Simulating Chaos in Deterministic Systems: The Double Pendulum

Joshua Kao
Department of Physics and Astronomy, University of Southern California, Los Angeles, California 90089, USA

(Dated: November 29, 2022)

Whereas the single simple pendulum is one of the most common systems used to demonstrate simple harmonic motion, a double pendulum consisting of two simple pendulums chained in succession exhibits widely different behavior. The double pendulum is a chaotic system, as even small differences in initial conditions will cause two almost-identical systems to diverge over time despite the system's deterministic nature. The goal of this paper is to analyze the factors that play into the double pendulum's chaos, using numerical simulations to observe the effects that different initial conditions have on the behavior over time. My findings agreed with existing literature in that higher initial angles resulted in faster exponential divergence. In addition, the sensitivity to error was so great that a simulation using constraints diverged from a simulation using the Lagrangian when given identical initial conditions.

Keywords: Nonlinear Dynamics, Chaos, Lagrangian Mechanics, Constrained Dynamics

I. INTRODUCTION

In physics, the term "chaos" refers to seemingly unpredictable or irregular behavior in an otherwise deterministic system, with the double pendulum system serving as a common example. The system itself seems simple, as it is made up of two simple pendulums both with fixed mass and radius. As shown in Figure 1, The first pendulum swings around a fixed pivot, while the second is attached to the mass of the first.

The system was first solved using Lagrangian mechanics, which operates off of the principle of least action. According to this approach, the system can be described in terms of an action integral that must remain stationary as time changes. Using this, yields a system of second-order differential equations that describe the motion.

The effects of small changes to mass ratio and starting angle were tested to see which of these parameters had the greatest effect on the divergence of two closely-related systems, using the Lyapunov exponent as a measure of chaos.

Next, a second simulation was implemented using force-based dynamics. This process views the double pendulum as a system consisting of two point masses constrained to be a fixed distance from each other. Given the instantaneous positions and velocities of each particle, it is possible to calculate the extra forces necessary to satisfy each constraint at the same time. What's important to note about this approach is that it is purely numerical and never arrives at any set of differential equations. Instead, this process is meant to allow the flexibility to add many different constraints and objects easily, which is why it is commonly implemented in game/physics engines.

The two simulations were also compared when given identical initial conditions. I found that the two approaches would eventually diverge just as if they had started with different initial conditions, offering my results to spur discussion on why this occurs.

II. LAGRANGE'S EQUATIONS

Solving for the equations of motion using the Lagrangian first requires identifying the system's degrees of freedom. In this case, the pendulum lengths are fixed and thus the system can be described using two generalized coordinates. From this point on, theta 1 will represent the angle the top pendulum makes with the vertical, with positive values corresponding to the counterclockwise direction as shown in Fig.1. Theta 2 is the second pendulum's angle to the vertical.

Using these coordinates, [2] demonstrates that we can now describe Cartesian coordinates of both masses:

$$x_1 = l_1 sin\theta_1$$

$$y_1 = -l_1 cos\theta_1$$

$$x_2 = l_1 sin\theta_1 + l_2 sin\theta_2$$

$$y_2 = -l_1 cos\theta_1 - l_2 cos\theta_2$$

The potential energy of the system can be written as:

$$U = -m_1 g l_1 cos\theta_1 - m_2 g(l_1 cos\theta_1 + l_2 cos\theta_2) \tag{1}$$

By taking the first time derivative, the kinetic energy of the system is:

$$T = \frac{1}{2}m_1 l_1^2\dot{\theta}_1^2 + \frac{1}{2}m_2(l_1^2\dot{\theta}_1^2 + l_2^2\dot{\theta}_2^2 + 2l_1 l_2\dot{\theta}_1\dot{\theta}_2 cos(\theta_1 - \theta_2)) \tag{2}$$

The Lagrangian is defined as the difference between kinetic and potential energy:

$$L = \frac{1}{2}(m_1+m_2)l_1^2\dot{\theta}_1^2 + \frac{1}{2}m_2 l_2^2\dot{\theta}_2^2 + m_2 l_1 l_2\dot{\theta}_1\dot{\theta}_2 cos(\theta_1 - \theta_2) \\ +(m_1+m_2)g l_1 cos\theta_1 + m_2 l_2 g cos\theta_2 \tag{3}$$

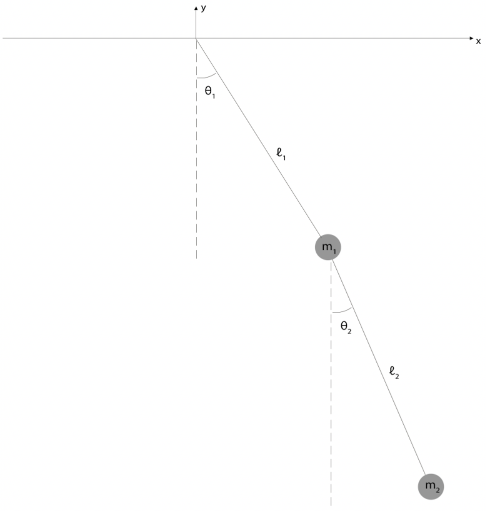

FIG. 1. A diagram of the system with each coordinate labelled as seen in [1].

Finally, the Euler-Lagrange equation is applied, which is based on the principle of the stationary action integral described in the introduction.

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{\theta}_j} - \frac{\partial L}{\partial \theta_j} = 0, j = 1, 2 \tag{4}$$

j corresponds to each generalized coordinate, in this case θ_1 and θ_2. For systems with higher degrees of freedom such as the triple pendulum and higher, the same equation can be applied.

Substituting the results into the Euler Lagrange Equation yields the following equations of motion:

$$\begin{aligned} \ddot{\theta_1} = (&-l_1 m_2 \dot{\theta}_1^2 sin(\theta_1 - \theta_2) cos(\theta_1 - \theta_2) \\ &+2l_1 m_2 \dot{\theta}_1 \dot{\theta}_2 sin(\theta_1 - \theta_2) cos(\theta_1 - \theta_2) \\ &+g m_2 sin\theta_2 cos(\theta_1 - \theta_2) \\ &-m_2 l_2 \dot{\theta}_2^2 sin(\theta_1 - \theta_2) \\ &-(m_1 + m_2) g sin\theta_1) \\ &\frac{1}{l_1(m_1 + m_2) - l_1 m_2 cos^2(\theta_1 - \theta_2)} \end{aligned} \tag{5}$$

$$\begin{aligned} \ddot{\theta}_2 = (&m_2 l_2 \dot{\theta}_2^2 sin(\theta_1 - \theta_2) cos(\theta_1 - \theta_2) \\ &+(m_1 + m_2) g sin\theta_1 cos(\theta 1 - \theta_2) \\ &+l_1(m_1 + m_2)\dot{\theta}_1^2 sin(\theta_1 - \theta_2) \\ &-2l_1(m_1 + m_2)\dot{\theta}_1 \dot{\theta}_2 sin(\theta_1 - \theta_2) \\ &-g(m_1 + m_2) sin\theta_2) \\ &\frac{1}{l_2(m_1 + m_2 - l_2 m_2 cos^2(\theta_1 - \theta_2}\end{aligned} \tag{6}$$

This is a pair of coupled, nonlinear second order differential equations, and explicit expressions for θ_1 and θ_2 cannot be written analytically. Instead, a numerical approach must be taken to analyse their behavior.

III. SIMULATING LAGRANGE'S EQUATIONS

In order to visualize the results, a program was written in C++ using SDL2, a graphics library that allows 2D rendering. The double pendulum was implemented as a class that stores its own mass and lengths, as well as its current values of $\theta_1, \theta_2, \dot{\theta}_1$, and $\dot{\theta}_2$. The simulation uses Euler's method to discretize time into timesteps using the following equations:

$$\theta_{next} = \theta + dt\dot{\theta}\dot{\theta}_{next} = \dot{\theta} + dt\ddot{\theta} \tag{7}$$

At each timestep in the simulation of length dt, the simulation uses Eq.5 and Eq.6 to calculate what the values for theta 1 double dot and theta 2 double dot should be, given its current position and velocity. The program then updates the first time velocity values and finally the position values to approximate what the system would look like when t reaches the next timestep. In all test simulations, the program was run with a timestep of 1/30,000 sec to get as much accuracy as possible. Also, all initial velocities were set to zero.

IV. MEASURING CHAOS

The Lyapunov Exponent is a characteristic of a system that describes the exponential divergence between two close trajectories in its state space [2, 3]. The authors of both [3] and [2] use the Lyapunov Exponent to get a measure of the chaos in the system. This paper makes use of the same formula presented by [4] to calculate the

Lyapunov Exponent λ using a series of intermediate values λ_i, given a total time interval of $\tau = n\Delta t$ and an initial distance of d_0:

$$\begin{aligned} \lambda_i &= \frac{1}{t_i - t_{i-1}} ln(\frac{d(t_i)}{d_0}) \\ \lambda &= \frac{1}{\tau}\sum_{i=1}^{n} ln(\frac{d(t_i)}{d_0}) \end{aligned} \tag{8}$$

While the mass of a pendulum is constant within trials, the following sections test differences in mass ratio across trials. Thus, mass has to be taken into account as a variable in the state space of the system along with position and velocity.

V. RESULTS

Using the program developed in section III, the effects of variations in mass ratio and starting angle were tested. The testing procedure followed a method suggested by [3], in which two simulations are run simultaneously with a small difference in one parameter. Once the Lyapunov Exponent reaches a point where the rate of growth is no longer exponential (when the state space difference reaches its maximum), the simulation is restarted with the next two closely related paths in some range of a different parameter. Figure 2 shows the Lyapunov Exponent calculated for an example trial.

Importantly, in all simulations run for this paper the Lyapunov Exponent was calculated once for every 250 iterations of the physics engine, for an overall frequency of 120 Hz compared to the actual engine's 30,000 Hz. From now on in plots of the Lyapunov Exponent, the term "iterations" will refer to the number of times the Lyapunov Exponent has been calculated, not total simulation iterations.

The first parameters tested were a θ_1 difference of 0.003 degrees across θ_1 from 0 to 180 degrees with step size of 1 degree. Figure 2 shows the average Lyapunov Exponents of these trials. Also tested was a mass ratio difference of 0.01 across mass ratios from 0.01 to 2.00 with step size 0.01. Last was mass ratio difference 0.01 across θ_1 from 0 to 180 degrees with 1 degree step size. Figure 3 shows the average Lyapunov Exponents calculated from these trials.

In all trials, λ was calculated to be greater than 0, which means that closely-related trajectories in the state space will diverge exponentially [3]. In addition, my findings agreed with those presented in [2] and [3] in that λ increased in magnitude with higher release angles. This means that the system is more sensitive to initial conditions and thus diverges faster when the phase space is higher.

VI. THE CONSTRAINT APPROACH

Rather than using degrees of freedom to find a system of generalized coordinates, using constrained motion takes a different approach. The double pendulum system can be viewed as a pair of point masses. Mass one is constrained to be a constant distance from the origin, while mass two is constrained to be a constant distance from mass one. Thus, the system consists of two particles and two constraints.

As shown in [5], the two constraints can be expressed as functions:

$$\begin{aligned} C_1 &= \frac{1}{2}(x_1^2 + y_1^2) - r_1^2 \\ C_2 &= \frac{1}{2}((x_1 - x_2)^2 + (y_1 - y_2)^2) - r_2^2 \end{aligned} \tag{9}$$

These functions take in the system "state vector" (a vector of length 4 that contains the positions of both masses) as a parameter and output a scalar. When the function evaluates to 0, the constraint is satisfied. The general goal of this approach is to evaluate the position, velocity, and external forces acting on each object and then calculate the "constraint forces" necessary to ensure each constraint remains satisfied at every iteration. These functions can be placed into one vector of length 2 and differentiated to find the Jacobian, J:

$$\boldsymbol{J} = \begin{pmatrix} x_1 & x_2 & 0 & 0 \\ x_1 - x_2 & y_1 - y_2 & x_2 - x_1 & y_2 - y_1 \end{pmatrix} \tag{10}$$

Newton's Second Law can be written in its matrix form:

$$\boldsymbol{\ddot{q}} = \boldsymbol{M^{-1}F} = \boldsymbol{M^{-1}(F_{ext} + F_C)} \tag{11}$$

where $\boldsymbol{F_C}$ is a vector of constraint forces necessary to keep each constraint satisfied. As shown in [5], Eq.10 can be substituted in to arrive at:

$$\boldsymbol{\ddot{C}} = \boldsymbol{\dot{J}\dot{q}} + \boldsymbol{JM^{-1}(F_{ext} + F_C)} \tag{12}$$

The first and second time derivatives of $\boldsymbol{C}$ must equal 0 in order for $\boldsymbol{C}$ to also remain 0. Using these identities and rearranging yields:

$$\boldsymbol{JM^{-1}F_C} = -\boldsymbol{\dot{J}\dot{q}} - \boldsymbol{JM^{-1}F_{ext}} \tag{13}$$

It can also be assumed that the constraint forces are always orthogonal to velocity and that $\boldsymbol{F_C}$ must be a multiple of $\boldsymbol{J}$ [5]. This yields the following matrix equation:

$$\boldsymbol{JM^{-1}J^T\lambda} = -\boldsymbol{\dot{J}\dot{q}} - \boldsymbol{JM^{-1}F_{ext}} \tag{14}$$

The left hand side of this equation is a matrix of known values multiplied by lambda, the vector of unknown constraint force magnitudes. The right side is a vector of known values. Given the system's current state vector, Eq.14 can be used to calculate the constraint forces necessary to make sure $\boldsymbol{C}$, $\boldsymbol{\dot{C}}$, and $\boldsymbol{\ddot{C}}$ are always equal to 0,

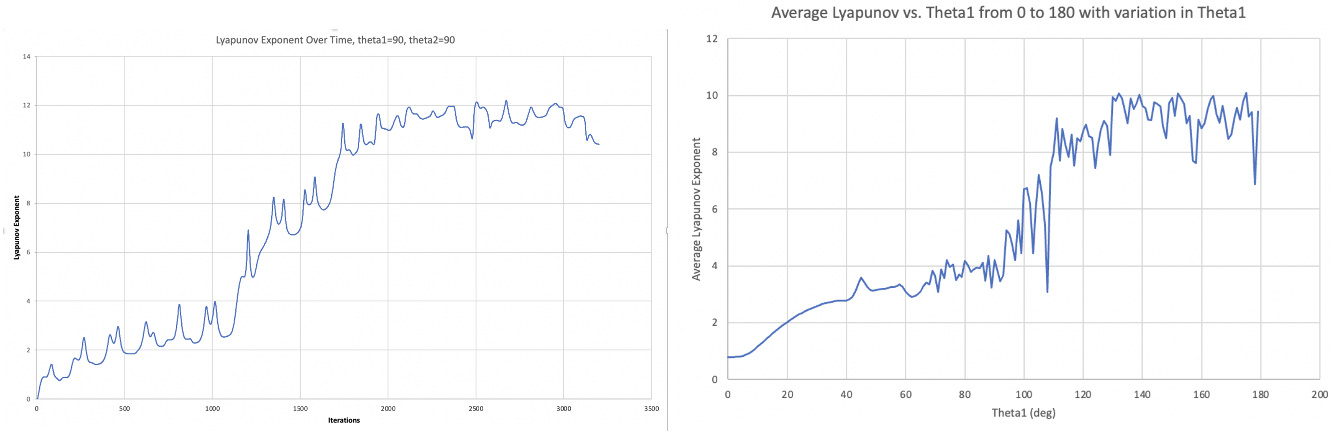

FIG. 2. **(Left)** The Lyapunov Exponent over time using Lagrange's equations with mass ratio 1.00, $\theta_1 = 90$ degrees, $\theta_2 = 90$ degrees, compared with a second system with the same parameters except for $\theta_1 = 90.003$ degrees. **(Right)** The average Lyapunov Exponent calculated for a series of trials with θ_1 from 0 to 180 degrees, and a starting difference in θ_1 of 0.003 degrees.

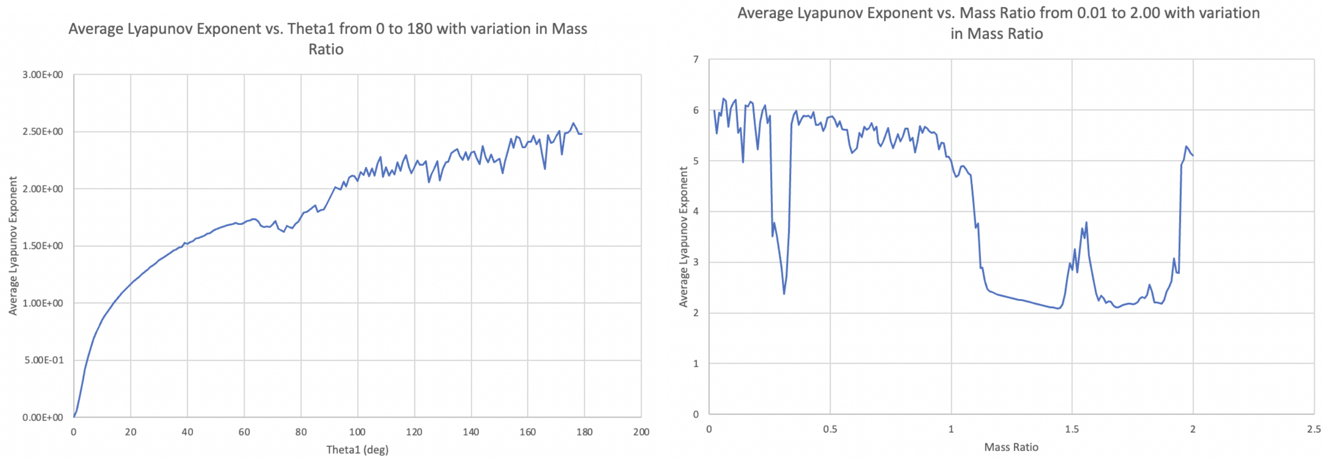

FIG. 3. **(Left)** The average Lyapunov Exponent calculated for a series of trials with θ_1 ranging from 0 to 180 degrees with mass ratios 1.00 and 1.01. **(Right)** The average Lyapunov Exponent calculated for a series of trials with mass ratio ranging from 0.01 to 2.00 with θ_1=90, θ_2 =0, and a mass ratio difference of 0.01.

satisfying the constraint.

VII. SIMULATING THE CONSTRAINT APPROACH

In order to compare the two methods of solving the double pendulum, both were implemented in the same program. The force-based approach takes in a system of point masses and constraint equations, and at each iteration plugs in the current system state vector and solves the linear system of equations in Eq.14. The linear algebra is handled by a matrix class that performs matrix multiplication and Gauss-Jordan Elimination for the final results.

Once the equation has been solved for the magnitude of the constraint forces, the program updates the acceleration vector for each particle in the system and then uses the same Euler ODE numerical solver described in Section III to step the system forward by 1/30,000 seconds.

The same system of tests used on the Lagrangian solution were again run to see if the two solutions would agree, with Figures 4 and 5 showing the results. While the plots for both formulations are noisy and don't agree entirely, all were remarkably similar and produced the same overall trends. The constraint approach was also able to reproduce the Lagrange-based approach's increased sensitivity at higher angles. The magnitudes of λ were also very close, with an average difference of 9.23% for the tests of angle vs angle, 10.43% for the tests of mass ratio vs angle, and 6.29% for the tests of mass ratio vs mass ratio.

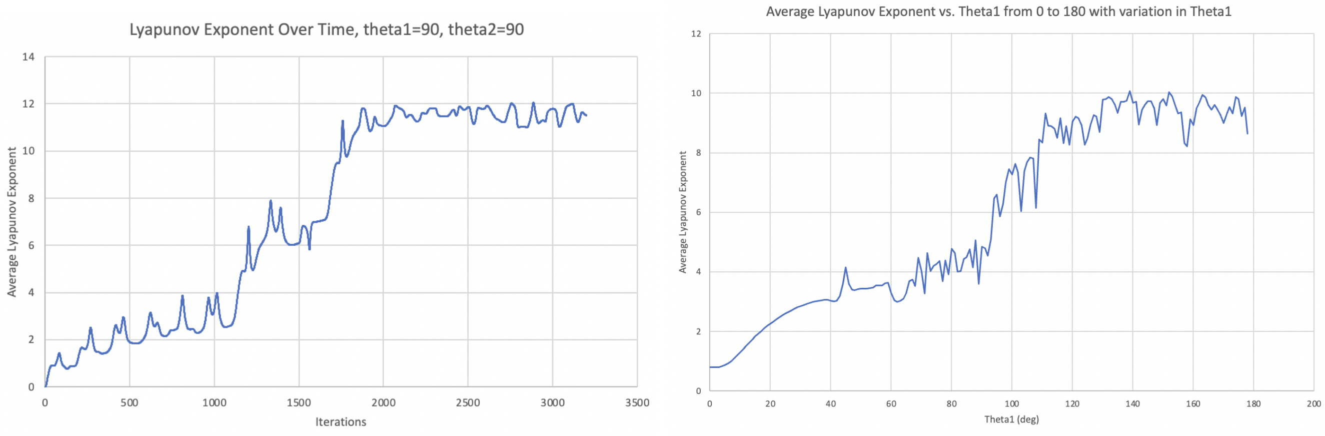

FIG. 4. **(Left)** The Lyapunov Exponent over time using constraints with mass ratio 1.00, θ_1 = 90 degrees, θ_2 = 90 degrees, compared with a second system with the same parameters except for θ_1 = 90.003 degrees. **(Right)** The average Lyapunov Exponent calculated for a series of trials with θ_1 from 0 to 180 degrees, and a starting difference in θ_1 of 0.003 degrees.

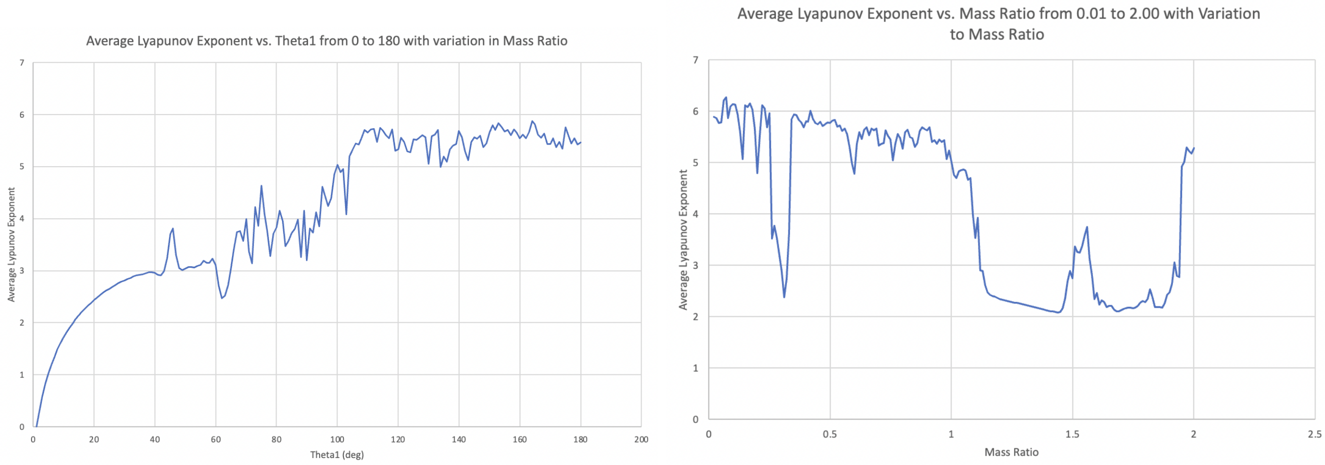

FIG. 5. **(Left)** The average Lyapunov Exponent calculated for a series of trials with θ_1 ranging from 0 to 180 degrees with mass ratios 1.00 and 1.01. **(Right)** The average Lyapunov Exponent calculated for a series of trials with mass ratio ranging from 0.01 to 2.00 with θ_1=90, θ_2 =0, and a mass ratio difference of 0.01.

VIII. COMPARING THE TWO METHODS

While the two simulations produce the same general behavioral trends when compared with themselves, it was noticed that two identical systems using the two different formulations would eventually diverge as if they had different initial conditions. However, an important note is that this phenomenon cannot be measured or described with the Lyapunov Exponent as used for prior trials. The definition of the Lyapunov Exponent requires a ratio to the initial distance in the state space, but in this case the initial state space distance is 0, suggesting undefined behavior. This also makes sense conceptually, as two trajectories deriving from the same point is physically impossible. Future research might uncover other ways of measuring exponential divergence more suited to computer simulations, but for now the phenomenon was analyzed by the state space distance alone. The state space distance for two sets of initial conditions was calculated for 4500 iterations and plotted in Figure 6. The results show an exponential increase in the state space difference over time, with the higher starting angle of θ_1=50 diverging faster than the lower angle of θ_1=30. This suggests that, just as with two closely related systems of the same simulation, a higher phase angle causes more sensitivity.

IX. MORE TESTING WITH THE CONSTRAINT APPROACH

The following two subsections consist of small tests that take advantage of the force-based model's ability to easily add extra masses and constraints to the system without the need for re-evaluating the differential equations. In code, this idea takes the form of different

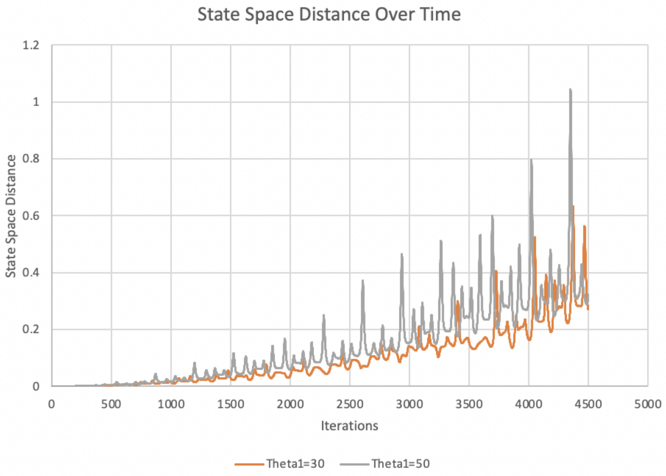

FIG. 6. The difference in state space between the Lagrangian and Constraint based simulations over time using two different sets of initial conditions. Both had mass ratio = 1 and $\theta_2 = 0$.

constraint objects, each with the ability to add their own rows to the master Jacobian matrix of the system.

A. The Influence of a Single Water Droplet

Gleick in his book "Chaos: Making a New Science" is cited as saying the following:

> "The dependence on initial conditions was so sensitive that the gravitational pull of a single raindrop a mile away mixed up the motion within two minutes." [3]

While this quote is meant to illustrate the unpredictability of the double pendulum, the validity of this statement can be easily tested using the simulation. First, two identical double pendulums were set up as in Section V. A third object of mass 0.034 g and was placed 1609 m away from the main system. At each step in the simulation, the first pendulum system was allowed to run normally while the second took into account the gravitational attraction of the droplet before doing its constraint calculation. For all trials, all masses were 1 kg. When the two systems were run, the rate of divergence seemed to largely depend on the initial angles given to the system. When the system behaved non-periodically at high angles (such $\theta_1 = \theta_2 = 90$), the two would diverge within less than 10 seconds. However, small oscillations would cause the two to stay in phase for the whole two minutes. With some testing, $\theta_1 = \theta_2 = 79$ was one such starting condition that diverged completely after around 9000 iterations, or 75 seconds and is plotted in Figure 7. This time is within the realm of "under two minutes" as described by Gleick, rendering his statement valid in light of the tests.

B. Simulating the Triple Pendulum

The force-based approach also makes it easy simulate triple, quadruple, etc. pendulum systems by adding an extra mass and then a constraint fixing its distance to the previous mass.

Due to the fact that these more complex systems have many possible parameters to change and are very computationally intensive, all masses were kept the same and only the tests for θ_1 were performed. These results can be seen in Figure Similarly to the double pendulum, the triple pendulum had higher values of λ with a higher starting value of θ_1, suggesting greater sensitivity to initial conditions at higher phase angles. Also, it is visible from the graphs and even visually apparent that having more degrees of freedom makes the system even more chaotic than the double pendulum, with each extra mass making the system more unstable. Compared to the plots in Figures 2 4, the Lyapunov reached a higher maximum value and rose faster than the double pendulums. Visually, at high release angles the triple pendulums would completely diverge within the first few oscillations. In general, these findings suggest that these more complex systems are even more sensitive to changes in initial condition than the double pendulum. The computational limitations of the simulations made it unfortunately unrealistic to test systems with more degrees of freedom than the triple pendulum. Even using 30,000 iterations per second, with more complex simulations the error involved with Euler's method would noticeably accumulate. This would cause the total energy of the system to slowly rise until the simulation broke down. However, using a smaller dt than 1/30,000 sec was impossible, as generating the graphs for this section already took over 1.5 hours to calculate. Decreasing the timestep for more accuracy would slow down the simulation even more.

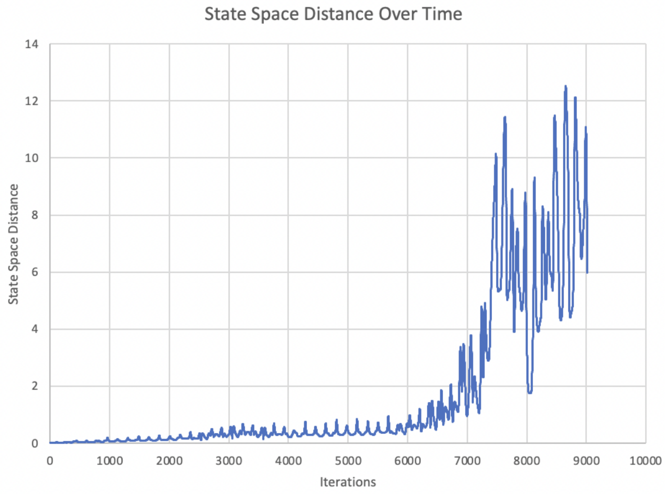

FIG. 7. The difference in state space between two constraint pendulums. Both had $l_1 = l_2 = 1$ and $\theta_1 = \theta_2 = 79$, but with one under the gravitational influence of an object of mass 0.034 g 1609 m away.

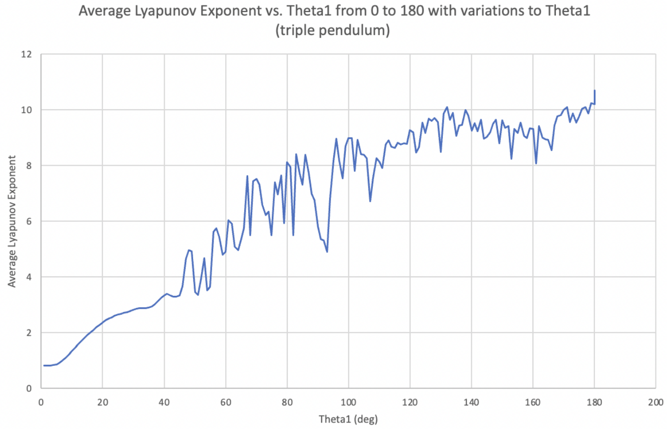

FIG. 8. λ calculated for the triple pendulum with θ_1 ranging from 0 to 180 degrees and a difference in θ_1 of 0.003 degrees. In all trials, all masses were 1 kg, $\theta_2 = \theta_3 = 0$ degrees.

X. DISCUSSION

Both the Lagrangian and force-based approach to the double pendulum system yield results that are mathematically valid and equally chaotic. Looking at Sections V and VII reveal that they show the same sensitivity to phase space and dependence on initial conditions. However, Section IX reveals that while they can both convincingly simulate the double pendulum, they are not the same.

The inevitable divergence between the two types of simulation are unavoidable, and further testing is needed to identify exactly where the divergence takes place. In theory, using the Lagrangian equations of motion should be the more "correct" out of the two approaches, as these are the true equations of motion. This means that with an infinitely small timestep, the Lagrangian approach would produce a perfectly physically accurate model of the system. However, the force-based approach is not physically wrong either, as constraint forces are not imaginary. In this system, the constraint forces are tension, and this approach takes those into consideration to model the motion.

Assuming the force-based approach is equally valid, the divergence could be a result of the two simulations responding to error in different ways. One possible area of examination is the fact that my Lagrangian approach uses polar coordinates internally to store its location, while the force-based one uses Cartesian coordinates internally which is then translated to polar for the user. While in a perfect world these should be identical, in practice factors such as floating-point accuracy and even the approximation of π used by C++ would be enough to throw off the models. Currently, this seems likely to be the cause between the two models' disagreement.

XI. CONCLUSIONS

Regardless of the discrepancy between the two simulations, this does not mean one simulation is superior to the other.

Whereas the Lagrangian method is widespread and proven to be physically sound, the principles involved make it impossible to implement entirely in code. Instead, it requires much predetermined math to arrive at a system of equations, which can then be given to the code to simulate.

It is for this reason that force-based dynamics is a more popular approach when it comes to physics and game engines, since its modularity allows the flexibility to add items and constraints without any recalculation. Though it does not respond to error the same way as the Lagrangian approach, the results from Section VII show how closely this approach is able to recreate the Lagrangian approach's behavior.

In the end, both methods are still merely approximations. They each have their different use cases, but more importantly they are both able to agree on the broader trends of the system's behavior. While not identical, they follow the same trends when it comes to the system's sensitivity to certain parameters: Both approaches yielded λ values greater than 0, showing exponential divergence between trajectories in the phase space. Furthermore, both approaches agreed with the results presented by [3] that larger phase spaces and higher release angles caused faster divergence.

As such, they are both valid methods when it comes to examining the phenomenon of chaos in physics.

CODE AVAILABILITY

Code is available at `https://github.com/joshjkao/constraineddynamics`. For more information on downloading and configuring SDL2, the graphics library used for this project, see the README.

[1] B. Yesilyurt, *Equations of Motion Formulation of a Pendulum Containing N-point Masses* 10.48550/arXiv.1910.12610 (2020).

[2] T. N. Anis Safitri and T. D. Chandra1, *Factors of length ratio and mass at chaos of double pendulum system* **2215**, 10.1063/5.0000582 (2020).

[3] J. Chen, *Chaos from simplicity : an introduction to the double pendulum* 10092/12659 (2008).

[4] S. Strogatz, *Nonlinear Dynamics and Chaos* (University Science Books, Boston, Massachusetts, Perseus Books, Boston).

[5] A. Witkin, *An Introduction to Physically Based Modeling: Constrained Dynamics*, (1997).