Physics Final Report: Simulating Three-Body Motion, Quantifying Chaos, and Finding Lagrange Points

Name: Hunter Frields

Capstone Advisor: Dr. Fisher

Grader: Dr. Gore

April 24th, 2023

1 Abstract

In this project, I numerically solve the n-body problem using two different numerical techniques: Forward Euler's Approximation and Runge-Kutta Approximation 4th Order (RK4). I compare an analytical solution for two-body motion with the Earth-Sun and "Equal Mass" systems to verify the n-body numerical solver works for two-body systems. I compare known perigee and apogee information about the Earth-Sun-Moon system to verify the correctness of the n-body numerical solver for three-body systems. I simulate n-body (verified for up to three-body) gravitational motion in 3-dimensional coordinates and output their motions into a 2D center-of-mass position graph and a three-dimensional absolute position graph. I use Lyapunov Exponents and a new method introduced in this paper, which I am calling the "Sticky Stability" method, to quantify chaotic behavior. The main goal is to find Lagrange Points for the Earth-Sun system using the James Webb Space Telescope (JWST) as a probe. The final result is that both methods of quantifying chaos show that the Earth-Sun-JWST system is more chaotic near gravitational attractors (massive bodies). The result also shows that Lyapunov Exponents cannot be used to find Lagrange Points using the assumptions made in this paper; however, the "Sticky Stability" method correctly identifies the Lagrange Points for the Earth-Sun system. The differences between the two methods of quantifying chaos are discussed more thoroughly in Section 3.1.6. Finally, contrary to the notion that a two-body system is never chaotic, both methods indicate that a two-body system is "chaotic". The reason the "Sticky Stability" method showed "chaoticness" was because the way it quantifies chaos does not follow the standard guidelines for Chaos Theory. For example, the "Sticky Stability" method is not generalizable for n-dimensional free space. The reason the Lyapunov Exponent method showed "chaoticness" was because the separation distance between separated probes is not constant and it also did not oscillate about the initial

separation distance (oscillations about initial separation distance converge to 0 very quickly). For an unreasonable amount of iterations, the Lyapunov Exponent for the ”Equal Mass” system gets close to 0 from the positive direction (the Lyapunov Exponent is always slightly positive); however, it still converges to a number slightly greater than 0 due to the fact that the oscillations do not occur at the initial separation distance. More information can be found in Section 6.8.

2 Introduction

The primary goal of this project is to create an n-body gravitational numerical integrator and a 3D simulation of an n-body gravitational system (tested for up to three bodies). The n-body problem was first introduced by Issac Newton when he came up with his famous gravitational equation for the force caused by one celestial body onto another (Equation 1). When Issac Newton derived the gravitational force equation, he realized that the equation was only analytically solvable for two-body systems, such as the Earth and Sun. For three-body systems or more, the only way to simulate their motions is through numerical integration. The three-body systems simulated in this project are the Sun-Earth-JWST system, the Sun-Earth-Moon system, and a "Chaotic Three-Body System" (Initial Conditions Provided in Appendix A). The second goal is to quantify Chaos using the known method, Lyapunov Exponents, and a new method introduced in this paper called the "Sticky Stability" method. That quantity representing Chaos will be plotted in a 2D Heat Map, which will represent the "chaoticness" of certain initial conditions for a probe. A probe is an object that has its initial conditions altered and its behavior analyzed when placed in a system of known objects. For example, in the Sun-Earth-JWST system, the JWST will be used as the probe.

The first notion of Chaos used in this paper is the Lyapunov Exponent, which is thoroughly described in Chapter 1 of *Chaos: An Introduction to Dynamical Systems* [1]. In reference to the book, Chaos theory is defined as the branch of mathematics that deals with complex systems whose behavior is highly sensitive to slight changes in initial conditions, so that small alterations can give rise to strikingly great consequences. Chaos theory has previously been used to explain irregularities in weather, stars, and blood vessels [2]. The second notion of Chaos, which is quantified using the "Sticky Stability" method, also attempts to describe the sensitivity of changing initial conditions on stability. The primary difference between the two is in the definition of "stability", which is thoroughly explained in Section 3.1.6.

Finally, the third goal of this project is to numerically determine the positions of Lagrange Points. Lagrange Points, as defined by NASA [5], are positions in space where the gravitational forces of a two-body system, like the Sun and the Earth, produce enhanced regions of attraction and repulsion. The result is that if a satellite is placed in a Lagrange point, then that satellite would remain in the same relative position to the Earth and Sun, which would conserve fuel. So, the goal is to numerically find the position of the Lagrange Points for the Earth-Sun system (Initial Conditions Provided in Appendix E). As mentioned earlier, the JWST will be used as a probe for finding the Lagrange Point, L2, in the Earth-Sun system because the JWST has an extremely small mass, so it has a negligible effect on the Earth and Sun, and it sits in L2. Therefore, the JWST is being used as a probe to quantify the "stability" of certain positions in the Earth-Sun system. It should be mentioned that a probe can also have a large mass; however, if a probe has a large mass compared to the other celestial objects in the system, then the quantification of the "stability" of particular initial conditions of the probe can not reasonably describe the stability of a system made up of the remaining

celestial objects. For example, in the Earth-Moon-Sun system, the Moon cannot be used as a probe to identify the "stability" of the Earth-Sun system since the Moon's mass will cause the entire system to behave differently when the Moon is given different initial conditions.

Now, as mentioned earlier, the reason that the Earth-Sun-JWST system (Initial Conditions Provided in Appendix D) was selected is that, in reality, the JWST is located at the Lagrange point, L2 (Seen in Figure 1), and the JWST's mass is negligible. Therefore, the known initial conditions of the JWST can be compared with that of the numerically calculated position of L2 of the Earth-Sun system. There are also analytical equations for calculating the position of Lagrange Points for a two-body system provided by Dennis Westra [6], which will be used in this project to test the other numerically calculated positions of the Lagrange Points (Provided in Section 3.1.7).

In order to verify the proper functionality of the different goals of this project, multiple verification steps were completed. First, after the numerical Euler Approximation solver is created to simulate the motion of n-bodies, it was verified using the analytical solution for the two-body problem (See Section 3.2.1) for both the Earth-Sun system and an "Equal Mass" system (Initial Conditions provided in Appendix B) moving at similar velocities. Following the verification that the n-body solver worked for two bodies, I had to verify that it also worked for three bodies using a custom-made analytical data tester function which is detailed in Section 3.2.2. Following that, I created a separate Lorenz Attractor system for two purposes: One was to verify that RK4 was implemented correctly for the n-body system and the second was to verify that the Lyapunov Calculation function was implemented properly (both are more thoroughly described in Sections 3.2.4 and 3.2.3 respectively). The reason the Lorenz Attractor system was selected as a tester system was that for certain initial conditions, the Lorenz Attractor has known Lyapunov Exponents. The comparison between the RK4 Lyapunov Exponent calculator and the known Lyapunov Exponents of the Lorenz System for initial conditions (p=10,r=28,b=8/3) can be seen in more detail in Section 3.2.3. The final verification step was to verify that the Largest Lyapunov Exponent 2D Map or the "Sticky Stability" 2D Map had its minimum value at the Lagrange Point, L2, in the Earth-Sun-JWST system with JWST as the probe. That verification step is detailed in Section 3.2.5 and discussed more thoroughly in Section 6.3, Discussion and Conclusions.

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

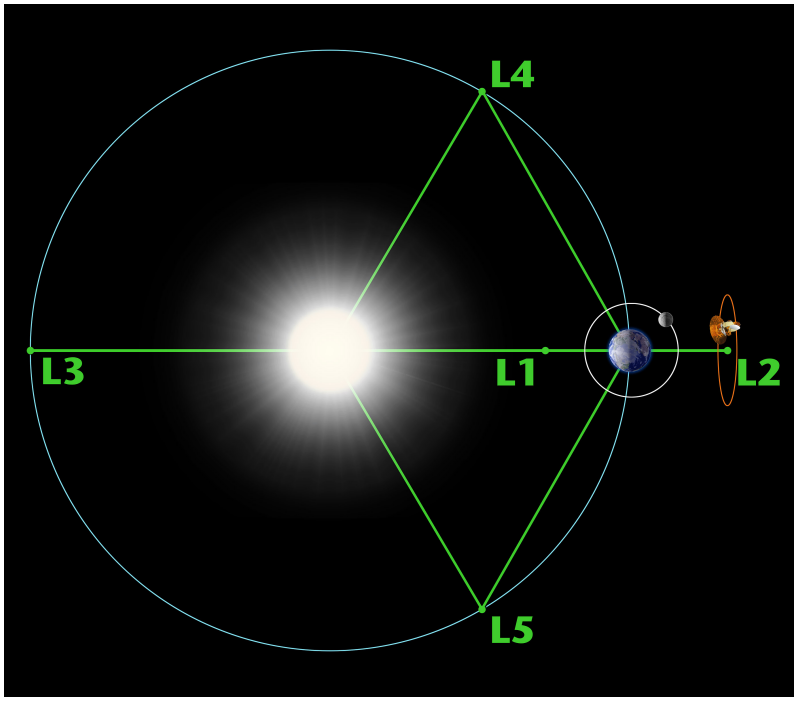

Figure 1: Earth-Sun-JWST System with Lagrange Points Marked (Image Provided From NASA [5])

3 Theory

3.1 Mathematical Modeling

3.1.1 N-body Euler Approximation Function

The first numerical approximation technique implemented in this project is Euler's Approximation. The pseudo-code for that algorithm is outlined in Section 4.2.1.

The generalized form of Forward Euler Approximation is:

$$\mathbf{y}(k+1) = \mathbf{y(k)} + h\mathbf{f}(t_k, y_k) \tag{2}$$

Where k is the iteration number, h is the timestep size, and f is the derivative of y with some dependence on t and y. For the use of my implementation, I had to convert this form to account for the fact I need to numerically determine both velocity and position where r represents the position vector and $\dot{r}$ represents the velocity vector:

$$\dot{\mathbf{r}}(t+\Delta t) = \dot{\mathbf{r}}\mathbf{(t)} + \Delta t \cdot \ddot{\mathbf{r}}\mathbf{(t)} \tag{3}$$

$$\mathbf{r}(t+\Delta t) = \mathbf{r(t)} + \Delta t \cdot \dot{\mathbf{r}}(t) \tag{4}$$

The order is reversed because, for Forward Euler's approximation, the position needs to be calculated from the previously known velocity. To implement Backward's Euler's approximation,the equations would be swapped, which would correlate to the mathematical equations:

$$\dot{\mathbf{r}}(t + \Delta t) = \dot{\mathbf{r}}\mathbf{(t)} + \Delta t \cdot \ddot{\mathbf{r}}\mathbf{(t)} \tag{5}$$

$$\mathbf{r}(t + \Delta t) = \mathbf{r(t)} + \Delta t \cdot \dot{\mathbf{r}}(t + \Delta t) \tag{6}$$

The major difference between the two solutions is that Equations 5 and 6 compute the future velocity and uses that to compute the future position rather than using the previous velocity to compute the future position. Generally, Backwards Euler Approximation yields greater results, so I thought it would be important to make the distinction between the two methods in this report.

3.1.2 N-body RK4 Approximation Function

Runge-Kutta is often significantly more accurate than both forms of Euler's Approximation; however, with that accuracy, it is also much more difficult to computationally solve. The RK4 version of Runge-Kutta is used in this project, which is the 4th-order version of Runge-Kutta expansion. The mathematical version of RK4 is outlined below for the given initial conditions:

$$\frac{dy}{dt} = f(t, y) \tag{7}$$

$$y(t_0) = y_0 \tag{8}$$

The classic Runge-Kutta method, called RK4, solves this by computing the following variables:

$$k_1 = f(t_n, y_n) \tag{9}$$

$$k_2 = f(t_n + \frac{h}{2}, y_n + \frac{hk_1}{2}) \tag{10}$$

$$k_3 = f(t_n + \frac{h}{2}, y_n + \frac{hk_2}{2}) \tag{11}$$

$$k_4 = f(t_n + h, y_n + hk_3) \tag{12}$$

Which leads to the approximation equations:

$$y_{n+1} = y_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)h \tag{13}$$

$$t_{n+1} = t_n + h \tag{14}$$

The previously described implementation can only be implemented directly for 1st-Order ODEs. Unfortunately, simulating n-body motion relies on a 2nd-order ODE, and thus, must require a simultaneous progression of both velocity and position. The mathematical process for the implementation is as follows for the given initial conditions:

$$\frac{d^2y}{dt^2}(t_0) = f(y_0) = acceleration(y_0) \tag{15}$$

$$\frac{dy}{dt}(t_0) = y_0' \tag{16}$$

$$y(t_0) = y_0 \tag{17}$$

The modified RK4 function solves this by computing the following variables:

$$k_{1v} = f(y_0) \tag{18}$$

$$k_{1x} = y_0' \tag{19}$$

$$k_{2v} = f(y_0 + k_{1x} * \frac{h}{2}) \tag{20}$$

$$k_{2x} = y_0' + k_{2v} * \frac{h}{2} \tag{21}$$

$$k_{3v} = f(y_0 + k_{2x} * \frac{h}{2}) \tag{22}$$

$$k_{3x} = y_0' + k_{3v} * \frac{h}{2} \tag{23}$$

$$k_{4v} = f(y_0 + k_{3x} * h) \tag{24}$$

$$k_{4x} = y_0' + k_{4v} * h \tag{25}$$

Which leads to the approximation equations:

$$y_{t_0+h} = y_0 + \frac{1}{6}(k_{1x} + 2k_{2x} + 2k_{3x} + k_{4x})h \tag{26}$$

$$y'_{t_0+h} = y_0' + \frac{1}{6}(k_{1v} + 2k_{2v} + 2k_{3v} + k_{4v})h \tag{27}$$

$$t_{n+1} = t_n + h \tag{28}$$

Where n is the iteration number and h is the timestep size. Equations 18-28 are then implemented as described in Section 4.2.2.

3.1.3 Two-Body Analytical Solution

One of my major tester functions is the two-body analytical tester function. $\mathbf{P(0)}_1$ is the position of Object 1 at t=0. $\dot{\mathbf{r}}\mathbf{(0)}_1$ is the velocity of object 1 at t=0. The same correlates with the definition of the position and velocity of Object 2 as well. **R(0)** is the position of the center of mass at t=0. Finally, $\dot{\mathbf{R}}\mathbf{(0)}$ is the velocity of the center of mass at t=0. The following equations are meant to define the positions and velocities of the objects in terms of the center of mass, which is important for solving the analytical solution.

$$\mathbf{r(0)}_1 = \mathbf{P(0)}_1 - \mathbf{R(0)} \tag{29}$$

$$\mathbf{r(0)}_2 = \mathbf{P(0)}_2 - \mathbf{R(0)} \tag{30}$$

$$\dot{\mathbf{r}}\mathbf{(0)}_1 = \dot{\mathbf{P}}\mathbf{(0)}_1 - \dot{\mathbf{R}}\mathbf{(0)} \tag{31}$$

$$\dot{\mathbf{r}}\mathbf{(0)}_2 = \dot{\mathbf{P}}\mathbf{(0)}_2 - \dot{\mathbf{R}}\mathbf{(0)} \tag{32}$$

Given those definitions, the angular momentum of the system at t=0:

$$\mathbf{L(0)} = \mathbf{r(0)}_1 \times \dot{\mathbf{r}}\mathbf{(0)}_1 * m_1 + \mathbf{r(0)}_2 \times \dot{\mathbf{r}}\mathbf{(0)}_2 * m_2 \tag{33}$$

Since the analytical solution is not solvable in terms of time, I had to find the theta based on the time that passed in my numerical solution. I did this by drawing a vector (I will call it Y) that points from object 1 to object 2 ($\frac{\mathbf{P(t)}_2 - \mathbf{P(t)}_1}{|\mathbf{P(t)}_2 - \mathbf{P(t)}_1|}$) and then based off of their positions from the numerical solution, I had to find the angle between the initial Y, which I called X ($\frac{\mathbf{P(0)}_2 - \mathbf{P(0)}_1}{|\mathbf{P(0)}_2 - \mathbf{P(0)}_1|}$), and the vector Y (which I would expect to line up with the analytical solution at that theta):

$$\theta = \cos^{-1} \frac{X \cdot Y}{|X||Y|} \tag{34}$$

I also had to find the reduced mass of the system:

$$\mu = \frac{1}{\frac{1}{m_1} + \frac{1}{m_2}} \tag{35}$$

I also had to define new vectors to define information about the system which I will label **r** (difference in position between the two masses) and the direction of rotation which I will label as $\hat{\theta}$:

$$\mathbf{r(0)} = \mathbf{P(0)}_2 - \mathbf{P(0)}_1 \tag{36}$$

$$\dot{\mathbf{r}}\mathbf{(0)} = \dot{\mathbf{P}}\mathbf{(0)}_2 - \dot{\mathbf{P}}\mathbf{(0)}_1 \tag{37}$$

$$\hat{\mathbf{r}}\mathbf{(0)} = X \tag{38}$$

$$\hat{\theta}\mathbf{(0)} = \frac{L(0)}{\left|L(0)\right|} \times \hat{\mathbf{r}}\mathbf{(0)} \tag{39}$$

From there, I solved this in terms of θ as the independent variable and I used the reduced solution of the two body problem in the form below to determine the motion (using the definitions of K and C as seen below) by plugging in all the previously determined variables:

$$r(\theta) = \frac{\frac{1}{K}}{1 - \frac{C}{K} * \cos\theta} \tag{40}$$

$$K = \frac{G \cdot m_1 \cdot m_2 \cdot \mu}{\left|\mathbf{L(0)}\right|} \tag{41}$$

$$A = \frac{1}{(\left|\mathbf{P(0)}_2 - \mathbf{P(0)}_1\right|} - K \tag{42}$$

$$B = \frac{-\dot{\mathbf{r}}\mathbf{(0)} \cdot \hat{\mathbf{r}}\mathbf{(0)}}{\hat{\theta}\mathbf{(0)} \cdot \left|\mathbf{P(0)}_2 - \mathbf{P(0)}_1\right|} \tag{43}$$

$$C = \sqrt{A^2 + B^2} \tag{44}$$

Using Conservation of Momentum, which will not be described here, I also know that the center of mass will move at constant velocity $\dot{\mathbf{R}}\mathbf{(0)}$, so the analytical equation for the center of mass is just:

$$\mathbf{R(t)} = \mathbf{R(0)} + \dot{\mathbf{R}}\mathbf{(0)} \cdot totalTimeElapsed \tag{45}$$

Equation 40 will be the distance between the two objects in space at the numerically calculated θ. The output of Equation 45 will be the position of the center of mass at the time the numerical calculation is at. These two equations can be put together to analytically determine an error between the numerical method and the analytical method as detailed in Section 3.2.1. For more information on how the Lagrangian is used to derive the above equations, read Chapter 4.2 in *Three Solutions to The Two-Body Problem* [4] by Frida Gleisner.

3.1.4 Largest Lyapunov Exponent Calculation

Lyapunov Exponents, as seen in Figure 3 indicated by the value λ, is a quantity that characterizes the rate of separation of infinitesimally close trajectories (it is equivalent to the natural log of the ellipse axes seen in Figure 2). The computational implementation of the algorithm is discussed more thoroughly in Section 4.4.

An initial uniform circle is created about the point of interest (initial position of JWST for example). The initial uniform circle is a representation of a set of slightly divergent initial conditions separated by a distance of $||\delta(0)||$, which, in the case of generating Figure 5, is 50 kilometers.

That means that the probe, the JWST for example, is placed at those slightly different divergent initial conditions and pushed forward in time. Each divergent JWST proceeds through time independently. That means there is only one JWST at a time in a single system. In the case of the "Equal Mass" system, the probe is the Earth_SunMass object. It is perturbed into a circle of divergent initial conditions. There are only two objects in a system at a time, but there are different "copies" of the system where the Earth_SunMass starts at a slightly different position.

The number of slightly divergent initial conditions is represented by the value numOfSamples. For my simulation, I decided to go with numOfSamples=20 as default, which means there are twenty equidistant setups of the different initial conditions of a probe along the uniform sphere at a distance of r=50km from the center point $x(0)$, which is also the original initial probe conditions.

The slightly divergent versions of the probe are pushed forward in time equivalent to a timeStep parameter. The first time step is called the "first iteration". The different system copies are iterated until the nth iteration, evolving the initial circle in an ellipse. The nth iteration is selected such that the ellipse stops evolving, which is the point at which the Lyapunov Exponent converges. Now, the most difficult part of this process is finding the ellipsoid that the uniform circle evolves into as depicted in Figure 2 over 3 iterations. The largest Lyapunov exponent is in the direction of maximum increase, which would be determined after finding the exponents for each direction. The largest Lyapunov exponent is the largest Lyapunov exponent out of all the slightly divergent initial conditions. The direction of the largest Lyapunov exponent is in the direction of maximum divergence. Optimally, integration would be done until t=infinity in order to find the exact largest Lyapunov Exponent, but that is not possible. The alternative is to approximate the largest Lyapunov Exponent by renormalizing nearby initial conditions each time step and do summations as follows:

$$h_1 = \frac{1}{l\tau} * \sum_{j=1}^{n} ln(a_j) \tag{46}$$

Where h_1 is the largest Lyapunov Exponent, l is the numOfIterations, τ = timeStep, $a_j = \frac{||\delta(t)||}{||\delta(0)||}$ before renormalization. As seen in Figure 4, after storing a particular a_j for a given iteration, the separation distance between two nearby trajectories is renormalized to the initial separation distance. This is done for enough iterations such that

the circle of initial conditions evolves into an ellipse and h_1 stops dramatically changing. The process described above is what is known as the Orbital Separation Method. It should be noted that an expansion on the OS method is the Gram-Schmitt Orthogonalization method, which is the one used to find all three Lyapunov Exponents for the Lorenz Attractor in Section 3.2.3. It should be noted that the other procedures are not described here since, for this computational project, it was not necessary to compute the other two for the n-body simulator. For more information on the derivation of the formula for the other Lyapunov Exponents, read Chapter 5.2 in *Chaos: An Introduction to Dynamical Systems* [1].

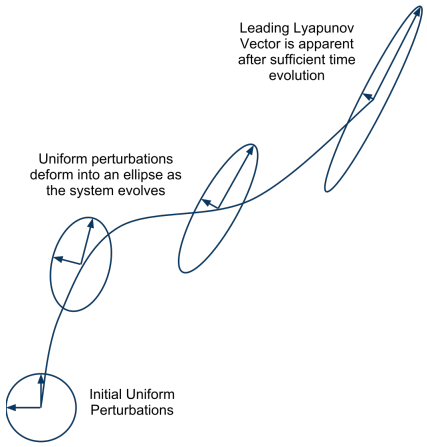

Figure 2: Progression of Divergent Initial Conditions over Time (Provided by Wikipedia, created by User: Yapparina [8])

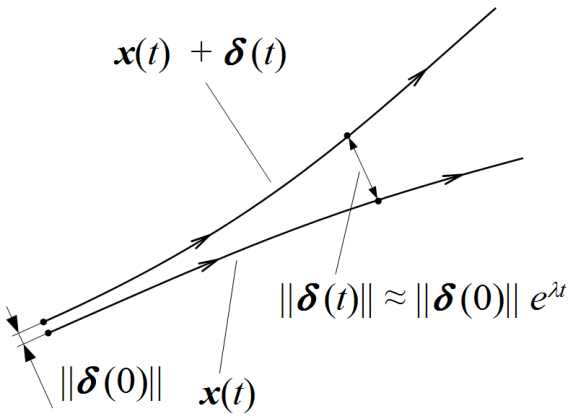

Figure 3: Separation of Nearby Initial Conditions Over Time (Provided by Wikipedia, created by User: Mrocklin [9])

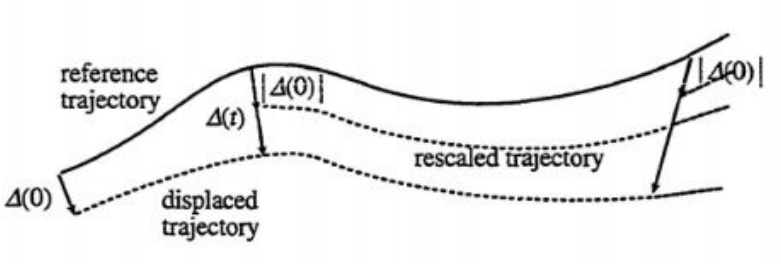

Figure 4: Renormalization of Nearby Initial Conditions Over Time (Provided by Chaos: An introduction to dynamical systems [1])

3.1.5 Normalizing Data For 2D Map Display

The normalization procedure for graphing the Lyapunov Exponents or "Sticky Stability" values is relatively simple. The exponents are normalized because the 2D Map associates values with shades of white and black. The grey scale goes from 0 to 1 (Black to White), which means that all of the values need to fit between 0 and 1 without losing relative information. The normalization procedure is as follows:

$$z_i = (x_i - min(x))/(max(x) - min(x)) \tag{47}$$

Where z_i is the normalized ith term, x_i is the ith Lyapunov Exponent or "Sticky Stability" value, min(x) is the minimum value in the 2D set, max(x) is the maximum value in the 2D set. An example of normalization would be:

$$(-5, -4, -2, -2, -2, 20, 30) -> (0.0, 0.028, 0.085, 0.085, 0.085, 0.714, 1.0) \tag{48}$$

This normalization equation is also slightly modified to account for extremely large values, such as when objects collide it may greatly overshadow all the other data, which can be seen in Figure 5, which is an example 2D Map where all the data is not visible. As seen in Figure 5, there is only a small white dot where the largest gravitational body is located (Most Chaotic Location). To adjust for this, I added a "contrast" term to the equation which allows for the largest values to be saturated such that the smaller values become more visible. The equation is now as follows:

$$z_i = saturation * (x_i - min(x))/(max(x) - min(x)) \tag{49}$$

With a saturation value of 50, the adjusted graph of Figure 5 can be seen in Figure 6.

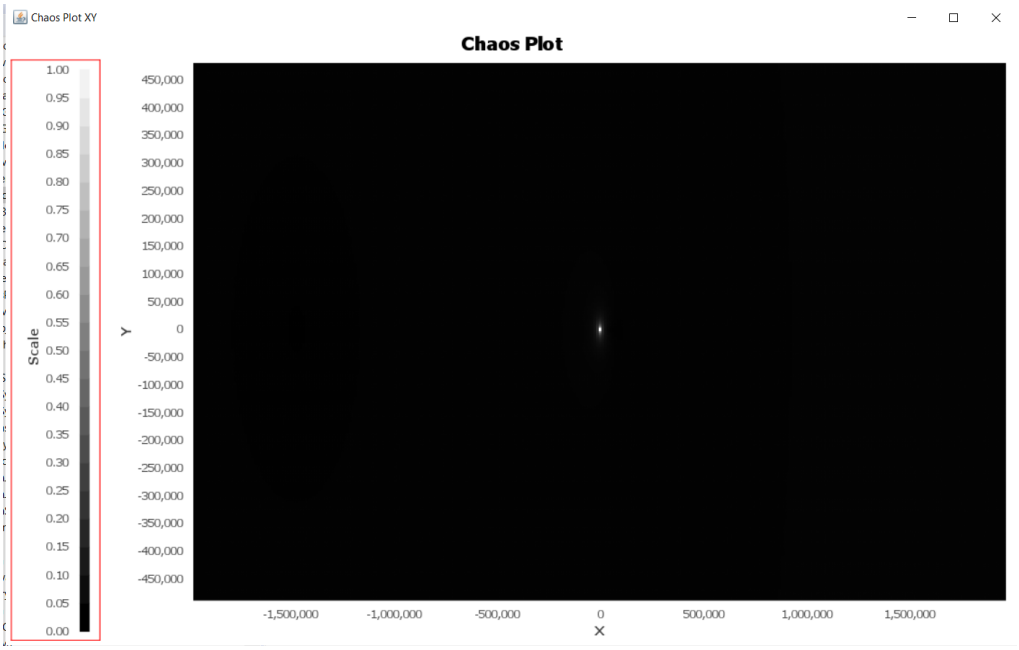

Figure 5: 2D Example Chaos Map with S=1

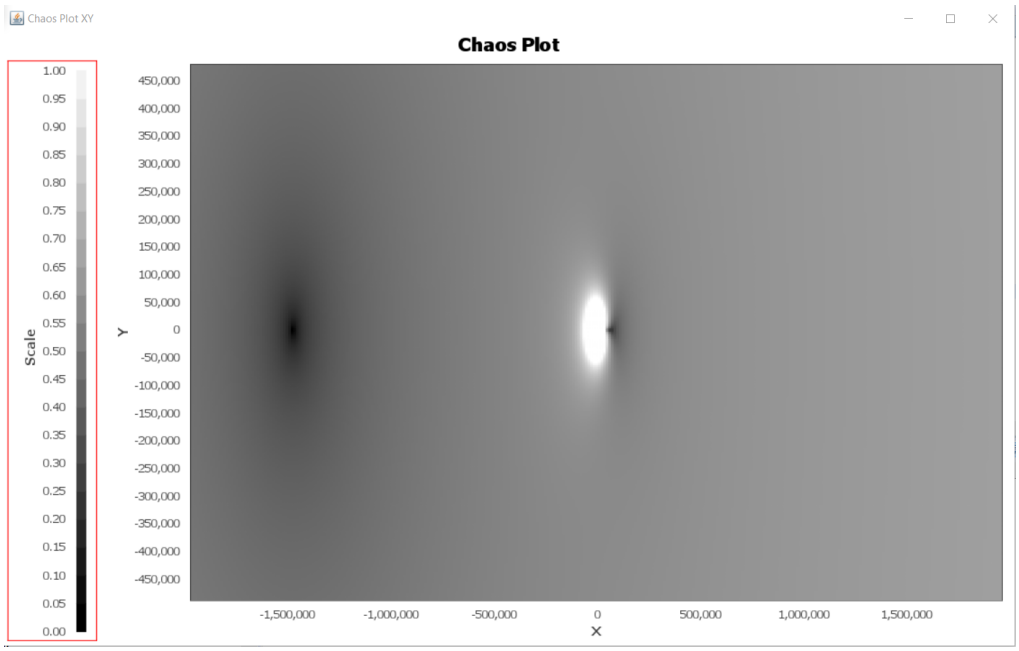

Figure 6: 2D Example Chaos Map with S=50

3.1.6 "Sticky Stability" Value Calculation

"Sticky Stability" is the method of quantifying chaos that I created specifically with the intention of finding Lagrange Points. As mentioned in the Introduction, Lagrange Points are positions in space that are nearly constant relative to the other bodies in the system. Lyapunov Exponents measure divergence in small perturbations in a probe's

initial conditions. Therefore, Lyapunov Exponents characterize "stability" as a probe having very little divergence with small changes in initial conditions (If you wiggle a probe, its motion over time will remain constant). The "Sticky Stability" method characterizes "stability" as a position in space where an object will "stick" to the other celestial objects and its relative distances to each object will remain relatively constant. For example, in the Sun-Earth-JWST system, the distance between the JWST and the Sun and the distance from the JWST to the Earth would be constant. Therefore, the definition of "chaotic behavior" for each quantifying method is very different. The "Sticky Stability" method is also not generalizable to n-body dimensional space. There is not a "Sticky Stability" value associated with each nth dimension. There is one "Sticky Stability" value associated with the entire position space. The Lyapunov Exponent is generalizable to nth-dimensional space and can also be applied across other non-position spaces (for example, the momentum space for a double pendulum). Therefore, while the "Sticky Stability" method is useful to quantify a particular definition of "chaotic behavior", it fails to satisfy the official definition.

In order to calculate the "Sticky Stability" value, the process is as follows:

$$v = \ln\left(\frac{1}{l\tau} * \sum_{j=1}^{l}\sum_{i=1}^{n}\frac{|d_i(0) - d_i(j\tau)|}{d_i(0)}\right) \tag{50}$$

Where l is the number of iterations. τ is the time-step size. n is the number of total objects in the system. $d_i(0)$ is the initial distance between the probe and celestial object, i. $d_i(j\tau)$ is the distance between the probe and celestial object, i, at time = $l\tau$. The value v describes the "average weighted" change in the position of the probe relative to the other celestial objects over time. The value v is larger for positions where a probe moves away from all the objects, but smaller where the probe moves such that its distances to all the other objects remains nearly constant. It is weighted so that the change in distance from a faraway object is less important than a change in distance to a close object. This prevents motions in ellipse patterns around large objects from "washing out" the data. For example, without the weighted term on the denominator, the movement of the Earth from its perigee to apogee would show that the JWST's distance from the Sun shifted dramatically even though it sat, correctly, in L2.

3.1.7 Analytical Calculation of Lagrange Points

According to the paper written by Dennis Westra [6], the analytical equation for L2 is as follows:

$$\vec{L_2} = \left(r - r\left(1 + \sqrt[3]{\frac{M_2}{3M_1}}\right), 0, 0\right) \tag{51}$$

Where r is the distance from M1 to M2. M1 is the larger mass and M2 is the smaller mass. The reason the x-position of $\vec{L_2}$ is the only value being calculated is that it is known that the other two values are 0 for L2 when the

two masses sit on the x-axis. This also assumes that M2 sits to the left of M1. The distance between M1 and L2 is subtracted from the distance between M1 and M2. The result is a position that lies to the left of M2, which is the proper location of L2.

The analytical equations for the other Lagrange Points in this coordinate system are:

$$\vec{L_1} = \left(r - r\left(1 - \sqrt[3]{\frac{M_2}{3M_1}}\right), 0, 0 \right) \tag{52}$$

$$\vec{L_3} = \left(r + r(1 + \frac{5}{12}\frac{M_2}{M_1}), 0, 0 \right) \tag{53}$$

$$\vec{L_4} = \left(\frac{1}{2}, 0, \sqrt{r^2 - (\frac{1}{2} * r)^2} \right) \tag{54}$$

$$\vec{L_5} = \left(\frac{1}{2}, 0, -\sqrt{r^2 - (\frac{1}{2} * r)^2} \right) \tag{55}$$

These equations assume that both masses sit on the x-axis and their velocities are only in the xz- plane. Note, when referencing the three-body coordinates, the plane of motion in this paper is the xz-plane; however, for the Chaos Graphs, the xz-plane is actually the xy-plane. So, in the case of the graph, the y-axis represents the changes in the z-position in the 3D graph positions.

3.2 Data Analysis

3.2.1 Two-Body Analytical Solution Comparison

In order to verify the correctness of the n-body numerical equation solver (both Euler and RK4), the following equations were used to average individual changes in values:

$$AveragePercentError = \frac{100}{n} * \sum_{j=1}^{n} \frac{|theoretical_i - experimental_i|}{|theoretical_i|} \tag{56}$$

The above equation is in its genetic form. That general form is used for the majority of the comparison functions in this project. For the particular case of the two-body solution, n=numberOfIterations and experimental correlates to a particular numerically calculated position (experimental is the analytical analog). The data resulting from this function can be seen in Section 5.1.1 and Section 5.1.2.

3.2.2 Three-Body Real-Data Comparison

The three-body real-data comparison function operates similarly to the two-body verification function. The primary difference is that n=3. There are only three comparisons made for each space object over 1 cycle (365 days in earth's orbit). The positions, for Earth for example, are calculated at t=0, t=365/2, and t=365 and compared with known perigee and apogee locations. The average percent error for each celestial object is then outputted. The data resulting from this function can be seen in Section 5.2.1.

3.2.3 Lorenz Attractor Lyapunov Exponent Values Verification

In order to verify that the generalized Lyapunov Exponent calculator was correct, it was tested against the known values for the Lorenz Attractor with initial conditions (p=10,r=28,b=8/3). The individual percent errors between the numerically calculated LCES (Lyapunov Exponents) and the analytical LCES are shown in Section 5.5. As mentioned earlier, the Gram-Schmitt Orthogonalization procedure was used in the case of the Lorenz Attractor in order to find all of the Lyapunov Exponents. In the case of the gravitational simulation, only the largest Lyapunov value was necessary. For more information on Gram-Schmitt Orthogonalization, reference *Chaos: An Introduction to Dynamical Systems* [1].

3.2.4 Lorenz Attractor RK4 Implementation Verification

The Lorenz Attractor relies on a 1st Order ODE, however, after taking the derivative of those functions I was able to create a 2nd Oder ODE describing the motion of the Lorenz Attractor. I then used the RK4 function from 3.1.2 to integrate the 2nd Order version of the Lorenz Attractor. The number of iterations it took for the Lorenz Attractor to converge to the proper Lyapunov Exponents (within 1%) was compared to the number of iterations Euler's Approximation took for the 2nd Order ODE. The results of that data collection are shown in Section 5.5.

3.2.5 Comparison between Minimum Lyapunov Exponent Position and Minimum "Sticky Stability" value to the Lagrange Points

This tester function is actually extremely simple. Using the Average Percent Error formula, I set n=3. The minimum value in either method is just the smallest value in the set of all the values in the 2D Chaos Map. It is the same value used in Equation 49 for normalization (min(x)). The minimum chaos value has a position attached to it. The sum of the percent errors of each direction (x,y,z) is then averaged for the predicted Lagrange Point at that (x,y,z) and returned as an average percent error. The only exception is when one of the analytical values is 0, that direction is discarded since it would yield infinity. Dealing with comparisons around 0 is complicated (since they give infinite percent error) and ignored in the percent error calculation. The results of this function are shown in Section 5.6 for

both methods.

3.3 Expected Results

The expectation of this simulation is that the 3D plot will correctly model the behavior of at least a three-body system. The expectation is that, for a two-body simulation, the aggregate percent error over 1 orbital period (365 days) is not greater than 1%. It is also expected that for the Earth-Sun-Moon system (Initial Conditions Provided in Appendix C), the percent error calculated for any celestial body does not exceed 1%. I also expect that the Sun will have the smallest percent error in its position due to the fact that the Sun barely moves at all. I also expect that the Moon will have the largest percent error due to the fact that it is so small and is affected by two net forces. I also expect that the smallest Lyapunov Exponent in the set of Lyapunov Exponents will correlate to a Lagrange point when given the initial velocity of the JWST. I also expect that the smallest "Sticky Stability" value will correlate to the Lagrange Point, L2, when given the JWST's initial velocity. The Lagrange point, L2, is stable, so it should be noticeable in both the 2D Map and the data. That also means that I expect the percent error between the numerically calculated location of L2 and the known location of L2 and JWST to be less than 1%. Finally, I expect that the Chaos plots will also be able to find the other Lagrange Points of the Sun-Earth system using a test probe, such as the JWST, within 1%.

4 Methods

4.1 Setting up Eclipse and JMonkey3D

In order to program a 3D and 2D graph, the first step was setting up Eclipse and Java. This was achieved by utilizing pre-existing 3D graph classes, which extend the functionality of the JMonkey3D Library available in the git repository [3]. While these functions improve visualization, they are not integral to the functionality of the project and will not be described in detail. For the 2D graph, the JFreeGraph library was implemented to display a line graph and a heat map (for Chaos Map).

4.2 Creating SpaceObject Class and SpaceObjectFast Class

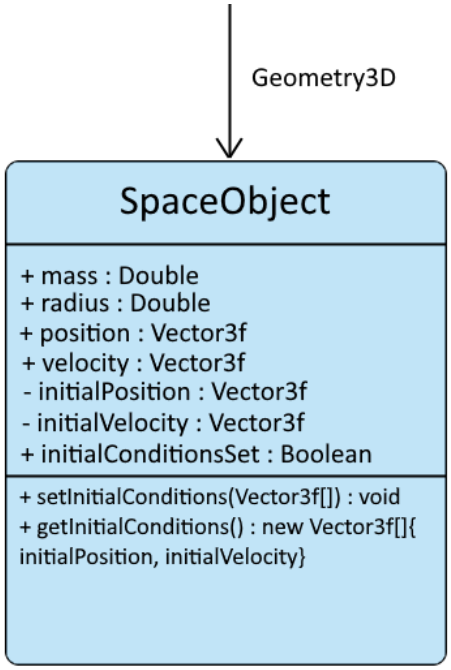

Figure 7: Space Object UML Diagram

The SpaceObject class and the SpaceObjectFast class represent all celestial objects that are simulated. The primary difference between the two classes is that the SpaceObjectFast class does not import the Geometry3D class. The Geometry3D class is a custom render class that implements Geometry from the JMonkey3D Library. The SpaceObjectFast class was introduced to address performance issues encountered while using RK4, which creates multiple copies of SpaceObjects. Switching to SpaceObjectFast classes for data calculation resulted in an approximately 200x speedup, as the regular SpaceObject creates a 3D mesh and geometry every time it is instantiated, which is time-consuming and only necessary for 3D rendering.

4.3 Setting up Euler's Approximation

```
Function eulerApproximation(spaceObjects, timeElapsed, timeStep) returns void
    inputs: spaceObjects (an arrayList containing spaceObject), a float timeElapsed representing simulation time passed,
                a float timeStep representing the smaller time step used in Euler Approximation

    local variables: timeCalculated (a float representing the total time calculated by Euler Approximation)

    timeCalculated ← 0

    Repeat
        eulerCalculation(spaceObjects, timeStep)
        timeCalculated += timeStep
    until timeCalculated + timeStep < timeElapsed

    //Call Euler Calculation to Handle Remaining Time Not Accounted for in Main Loop: timeElapsed - timeCalculated
    eulerCalculation(spaceObjects, timeElapsed - timeCalculated)

endfunction
```

Figure 8: Euler Approximation Pseudo-code

```
Function eulerCalculation(spaceObjects, timeStep) returns void
    inputs: spaceObjects (an arrayList containing spaceObject),
                a float timeStep representing the smaller time step used in Euler
                                                    Press Esc to exit full screen

    local variables: acceleration (a Vector3f representing the acceleration of each spaceObject)

    For Each SpaceObject s1 in spaceObjects Do
        acceleration ← (0,0,0)
        For Each SpaceObject s2 in spaceObjects Do
            If s2 != s1:
                //accelerationEffect uses the general newton's force equation of the force exhibited on s1 by s2
                acceleration ← acceleration + accelerationEffect(s2,s1)
            Endif
        Endfor

        s1.position ← s1.position + s1.velocity • timeStep
        //This calculates the change in s1.velocity after s1.position so that it resembles the forward Euler approximation.
        //These lines would be reversed for Backwards Euler Approximation
        s1.velocity ← s1.velocity + acceleration • timeStep
    Endfor

endfunction
```

Figure 9: Euler Calculation Pseudo-code

The Euler Approximation function simulates n-body motion. The pseudo-code for the algorithm is outlined in both Figure 7 and Figure 8. The Euler Approximation was developed with two time parameters instead of just the time-step as seen in its equation equivalent in Section 3.1.1, to accommodate the real-time game loop in the JMonkey3D software used for displaying the 3D simulation. The function integrates using the regular time-step until the time passed is the same as what passed in the simulation, as shown by Equations 3 and 4 that correlate to Lines 16 and 19 in Figure 8.

4.4 Two-Body Analytical Verification

I tested the two-body solution using the analytical solver described in Section 3.1.3, Equations 40-45. The comparison equation in Section 3.2.1 was utilized to record accumulated error between the numerical and analytical solutions.

I encountered a major setback with the Euler Calculation function when attempting to achieve the required accuracy for three-body motion. Specifically, I observed large errors for the Moon's position over time when using Euler's Approximation on the Earth-Moon-Sun system. To address this issue, I opted to implement RK4, which is known to be superior to Euler's Approximation in terms of conserving energy and momentum (and, as a result, position).

4.5 Setting up RK4 Approximation

After creating Euler's Approximation and discovering its limitations in simulating three-body motion, I decided to implement the Runge-Kutta 4 (RK4) method. The implementation of RK4 is similar to Euler's Approximation in that it takes in two time parameters (time-step and time-elapsed). However, it uses Equations 18-28 from Section 3.1.2 for numerical timestep calculations.

Regarding setbacks and resolutions, I encountered several challenges when implementing RK4. One significant issue was the difficulty in modifying the given first-order ODE version of RK4 to simulate gravity, which relies on an acceleration solely dependent on position. The modifications made can be seen in Section 3.1.2.

4.6 Three-Body Real-Data Tester Verification

```
Function compareToEarthMoonSunSolution(spaceObjects, totalTimeElapsed, tpf) returns Double
    inputs: spaceObjects (an arrayList containing spaceObject), totalTimeElapsed       array
                (float representing the entire in-simulation time passed),
                    tpf (a float representing the real-time refresh rate of the screen
                        (used for drawing to screen panel))

    Earth = spaceObjects[0]
    Moon = spaceObjects[2]

    ErrorE1 = getPercentError(Earth.eccentricity, 0.01671)
    ErrorM1 = getPercentError(Moon.eccentricity, 0.0549)

    If totalTimeElapsed == 0 days:
        ErrorE2 = getPercentError(Earth.position,EarthApogeeLocation)
        ErrorM2 = getPercentError(Moon.position,MoonApogeeLocation)
    Endif
    ElseIf totalTimeElapsed == 365/2 days:
        ErrorE3 = getPercentError(Earth.position,EarthPerigeeLocation)
        ErrorM3 = getPercentError(Moon.position,MoonPerigeeLocation)
    Endif
    ElseIf totalTimeElapsed == 365 days:
        ErrorE4 = getPercentError(Earth.position,EarthApogeeLocation)
        ErrorM4 = getPercentError(Moon.position,MoonApogeeLocation)
    Endif

    #Uses process described in Section 3.2.2 to obtain a percent error for each object

    return #A percent error array indexed by object

endfunction
```

Figure 10: Experimental-Data Tester Function Pseudo-code

The custom three-body real-data tester verification was necessary to assess the accuracy of the simulation for three or more bodies. A verification for the Earth-Moon-Sun configuration was developed for this project. The pseudo-code for the real-data tester implementation can be found in Figure 9, and the process is described in Section 3.2.2.

4.7 Creating Lorenz Attractor Simulation to Verify Lyapunov Exponent Calculation

The Lorenz Attractor system was built in order to verify the correctness of the Largest Lyapunov Exponent function. J. C. Sprott provides known Lyapunov Exponents for initial conditions (p=10, r=28, b=8/3) of the Lorenz Attractor [7]. The associated Lyapunov Exponents are (0.906, 0, -14.572), where the x-term is the leading Lyapunov Exponent, which is the only one being calculated for the 2D Chaos Map in order to save computational time. The results obtained using RK4 and the Lyapunov Exponent calculation can be seen compared to the expected Lyapunov Exponents in Figure 29. The comparison is discussed in Section 5, Results and Conclusions.

4.8 Creating 2D Chaos Map

```
Function chaosMap(testedObject, spaceObjects, xLimit, yLimit, resolution=10) returns Double[][]

    inputs: testedObject (SpaceObject underwhich the small perterberations will be tested),
    spaceObjects (arrayList of all spaceObjects), xLimit (xLimit consists of two doubles which specifies the
    x-bounds of the map, yLimit (yLimit consists of two doubles which specifies the y-bounds of the map,
    resolution (resolution is the number of different coordinates the testObject is placed at, for a
        resolution of 10, it is placed at coordinates xLimit/10 * i for i from 0 to 10)

    Double[][] lyapunovValues;
    For X in resolution:
        for Y in resolution:
            testObject.position = (xLimit/resolution * X, yLimit/resolution * Y)
            lyapunovValues[X][Y] = lyapunovCalculation(testObject,spaceObjects,20,0.1)

    return lyapunovValues


endfunction
```

Figure 11: Lyapunov Exponent Functions Pseudo-code

The 2D Chaos Map pseudo-code can be found in Figure 11. The function generates a grid of values based on the chosen resolution. For instance, if the resolution were 10, it would populate the entire area with a 10x10 grid of calculated values. Afterward, it uses the normalization function from Section 3.1.5 before displaying it to the JFreeGraph heat map. The lyapunovCalculation Function returns the output of either the Largest Lyapunov Exponent Calculation (Section 3.1.4) or the "Sticky Stability" Value Calculation (Section 3.1.6), depending on the graphed option. The 2D Chaos Map also takes in an initial velocity for the probe being tested. That is because the graph only shows the results of altering the position in the xy-plane. The Chaos Map, therefore, is sensitive to changes in velocity as well, but it is not directly shown. Figures 32 and 33 show the same scale (entire Earth orbit) where the plots look entirely different. That is because the probe, the JWST, starts with a different initial velocity (one that is most stable at the associated Lagrange Point).

4.9 Implementing Largest Lyapunov Exponent Calculation for 2D Chaos Map

The Largest Lyapunov Exponent Calculation pseudo-code can be seen in Figure 11. The mathematical model, provided in Section 3.1.4, was implemented into Line 26 of the pseudo-code. The normalization function from Section 3.1.5 was used to normalize the data before graphing. Whenever the Lyapunov Exponent Calculation is used, the title of the 2D Chaos Map is "Lyapunov Exponent. "

Setbacks and Resolutions: The most significant challenge I encountered while implementing the Lyapunov Exponent function was the lack of a way to test if I was getting the correct data. Initially, I was unsure which parameters to use, and I was uncertain if the graph I was receiving was accurate. In order to verify that my Lyapunov Exponent Calculation implementation was functioning correctly, I decided to create the Lorenz Attractor (Section 4.6). This was the primary obstacle of the project, and for a long time, I was uncertain if I would be able to get the Lyapunov Exponent calculation to work properly.

4.10 Implementing "Sticky Stability" Value Calculation for 2D Chaos Map

In Section 3.1.6, the mathematical model for the "Sticky Stability" value calculation is described. The code for "Sticky Stability" is similar to the pseudo-code in Section 4.8, but with two key differences: the 2D Chaos Map is labeled "Sticky Stability" and the mathematical model for the value calculation is different.

4.11 Creating the Lagrange Point Finder for 2D Chaos Map

The Lagrange point finder is an important aspect of the 2D Chaos Map project. Its purpose is to find the most stable points in the system numerically. The function takes in an array of chaos values from the 2D Chaos Map and identifies the smallest chaos value, whether from Sticky Stability or Lyapunov. For the Lagrange Point finder to

function correctly, the probe used must have an initial velocity that is stable in the Lagrange points. This is necessary to distinguish one Lagrange point from another. To increase accuracy, the function looks ”near” the Lagrange point rather than attempting to search for L2 over the entire Earth orbit, as shown in Section 5.6. By using an initial velocity, the Lagrange Point Finder function finds the minimum value from the chaos map grid and returns a position associated with that minimum value. The results of the Lyapunov Method are presented in Section 5.8, Figures 45 and 46, while the results for the ”Sticky Stability” Method are presented in Section 5.8, Figures 43 and 44.

4.12 Creating the Velocity Finder Function for Lagrange Points Using ”Sticky Stability” Value

The Velocity Finder function is a mathematical function that does not rely on any specific mathematical model. It takes an input position and tests an array of velocities at that position. The function operates differently from the Chaos Map discussed in Section 4.2, in that it modifies the velocity only in the xy-plane. Additionally, unlike the Chaos Map, the function does not output the ”Sticky Stability” values to a plot. Instead, the function returns only the velocity that results in the smallest chaotic value given the input position. That associated velocity could also be called the most ”stable” velocity for the input position.

The Velocity Finder function uses the analytical equations for the Lagrange points outlined in Section 3.1.7 in order to test different velocities. The equivalent of the ”resolution” for the ”Chaos Graph” is ”uncertainty” for the Velocity Finder. The velocity steps taken determine how accurate the velocity found is. The uncertainty selected for the Lagrange Point velocities was 0.001. That meant that over the xy-plane, it would step 0.001 km/s over a given range. In the case of the Lagrange Points (found using the process described in Section 3.1.7), the range was selected to be between -500 and 500 since all of the optimal Lagrange Point velocities were within that range.

This function was created to demonstrate that as long as the position or velocity being tested is known, the other value can also be calculated. In the Chaos Map, the initial velocity of a probe is given, and the most stable position is returned. On the other hand, in the velocity finder function, the initial position of a probe is given, and the most stable velocity for that position is returned.

5 Data

5.1 Data Results of Two-Body Analytical Function

5.1.1 Earth-Sun System

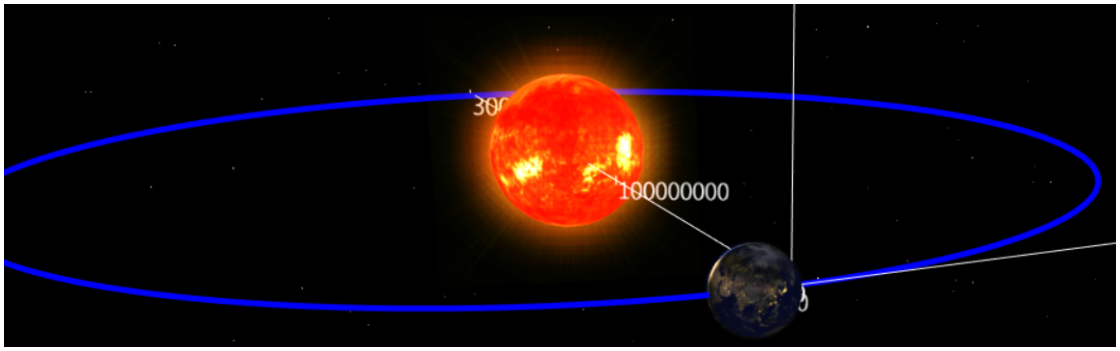

Figure 12: Sun-Earth 3D Position Graph

Analytics:

Angular Momentum of C.O.M.: 2.617227539004119E40
Position of C.O.M.: (1.4711971827916604E8, 0.0, 2664.162498363986)
Analytical Eccentricity of Earth: 0.01670800136304142
Numerical Position of Earth: (159640.77512880325, 0.0, -6791621.318804197)
Analytical Position of Earth: (159638.59701395035, 0.0, -6791578.119149687)
Numerical Position of Sun: (1.4712015952068493E8, 0.0, 2684.5620578675707)
Analytical Position of Sun: (1.4712015952069175E8, 0.0, 2681.4369445095)
Percent Error For Earth: 2.473866076739127E-8%
Simulation Time Elapsed: 8264.86611111111 hours

Figure 13: Sun-Earth Analytical Comparison Data

5.1.2 Equal Mass System

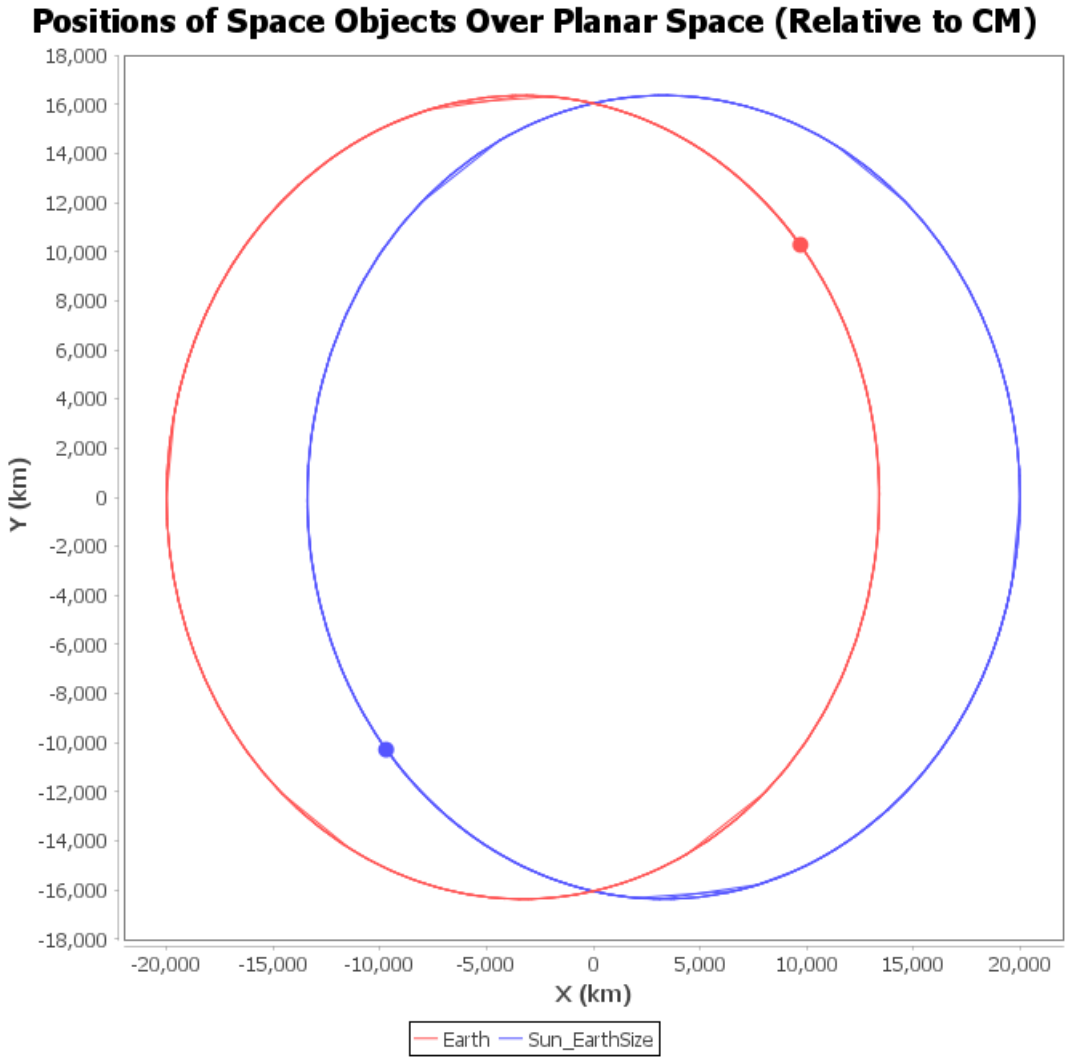

Figure 14: Equal Mass 2D C.O.M. Plot

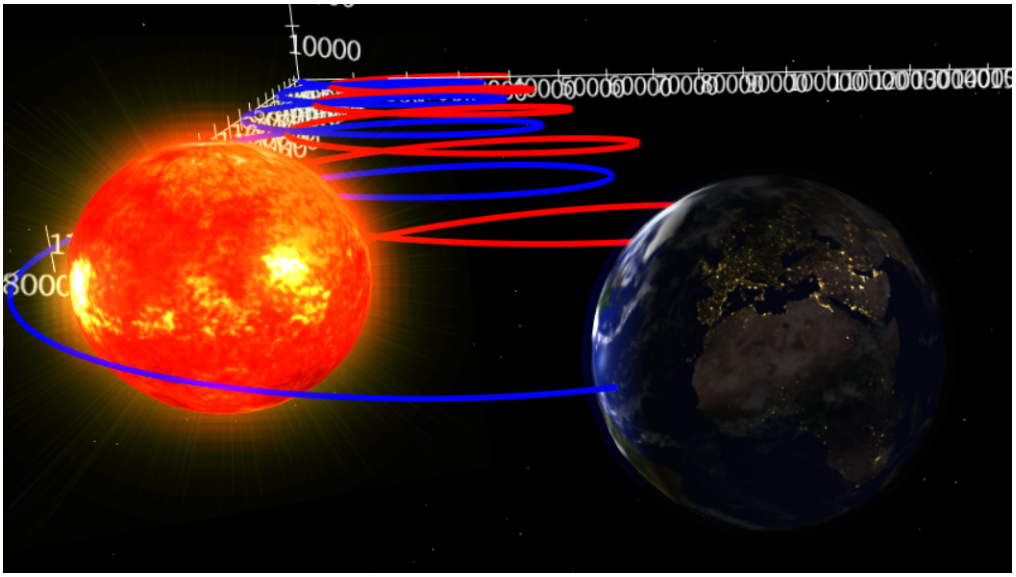

Figure 15: Equal Mass 3D Absolute Position Graph

Analytics:

Angular Momentum of C.O.M.: 4.7777519999999995E35
Position of C.O.M.: (19999.999999999884, 0.0, 1674825.6805235092)
Analytical Eccentricity of Earth: 0.19719394632072235
Numerical Position of Earth: (193.68902494477717, 0.0, 1672347.5807702912)
Analytical Position of Earth: (192.368039688532, 0.0, 1672364.4839563065)
Numerical Position of Sun_EarthSize: (39806.31097505499, 0.0, 1677303.7802767272)
Analytical Position of Sun_EarthSize: (39807.63196031147, 0.0, 1677321.0160436926)
Analytical C.O.M. Position: (20000.0, 0.0, 1674842.7499999995)
Numerical C.O.M. Position: (19999.999999999884, 0.0, 1674825.6805235092)
Analytical C.O.M. Velocity: (0.0, 0.0, 0.9999999999999998)
Numerical C.O.M. Velocity: (0.0, 0.0, 0.9999999999999991)
Percent Error For Distance Between Objects: 0.00666970218465162 4%
Percent Error For C.O.M: 0.0010190235925384 35%
Simulation Time Elapsed: 465.2340972222222 hours

Figure 16: Equal Mass Analytical Comparison Data

5.2 Data Results of Three-Body Real-Data Tester Function

5.2.1 Earth-Sun-Moon System

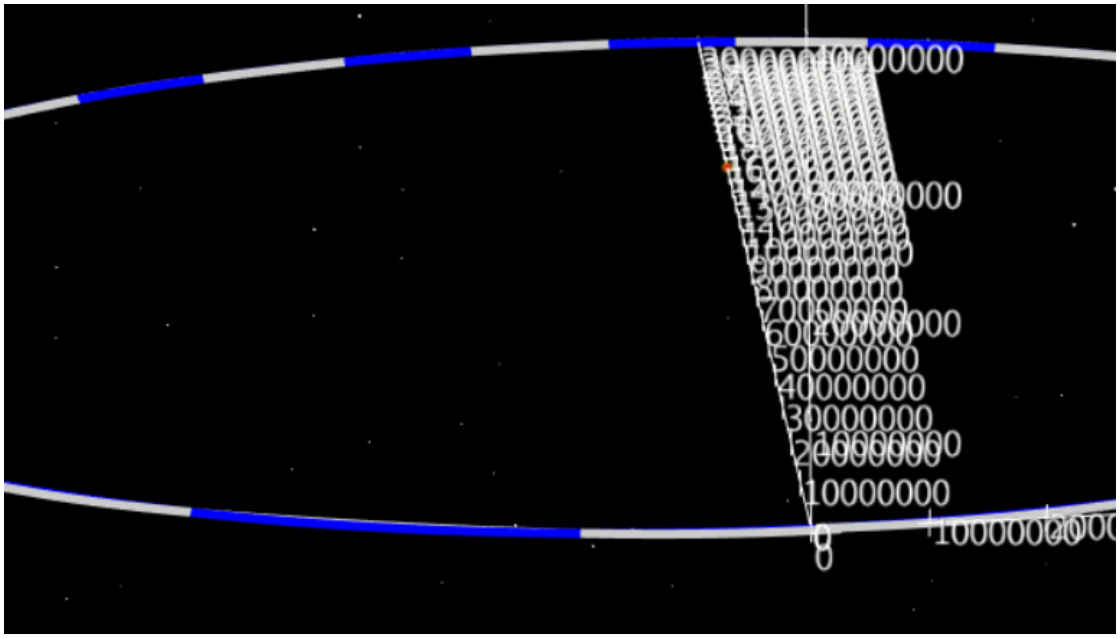

Figure 17: Sun-Moon-Earth 3D Position Graph

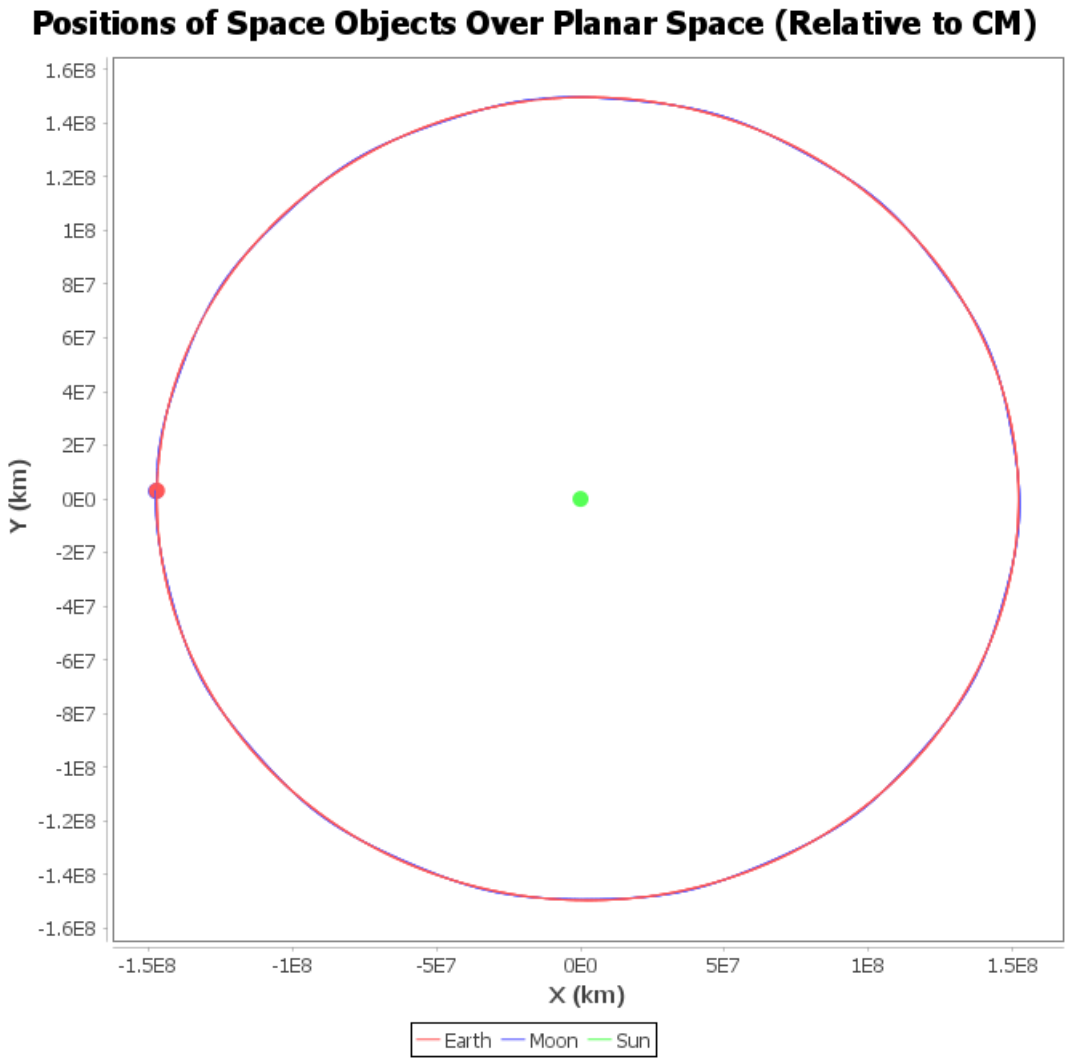

Figure 18: Sun-Moon-Earth 2D C.O.M Position Graph

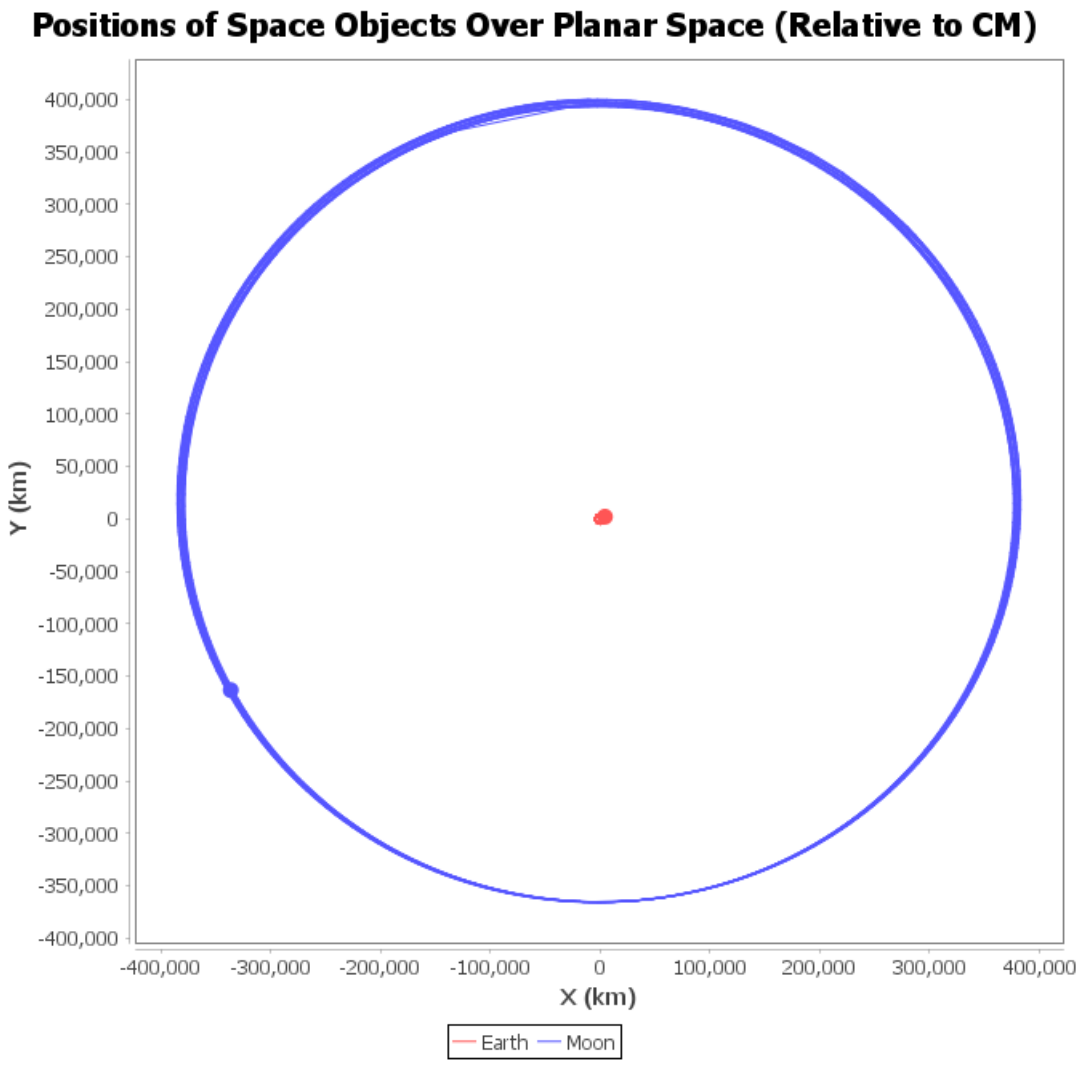

Figure 19: Earth-Moon 2D C.O.M Position Graph

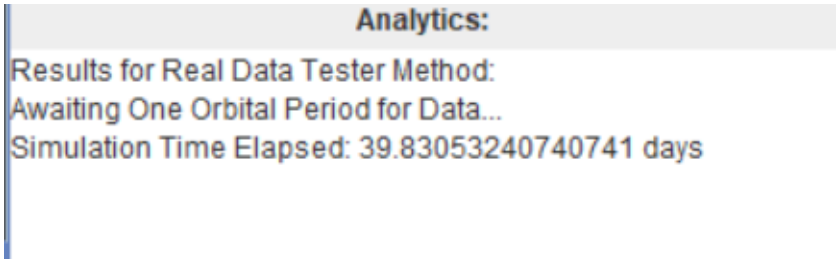

Figure 20: Sun-Earth-Moon Analysis Function Before 1 Cycle (365 days)

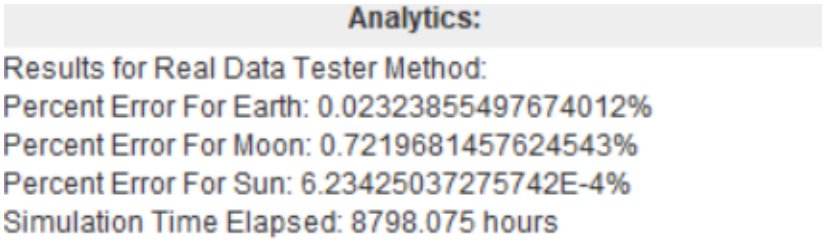

Figure 21: Sun-Earth-Moon Analysis Function After 1 Cycle (365 days)

5.3 Simulation Results of Earth-Sun-JWST System

5.3.1 Assuming Normal Earth Mass

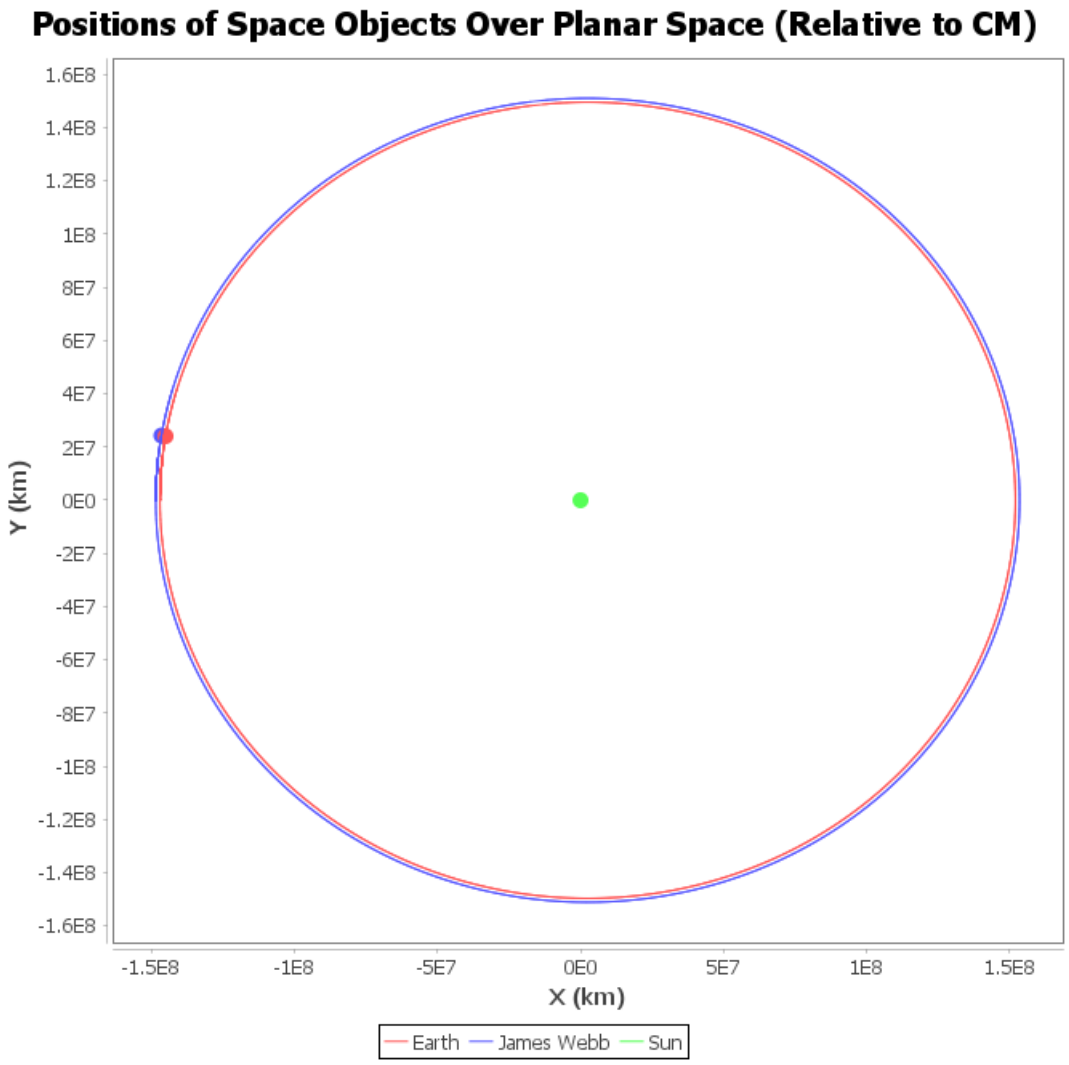

Figure 22: Sun-Earth-JWST 2D C.O.M. Graph

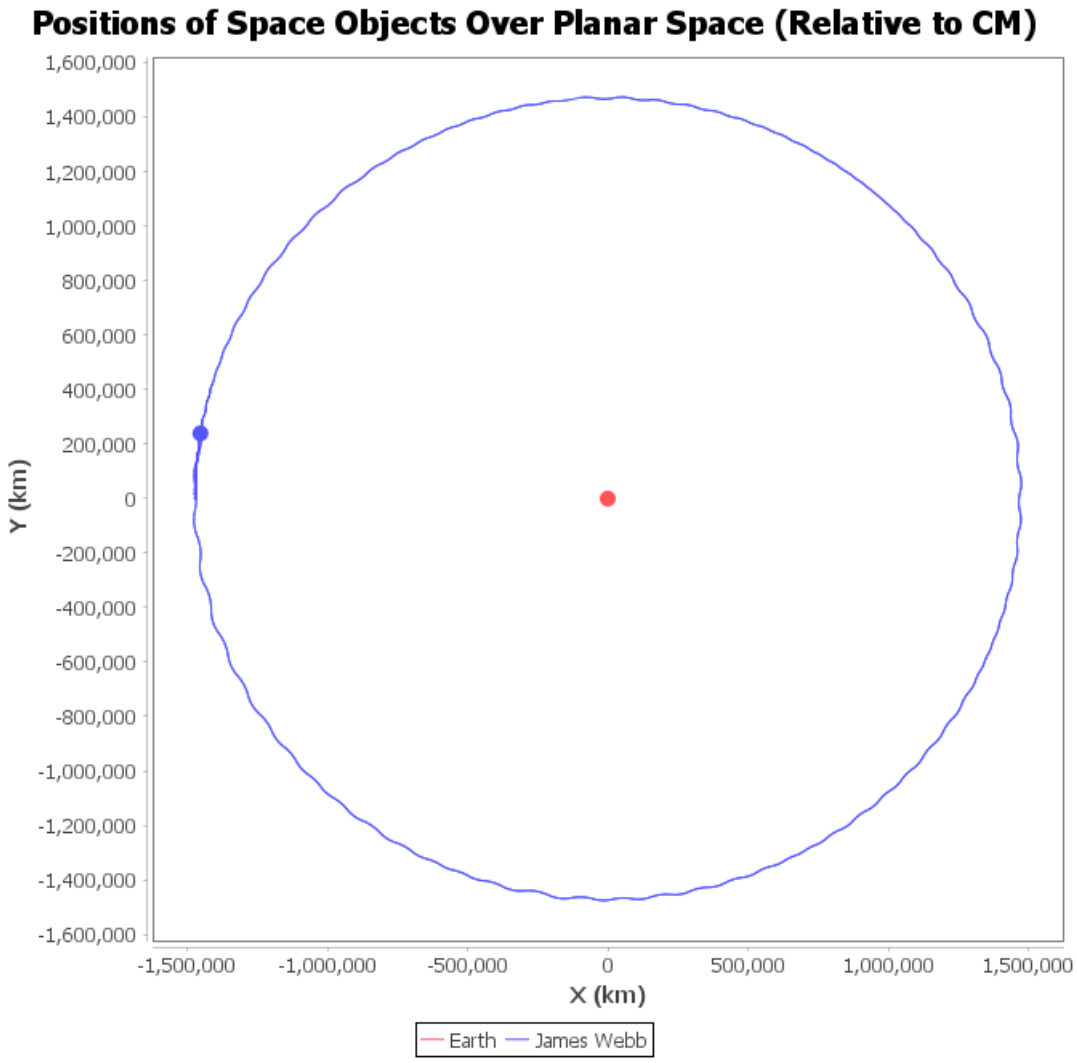

Figure 23: Earth-JWST 2D C.O.M. Graph

5.3.2 Assuming Zero Earth Mass

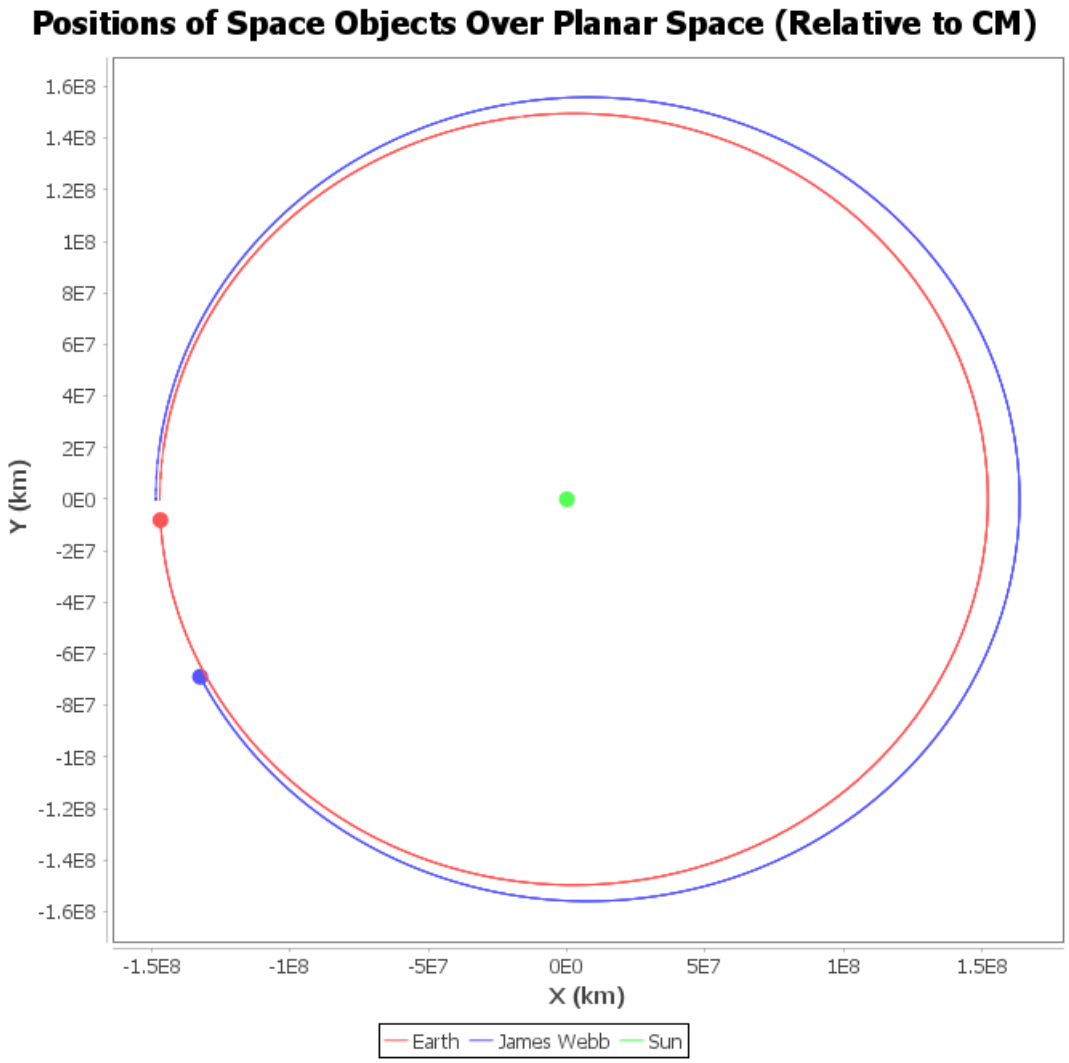

Figure 24: Sun-Earth(Zero Mass)-JWST 2D C.O.M. Graph

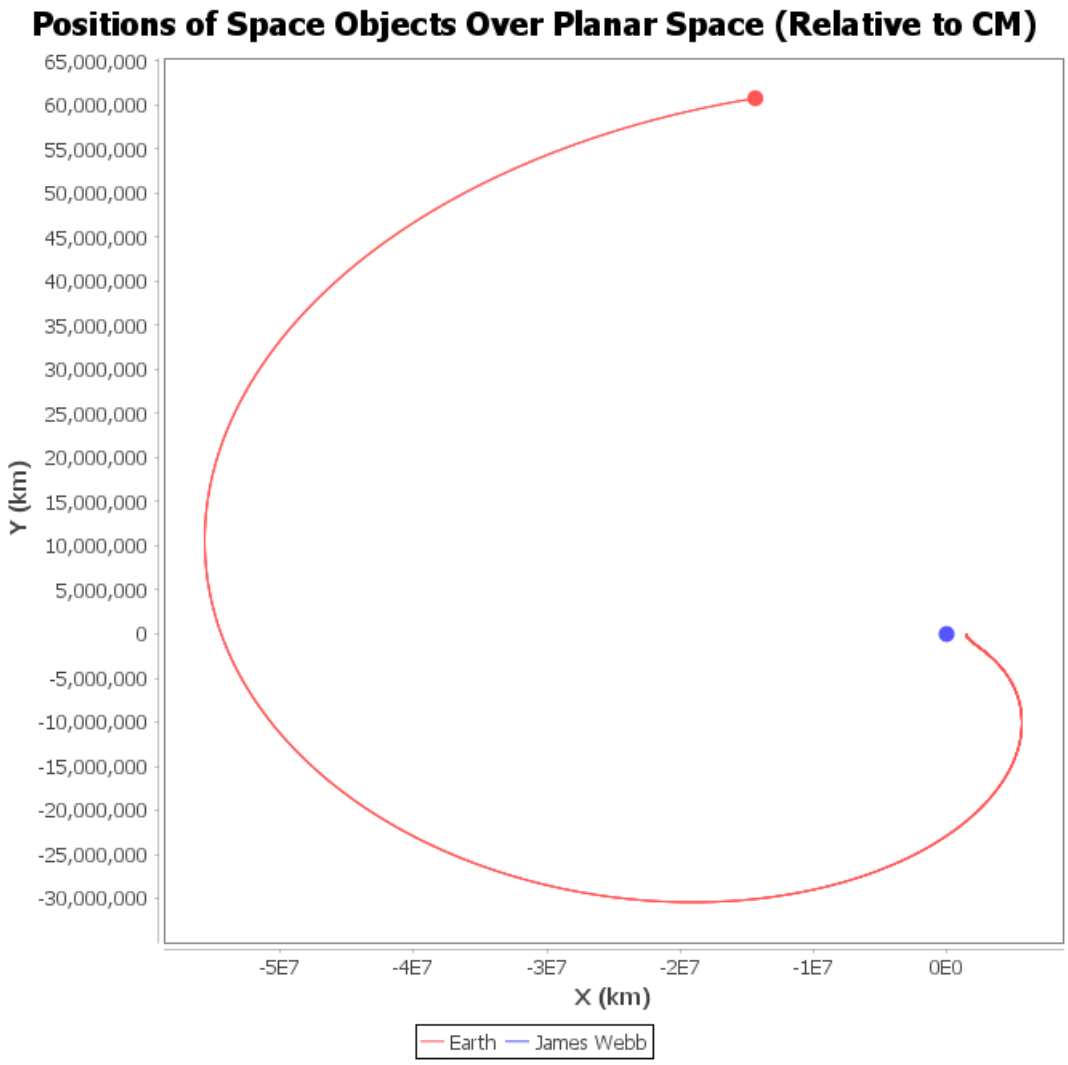

Figure 25: Earth(Zero Mass)-JWST 2D C.O.M. Graph

5.4 Simulation Results of "Chaotic System"

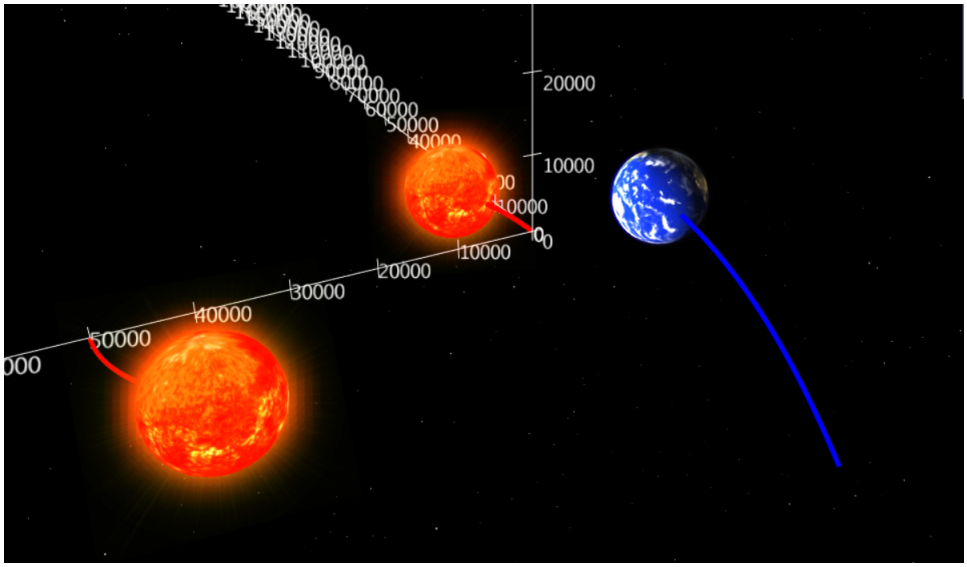

Figure 26: Chaotic Simulation Initial 3D Position Graph

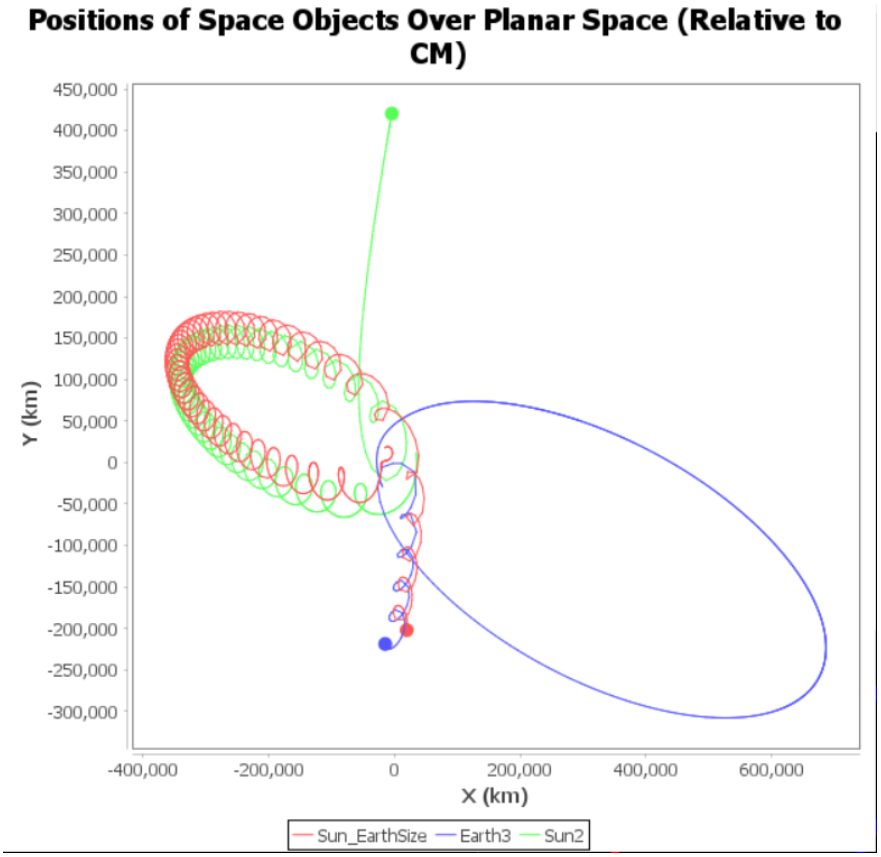

Figure 27: Chaotic Simulation After Some Time 2.D. Center of Mass

5.5 Simulation of Lorenz Attractor

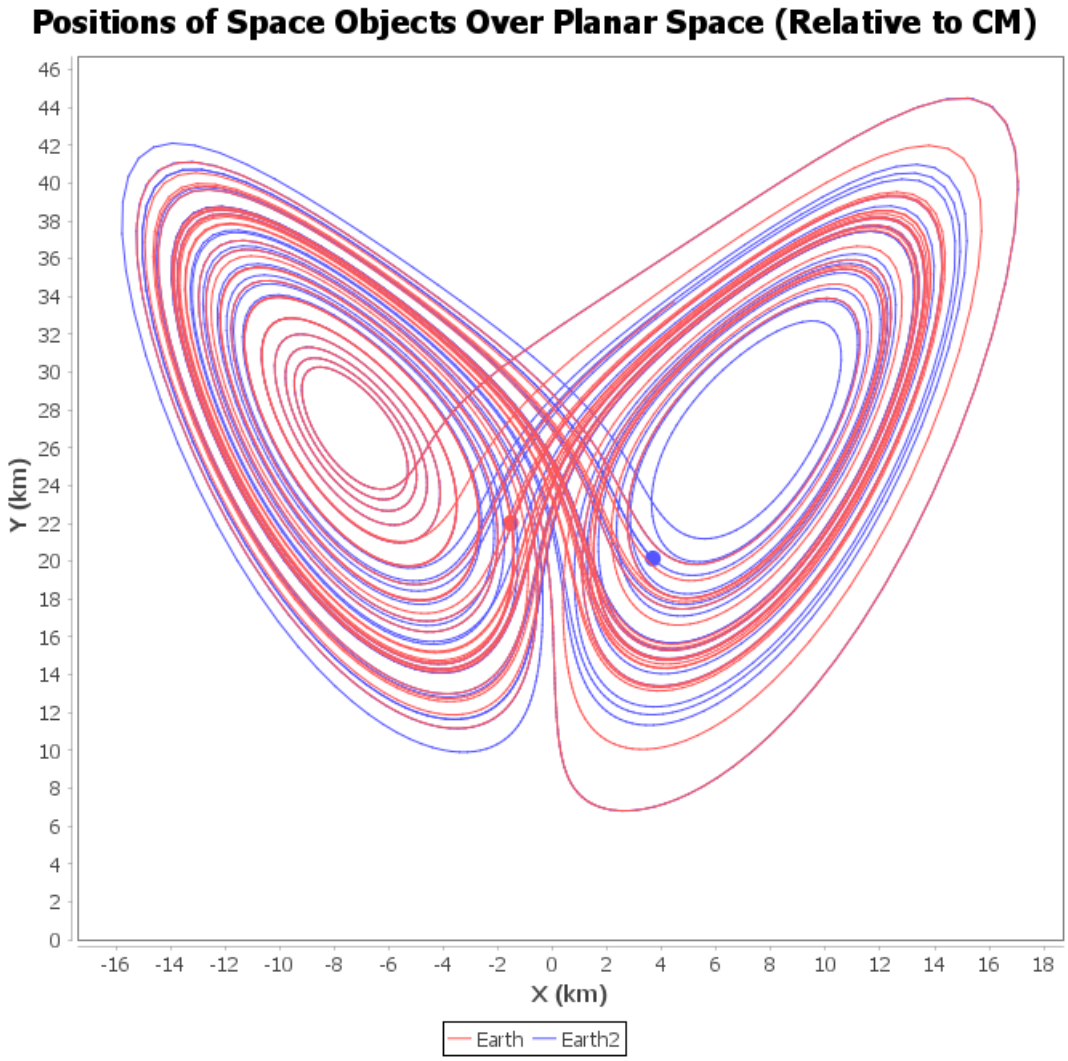

Figure 28: Lorenz Attractor Two Initial Conditions Initially Separated by Small Epsilon

```
Numerical LCEs: (0.8996115833410924, 0.005347161337803071, -14.533918674216743)
Analytical LCEs: (0.9058, 0.0, -14.572)
Perecent Error LCE1: 0.6831990129065666
Perecent Error LCE2: Infinity
Perecent Error LCE3: 0.26133218352495474
```

Figure 29: LCE Results for Lorenz Attractor Comparisons

5.6 2D Chaos Map for Earth-Sun-JWST with JWST as Probe

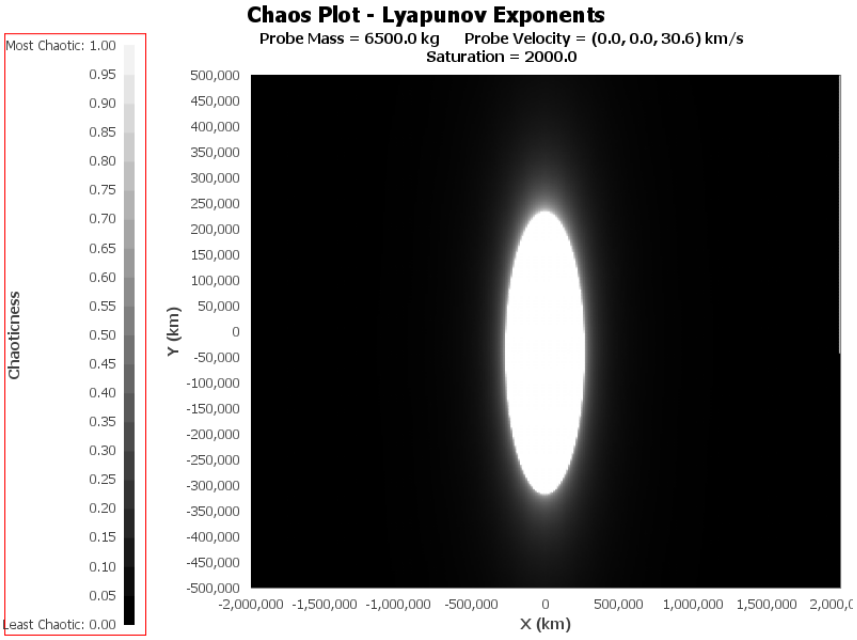

Figure 30: Lyapunov Exponent Plot Around Earth (Using Stable Velocity for L2 and L1). Graph Samples=200,000. Circle Samples=20. Delta=50km. Time-step=500s. Iterations=500

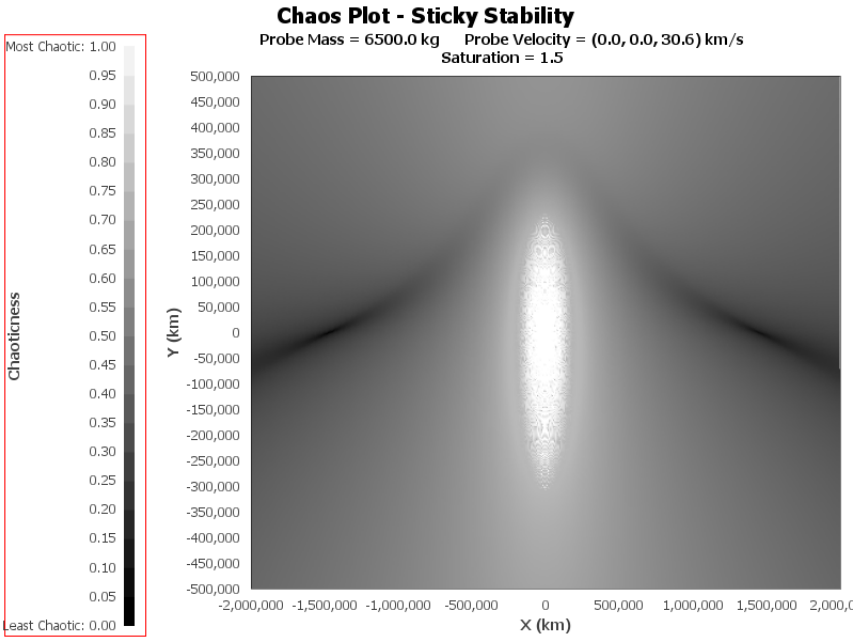

Figure 31: "Sticky Stability" Plot Around Earth (Using Stable Velocity for L2 and L1). Graph Samples=200,000. Time-step=500s. Iterations=500

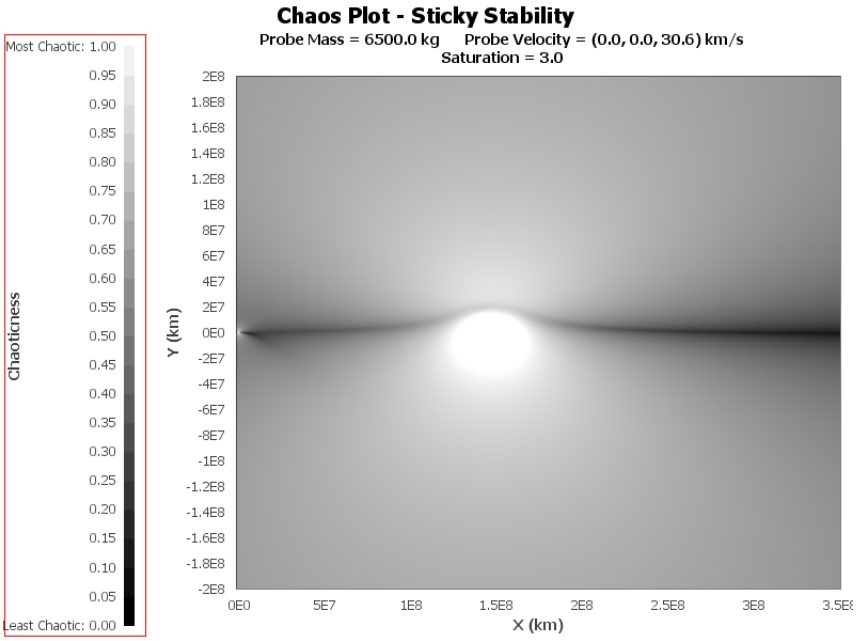

Figure 32: "Sticky Stability" Plot Across Entire Earth Orbit (Using Stable Velocity for L2 and L1). Graph Samples=10,000,000. Time-step=500s. Iterations=500

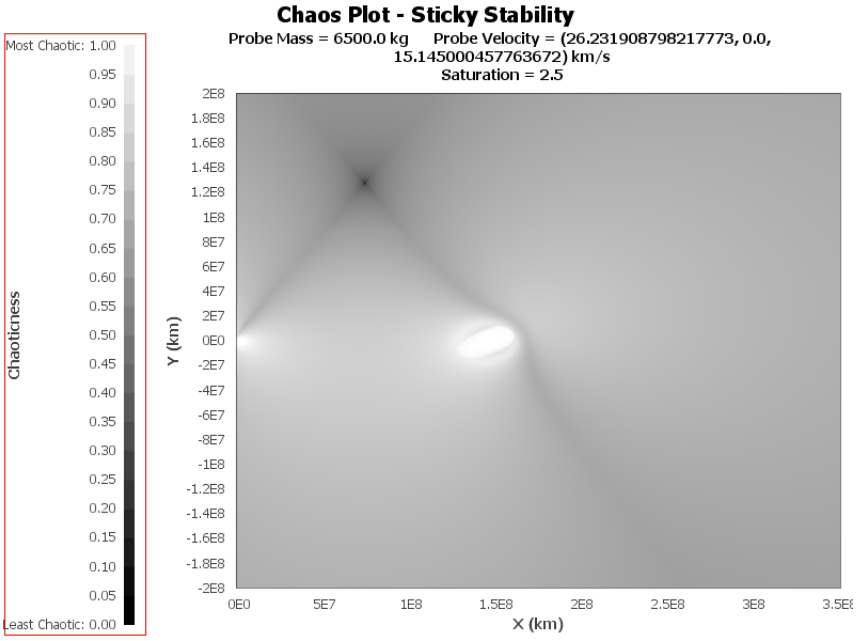

Figure 33: "Sticky Stability" Plot Across Entire Earth Orbit (Using Stable Velocity for L4). Graph Samples=10,000,000. Time-step=500s. Iterations=500

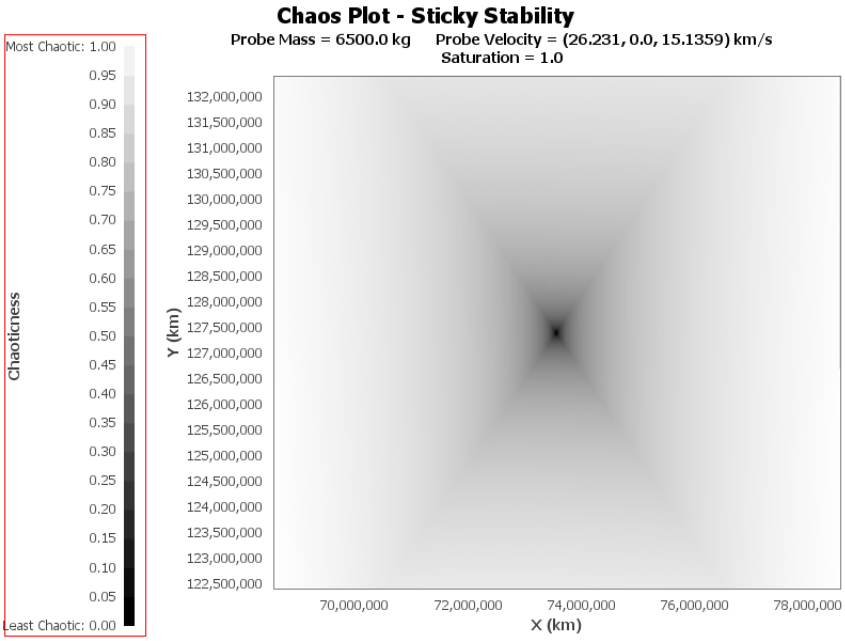

Figure 34: "Sticky Stability" Around L4 (Using Stable Velocity for L4). Graph Samples=200,000. Time-step=500s. Iterations=500

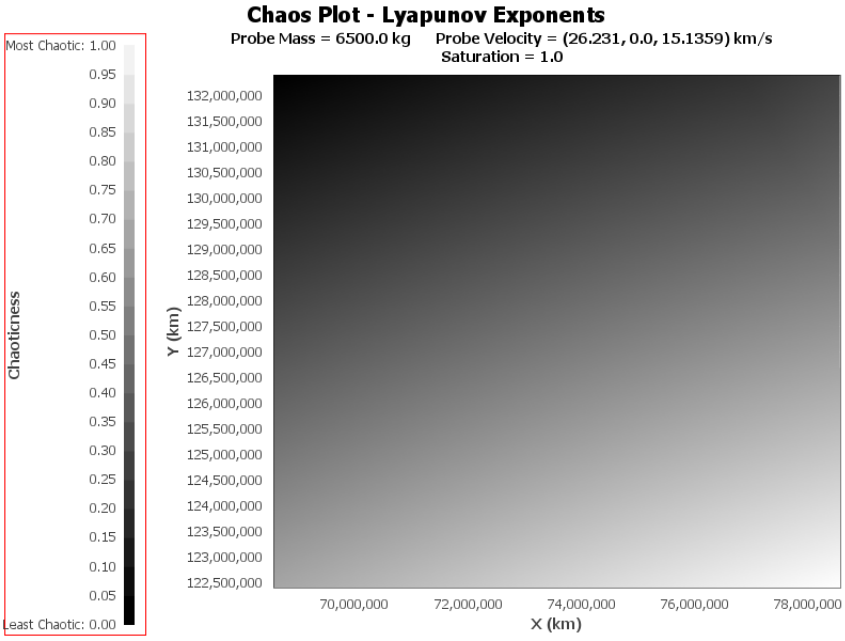

Figure 35: Lyapunov Exponent Plot Around L4 (Using Stable Velocity for L4). Graph Samples=200,000. Circle Samples=20. Delta=50km. Time-step=500s. Iterations=500

5.7 2D Chaos Map for "Chaotic System" with Massless Probe as Probe

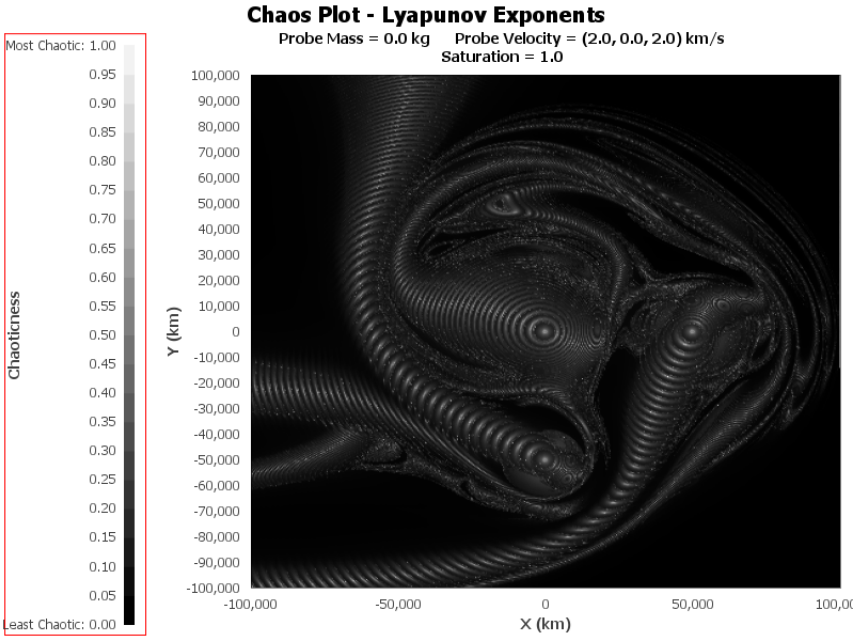

Figure 36: Chaotic System Lyapunov Exponent Plot. Graph Samples=200,000. Circle Samples=20. Delta=50km. Time-step=500s. Iterations=500

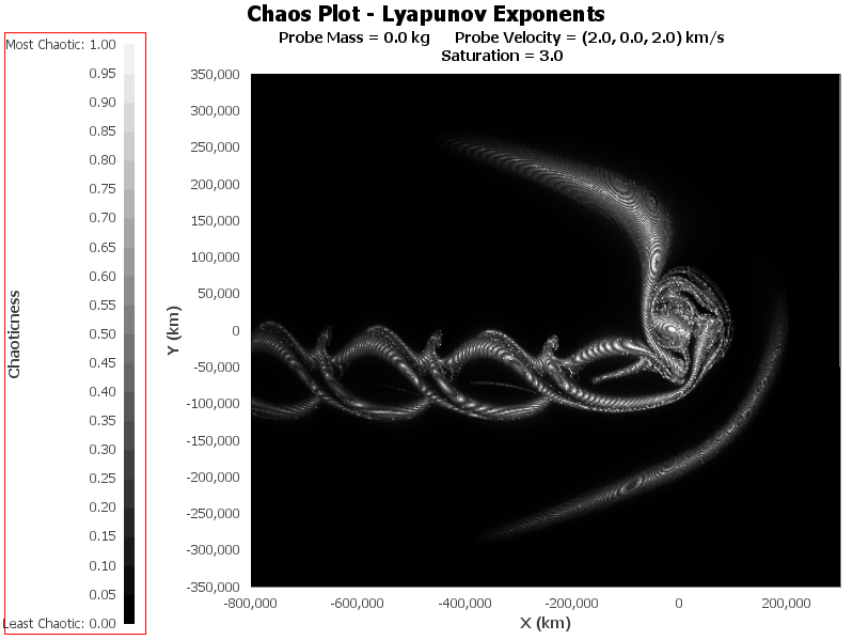

Figure 37: Zoomed Out Chaotic System Lyapunov Exponent Plot. Graph Samples=200,000. Circle Samples=20. Delta=50km. Time-step=500s. Iterations=500

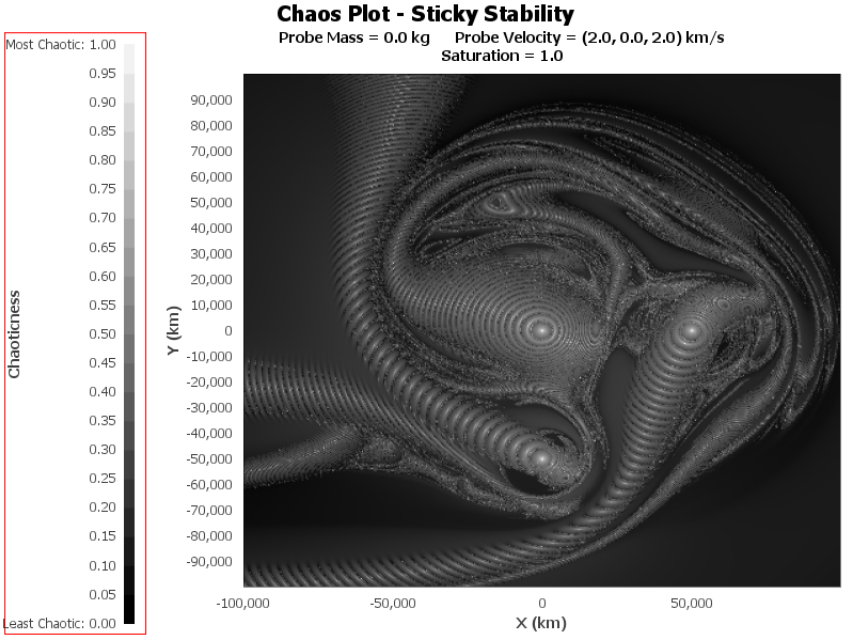

Figure 38: Chaotic System "Sticky Stability" Plot. Graph Samples=200,000. Time-step=500s. Iterations=500

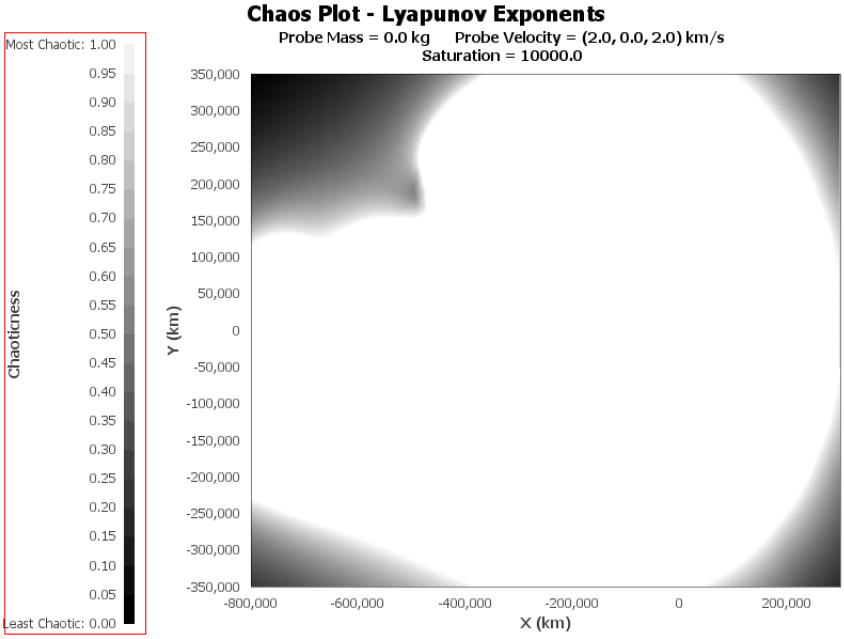

Figure 39: Chaotic System Lyapunov Exponent Plot with large Saturation. Graph Samples=200,000. Circle Samples=20. Delta=50km. Time-step=500s. Iterations=500

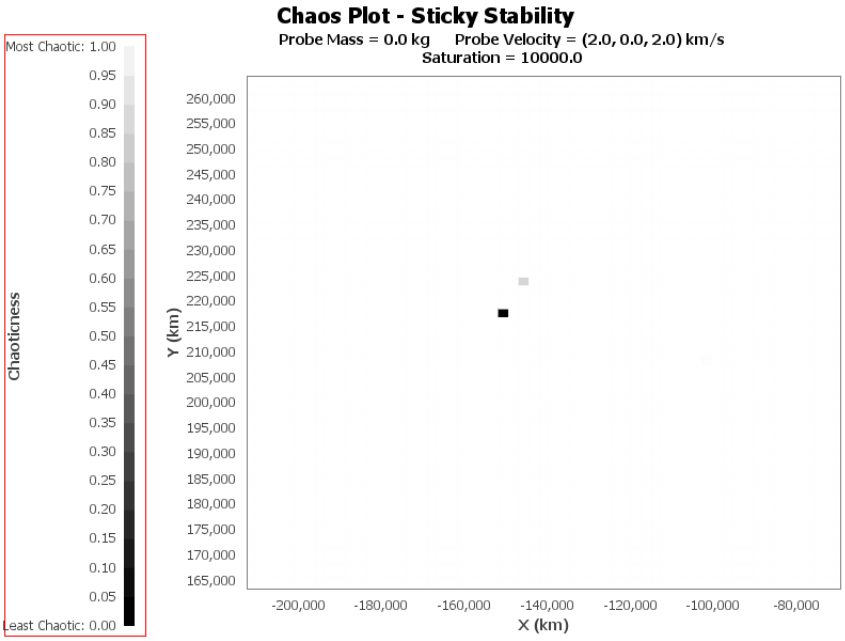

Figure 40: Chaotic System "Sticky Stability" Plot with large Saturation. Graph Samples=200,000. Time-step=500s. Iterations=500

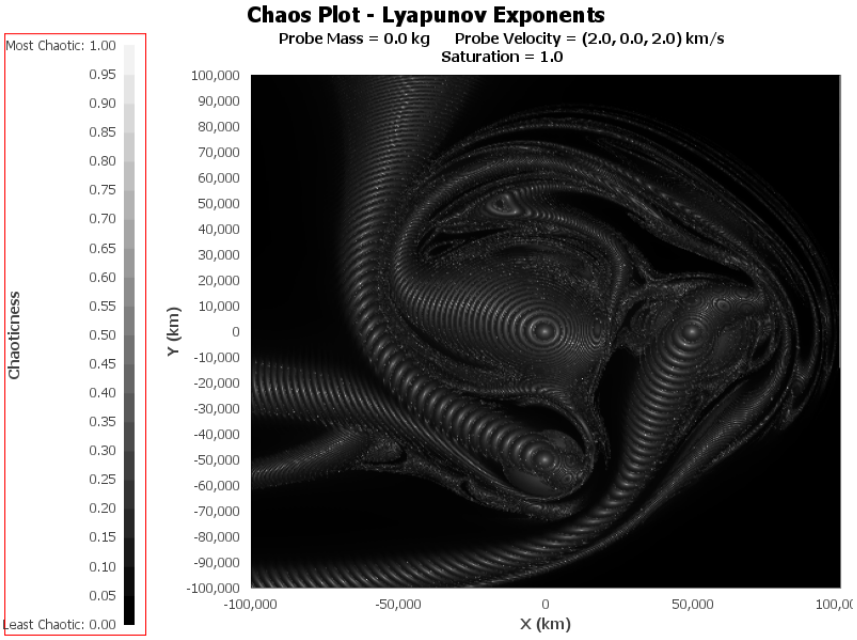

Figure 41: Chaotic System Lyapunov Exponent Plot. Graph Samples=200,000. Circle Samples=20. Delta=500km. Time-step=500s. Iterations=500

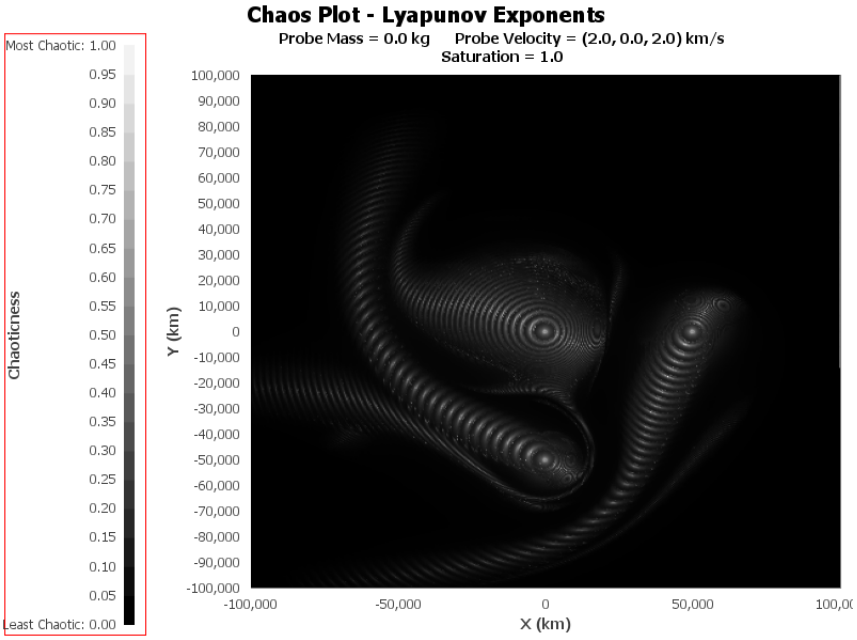

Figure 42: Chaotic System Lyapunov Exponent Plot. Graph Samples=200,000. Circle Samples=20. Delta=50km. Time-step=500s. Iterations=50

5.8 2D Chaos Map for "Equal Mass" System with Sun_EarthSize as Probe

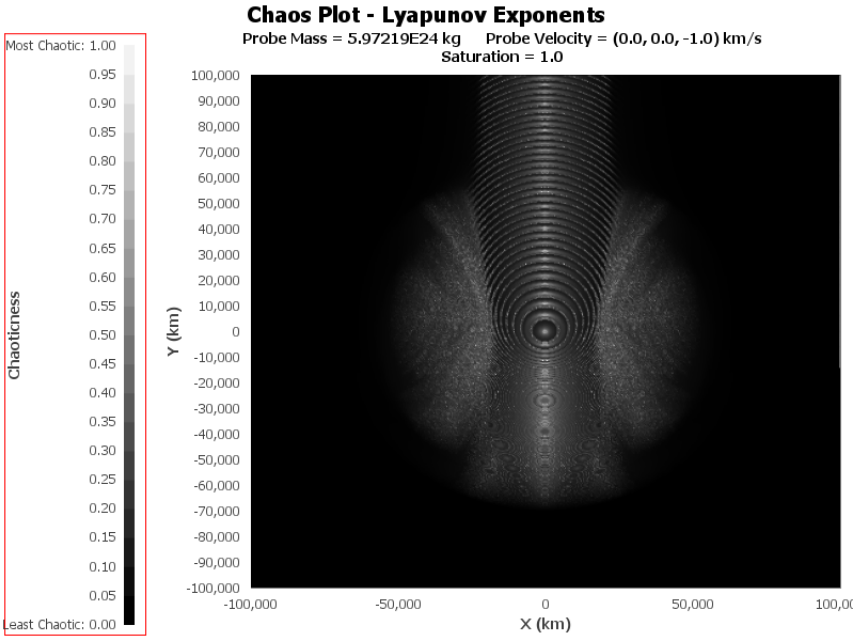

Figure 43: "Equal Mass" Lyapunov Exponent Plot. Graph Samples=200,000. Circle Samples=20. Delta=50km. Time-step=500s. Iterations=500

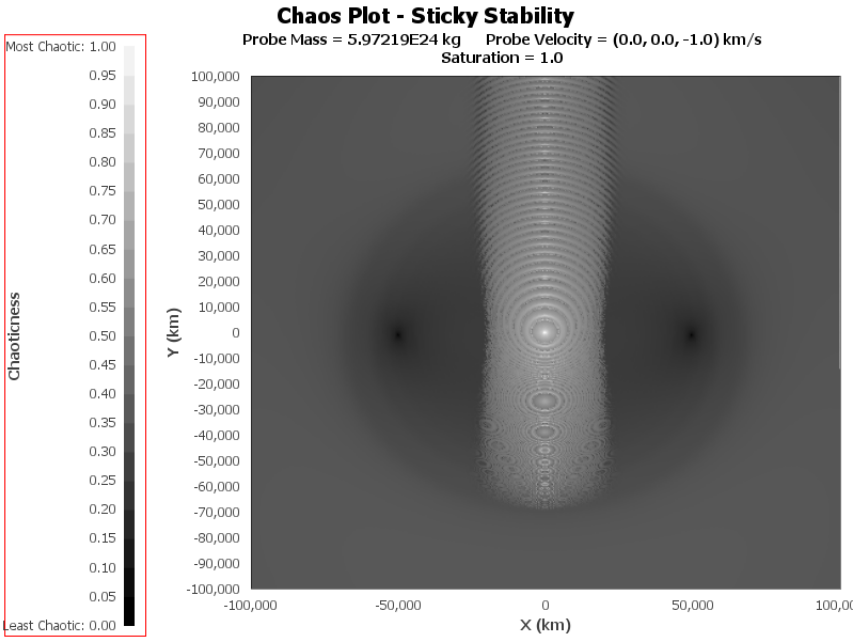

Figure 44: "Equal Mass" "Sticky Stability" Plot. Graph Samples=200,000. Time-step=500s. Iterations=500

5.9 Results for Lagrange Point Finder

```
Analysis of Lagrange Point Finder for L2:
Results of Sticky Stability Function:
Theoretical: (-1471603.356885612, 0.0, 0.0)
 Numerical: (-1468750.000000004, 0.0, -8.703864295966923E-10)
Percent Error: 0.19389442625672434

Associated Lyapunov Exponents:
The associated Largest Lyapunov Exponent for (-1468750.000000004, -8.703864295966923E-10, 0.0) is 4.1155505785772526E-8
The associated Largest Lyapunov Exponent for (4464.2857142812045, -8.703864295966923E-10, 0.0) is 0.018127502059364253
```

Figure 45: L2 Finder: Minimum Function Results From Figure 31: "Sticky Stability"

```
Analysis of Lagrange Point Finder for L4:
Results of Sticky Stability Function:
Theoretical: (7.356008E7, 0.0, 1.2740979596883123E8)
 Numerical: (7.357124221428667E7, 0.0, 1.2740979856690758E8)
Percent Error: 0.015174282418765842

Associated Lyapunov Exponents:
The associated Largest Lyapunov Exponent for (7.357124221428667E7, 1.2740979856690758E8, 0.0) is 1.0207914750117866E-8
The associated Largest Lyapunov Exponent for (6.857124221428572E7, 1.2242098424923404E8, 0.0) is 1.1697483451029391E-8
```

Figure 46: L4 Finder: Minimum Function Results From Figure 34: "Sticky Stability"

```
Analysis of Lagrange Point Finder for L2:
Results of Lyapunov Exponent Function:
Theoretical: (-1471603.356885612, 0.0, 0.0)
 Numerical : (-1993690.8517350159, 0.0, 498417.7215189847)
Percent Error: 35.47746017339276

Associated Lyapunov Exponents:
The associated Largest Lyapunov Exponent for (-1993690.8517350159, 498417.7215189847, 0.0) is 2.031442502843173E-8
The associated Largest Lyapunov Exponent for (5.155015969648957E-9, 49050.63291139109, 0.0) is 0.029432741685918223
```

Figure 47: L2 Finder: Minimum Function Results From Figure 30: Lyapunov Exponents

```
Analysis of Lagrange Point Finder for L4:
Results of Lyapunov Exponent Function:
Theoretical: (7.356008E7, 0.0, 1.2740979596883123E8)
 Numerical : (6.857124221428572E7, 0.0, 1.3239861288458113E8)
Percent Error: 6.781990701633663

Associated Lyapunov Exponents:
The associated Largest Lyapunov Exponent for (6.857124221428572E7, 1.3239861288458113E8, 0.0) is 8.95620789517441E-9
The associated Largest Lyapunov Exponent for (7.854892078571619E7, 1.2242098424923404E8, 0.0) is 1.1697483451029391E-8
```

Figure 48: L4 Finder: Minimum Function Results From Figure 35: Lyapunov Exponents

5.10 Results for Optimal Velocity Finder

```
Theoretical L1 Position: (1471603.356885612, 0.0, 0.0)
The most stable velocity for probe in Lagrange Point L1 is: (0.0, 0.0, 29.994) +/- 0.001

Theoretical L2 Position: (-1471603.356885612, 0.0, 0.0)
The most stable velocity for probe in Lagrange Point L2 is: (0.0, 0.0, 30.595) +/- 0.001

Theoretical L3 Position: (2.942405040509E8, 0.0, 0.0)
The most stable velocity for probe in Lagrange Point L3 is: (0.0, 0.0, -30.072) +/- 0.001

Theoretical L4 Position: (7.356008E7, 0.0, 1.2740979596883123E8)
The most stable velocity for probe in Lagrange Point L4 is: (26.231, 0.0, 15.135) +/- 0.001

Theoretical L5 Position: (7.356008E7, 0.0, -1.2740979596883123E8)
The most stable velocity for probe in Lagrange Point L5 is: (-26.23, 0.0, 15.153) +/- 0.001
```

Figure 49: Given Known Lagrange Points, Finds the Velocity That Returns The Minimum "Sticky Stability" Value (Best Velocity For Object To Have in Lagrange Point)

6 Discussion and Conclusions

6.1 Two-Body Analytical Tester Function

6.1.1 Earth-Sun System Two-Body Test

The Earth-Sun System was created for the sole purpose of verifying the numerical n-body solver for two-body systems. Specifically, the Earth-Sun was meant to represent a system where one mass is much larger than the other mass, which can be solved using the analytical equations from Section 3.1.3 (Equations 40, 45). The data for the analytical equation can be seen in Figure 13. The total time that the simulation was run over was 8,260 hours, or approximately 365 days (1 cycle). According to Section 3.3, Expected Results, it was expected that the aggregate average percent error should be less than 1%. The actual average percent error for Earth was 2.47E-8%, which is significantly smaller than what the expected was. This was run for a time-step size of 500 seconds. It should also be noted that Figure 13 reports the analytical eccentricity of the Earth, which is 0.01671. According to NASA, the actual eccentricity of the Earth is 0.01671. Therefore, the percent error between the analytical and numerical eccentricity of the Earth is 0%. The numerical and analytical positions of both the Earth and the Sun are reported as well. Only the percent error for the Earth's position is reported because the Sun was stationary in the assumption. The 3D output of the simulation can be seen in Figure 12.

6.1.2 "Equal Mass" System Two-Body Test

The "Equal Mass" system was created to test the n-body solver against the analytical solution for similar masses. The "Equal Mass" solution was created to test when the masses are equal since the assumption that one body is much larger than another is not accurate for some systems. For the Equal Mass system, the percent error for the distance between the objects and the percent error for the center of mass position is reported. The reason those percent errors are reported as opposed to the percent error in the positions is that the objects are close to each other and the percent error as they get further from the origin would get smaller just because the distances would be so large. Using the distance between them and the location of the C.O.M. gives a good error value for the motion of the system. As can be seen, both errors are much less than 1%, which was the requirement to proceed to the 2D Chaos Map and the expected result. If this same simulation is run for Euler's Approximation for the same time period, the error for the distance between objects is approximately 3%. That is one of the major reasons I decided to implement RK4 for all of the numerical calculations. Accumulating 3% error average after only 465 hours is a lot. The reason it is so large is that the objects are close to each other and are moving very fast. RK4 is significantly better at conserving energy and momentum, so it manages to keep the objects from "falling" into each other.

The 2D center of mass plot can be seen in Figure 14. As shown, the center of mass plot shows two objects

orbiting. The eccentricity of those orbits are both approximately 0.197, which is the eccentricity of a slight ellipse. When the objects move about the center of mass, they orbit in ellipsoid patterns given their initial conditions (Provided in Appendix B). It is interesting how both objects start with different initial velocities, but their eccentricities and orbits look identically reversed on the Center of Mass plot.

The 3D plot can be seen in Figure 15. The difference in the paths of the objects is more apparent in the absolute position graph. Seeing how the path of the Sun_EarthMass and Earth are not identical in the absolute position graph, but appear to be identical in the C.O.M. plot is very interesting.

6.2 Validity of N-Body Numerical Solvers For Two-Bodies

I found that the average error accumulated over 365 days for the Earth-Sun system and the Equal Mass system was less than 1% for a time-step size of 500s for RK4. Based on the results of the two two-body system analytics, only RK4 measures position accurately within 1% given a timestep size of 500s. The reason is that in the "Equal Mass" system, Euler's Approximation accumulates over 3% error for the same time period that RK4 only accumulates 0.006% error. 3% error is larger than what I expected and larger than what I would want when attempting to find stability. Therefore, I decided to do all of the numerical integrations using RK4 as soon as I validated that the RK4 function was implemented correctly.

6.3 Earth-Sun-Moon System Three-Body Test

In order to verify that the n-body numerical solver worked, I created the Earth-Sun-Moon system to see if the numerical solver would get less than 1% error for each object using the real-data tester function described in Section 4.5. As can be seen in Figure 20, before 1 cycle is completed, the real-data tester method just reports that not enough time has passed and it reports the simulation time elapsed. Figure 21 shows what the real data tester method outputs after 1 cycle is completed. It can be seen that the percent error for the Earth is 0.02%, the percent error for the Sun is 6.23E-4%, and the percent error for the Moon is 0.72%. As expected, the Moon had the largest percent error and the Sun had the smallest percent error. Using RK4 and a time-step size of 500s, the Moon's percent error over 365 days was barely under the requirement of 1%, however, it did pass. The results also were within expectations. The moon is small and is very close to the Earth. As seen from the "Equal Mass" simulation, when objects are close in mass and position, the error is much larger. For example, Euler's Approximation reported 40% error for the same time period for the Earth. RK4 was good enough such that it got the Moon's position error under 1% for the entire cycle.

Figure 17 shows the output of the 3D simulation for the Earth-Sun-Moon system. The grey path and blue path alternates visibility because the Moon revolves around the Earth (blue path). The center of that ellipse, while not super visible, is the Sun. Figure 18 shows the same path in the C.O.M. frame. Now, since the center of mass position is

almost the sun position, the graphy looks almost identical to the one seen in Figure 17. The revolution of the Moon around the Earth is not very apparent in that graph. Figure 19 shows the path of the Moon and the Earth in the center of mass frame without the Sun included. The revolution of the Moon around the Earth is very apparent in that Figure. It can also be seen that the motion of the Moon around the Sun gets wider lines as the simulation progresses because of accumulated errors. In order to reduce the inaccuracy of the Moon's motion, the time-step size could be reduced; however, the total error was less than 1% for the Moon, so the time-step size of 500s was kept.

6.4 Validity of N-Body Numerical Solvers For Three-Bodies

Based on the results of the three-body system real-data test, once again, only the RK4 function was valid for a timestep size of 500s. Therefore, the n-body numerical integrator was validated to work for three-body systems or less with accuracy and precision.

6.5 Results of Earth-Sun-JWST Simulation

One of the main simulation goals of this project was to simulate the motion of the JWST in the Lagrange Point, L2. As seen in Figure 1, L2 sits behind the Earth during its entire orbit. As can be seen in Figure 22, the JWST sits behind the Earth for the entire orbit around the Sun. Figure 23 shows the center of mass graph with just the Earth and JWST. It might appear as if the JWST is orbiting the Earth, but that is not the case. It looks like that because the JWST is always behind the Earth in the same spot, so when it rotates around the Sun, the JWST appears to rotate around the Earth in the center of mass frame. Figure 23 also shows how the JWST's position becomes more unstable over time. That is because L2 is only semi-stable. Over time, JWST would fall out of L2. That "squiggle" motion would get more aggressive after more time passes, which would eventually lead to a divergence in pathing.

In order to show that the effect of the Earth matters in the behavior of L2 and the JWST, another simulation was completed with the Earth having zero mass. The position graph for that can be seen in Figure 24. As can be seen, the Earth takes the same path that it did before; however, the JWST immediately revolves around the Sun in a larger orbit with a longer period. That is because without the effect of the earth, the JWST's position reduces to the regular centripetal acceleration dependence where objects with larger velocities and larger radii have larger periods. Figure 25 shows the C.O.M. frame for just the Earth (Zero Mass) and the JWST. This graph just shows that the distance between the Earth and the JWST increases over time as opposed to remaining constant. The reason the graph is centered on the JWST is that the mass of the Earth in this scenario is zero.

6.6 Results of Lorenz Attractor Simulation

In order to verify that the Lyapunov Exponent function worked, I tested it against the known Lyapunov Exponent values for the Lorenz Attractor as detailed in Section 4.6. The numerical LCES and analytical LCEs are shown in Figure 29 as well as their percent errors. The percent error for LCE1 was 0.68%, which is a really good approximation. The percent error for LCE2 was Infinity. The reason it was Infinity was that the expected value was 0, so the percent error formula would divide by 0, which is being assumed to be infinity by the Java interpreter. This value should be ignored since, as can be seen by the numerical calculation, LCE2 is also really close to being zero, even if that "closeness" cannot be quantified using percent error. LCE3 had a percent error of 0.26%, which is also a really close approximation. The results of the LCES showed that the numerical Lyapunov Exponent finder was correct, which meant that it would also work accurately for the n-body celestial body simulator. For the purpose of simplifying the computational cost, only LCE1 is calculated for the n-body simulator as noted in Section 3.1.4. The 2D position graph for two close initial conditions after time can also be seen in Figure 28. Figure 28 shows how two slightly different initial conditions lead to entirely different positions after a long enough time.

6.7 Results of 2D Chaos Map for "Chaotic System" System

The "Chaotic System" simulation (Initial Conditions Provided in Appendix A) was created because I wanted a system where a lot of objects were close together and a three-body system is known to be chaotic. The "Chaotic System" was also created to determine the differences between the Lyapunov Exponents and the "Sticky Stability" method outputs. It was also used to determine the effect of manipulating $\delta(0)$ in the Lyapunov Exponent calculation was well as the number of iterations completed. The initial positions and motions of the celestial objects over a short period of time can be seen in Figure 26. The positions and motions of the celestial objects over a long period of time can be seen in Figure 27.

6.7.1 Analysis of Differences Between Lyapunov Exponents and "Sticky Stability" Values

The Lyapunov Exponent output of the "Chaotic System" can be seen in Figure 36. The number of graph samples, or position samples taken in the graph, was selected to be 200,000 (enough samples to come out as 4K resolution when displayed in a paper). The number of circle samples, or number of probes tested around the original probe location with an initial separation distance of $\delta(0)$, is equal to 20 (enough samples to conserve the shape of a circle without costing too much in computational time). The $\delta(0)$ was selected to be 50km because 50km is much smaller than the distances between the objects, but not so small that the numerical integrator might get numbers that it rounds to 0. The time-step size was selected to be 500s because that time-step based the two-body and three-body tests using RK4. Finally, the number of iterations was selected to be 500 because that was the required number of

iterations to find Lagrange Point, L2, which is discussed more in Section 6.9.1. It should also be mentioned that the mass-less probe is being initialized at each position with an initial velocity of (2,0,2) km/s. The most important detail in the graph is that at the position of the large celestial objects, the Lyapunov exponent for the mass-less probe is at its maximum. This makes sense because if a mass-less probe is placed near massive objects, small changes in initial conditions would be most affected. The small formations further from the celestial "attractors" are unstable regions created from the effect of the different gravitational components of the objects. The zoomed-out plot of Figure 36 with a saturation value of 3 can be seen in Figure 37. Based on the motions of the objects depicted in Figure 27, it seems that the "chaoticness" has a sort of loop motion in the same direction as the motion of the objects.

The Lyapunov Exponent output of the "Chaotic System" with $\delta(0)$ = 500 km can be seen in Figure 41. The difference between the output of Figure 36 and Figure 41 is very minimal even though Figure 41 was calculated with a delta 10x larger. The reason is that 500 km is still a small delta compared to the distances between the objects. Therefore, 50 km is a valid delta because even 500 km works for a "Chaotic System" where the distances between the objects are all about 50,000 km.

The Lyapunov Exponent output of the "Chaotic System" with Iterations = 50 can be seen in Figure 42. The differences between Figure 42 and Figure 36 are very apparent. The main reason for their differences is that Figure 42 is calculated using a much shorter total time. The more time iterated over, the more information about the Lyapunov Exponents is obtained. As can be observed, near the "attractors" the data is practically identical. As the distance grows, the information becomes less known with less iterations. Therefore, 500 iterations is effective for testing near gravitational "attractors", but it may have less information about points located far away from attractors. For computational efficiency, 500 iterations was selected because it included more information further away from the attractors, when compared to 50 iterations, without significant computational costs.

The "Sticky Stability" value output can be seen in Figure 38. The number of circle samples and delta are not included in the description of Figure 38 because the "Sticky Stability" method does not require these parameters as detailed in Section 3.1.6. Now, as mentioned earlier, the "Sticky Stability" method and the Lyapunov Exponent method have entirely different defintions of "stability" and "chaotic". Lyapunov Exponents measure the divergence in slightly different initial conditions for a probe over time. "Sticky Stability" measures the divergence of a probe from the other celestial objects in the system over time (so no circle of initial conditions). However, Figure 38 looks surprisingly similar to the Lyapunov Exponent graph in Figure 36. The major difference seen between these two graphs would be the "depth". The Lyapunov Exponent graph shows large contrast between the points, which means that the difference between the largest values and smallest values is large. The "Sticky Stability" plot shows low contrast between the points, which means that the difference between the largest values and smallest values is small. Surprisingly they are similar, but they also demonstrate some major differences.

The major differences between the two plots are highlighted in Figures 39 and 40. When the saturation value

is blown up to a very large number, most of the large values are saturated to ”full white” such that the smallest end up being more obvious. The saturated plot of Lyapunov Exponents indicates that the smallest value is somewhere along the edge of the graph. The saturated plot of the ”Sticky Stability” method indicates that the smallest value is a single point somewhere near the attractors. This makes sense because Lyapunov Exponents characterize ”stability” as having small divergence between closely separated probes. If a probe is placed significantly far away from the attractors, then the force is smaller, which also means that the separation between the close probes would also be smaller. For the ”Sticky Stability” method, ”stability” is characterized as a position where the probe ”sticks” to the other objects. Therefore, going really far away would mean that the distance to the other objects would greatly increase over time, which would be characterized as unstable. Therefore, the ”Sticky Stability” method would converge to some point near the attractors. The results for that are within expectations.

6.8 Results of 2D Chaos Map for ”Equal Mass” System with Sun_EarthSize as probe

As mentioned in ”Expected Results”, it was expected that a two-body system would indicate that it is not chaotic. The Lyapunov Exponent output of the ”Equal Mass” System (Initial Conditions Provided in Appendix B) shows the opposite (shown in Figure 43). It shows that moving the Sun_EarthSize is more stable further away from the Earth object (positioned at 0,0). The ”Sticky Stability” method shows similar chaotic behavior, but it indicates that the most stable points are to the left and right of the Earth (can be seen as the darkest points) seen in Figure 44. This completely contradicts the known assumption that two-body systems are non-chaotic; however, that also assumes that the previous numerical definitions of Chaos are complete. Both Lyapunov Exponents and the ”Sticky Stability” method aim to quantify chaos; however, the way the methods were executed was insufficient for quantifying chaos for non-chaotic systems. Lyapunov Exponents are built on the assumption that the system being analyzed behaves chaotically, which means that the divergence between the objects over time can be characterized exponentially, at least over time. For a two-body system, the distance over time is cyclical, not exponential. The exponential fit is incorrect. While the separation distance between nearby initial conditions in a two-body system is bounded (no divergence), it still oscillates due to the differences in periods for the divergent conditions. As mentioned in the introduction, if the oscillations were around the initial separation distance, the Lyapunov Exponent would converge to 0 without a lot of iterations, which would indicate asymptotically stable behavior. For unknown reasons (possibly due to differences in period or wrong reference frame), the oscillations occur around a value slightly greater than the initial separation distance. Therefore, the Lyapunov Exponent ends up being approximately 0, but greater than 0, at every point in the 2D Chaos Map, but only after an extremely long integration time period. That means that in the 2D map, each value converges to a number (a slightly different number) close to 0 at a slightly different rate. The Lyapunov Exponent is

also slightly positive since the average separation distance is greater than the initial separation distance.

The "Sticky Stability" method of quantifying chaos looks for points where the objects "stick" together. In a two-body system, while it may not act "chaotic" in the sense that they experience exponential divergence, there are positions where the Earth_SunMass can be placed such that the C.O.M. orbit for both objects overlap, which would make the distances between the objects nearly constant. The initial conditions in Appendix B for the "Equal Mass" system produce a C.O.M. orbit that looks like two ellipses orbiting about different points as seen in Figure 14. That behavior will be consistent throughout time. It can be seen from Figure 14 that the setup would cause the distance from the objects to vary over time. There would be an upper and lower bound, but that variation is what allows "Sticky Stability" to find "more stable" positions even if the system itself is non-chaotic. The "Sticky Stability" method also does not follow the same definition of stability and "chaoticness" that Chaos Theory uses to establish that a two-body system is non-chaotic.

This shows that quantifying chaotic behavior can be difficult, even with the use of Lyapunov Exponents. Chaotic systems have exponential divergences. Differences in initial conditions in the two-body system lead to oscillations in the separation distance (that are bounded and are not exponentially diverging or converging). The Lyapunov Exponent, integrated over a short time, "interprets" that small difference in average separation distance and initial separation distance as exponential divergence. After the average separation distance converges (the oscillations in the separation distance yield a stable average), more iterations will cause the Lyapunov Exponent to converge from a slightly positive number to 0 very slowly. Plotted in a normalized graph, it appears to depict chaotic behavior; however, that is not actually what is happening.

6.9 Results of 2D Chaos Map for Earth-Sun-JWST System

6.9.1 Identification of L2

The primary goal of this project was to correctly find Lagrange Points using Lyapunov Exponents and the "Sticky Stability" method, specifically to find L2, which is where the JWST sits.

The Lyapunov Exponent plot around the Earth can be seen in Figure 30. As it can be seen, there are no Lagrange points seen anywhere in the graph. Both L1 and L2 should, theoretically, sit around 1.5 million km on both sides of the Earth; however, what is there is complete black. Raising the saturation just increases the size of the instability near Earth, but it never reveals stability at any particular position. As mentioned earlier, the Lyapunov Exponents indicate that the most stable position around the Earth is the furthest away position. This can also be seen in the data in Figure 47. The percent error for L2's position was 35%. The reason was because the numerically calculated position was just the edge of the graph. The reason the largest Lyapunov Exponent could not find L2 is most likely due to the fact that more of the n-dimension Lyapunov Exponents would be needed. Since L2 is a saddle point, that means out of

its three Lyapunov Exponents, at least one is positive and one is negative. In order to identify it, some ”chaotic” map following the gradient flow of the n-dimensional Lyapunov Exponents would be needed in order to possibly find the saddle point.

The ”Sticky Stability” plot around Earth can be seen in Figure 31. The most apparent difference between the Lyapunov Exponent plot and the ”Sticky Stability” plot is that there is very apparent ”stable” locations at the positions of L1 and L2. The velocity being inputted in this graph is that of the JWST, which is stable at L2. Since that velocity is very similar to the one that makes L1 most stable, the graph ends up finding L1 as well, albeit with a larger value than L2. The results for that can be seen in Figure 45. The Lagrange Point, L2, was found within 0.19% error. That makes sense since the definition of stability for ”Sticky Stability” is a point that sticks to the other celestial objects in a system. That is also very similar to the definition of a Lagrange Point.

6.9.2 Identification of L4

The Lyapunov Exponent plot around L4 can be seen in Figure 35. As it can be seen, there is no Lagrange point detected. It is just a gradient of stability away from the Sun. This makes sense, because as mentioned earlier, Lyapunov Exponents are smallest when far away from ”attractors”. The data for the minimum calculation can be seen in Figure 48. The percent error is not too large, but if you look closely, the point it selected as the ”most stable” is just the furthest point away from the Sun. Therefore, the Lyapunov Exponents did not detect the Lagrange Point, L4.

The ”Sticky Stability” plot around L4 can be seen in Figure 34. The Lagrange Point, L4, is visually identifiable. It is in the exact center of the plot (the position and velocity were chosen as the Earth’s position and velocity if it was 60 degrees further in its orbit because that is where L4 sits). The data for the minimum calculation can be seen in Figure 46. The percent error for L4’s position is 0.015%, which is really good given that the number of samples observed was only 200,000.

6.9.3 Comparision of Lagrange Points

Another notable observation is that L4 appears to be much larger than L1 or L2. Figure 32 shows the ”Sticky Stability” plot for the entire Earth orbit (Sun is at the center) for the stable velocity to sit at L2. Therefore, L1 and L2 should be visible. Technically, L2 can be found numerically if the minimum value is found, but visually it is almost impossible to spot L1 and L2 since they are so small. The Earth is located on the very left-hand edge of the graph and L1 and L2 are small dots around the Earth. They are practically imperceptible on that graph.

On the other hand, the graph of the entire Earth orbit (Sun is at the center) using ”Sticky Stability” for the stable velocity to sit at L4 is shown in Figure 33. L4 is extremely visible even on that scale. The primary reason L4 is visible on that scale and L2 is not is because L2 is only semi-stable and it is a very small ”window of stability”. L4 acts as a

sink where objects fall into L4. L4 is fully stable, so it has a much larger effect radius.

6.10 Results of Velocity Finder For Earth-Sun Lagrange Points

Due to extra experimentation, I also developed an ”Optimal Velocity Finder” that takes in a position. The primary reason was that I wanted to show that the ”Lagrange Point Finder” method can take in an initial velocity for a probe and return a stable initial position. The ”Optimal Velocity Finder” takes in an initial position and it returns the most stable initial velocity. I wanted to show that the calculation works both ways. The results for the optimal velocity finder given the theoretical Lagrange Point locations can be seen in Figure 49. The found L2 velocity is extremely similar to the velocity of the JWST. The L4 velocity is the same as the velocity of the Earth 60 degrees into its orbit clockwise.

6.11 Summary

Overall, this project shows that ”chaoticness” and ”stability” can be described in different ways. The definition of ”stability” for Lyapunov Exponents, contrary to the original expectation, does not identify Lagrange Points based on the theoretical calculations made in this project. The definition of ”stability” for ”Sticky Stability” can correctly identify Lagrange Points. Both methods of quantifying chaos indicate that a two-body system is chaotic, which is false. For an extremely large amount of iterations, the largest Lyapunov Exponent does converge close to 0 for a two-body system, but the graph itself indicates ”chaoticness” due to differences in those small numbers. This project has also presented a numerical way for numerically finding stable Lagrange Points or any stable ”sticky” points in an n-body system (verified for three-body).

For further expansion, this project could be expanded to produce a Chaos Plot by testing 3D initial conditions (Currently only two-dimensional coordinates were tested). The Chaos Plot could also be modified to test different initial velocities and initial positions simultaneously in order to find the Lagrange Points without knowing an initial velocity or position. The other two Lyapunov Exponents could be found and perhaps a gradient flow could be calculated around a Lagrange Point (looking for a negative, positive, and positive or negative set of Lyapunov Exponents). Perhaps the Lyapunov Exponent could be calculated using a similar method integrated with respect to a different coordinate system. The Lyapunov Exponent was found in this paper integrating over time; however, it could be possible that the largest Lyapunov Exponent would completely converge to 0 in the angle-space instead of the time-space. It is also possible that a Lagrange Point position might not be directly identifiable given the calculation method for the three Lyapunov Exponents; however, given the position of the Lagrange Point, L2, the associated Lyapunov Exponents will have a negative and positive number. Another possible expansion would be to combine the ”Sticky Stability” method to find Lagrange Points and then use the Lyapunov Exponents to indicate whether a specific Lagrange Point is asymp-

totically stable, convergent stable, spiral stable, a saddle point, etc. There are multiple expansions that can be made to this project in the future.

7 Appendices

7.1 Appendix A: "Chaotic System" Initial Conditions

Chaotic System

Name	Mass (kg)	Initial Position (km)	Initial Velocity (km/s)
Earth	5.97219E+24	(0,0,-50000)	(-3,0,5)
Sun_EarthSize	5.97219E+24	(0,0,0)	(0,0,3)
Sun_EarthSize2	5.97219E+24	(50000,0,0)	(0,0,-2)
Massless Probe	0	N/A	(2,0,2)

Table 1: Initial Conditions for "Chaotic System"

7.2 Appendix B: "Equal Mass" System Initial Conditions

Equal Mass System

Name	Mass (kg)	Initial Position (km)	Initial Velocity (km/s)
Earth	5.97219E+24	(0,0,0)	(0,0,3)
Sun_EarthSize	5.97219E+24	(40000,0,0)	(0,0,-1)

Table 2: Initial Conditions for "Equal Mass" System

7.3 Appendix C: Earth-Sun-Moon System Initial Conditions

Earth-Sun-Moon System

Name	Mass (kg)	Initial Position (km)	Initial Velocity (km/s)
Earth	5.97219E+24	(0,0,0)	(0,0,30.29)
Sun	1.9891E+30	(147120163,0,0)	(0,0,0)
Moon	7.34767E+22	(0,0,405402)	(-0.996,0,30.29)

Table 3: Initial Conditions for Earth-Sun-Moon System

7.4 Appendix D: Earth-Sun-JWST System Initial Conditions

Earth-Sun-JWST System

Name	Mass (kg)	Initial Position (km)	Initial Velocity (km/s)
Earth	5.97219E+24	(0,0,0)	(0,0,30.29)
Sun	1.9891E+30	(147120163,0,0)	(0,0,0)
JWST	6500	(-1471603.356,0,0)	(0,0,30.6)

Table 4: Initial Conditions for Earth-Sun-JWST System

7.5 Appendix E: Earth-Sun System Initial Conditions

Earth-Sun System

Name	Mass (kg)	Initial Position (km)	Initial Velocity (km/s)
Earth	5.97219E+24	(0,0,0)	(0,0,30.29)
Sun	1.9891E+30	(147120163,0,0)	(0,0,0)

Table 5: Initial Conditions for Earth-Sun System

References

[1] Alligood, K. T., Sauer, T., & Yorke, J. A. (2000). *Chaos: An introduction to dynamical systems*, Springer.

[2] Runco, M. (n.d.). *Chaos theory*. Chaos Theory - an overview, ScienceDirect Topics. Retrieved September 26, 2022, from https://www.sciencedirect.com/topics/agricultural-and-biological-sciences/chaos-theory#:~:text=Molecular%20biologists%20see%20chaos%20as,in%20all%20forms%2C%20including%20fluids

[3] https://github.com/JackFrost213/PhysicsCapstone

[4] Gleisner, Frida. "Three Solutions to the Two-Body Problem." Diva Portal, 2013, http://www.diva-portal.org/smash/get/diva2:630427/FULLTEXT01.pdf.

[5] NASA, NASA, https://www.nasa.gov/.

[6] Westra, Dennis. "Lagrangian Points." Vienna Homepage, University of Vienna, 5 July 2017, https://www.mat.univie.ac.at/∼westra/lagrangepoints.pdf.

[7] Sprott, J. C. "Lyapunov Exponent and Dimension of the Lorenz Attractor." Sprott Physics, 2 Oct. 1997, https://sprott.physics.wisc.edu/chaos/lorenzle.htm.

[8] Yapparina. (2015, August 16). "Orbital Instability". Creative Commons. Wikipedia.

[9] Mrocklin. (2010, November 8). "LyapunovDiagram.svg". Creative Commons. Wikipedia.