename provided as a command line argument.

The C++ code was also parallelised to improve the efficiency of the simulation. This only has a minimal effect on the total time to run the simulation because the main cost comes from the large number of iterations over which the simulation is run. This cannot be parallelised because each iteration is dependent on the last iteration. However, the parallelisation does slightly improve the overall efficiency of the script. It is used to increase the efficiency of the calculations for the acceleration, positions, and velocities for each of the planets as these calculations can be run simultaneously. This would become more important if the number of bodies in the simulation were increased.

In the main Jupyter notebook, the corresponding C++ code is called through one of the functions as a subprocess. This is currently written to run using WSL. If WSL is not downloaded on a user's machine, the path should be adapted to match their system. The main Jupyter notebook also calls another external file ("my_animate_orbits.py"), with the option to call a similar script "animate_orbits.py". These scripts must be saved in the same folder as the Jupyter notebook, or have their paths manually updated within the Jupyter notebook. The running of these scripts is the slowest part of the program, as the positions of the bodies are plotted as an animation and saved to a gif. The number of frames can be varied to balance computationally efficiency and accuracy of the plots.

The rest of the Jupyter notebook is used for analysis of the data saved in the CSV files produced in the C++ script.

METHOD

For the model written in the C++ script, the Verlet leap-frog method was used - a second-order numerical technique which is a more complex but accurate method than the common forward Euler method. The forward Euler method is a first order method that can produce unstable results, because it does not guarantee energy conservation within a system. [2] For example, if applied to gravitational systems, the orbits can spiral inwards or outwards. However, in the Verlet method, energy is conserved, which allows it to be used to produce plausible results throughout long simulations.

The Verlet leap-frog process involves finding the initial force and acceleration acting on each body under some set of initial conditions.

The gravitational force vector acting on a body from another celestial body is calculated using equation 1:

$$\mathbf{F}_{kl}(t) = G\frac{m_k m_l}{\mid \mathbf{r}_{kl}(t) \mid^2}\hat{\mathbf{r}}_{kl}(t) \tag{1}$$

where G is the gravitational constant (6.674x10^{-11} $\mathrm{m}^{-3}\,\mathrm{kg}^{-1}\,\mathrm{s}^{-2}$) , m_k and m_l are the masses of the two bodies in kg, $\mathbf{r}_{kl}(t)$ is the position vector between the bodies at time t, and $\hat{\mathbf{r}}_{kl}(t)$ is the unit position vector, of length 1, pointing in the direction between the two bodies at time t.

The total force vector acting on a body is the sum of the individual force vectors from each other celestial body in the system. The x- and y-components of the force vector can be calculated separately because they are independent.

This total force vector can then be used to calculate the acceleration, $a_k(t)$ of a body, calculated using equation 2:

$$a_k(t) = \frac{\mathbf{F}_k(t)}{m_k} \tag{2}$$

This value for the acceleration, along with the values for the position and velocity of the body can be used to calculate the new position of the celestial body, using equation 3:

$$\mathbf{r}_k(t+\delta t) = \mathbf{r}_k(t) + \mathbf{v}_k(t)\delta t + \frac{1}{2}\mathbf{a}_k(t)\delta t^2 \tag{3}$$

where δt is the time-step of the simulation, in this case set to 3600 s (1 hour).

Before calculating the updated velocity or acceleration, the time-step is updated, as shown in equation 4:

$$t \leftarrow t + \delta t \tag{4}$$

The new acceleration is now calculated using equations 1 and 2, inputting the updated position vector calculated in equation 3.

The new velocity, $v_k(t)$ is then calculated using equation 5, which depends on both the current and previous values for the acceleration.

$$\mathbf{v}_k(t) = \mathbf{v}_k(t-\delta t) + \frac{\delta t}{2}[\mathbf{a}_k(t) + \mathbf{a}_k(t-\delta t)] \tag{5}$$

This process is repeated for each iteration, first calculating the new position using the previous acceleration, then using this new position to calculate the new acceleration. These updated values for the position and acceleration are used alongside the previous acceleration to update the value for the velocity.

Once the positions for each of the three bodies across the length of the simulation have been found, the eccentricity of the Earth's orbit around the Sun is calculated. This is done using equation 6:

$$e = \frac{r_a - r_p}{r_a + r_p} \tag{6}$$

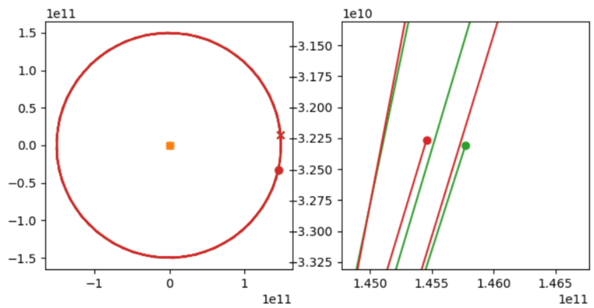

FIG. 1. Single frame of screenshot of gif animation. Left panel shows the orbit of the Earth around the Sun, and the right panel shows the orbit of Moon around the Earth.

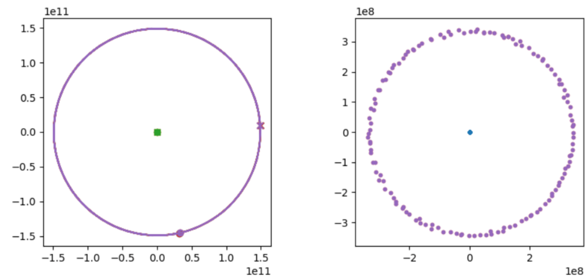

FIG. 2. Single frame of screenshot of updated version of gif animation. Left panel shows the orbit of the Earth around the Sun, and the right panel shows the relative positions of the Moon compared to the Earth.

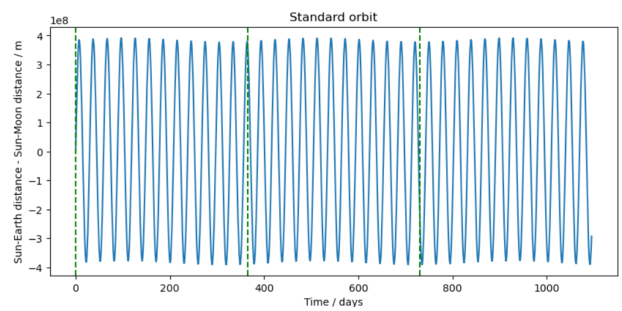

FIG. 3. Plot of the difference between Sun-Earth and Sun-Moon distances against time. The vertical green dashed lines represent a year.

where r_a is the radius of apoapsis (the furthest distance between the two bodies) and r_p is the radius of periapsis (the smallest distance between the two bodies).

The time period of the orbit of the the Moon was also found, by calculating the difference between the Sun to Moon distance and the Sun to Earth distance. The time period of the orbit was then found by calculating the average time between peaks. It is worth noting that, because it is calculating the difference between the Sun-Earth and Sun-Moon distances, this time period is not the time of one orbit of the Moon around the Earth (which is 27.3 days), but rather the time between consecutive full Moons (known as a synodic month). [10]

RESULTS/DISCUSSIONS

A single frame of the output animation gifs for the two animation python scripts are shown in figures 1 and 2. It can be seen in the left-hand side of both figures that the Earth has a very circular orbit around the Sun, as expected. The right-hand side of figure 1 shows the Moon orbiting the Earth as the Earth orbits the Sun whereas figure 2 shows all the positions of the Moon relative to the Earth. This shows that the Moon also follows a cyclical orbit around the Earth.

In this simulation, the eccentricity of the Earth around the Sun was calculated as 0.0011. This is a significant underestimate, as the actual value is estimated to be 0.017. [11] The radius at apoapsis was underestimated with a value of 149.5 million km, compared to 152.1 million km, and the radius at periapsis was overestimated with a value of 149.2 million km, compared to 147.1 million km. [11] However, for a relatively simple simulation, these values demonstrate a reasonable degree of accuracy.

Figure 3 shows the orbit of the Moon relative to the Earth. The green dashed lines mark when a year has passed within the simulation. It can be seen that the Moon orbits approximately 12 times throughout the year, with the average time period calculated as 29.7 days. This is an accurate prediction, with the true value being 29.5 days.

The discrepancies between the simulation and the true values may be due to not accounting for the gravitational effect of the other planets in the Solar System, which may provide a slight force on the Earth during its orbit. Furthermore, there is an innate inaccuracy with the calculation of the Sun's position. Figure 4 shows the orbit of the Sun throughout the simulation, with the orbit "bouncing" to larger y values without ever going below zero on the x-axis This is likely due to incorrect initial conditions, as the Sun should have a small but non-zero velocity.

Figure 5 shows the variation of the eccentricity of the Earths orbit around the Sun as the mass of the Sun was varied in the simulation from 0.7 to 1.3 solar masses in steps of 0.025. The eccentricity has a clear maximum when the solar mass is smallest, dropping to a minimum when the Sun is 1.0 solar mass, before increasing again, albeit at a slower rate than it decreased before the minimum.

Figure 6 shows a plot of the Earth's orbit around the Sun for different relative masses of the Sun, at 0.7, 1.0, and 1.3 solar masses, with the habitable zone of the Sun also plotted.

The plot suggests that the Earth is only just contained within the habitable zone. However, the model used to calculate this zone ignored the effects of water clouds on the Earth's surface which help reduce the temperature of the Earths surface by reflecting some of the light from the Sun.[8] If this were accounted for, the inner bound of the habitable zone would be pushed further in.

It can be seen from the simulations that a less massive Sun produces an orbit with a significantly larger eccentricity, with

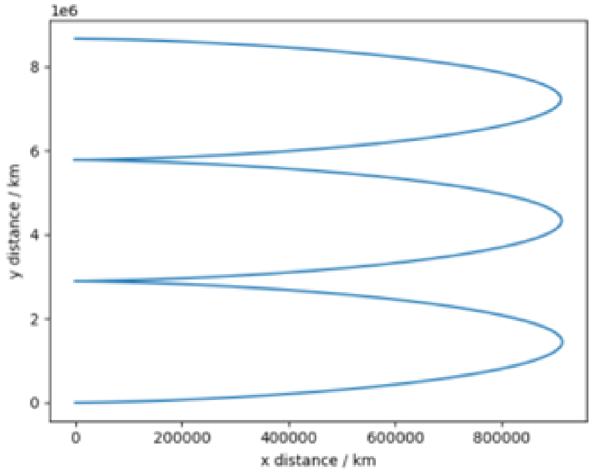

FIG. 4. Plot of the Sun's position over the course of a simulation. This is a clear inaccuracy in the method used in this simulation. However, the effect should be minimal, as the Sun has significantly greater mass then both the Earth and the Moon, so does not move much.

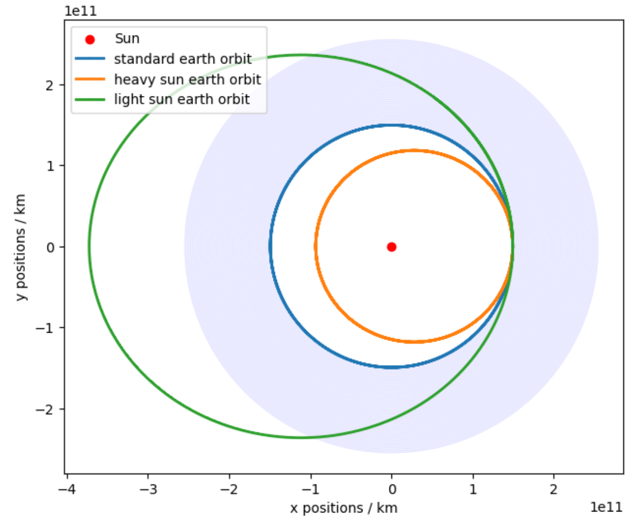

FIG. 6. Earth's different orbits around the Sun with the mass of the Sun set to 0.7, 1, and 1.3 solar masses respectively. The pale blue zone indicates the Sun's habitable zone.

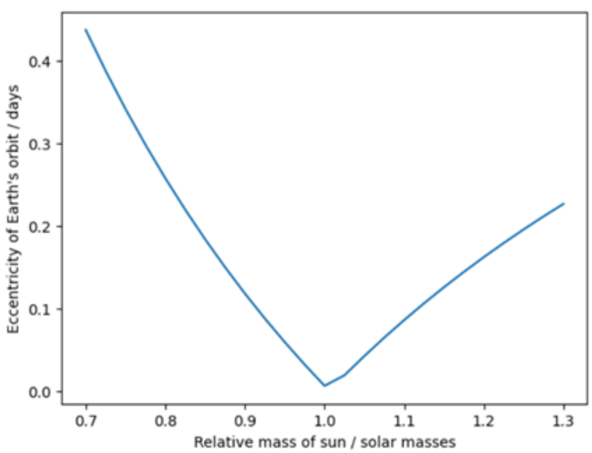

FIG. 5. Variation of the eccentricity of the Earth's orbit of the Sun as mass of the Sun increases from 0.7 to 1.3 solar masses in steps of 0.025. There is a clear minimum when the Sun is at 1.0 solar mass.

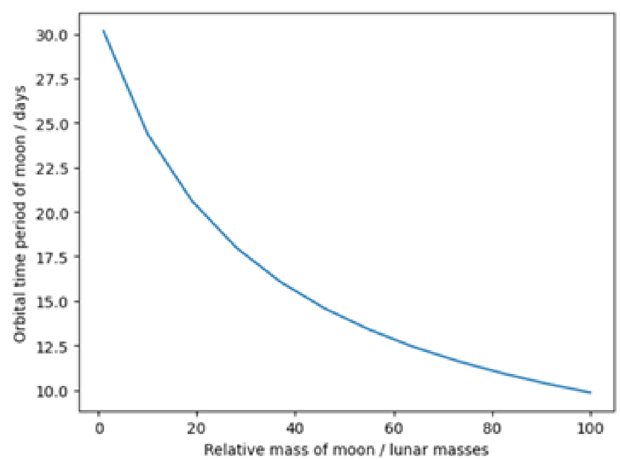

FIG. 7. Variation of the time period of the Moon's orbit of the Earth as mass of the Moon increases from 1 lunar mass to 100 lunar masses.

a value of 0.427 when the mass of the Sun is set to 0.7 solar masses. This causes the orbit to extend to much greater distances than the standard Earth orbit, reaching a maximum radius of $3.724\text{x}10^{11}$ m which is outside the Sun's habitable zone.

This suggests that if the Sun had significantly less mass, the Earth would leave the Sun's habitable zone. This would mean it would likely not have the correct conditions for life, as the surface temperature would decrease significantly while it is at much greater distances from the Sun, causing any water to freeze.

Similarly, if the Sun were more massive than it currently is, the Earth's orbit would go much closer to the Sun. The eccentricity of the orbit would again increase, with a value of 0.232 when the mass of the Sun is set to 1.3 solar masses. Furthermore, the Earth would once again leave the Sun's habitable zone for part of its orbit, as the radius would reach a minimum of $9.328\text{x}10^{10}$ m. This would result in much greater temperatures, potentially leaving the Earth with conditions more similar to those of Venus, which has an average orbit radius of $1.082\text{x}10^{11}$ m. [12].

The second major focus of the code is how the orbit of the Moon would vary if it had more mass. Figure 7 shows a comparison of the simulated time period of the Moon as it increases from a relative mass of 1 lunar mass to a relative mass of 100 lunar masses. The time period of the Moon's orbit around the Sun decreases significantly as the mass of the Moon increases, from 29.70 days to 9.81 days.

If the Moon were approximately 100 times heavier than its

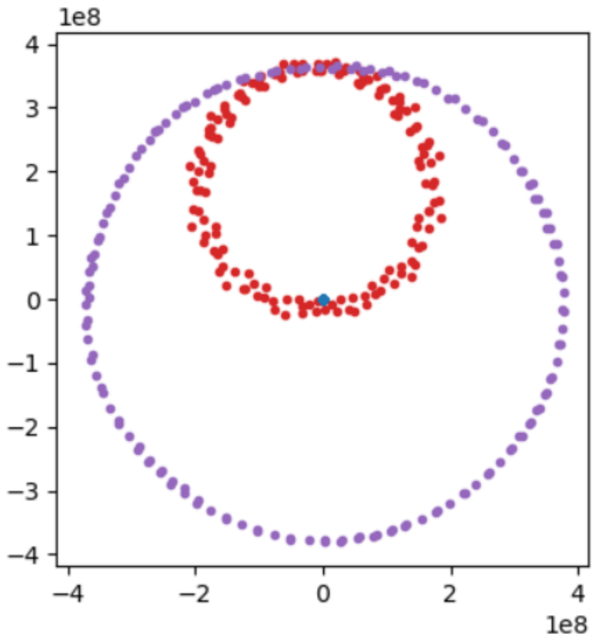

FIG. 8. Positions of the Moon relative to the Earth with the lunar mass set to 73, and the animation set to 273 frames shown in red and the standard orbit shown in purple. The more massive Moon-Earth system becomes a binary system, with the Moon orbiting very close to the Earth.

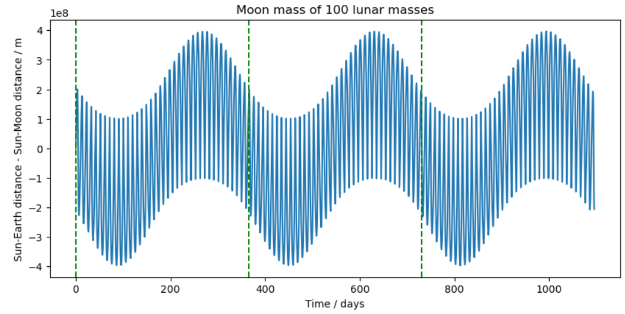

FIG. 9. Plot of the difference between Sun-Earth and Sun-Moon distances against time for a system with the relative lunar mass set to 73 and the relative solar mass as 1. The vertical green dashed lines represent a year.

current mass, it would have a similar mass to the Earth. This would change the Earth-Moon system from the Moon orbiting the Earth to a binary system, with both bodies orbiting each other. This can be visualised in figure 9, where the Moon and Earth orbit much closer together. The time period of the orbit of the Moon also decreases significantly, in this simulation to a value of 9.81 days.

Figure 8 shows a plot of the positions of the Moon relative the Earth with the mass of the Moon set to 73 lunar masses. It can be seen that the Moon swings into and away from the Earth with a much larger eccentricity. If the figure had the centre of mass of the system rather than the Earth at the centre of the plot it would be clearer to see the interaction and symmetry between the orbits of the two bodies, showing a binary system. This can also be seen in figure 9 which shows the variation in the difference between the Sun-Earth and Sun-Moon distances.

The simulation likely is not a particularly accurate depiction of a how a binary system would orbit the Sun as the same initial conditions were used as all the other simulations. This could be why the bodies approach so close together, as the initial velocity of the Moon is too low to maintain a full, more-probable orbit radii.

It is worth noting that if the animation of the Moon orbiting the Earth produced in the code does not show the Moon fully orbiting the Earth but rather staying above the Earth, as shown in figure 10 it is likely because the frame rate is set too low. This phenomenon is similar to the strobe effect, which can make helicopter blades appear as though they are stationary in a video of a helicopter flying. This effect is due to the frame rate of the camera being the same as the rotational frequency of the blades. If this occurs in the code, it is worth changing the number of frames to a higher value and re-running the production of the animation.

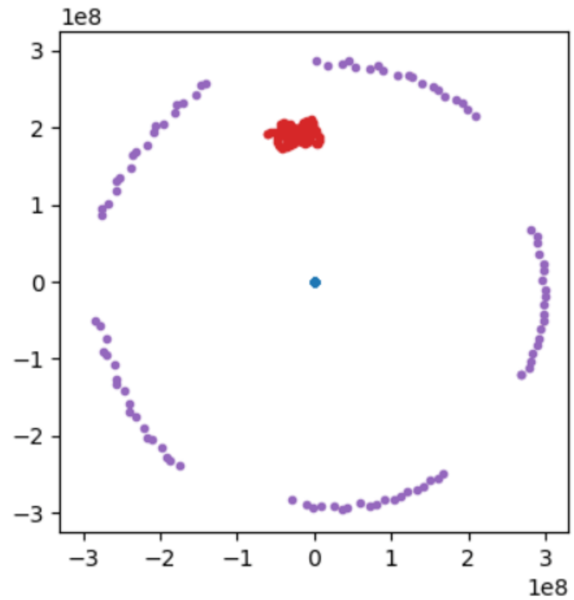

FIG. 10. Positions of the Moon relative to the Earth with the lunar mass set to 73 shown in red and the standard orbit shown in purple. This number of frames means the animation only plots a point every 10.95 days of the simulation, so does not capture an accurate path for the Moon's orbit.

Overall, these are all very basic simulations, mostly of hypothetical scenarios. When the mass parameters are varied, the same initial conditions as the standard simulation are used which is inaccurate. If the Sun had a different mass this would have affected the formation of the Solar System, meaning the Earth and moon would not have the same initial conditions they have now. The layout of the Solar System has taken billions of years to form and a slight change in any of the initial conditions of the bodies in this system would affect their current orbits. It is also worth noting that orbits change over time, which adds extra evidence that the current initial conditions would differ if the mass of the bodies were different. However, these simulation provide a useful resource to teach the basics about n-body simulations, and can provide a solid base for simulations that take more variables into account, such as the interactions with the other bodies in the Solar System. Furthermore, with further work, the code can be adapted to explore how the habitability of exoplanets may vary and determine what the habitability depends on and be used to simulate other systems of celestial bodies.

CONCLUSIONS

Overall, the simulations provide a solid base for showing how n-body simulations are constructed and work. Results for the eccentricity of the Earth round the Sun were fairly inaccurate with a value of 0.001 compared to the true value of 0.017. This major discrepancy is likely due to the exclusion of the other bodies in the Solar System, which would affect the calculated force vector on each body. Furthermore, the initial conditions for the Sun are incorrect as the velocity should be non-zero. However, the time period of the Moon around the Earth was calculated as 29.7 days which is reasonably accurate compared to the true value of 29.5 days.

The simulation also compares how these values vary with different relative masses of the Moon and Sun, with the eccentricity at a minimum under standard conditions, and the time period of the Moon's orbit decreasing as its mass increases.

In conclusion, this report introduces a range of learning topics, in particular how n-body simulations can be used to model orbits celestial bodies and model the habitability of planets and exoplanets.

REFERENCES

[1] R. Spurzem, *Journal of Computational and Applied Mathematics*, 1999, **109**, 407-432.

[2] G. Schutz, *Phase transitions and Critical Phenomena*, 2001, **19**, 1-251.

[3] L. Nyland, M.Harris, J. Prins *GPU Gems*, 2009, **3**, 677-695.

[4] A. Elcock, D. Sept, J. McMammon, *Journal of Physical Chemistry*, 2001, **105**, 1504-1518.

[5] J. Barnes, P. Hut *Nature*, 1986, **324**, 446-449.

[6] C. McKay, *PNAS*, 2014, **35**, 12628-12633.

[7] *Calculating the Habitable Zone*, Planetary Biology, `https://www.planetarybiology.com/calculating_habitable_zone.html`, (accessed 09/12/2024)

[8] R. Kopparapu et al., *Astrophysical Journal*, 2013, **765**, 131.

[9] A. Elcock, D. Sept, J. McMammon, *Astronomy and Astrophysics*, 2016, **597**.

[10] *Eclipses and the Moon's Orbit*, NASA, `https://eclipse.gsfc.nasa.gov/SEhelp/Moonorbit.html`, (accessed 10/12/2024)

[11] *Earth Fact Sheet*, NASA, `https://nssdc.gsfc.nasa.gov/planetary/factsheet/Earthfact.html`, (accessed 10/12/2024)

[12] *Venus Fact Sheet*, NASA, `https://nssdc.gsfc.nasa.gov/planetary/factsheet/venusfact.html`, (accessed 10/12/2024)