Mini-Project 1: A Circular Solution to the N-Body Problem

Mathematical Physics 1

Dhruv Aryan

March 13, 2022

PHY-1001-1

Introduction

The n-body problem has been a major fascination amongst physicists for centuries. It is the problem of predicting the motion of an n number of bodies interacting gravitationally with each other. The case of n = 1 is trivial, and we can calculate the motion in the case of n = 2 with complete precision, so the problem is mainly used to describes situations where three or more masses are involved. It was proved that there is no generalised analytical solution when three or more bodies are involved, so we instead resort to numerical approaches.

In this project, I want to analyse the motion of n bodies placed on a circle. More specifically, I aim to create initial conditions so that the bodies move on that same circular path, i.e. the bodies undergo circular motion, with the net gravitational force on each particle acting as the centripetal force, and then observe the relationship between these initial conditions and the number of bodies, if such a relationship exists.

Background

For this project, I will be using Newtonian mechanics.

Newton's law of universal gravitation

Newton's law of universal gravitation is given by the following mathematical expression:

$$F_g = \frac{Gm_1m_2}{r^2} \tag{1}$$

where G is the gravitational constant ($G \approx 6.674 \times 10^{-11}$), m_1 and m_2 are the masses of two bodies, r is the distance between their centres of masses, and F_g is the force acting on each mass due to the other's gravitational field.

Centripetal force

Let us consider a body moving in circular motion. A body must have some centripetal force acting on it in order to undergo circular motion. The centripetal force acting on it is given by

$$F_c = \frac{mv^2}{r} \tag{2}$$

where m is the mass of the body, v is its tangential velocity (i.e. the velocity of the body tangential to its circular path), r is the radius of its circular path and F_c is the centripetal force acting on the body.

Circular motion of n-bodies

If n bodies are arranged in a circle, then the net force acting on each body will be the vector sum of all the individual forces acting on it due to each of the other bodies. These individual forces will be given by equation (1). Since the arrangement of other masses are symmetric with respect to any specific mass, the direction of this net force must be radially inwards. Hence, if we want to obtain circular motion, then this net force must be the centripetal force. From equation (2), we know that this implies that each body must have some tangential velocity. Therefore, we get the following equation:

$$F_{net} = \frac{mu^2}{r} \tag{3}$$

where F_{net} is the net force acting on each body, m is the mass of each body, u is the magnitude of the tangential velocity of each body, and r is the radius of the circle on which the bodies lie, and on which they will travel. If a_{net} is the net acceleration of each body, then, on rearranging equation (3), we get

$$u = \sqrt{ra_{net}} \tag{4}$$

Methodology

For this project, I will be running simulations using the programming language **Python** and the library **VPython** for animations. All simulations are run with the centre of mass of the system as the frame of reference. This is set in the program by taking the frame about which the angular momentum remains 0 throughout the simulation. The numerical method used to solve differential equations will be the **leapfrog** method.

Leapfrog method

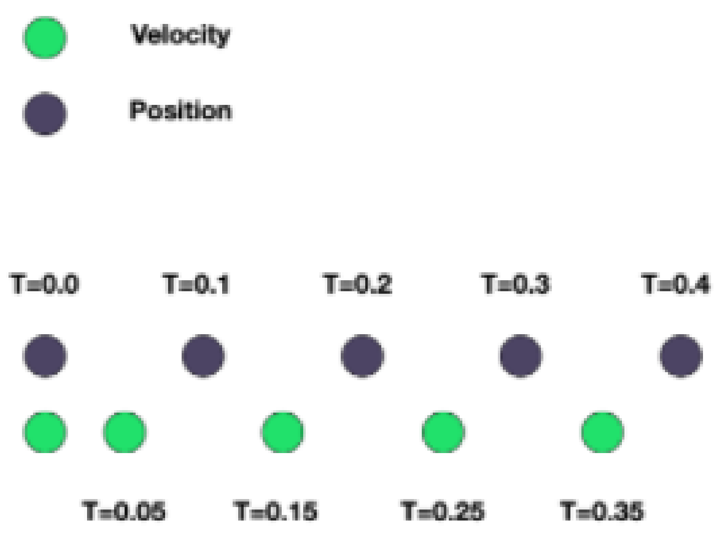

Figure 1: Diagram showing the idea of the leapfrog method

The leapfrog method takes advantage of the fact that the velocities evaluated at the midpoint of each time interval are bound to be more accurate than the velocities evaluated at the start of each time interval. The leapfrog method uses the following iterative equations:

$$x_i(t+\Delta t) = x_i(t) + v_i\left(t + \frac{\Delta t}{2}\right)\Delta t \tag{5}$$

$$v_i\left(t + \frac{\Delta t}{2}\right) = v_i\left(t - \frac{\Delta t}{2}\right) + a_i(t)\Delta t \tag{6}$$

We need one more calculation to implement this algorithm:

$$v_i\left(\frac{\Delta t}{2}\right) = v_i(0) + a_i(0)\frac{\Delta t}{2} \tag{7}$$

In the above equations, we can get a_i using equation (1).

Initial conditions

For n bodies lying on a circle in the x-y plane, we can use polar coordinates to find the coordinates of the initial position of each body. The following expression gives us the initial positions of n bodies lying on a circle in the x-y plane:

$$(x_k, y_k, z_k) = \left(r\sin\frac{2k\pi}{n}, r\cos\frac{2k\pi}{n}, 0\right) \tag{8}$$

where (x_k, y_k, z_k) are the coordinates of the initial position of the kth body and r is the radius of the circle that the bodies lie on, which I have taken as 2AU[1]. Hence, $r = 2.992 \times 10^{11}$ m.

We can then use the net acceleration defined in the computer program to give us the magnitude of the initial velocity, and the direction of the velocity of the kth body is given by

$$\mathbf{v_k} = \left(-u\cos\frac{2k\pi}{n}, u\sin\frac{2k\pi}{n}, 0\right) \tag{9}$$

where u is the magnitude of the initial velocity given by equation (4) and $\mathbf{v}$ is the vector form of the initial velocity of the kth body.

For all the simulations, the total mass of the system is kept constant at 600 times the mass of Jupiter[2]. Hence, the total mass of the system is 1.14×10^{30}.

[1] 1 AU = 1 Astronomical Unit = 1.496×10^{11} m

[2] Mass of Jupiter = 1.9×10^{27} kg.

Results and Analysis

Orbits observed

Simulating first for $n = 2$, we get the following after letting the simulation run for some time:

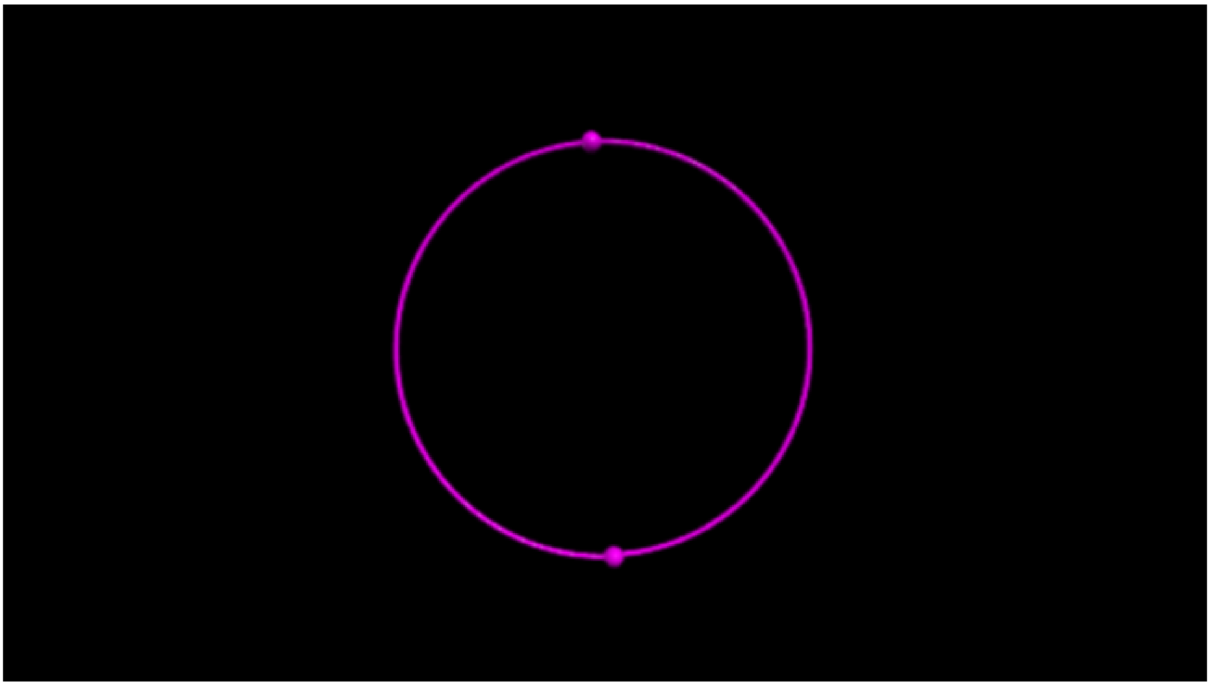

Figure 2: Orbits obtained for a 2 body system

Simulating for $n = 3$, we get the following after letting the simulation run for a small amount of time:

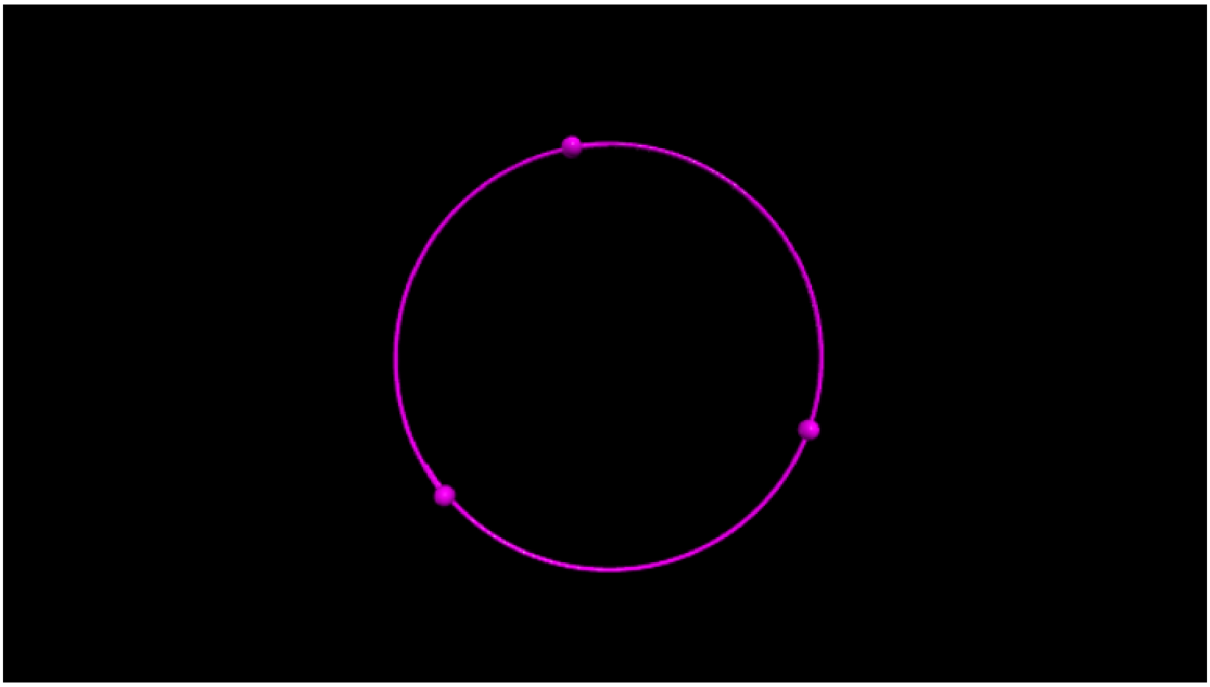

Figure 3: Orbits obtained for a 3 body system after a small amount of time

This shows that the system is largely stable, which fits with the predictions of Newtonian mechanics.

However, if we let the simulation run for a large amount of time, we get the following:

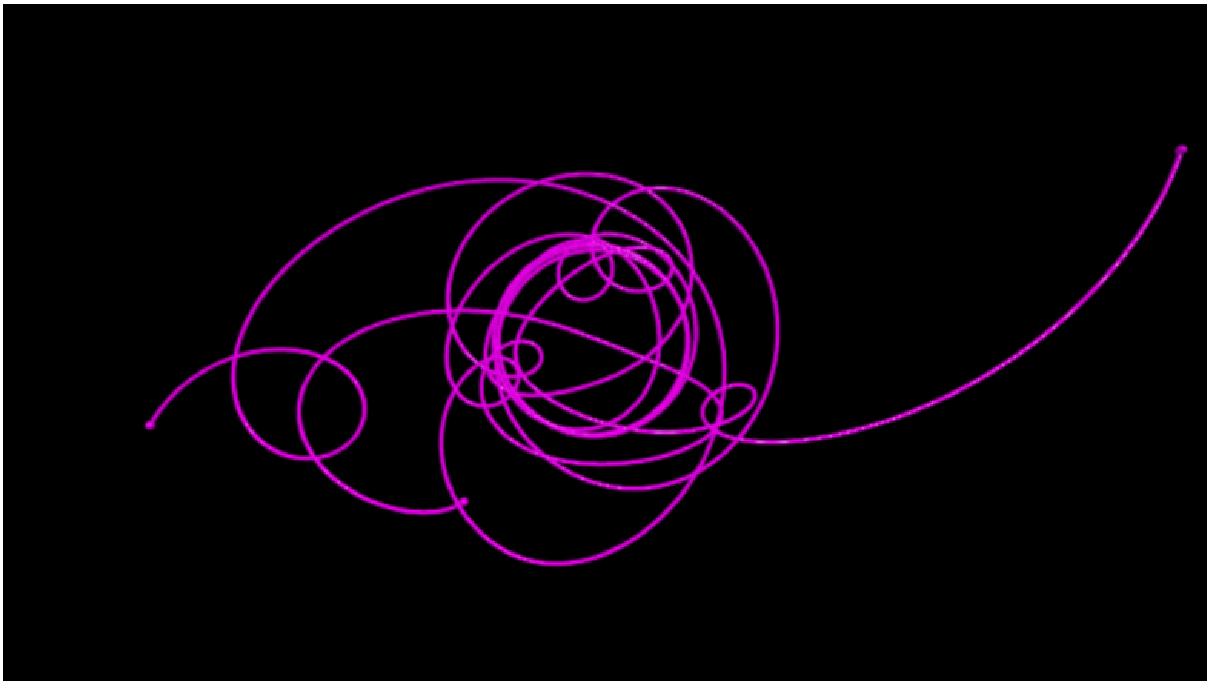

Figure 4: Orbits obtained for a 3 body system after a large amount of time

This chaotic motion mainly arises in the simulation as a result of the accumulation of errors in the program due to the leapfrog method, since we use discrete time steps.

We get similar results for all values of n.

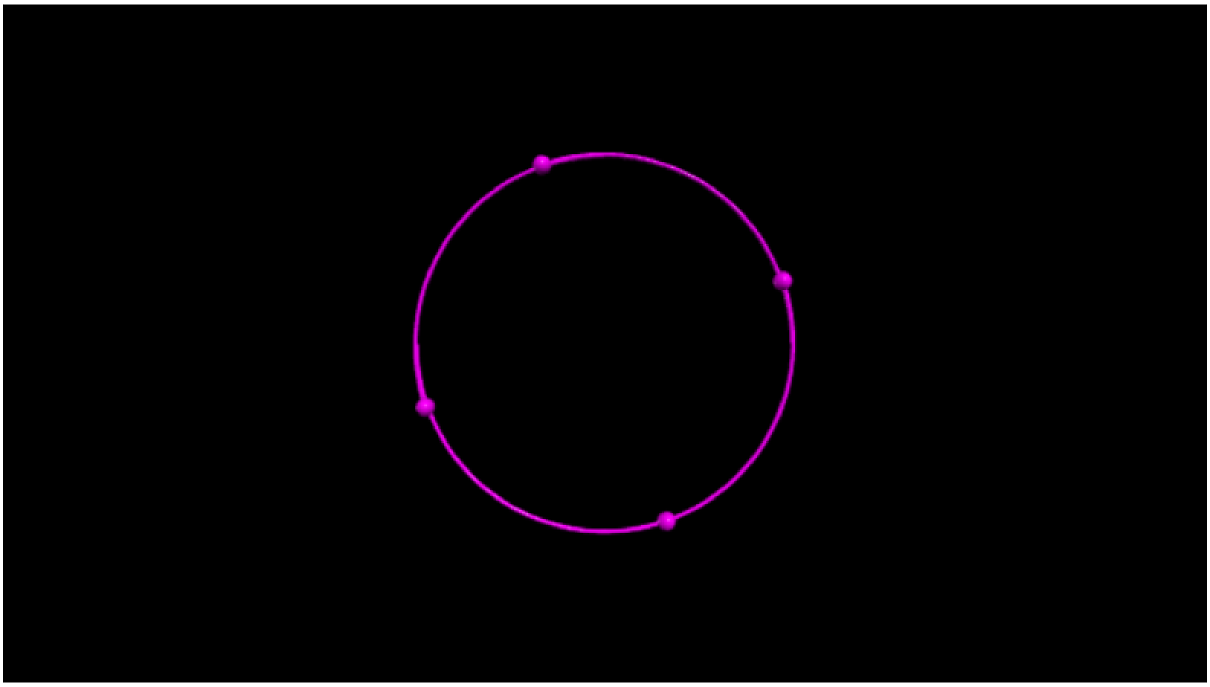

Figure 5: Orbits obtained for a 4 body system after a small amount of time

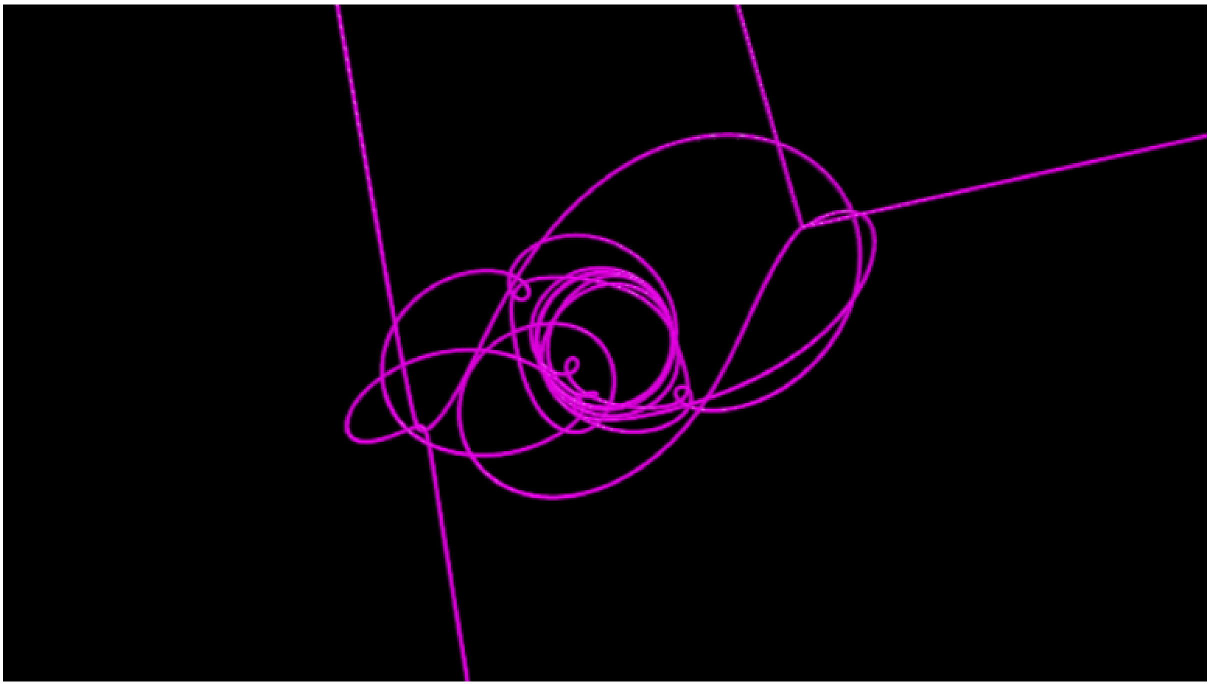

Figure 6: Orbits obtained for a 4 body system after a large amount of time

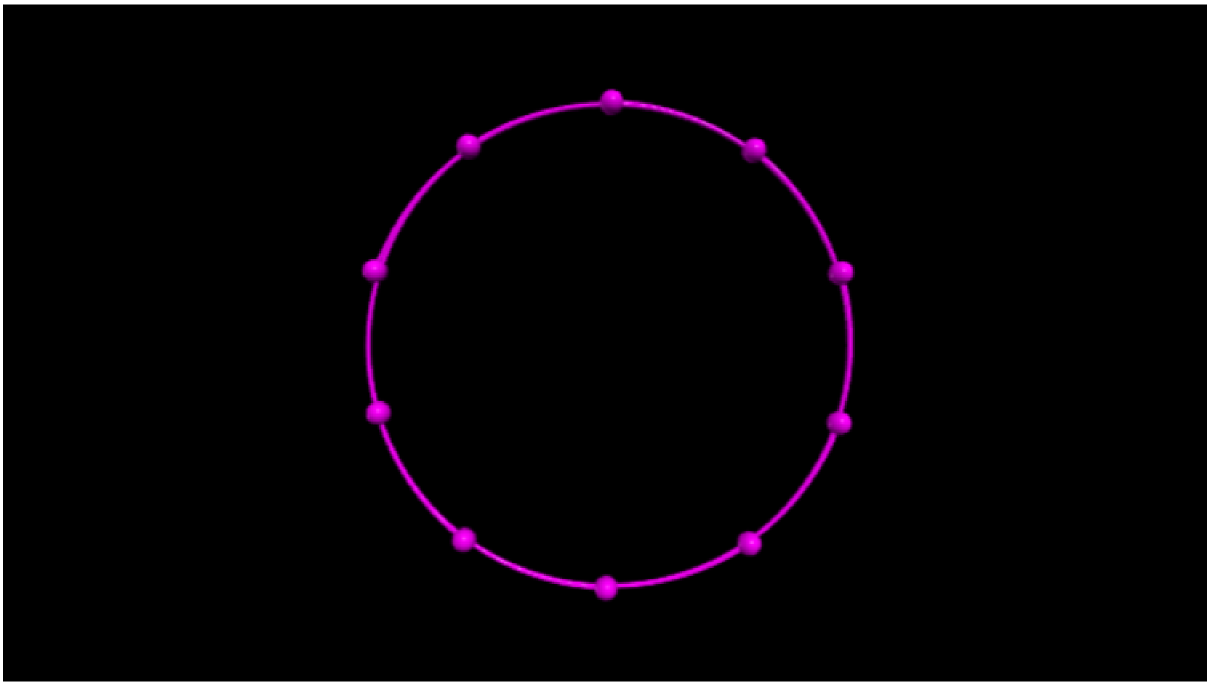

Figure 7: Orbits obtained for a 10 body system after a small amount of time

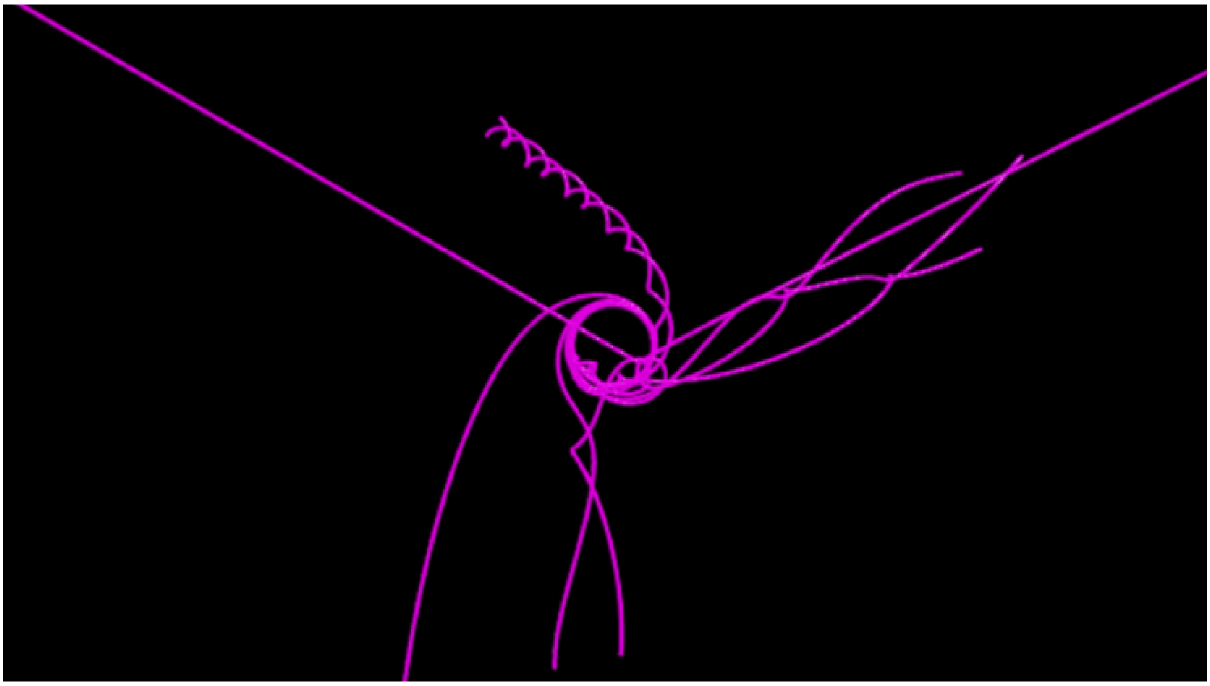

Figure 8: Orbits obtained for a 10 body system after a large amount of time

Analysis of initial velocities

We can plot the values of u that we obtain from the program in order to find out if there is a direct relationship between u and n. We get the following graph:

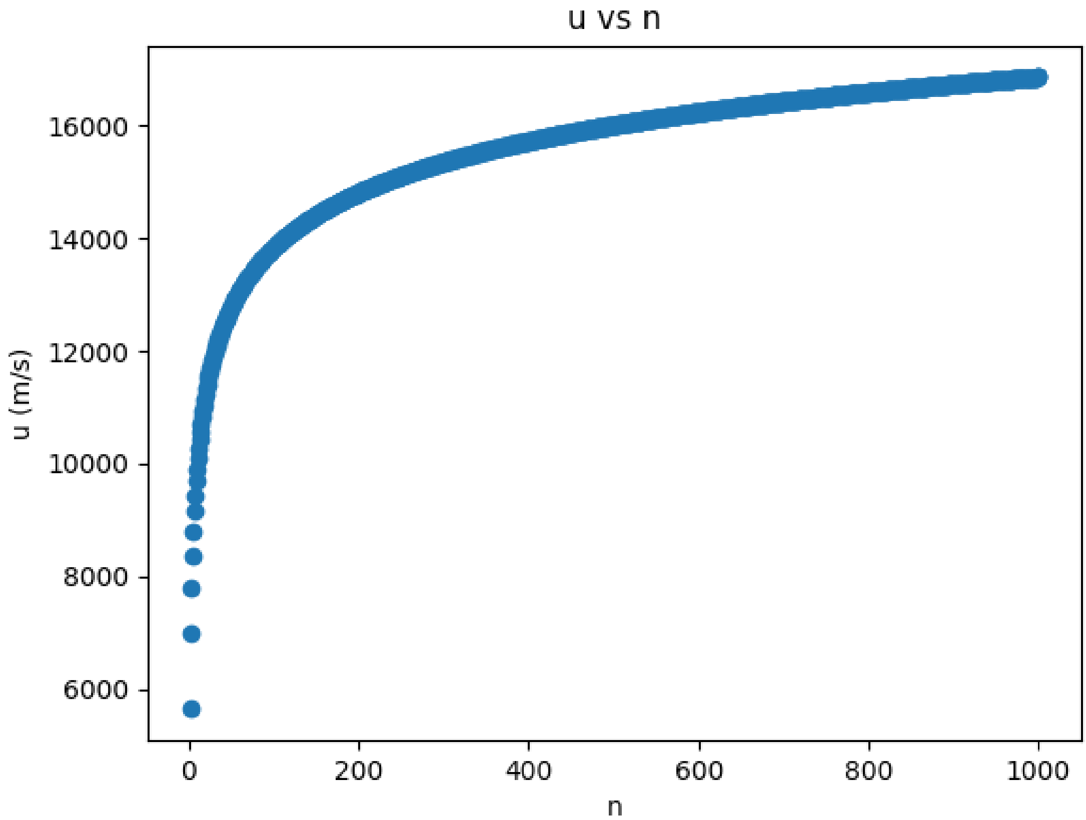

Figure 9: Graph of the magnitude of the initial velocity of each body against the number of bodies in the system

Clearly, there seems to be a direct relationship between u and n.

Taking $\log u$ against $\log(\log n)$, we get a straight line:

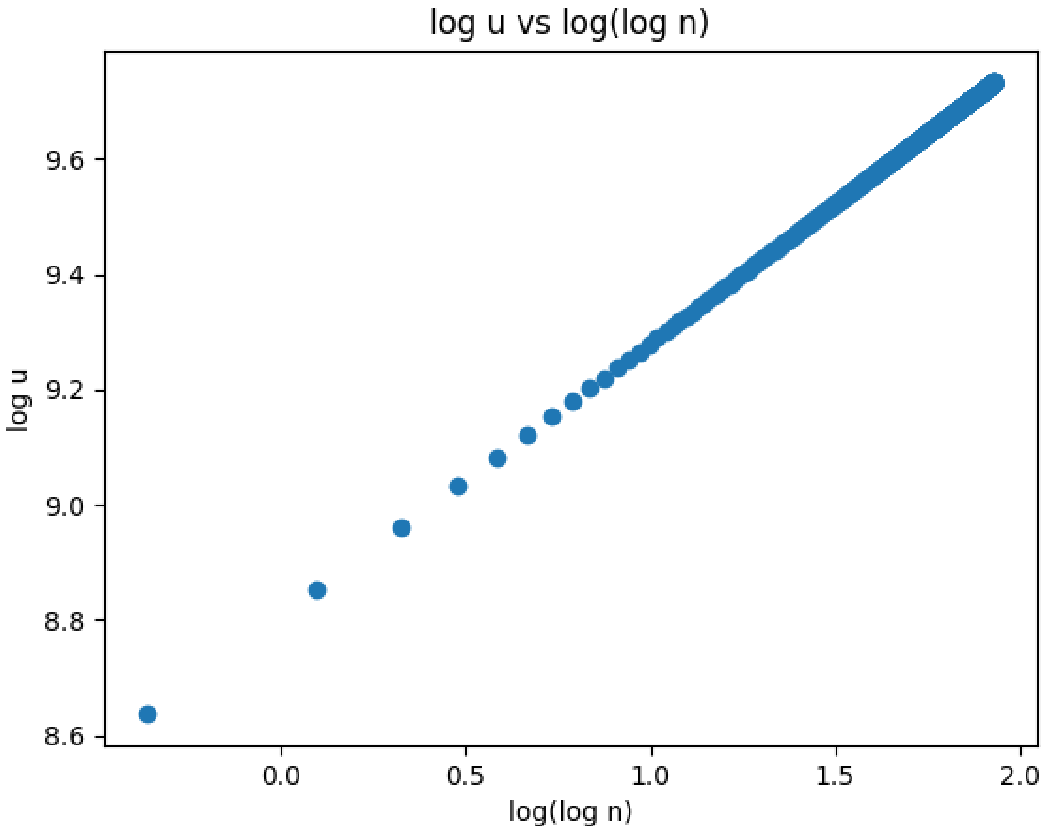

Figure 10: Graph of the log of the magnitude of the initial velocity of each body against the log of the log of the number of bodies in the system

Drawing a line of best fit over this graph, we get the following:

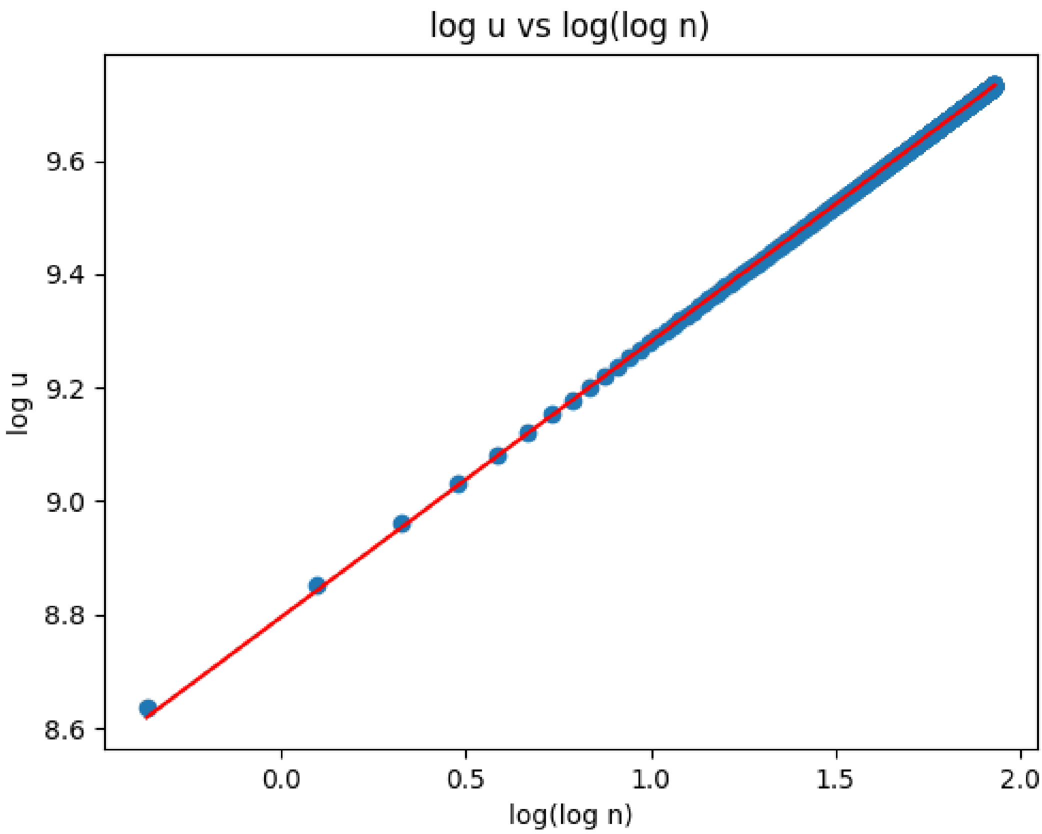

Figure 11: The best fit line obtained from the previous graph

Here, $m = 0.48$ and $c = 8.80$, where m and c are the slope and y-intercept of the best fit line respectively. Using the equation of a line, this gives us the following equation:

$$\log u = 0.48 \log(\log n) + 8.80 \tag{10}$$

Rearranging and simplifying this equation, we get

$$u = 6610(\log n)^{0.48} \tag{11}$$

This gives us a clear expression for u in terms of n.

Conclusions

We have used Newton's law of universal gravitation to simulate what happens when n bodies placed on a circle interact with each other. These simulations reveal that we should get a stable orbit regardless of the number of bodies, as long as there is a tangential initial velocity given to each mass with a magnitude of $6610(\log n)^{0.48}$, where n is the number of bodies. This expression for the velocity may not represent the real relationship between u and n, but this project tells us that it is, at the very least, a highly accurate approximation.

Discussion

1. Looking at the values of u for only large values of n, we get the following graph:

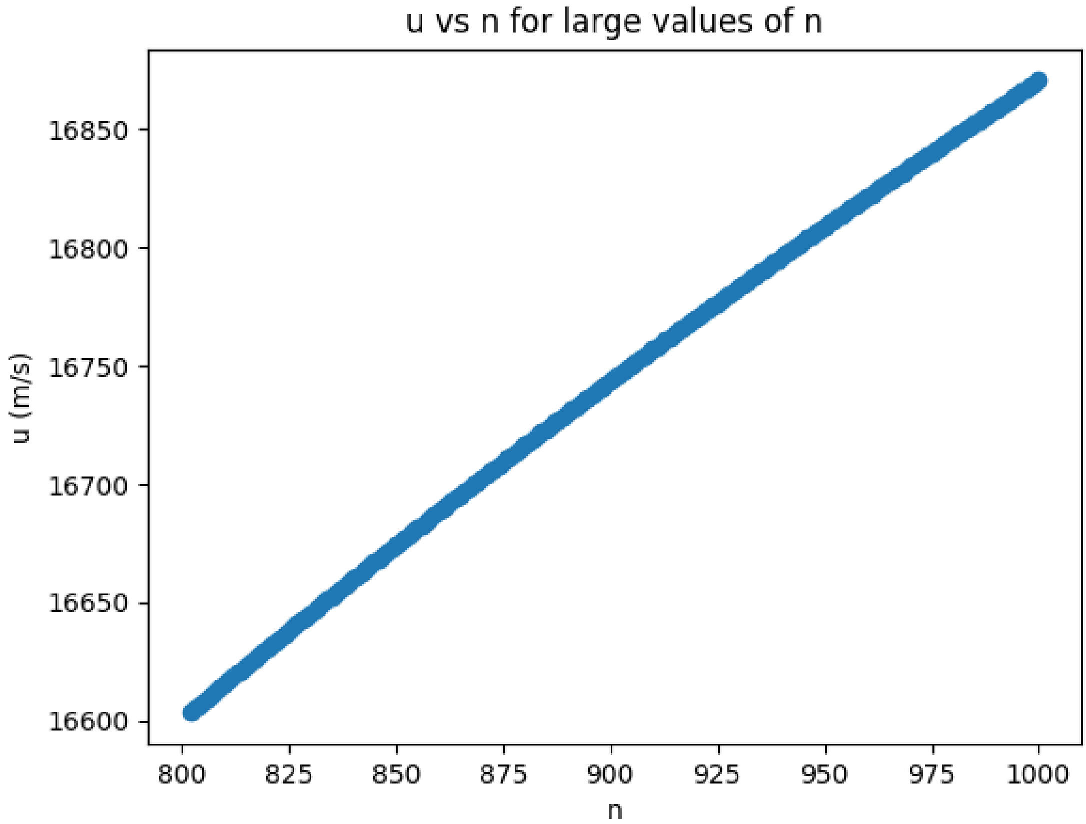

Figure 12: The graph obtained when only large values of n are considered

   This almost appears like a straight line, suggesting that for large values of n, u is directly proportional to n.

2. This project's results show a specific solution to the n-body problem where stable orbits are obtained. With further research, this can have many potential applications in reducing more complex n-body systems into these types of circular systems.