Chaotic Nature of Physical Systems: Conservative Chaos and Hénon-Heiles Hamiltonian

by

Yakup Emre Şahin

A report submitted for the EE492 senior design project class
in partial fulfilment of the requirements for the degree of

Bachelor of Science
(Department of Electrical and Electronics Engineering)
in Boğaziçi University

4 June 2023

Principal Investigator: Prof. Dr. Yağmur Denizhan

Acknowledgments

I would like to express my deep gratitude to Professor Yağmur Denizhan, my research supervisor, for her patient guidance, enthusiastic encouragement and useful critiques of this research work. This project would not have been possible without her constant feedback and support.

Additionally, I wish to thank my parents for their support and encouragement throughout my study.

Abstract

As a deterministic source of unpredictability, chaos has been one of the concepts that have changed our perception of nature and shaped the ideas of engineering. As first pointed out by Hénon and Heiles in 1964, chaos can emerge even in very simple, low-dimensional, conservative systems. In the days that the scope of our engineering challenges extends out to outer space where dissipative forces disappear and we are left with a conservative void, the study of conservative chaos becomes more and more important. The purpose of our project is to investigate the common properties of conservative chaos and how to assess the chaoticity of such systems/trajectories by studying the same model Hénon and Heiles used in their original work. We reproduce the results made on this model previously and discuss that the judgement of chaoticity is highly dependent on the methods used in the assessment process.

Contents

List of Figures

List of Tables

1 Introduction

Chaos, with its simple description of sensitive dependence on initial conditions, is one of the breakthroughs made at the intersection of science and mathematics. The fact that even deterministic systems can be unpredictable has changed the scientific approaches in almost every discipline. Soon after, scientists realized lots of systems that have been studied for thousands of years show chaotic properties and this fact had not been observed by then due to a lack of precision in the measurements. One of these systems was the three-body problem. Since Newton, these problem has been widely worked on yet not given any solid explanations. It was such an important issue that Oscar II, King of Sweden, challenged the brilliant minds of the globe by establishing a prize on the problem in 1887. French mathematician and philosopher Henri Poincaré (1854-1912) approached the problem in terms of the existence of periodic orbits, more precisely non-existence of periodic orbits. Through his work, he has become one of the founders of modern chaos theory, also winning the prize. It is now well known that n-body systems with $n > 2$ usually have chaotic characters.

After Poincaré, the problem of n-body systems has continued to be studied in different aspects, such as gravitational couplings of bodies with specific dynamics. In 1963, two scientists from Princeton University Observatory, Michel Hénon and Carl Heiles conducted a series of numerical experiments on an approximate system suggested by themselves while trying to explain the galactic motion of stars. They discussed the problem in terms of the existence of a third integral of motion in addition to the energy and the z-component of the system's angular momentum. Their study found that the third integral of motion existed only for a limited number of initial conditions [1]. From a modern perspective, initial conditions that do not have the third integral of motion correspond to chaotic orbits. The system they proposed is called the Hénon Heiles system and this system has been widely studied [2],[3] due to its chaotic characteristics despite its simplicity.

The chaos notion in this system is specifically interesting since chaos has been popularized primarily on dissipative systems, with some typical examples such as turbulent flow of fluids and weather dynamics, primarily due to observable strange attractors and low-dimensional chaos in dissipative systems [4]. However, conservative systems can also exhibit chaotic behaviours, like in the example of a double pendulum. In this case, phase space does not have any strange attractors, yet some regions with positive Lyapunov exponent, state that adjacent trajectories diverge from each other. In recent decades, scientists observed that conservative systems of astrophysical bodies like planets, stars, and galaxy clusters indeed have positive Lyapunov exponents. Some planets in the solar system also have chaotic behaviour, so that initial position uncertainty of 1 km may lead to an average error of $5 \cdot 10^8$ km in 100 million years [5]. Such observations once again demonstrated the importance of the studies on conservative chaos. Scientists made further studies on the idea of controlling systems with conservative chaotic characteristics [6, 7].

2 Theoretical Background

A Brief Summary of System Dynamics

In this section, we intend to give a sufficient yet not too comprehensive introduction to the theory of nonlinear dynamics with a focus on the model at our hand. This underappreciated field has been categorized under different main disciplines so far, however, we find it more meaningful to label it as applied mathematics since the study of nonlinear systems can be thought of as applying some rigorous mathematical scheme to any model from any discipline as one sees fit. Our approach will mostly reveal some fundamental ideas of the nonlinear theory so that a technically-equipped reader who has never encountered such discussion before can follow the flow of argumentation here.

Scientific modelling has started to be fruitful mainly after the construction of calculus. Since then, the study of differential equations has been the most crucial for the analysis of systems progressing in time, usually called dynamical systems, as they define how some variables which are defining the behaviour of a system change with time. Before proceeding let us make a few remarks here. Philosophically speaking, when we talk about a system, it is the model we devised to describe this system that we imply. All of the system properties are, actually, attributes of our models. However, we choose to call it the *system* rather than the *model of the system*, in short.

The collection of these variables is called the phase variables or the phase $\mathbf{q}$ and it is represented by a vector in the vector space known as the *phase space*. The equation governing how the phase alters in time is called the dynamics equation. If the dynamics equation of a system (model) is time-dependent, i.e. $\dot{\mathbf{q}} = \mathbf{q}/dt = \mathbf{f}(\mathbf{q}, t)$, such systems are said to be *non-autonomous*. As might be expected, a system with a time-independent dynamics equation, i.e. $\dot{\mathbf{q}} = \mathbf{f}(\mathbf{q})$, is called an *autonomous* system. In our study, we will deal with autonomous systems. As a categorization based on the continuity of the time-independent variable, discrete-time systems are called *maps* and continuous-time systems are called *flows*. Our focus on this project will be on flows. Furthermore, our dynamic equation $\mathbf{f}(\mathbf{q})$ is guaranteed to be infinitely differentiable. This condition is crucial since some of the theorems and analyses we use are only valid for smooth vector fields.

Overall, we can represent an autonomous nonlinear flow with a smooth vector field as the following where n is called the dimension (order) of the system.

$$\dot{\mathbf{q}}(t) = \mathbf{f}(\mathbf{q}(t)) = \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \vdots \\ \dot{q}_n \end{bmatrix} = \begin{bmatrix} f_1(\mathbf{q}) \\ f_2(\mathbf{q}) \\ \vdots \\ f_n(\mathbf{q}) \end{bmatrix} \tag{1}$$

After this point, it is only left to solve the dynamic equation to get the time evaluation of the systems. Our analytical tools are sometimes powerful enough to provide us with exact solutions, however; in the nonlinear case, we are limited to a few analytically solvable solutions. Hence, we can only implement numerical methods to simulate the behaviour of the system with a discrete-time approximation. We, then, obtain the phase trajectory depending on the specified initial conditions $\mathbf{q}_0 = \mathbf{q}(t = 0)$ on the space of phase variables, previously defined as phase space. Consider a continuous set of initial conditions with an initial phase space volume specified to the set and we track the set of trajectories starting from this set. If the volume of this set is growing, such a system is called a growing system. Growing systems, in general, result in catastrophes. If the phase volume is shrinking, these kinds of systems are called *dissipative systems*. Most of the systems around us are, in fact, dissipative due to the dissipative forces surrounding us such as friction. Lastly, phase volume-preserving systems are called *conservative systems*. Conservative systems have conserved quantities called *integrals of motion*, i.e. energy for conservative physical systems. We will focus on these integrals of motion discussion further when we talk about the model we are interested in.

Dissipative systems can have regions where the trajectories are pulled towards, called *attractors*. These attractors can be point attractors (equilibrium points), periodic attractors (limit cycles), and chaotic attractors (strange attractors). On the other hand, conservative systems cannot have attractors since their phase volume could not be preserved otherwise. The trajectories of the conservative systems are bound to the surface of the fixed *conserved quantity* manifold. Such manifolds are called *invariant tori*. The phase space motion of the conservative systems on the surface of the invariant torus can be envisaged by the motion of an ant, swirling from one side to the other on the surface of a doughnut. Therefore, in such systems, it is typical to observe periodic or quasi-periodic orbits. Quasi-periodic orbits can be roughly described as orbits that swirl just like periodic orbits yet in every swirl they are turned next to the previous swirl and a closed path is never obtained. These orbits are bound to the surface of the invariant torus. We also know that conservative systems can have chaotic orbits yet these orbits do not converge to an attractor.

In this study, our main focus will be on a specific Hamiltonian system, called Hénon-Heiles systems. Hamiltonian systems are a special subgroup of conservative systems, which have mostly originated from physical models. Therefore, our brief discussion of the system dynamics and the chaos of conservative systems can be generalized further.

Chaos: A Deterministic Source of Unpredictability

Chaos, by its simplest explanation, is the divergence of nearby trajectories of a system. Being a concept beyond stability, it represents the long-term unpredictability of the behaviour of a system due to our observational limitations. However, this unpredictability does not stem from any probabilistic nature of the system (model of the system). Chaos is perfectly deterministic, yet the unpredictability is inherently embedded in the fact that our measurements are compelled to an error margin. Chaos, in fact, is the feature of a trajectory. The question is whether or not any infinitesimal perturbation to the trajectory in hand would grow or be suppressed. Systems exhibiting chaotic trajectories basically are called chaotic systems even as all of the trajectories don't require to be chaotic. In chaotic systems, even the slightest uncertainty would grow exponentially if one waited enough. The measure of this

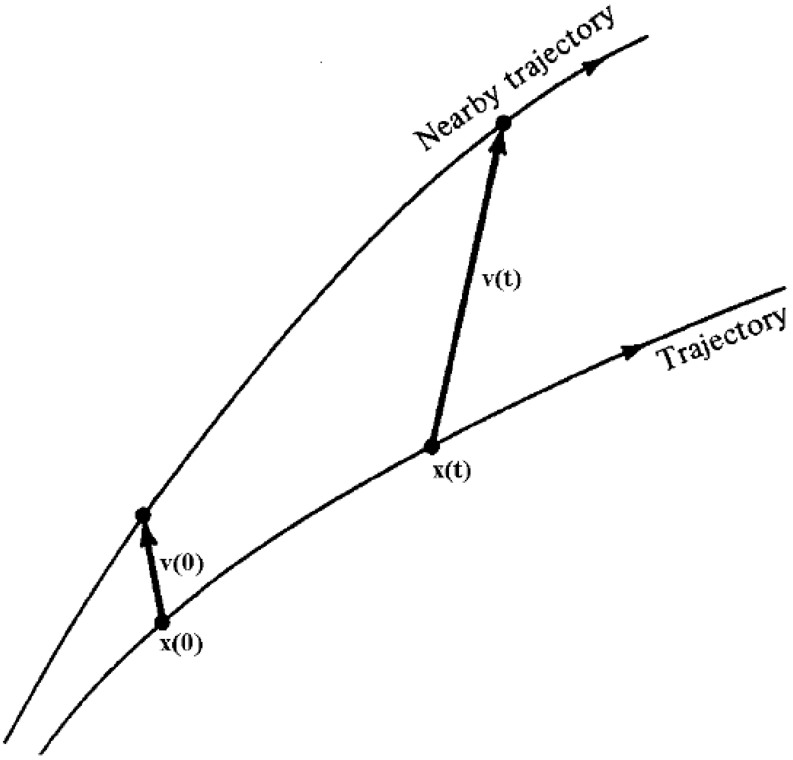


Figure 1: Exponential divergence of nearby trajectories[1]

exponential growth of the uncertainty is called *Lyapunov characteristic exponents* (LCE).

For an n-dimensional system, there exists n Lyapunov characteristic exponents

$$\{\lambda_1, \lambda_2, \ldots, \lambda_n\}$$

Each represents the divergence in a specific manifold. Calculation of individual LCE is a very tricky business. Also, while deciding whether or not a trajectory is chaotic, it is more than enough to determine the maximum LCE. Maximum LCE will eventually be the dominating factor of the exponential divergence in the long run. In other words, maximum LCE approximately tells us how the magnitude of any perturbation to the trajectory would change.

$$|\mathbf{v}(t)| \approx e^{\lambda_{\max} t} |\mathbf{v}(0)| \tag{2}$$

When the maximum LCE ($\lambda_{\max}$) is known, deciding the chaoticity of the trajectory is very trivial. If the maximum LCE is non-positive, it points out that perturbation does not grow. They preserve their magnitude at most. In short, the trajectory is *ordered.* If the maximum LCE is positive, then the nearby trajectories diverge and how fast they diverge from each other is controlled by the value of the maximum LCE. The trajectory is *chaotic* and any uncertainty in the measurements will not be suppressed in the long run.

Since Lyapunov characteristic exponents are the set of divergence exponents, their sum can reveal if the phase volume shrinks, grows or remains the same.

[1] Strogatz, 2003

$\Lambda = \sum \lambda_i$	system type
$\Lambda < 0$	dissipative system
$\Lambda = 0$	conservative system
$\Lambda > 0$	growing system

Table 1: System Types and Sum of Lyapunov Characteristic Exponents

Here, it is important to make a few remarks on the fact that the system type and the chaoticity are two different things. One is a measure of how the phase volume changes as time passes whereas the other is whether the phase volume grows in any of the LCE manifolds. A system with one positive LCE and a few negative LCEs can still have a shrinking phase volume if the sum of LCEs is negative. In this case, the phase volume will be elongated in the positive LCE manifold yet shrink more in other manifolds so that the overall n-dimensional phase volume will be declining. This case is called *dissipative chaos* and it is the commonly studied version of chaos since in our daily lives, systems are doomed to lose energy by dissipative forces. Conversely, a growing system must be chaotic to be able to keep Λ at a positive value.

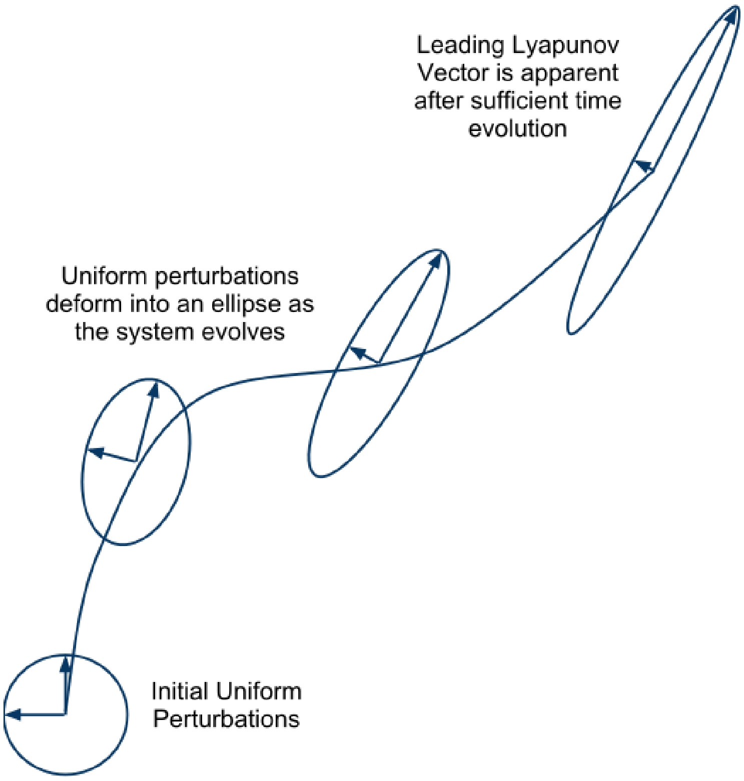
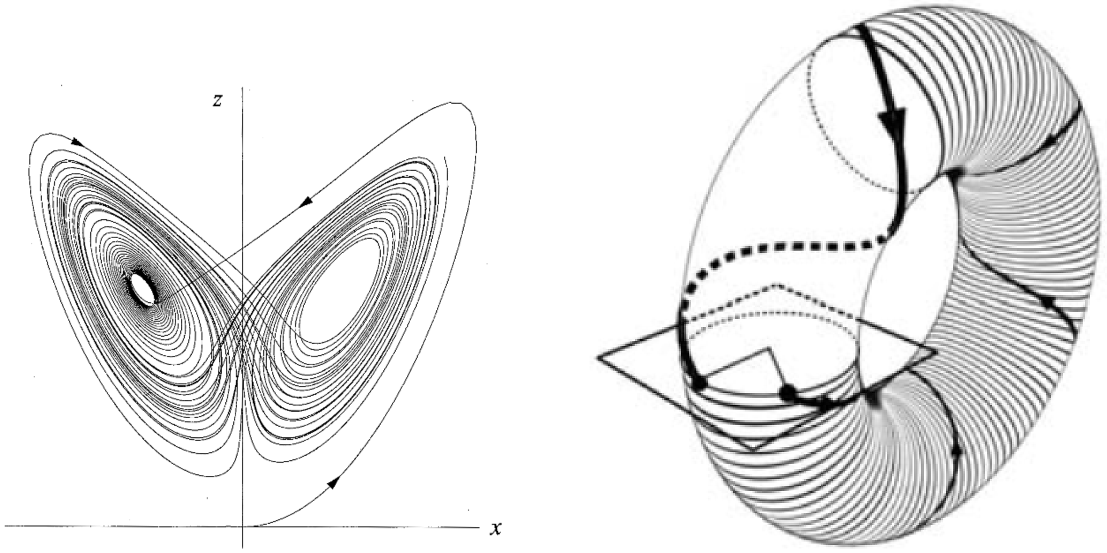

Leading Lyapunov Vector is apparent after sufficient time evolution
Uniform perturbations deform into an ellipse as the system evolves
Initial Uniform Perturbations

Figure 2: Phase volume changes in LCE manifolds [2]

The discussion of chaos in conservative systems is a bit more counter-intuitive to grasp compared to the others because we know that the Lyapunov characteristic exponents must add up to zero. This is another explanation of why the trajectories move on the surface of an **invariant** torus in conservative systems. However, the

[2] Wikipedia



(a) Lorentz Attractor

(b) Invariant Torus

Figure 3: Examples of Dissipative vs. Conservative Chaos[3]

invariant torus does not put any constraints on how the trajectories move. The answer merely depends on the maximum Lyapunov exponent. Positive $\lambda_{\max}$ indicates a chaotic trajectory and non-positive $\lambda_{\max}$ indicates an ordered trajectory yet bounded to the surface of an invariant torus in either case. The only condition for the conservative chaos is that the amount of divergence in the chaotic manifold must be compensated by the convergence of other (negative) LCEs exactly so that the cross-sectional area inside the invariant torus is preserved. There are further properties of the chaos in the Hénon-Heiles system, common to Hamiltonian systems.

Hamiltonian Systems: The Way Nature Speaks

Hamiltonian systems are very special in terms of their importance for the understanding of nature even though they are only a specific sub-category of conservative systems. They are defined by a specific set of phase variables and a particular rule that interlinks the choice of phase variables and the dynamics equation. Before going deep into the technical part, let's talk about where Hamiltonian systems come from. Since Sir William Rowan Hamilton introduced it in 1833 as a reformulation of classical mechanics, Hamiltonian mechanics has been a fundamental tool for physics. Even today, it is used for the most technical and theoretical analysis of the quantum realm. Hamiltonian mechanics provides a new perspective of reality by offering the concept of conjugate momenta and simplifying the analysis of a physical model in the configuration space. Configuration space is a very close notion to the phase space of the system dynamics. In short, Hamilton interpreted the functioning of the most fundamental nature (physics) in a way that the physics methods are compatible with the methods of system dynamics. Frankly speaking, it would not be too ambitious to argue that the mathematical foundation of system dynamics as an applied

[3] (a) Strogatz, 2003 (b) Gruiz and Tél, 2006

mathematics branch is based on this revolution.

Now, let's focus on the mathematical aspect of Hamiltonian systems. In such systems, phase variables consist of the canonical coordinates $\boldsymbol{q}$ and their conjugate momenta $\boldsymbol{p}$. These canonical coordinates can be Cartesian position elements as well as angle as an angular position variable as in the example of a simple pendulum. The conjugate momenta correspond to the physical momenta concept in classical mechanics. However, in Hamiltonian mechanics, it is a more generalized version defined by a quantity (in classical physics) or operator (quantum physics) called Hamiltonian. Hamiltonian is a measure of the total energy of the system and one can define the systems by accounting for the possible energy sources and forming a Hamiltonian. In general, a Hamiltonian consists of the kinetic and the potential part of a system. For example, by choosing the phase variables $\boldsymbol{\pi} = \{h, p\}$, the Hamiltonian for a falling ball is $H = T + V = p^2/2m + mgh$. In general, the Hamiltonian formalism of a N-dimensional physical system is given by

$$H(\boldsymbol{\pi}) = H(\boldsymbol{q}, \boldsymbol{p}) = H(q_1, q_2, \ldots, q_N, p_1, p_2, \ldots, p_N)$$
$$\dot{p}_i = -\frac{\partial H}{\partial q_i}, \; \dot{q}_i = \frac{\partial H}{\partial p_i}$$

Notice that the system dimension n is $2N$. N-dimensional physical system means that this system has N spatial dimensions and it is different from the dimension of the system. The form of a Hamiltonian system in the language of system dynamics is

$$\boldsymbol{\pi} = \begin{bmatrix} q_1 & \ldots & q_N & p_1 & \ldots & p_N \end{bmatrix}^T$$

$$\dot{\boldsymbol{\pi}} = \boldsymbol{f}(\boldsymbol{\pi}) = \boldsymbol{f}(\boldsymbol{q}, \boldsymbol{p}) = M_N \nabla_{\boldsymbol{\pi}} H(\boldsymbol{\pi})$$
$$= \begin{bmatrix} 0_{NxN} & \mathbb{I}_N \\ -\mathbb{I}_N & 0_{NxN} \end{bmatrix} \begin{bmatrix} \partial H/\partial q_i \\ \partial H/\partial p_i \end{bmatrix} = \begin{bmatrix} \partial H/\partial p_1 \\ \vdots \\ \partial H/\partial p_N \\ -\partial H/\partial q_1 \\ \vdots \\ -\partial H/\partial q_N \end{bmatrix}$$

and the dimension of the system is double the spatial dimension of the physical system. M_N here is called the symplectic matrix and it is the piece that gives its special character to Hamiltonian systems. It means that the time evolution of phase variables, canonical coordinates and conjugate momenta, is not independent. The dynamics of the canonical coordinates are anti-symmetric with the dynamics of the conjugate momenta. The choice of *conjugate* title of the momenta is, therefore, not random but rather expresses the symplectic nature of Hamiltonian formalism. This inherent structure provides Hamiltonian systems with distinctive properties. Firstly, we can check the phase space volume preservingness of Hamiltonian systems.

$$\begin{aligned}
\frac{d}{dt}\oint_{\partial V} d\boldsymbol{\pi} &= \oint_{\partial V} \frac{d\boldsymbol{\pi}}{dt} \cdot d\hat{\boldsymbol{n}}_{\partial V} \\
&= \oint_{\partial V} (M_N \nabla_{\boldsymbol{\pi}} H(\boldsymbol{\pi})) \cdot d\hat{\boldsymbol{n}}_{\partial V} \\
&= \int_V \nabla_{\boldsymbol{\pi}} \cdot (M_N \nabla_{\boldsymbol{\pi}} H(\boldsymbol{\pi}))\, dV \\
&= \int_V \sum_{i=1}^{N}\sum_{j=1}^{N} \left(\frac{\partial^2 H}{\partial q_i \partial p_j} - \frac{\partial^2 H}{\partial p_i \partial q_j} \right) dV \\
&= 0
\end{aligned}$$

This is called Liouville's theorem and shows that Hamiltonian systems are conservative. Conserved quantities of Hamiltonian systems (mechanical systems) are called integrals of motion. They are defined as time-independent functions whose values are preserved along the trajectories in the phase space. Hamiltonian of a Hamiltonian system gives the most obvious integral of motion: energy.

Another important property root in the symplectic structure is that their Lyapunov characteristic exponents come in positive and negative pairs.

$$\lambda = \{\lambda_1, -\lambda_1, \ldots, \lambda_N, -\lambda_N\}$$

This can be thought of as the following. LCEs amount to the divergence in LCE manifolds and these manifolds must be independent linear combinations of the phase variables. Therefore, they are automatically affected by the symplectic form of the phase variables. Even though we will not provide detailed mathematical proof of why LCEs come in pairs in Hamiltonian systems, one can grow an intuitive understanding that the divergence at some level in one manifold would mean convergence in the opposing manifold at the same level. This attitude is also consistent with the fact that phase volume must be preserved. This condition implies a deep fact about the ordered trajectories of Hamiltonian systems. If the sum of LCEs is 0, λ_{max} is non-positive, and LCEs are pair-wise, then all of the LCEs must be zero for ordered orbits.

Let's end our discussion about the Hamiltonian systems with a final comment. In Hamiltonian systems, two of the LCEs must be zero. This is because in systems with conserved phase space volumes trajectories cannot cross each other, and they move on the surface of an invariant torus. In 1983, Haken [8] provided a mathematical proof of this property, generalized to all phase volume-preserving systems. Combined with the pair-wise LCE nature, it requires 2 zero LCEs for Hamiltonian systems.

A Benchmark System: Hénon-Heiles Model

Hénon Heiles system is a model that describes the motion of a star in a galactic potential. For the sake of simplicity, Hénon and Heiles selected an axisymmetric potential, yielding planar motion. The (gravitational) potential energy of the star in the Cartesian coordinates is as the following.

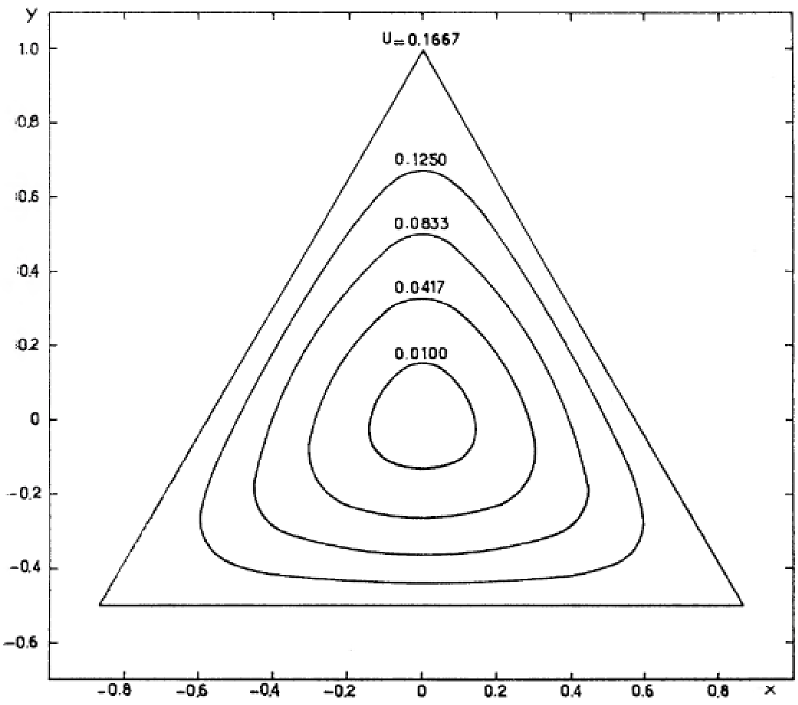


Figure 4: Equipotential lines for Hénon-Heiles system[4]

$$V(x,y) = \frac{1}{2}(x^2 + y^2) + \lambda\left(x^2y - \frac{y^3}{3}\right) \tag{3}$$

As it were the case in the previous studies on the investigation of chaotic orbits for the Hénon Heiles system, we also consider the nonlinear coefficient λ to be considered 1. Yet we will keep showing the coefficient λ explicitly for the completeness of our discussion. Yet remember that all of our results were obtained for $\lambda = 1$. This potential can have bounded trajectories only for energy values less than 1/6. After that value, trajectories can escape to infinity.

The Hamiltonian ($H = T + V$) of a system with the Hénon-Helies potential then becomes

$$H = \frac{1}{2}(p_x^2 + p_y^2) + \frac{1}{2}(x^2 + y^2) + \lambda\left(x^2y - \frac{y^3}{3}\right) \tag{4}$$

The dynamical equations governing this system can be set with the set of phase variables $\boldsymbol{\pi} = \{x, y, p_x, p_y\}$ and using the classical Hamiltonian mechanics' formulations $\dot{q}_i = \partial_{p_i}H$ and $\dot{p}_i = -\partial_{q_i}H$

[4] Hénon and Heiles, 1964

$$\begin{aligned}\dot{x} &= p_x\\ \dot{y} &= p_y\\ \dot{p}_x &= -x - 2\lambda xy\\ \dot{p}_y &= -y - \lambda(x^2 - y^2)\end{aligned} \tag{5}$$

When this model was first introduced by Hénon and Heiles in 1964, they devised this specific model, which results in planar trajectories in space, to simplify their objective, which was to find a second integral of motion of such a system. The integrals of motion are conserved quantities of mechanical systems as mentioned earlier. One obvious integral of motion is the energy for Hamiltonian systems, which is directly given by the Hamiltonian operator. In three dimensions with no axisymmetric potentials, the angular momentum is, analytically, known to be an integral of the motion. However, in the system of Hénon and Heiles, angular momentum is not one since the movement is constrained to a plane.

Hénon and Heiles studied the existence of isolating integrals of motion for fixed values of energy. Isolating integrals of motion can be thought of as the integrals of motion which constrain the shape of orbits to lower dimensional hypersurfaces. This is exactly what fixing the energy does to orbits. They did not expressly consider or notice the relationship between the number of isolating integrals of motions and the chaoticity of the orbits. This has been a hot topic in physics and mathematics up to the present as the endeavour of analytically studying the existence of *invariant, regular foliations* and it is called the study of *integrability.* Now, we know that regular/ordered orbits require N isolating integrals of motion in a n-dimensional system where we again recall that $n = 2N$. Even though integrability is an abstruse subject and is not directly a part of the objective of our project, this discussion here is only brought up to point out the deeper link between chaoticity and integrability [9].

Hénon and Heiles, in their paper, tried to observe this correlation between the characteristics of a trajectory and the existence of further integrals of motion for that trajectory. It is theoretically known that for a closed mechanical system, the number of independent integrals of motion is $2N - 1$ where N is the spatial dimension, which is $N = 2$ in our case. The idea on their paper was to handle the problem in a numerical manner, rather than trying to find the other integrals, analytically.

After the original work of Hénon and Heiles, the very same thing has been investigated widely in the study of nonlinear dynamics, only using different methods. The main conclusion to be gathered in such a study is that the existence of further integrals of motion depends on the values of energy, the default integral of the system. In numerical simulations made in all the studies on this model, non-ordered trajectories have been observed more and more as the energy increases.

To sum up, this system has one obvious (isolating) integral of motion: energy.

$$I_1 = \frac{1}{2}(p_x^2 + p_y^2) + V(x, y) \tag{6}$$

This condition enables us to analyze the 4-dimensional system in 3-dimensional hypersurfaces with constant energy values. Constant energy slices correspond to some specific region in the phase space, hence fixing the energy means that we are left with 3 degrees of freedom the specify the initial condition. To our knowledge, no one has been able to find a further integral of motion analytically. Hence, as Hénon and Heiles did, the common approach to the rest of the problem is utilizing numerical methods.

Lastly, we know that a 4-dimensional Hamiltonian system like the Hénon-Heiles system must have 2 LCEs with zero values. Pair-wise LCE nature of Hamiltonian systems then leaves us two options: ordered trajectories with LCEs $\{0, 0, 0, 0\}$ and chaotic trajectories with LCEs $\{0, 0, \lambda, -\lambda\}$. Our project is conducted based on this result.

Problem Statement

This project aims to study the general properties of the H ́enon-Heiles system, one of the mechanical systems that exhibit chaos - most specifically, the phenomenon of conservative chaos. The obvious objective is to study if this system can exhibit chaotic behaviour. However, knowing that the H ́enon-Heiles system can indeed have chaotic orbits, our main concentration will be mostly on the question of how can we assess this chaoticity. For this purpose, theoretical computations on the H ́enon-Heiles Hamiltonian are to be used to construct a numerical scheme to investigate the chaotic nature of the phase space. Then, the numerical scheme will be run using computer tools to obtain the phase space properties, providing information about the system characteristics in different regions of the phase space and how the transition occurs between these characteristically different regions. We will make use of different methods of assessing chaoticity and discuss how they are advantageous to each other and how the drawbacks of the methods constrain our ability to come up with a conclusion.

3 Methods

Method of Poincaré Surface

The method of Poincaré Surface is one of the commonly used methods for the analysis of high-dimensional dynamical systems. The idea is to observe the motion of a swirling trajectory by placing a surface/section in the n-dimensional phase space that the trajectory would pierce as it swirls and tracking the piercing points on this surface, which is called Poincaré Surface. The map that describes where the orbit pierces the Poincaré surface is called the Poincaré map, $\mathcal{P}$.

$$\boldsymbol{x}_{k+1} = \mathcal{P}(\boldsymbol{x}_k)$$

Note that the Poincaré map only accounts for the piercings in one specified direction, only from one side of the surface to the other, so that a periodic orbit would generate a Poincaré map of a kind $\mathcal{P}(\boldsymbol{x}_k) = \boldsymbol{x}_k$. Hence a single point on the Poincaré surface indicates a periodic orbit. For example, if there are two points on the Poincaré Surface, it means that this orbit closes its path on its own after two swirls and such orbits are called period-2 orbits. Generalizing this behaviour, period-n orbits are trajectories that finish one period of their overall behaviour after n piercing through the Poincaré surface, i.e., $\mathcal{P}^n(\boldsymbol{x}_k) = \boldsymbol{x}_k$. Note that this period-$n$ concept is related to the choice of Poincaré surface and regardless of how many swirls with respect to that Poincaré surface is required to complete one period, they are all periodic orbits.

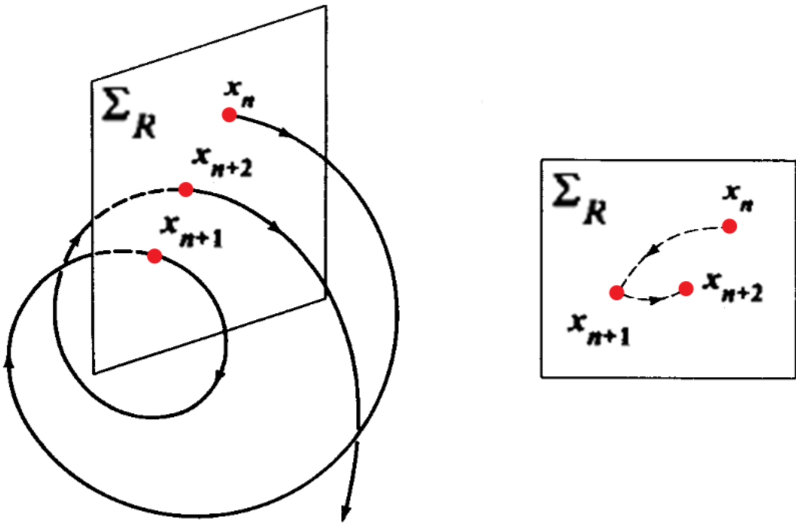


Figure 5: Poincare map of a swirling motion[5]

What if there are infinitely many piercing points on Poincaré surface as $t \to \infty$? This situation may be signalling two scenarios: quasi-periodic orbits or chaotic orbits. Quasi-periodic orbits can be better understood by the comparison of them with n-period orbits. For an n-period orbit, each swirl, actually, means a phase contribution (rate of rotation) added to the motion of the piercing points on the Poincaré surface such that after n of these phase contributions are added one after the other, the net

[5] Lieberman and Lichtenberg, 1992

phase becomes 2π and one period is completed. If the rate of rotation is a *rational* multiple of 2π, then the orbit is a periodic orbit. On the other hand, a rate of rotation which is an *irrational* multiple of 2π means that the trajectory will never be able to close on itself even if it swirls until infinity. Such orbits are called *quasi-periodic* orbits. This is why they generate Poincaré surfaces with infinite piercing points, however; note that they are still ordered, not chaotic. In other words, a small disturbance to a quasi-periodic orbit would produce another quasi-periodic orbit, only shifting a little. Therefore, the pattern quasi-periodic orbits make on the Poincaré surface is usually an open or closed path. Whereas Poincaré surface of a chaotic orbit consists of randomly scattered piercing points. Any infinitesimal perturbation also scatters and there exists no pattern on the Poincaré surface. This is the how method of Poincaré surface can be used to assess chaoticity.

We know that the Hénon-Heiles system is a 4-dimensional Hamiltonian, conservative, system with an analytically known isolating integral of motion: energy. By fixing energy, the motion of an arbitrary trajectory is confined to a 2-dimensional hypersurface. Moreover, we can set a 1-dimensional Poincaré surface to observe the whirling behaviour of the system. Ultimately, we obtain a 2-dimensional scheme to track the attitude of trajectories. In our analysis, the Poincaré surface we choose is the line of $x = 0$. We could have chosen any 1-dimensional surface in the 4-dimensional phase space yet the choice of $x = 0$ brings convenience in terms of easing the computations and enabling physically relatable envisagement.

Maximum Lyapunov Exponent via Variational Techniques

Another method to assess the chaoticity of an orbit is to calculate its maximum Lyapunov exponent. There are different methods to calculate max. LCE and here we will use a relatively old but conceptually easy-to-follow method by Benettin et al. [10] which is based on variational approach. Let's introduce a deviation to the initial condition of the trajectory at hand and try to find a time-evolution scheme for the deviation vector $\boldsymbol{v} = [\delta x \, \delta y \, \delta p_x \, \delta p_y]^T$ by means of variational equations.

$$\frac{d\boldsymbol{v}}{dt} = M_N \cdot P \cdot \boldsymbol{v}$$

where M_N is the symplectic matrix and $P = P_{ij} = \partial^2 H / \partial q_i \partial q_j$. Making the calculations, one can easily find the so-called *variational dynamics.*

$$\boldsymbol{v} = \begin{bmatrix} \delta x \\ \delta y \\ \delta p_x \\ \delta p_y \end{bmatrix} \Rightarrow \dot{\boldsymbol{v}} = \begin{bmatrix} \delta p_x \\ \delta p_y \\ \delta x - 2x\delta y - 2y\delta y \\ -\delta y - 2x\delta x + 2y\delta y \end{bmatrix}$$

Maximum Lyapunov exponent is a measure of how the magnitude of the deviation vector exponentially changes over time. A numerical (discrete) approximation of this measure can be realized by evaluating the deviation vector alongside the original trajectory and recording the magnitude of the deviation vector at every time step. Yet, at the same time, we must be sure that the probable exponential growth should not cause any computational errors by increasing too fast. For this reason, we need to normalize the deviation vector to its initial magnitude at each step to keep it in the vicinity of our trajectory.

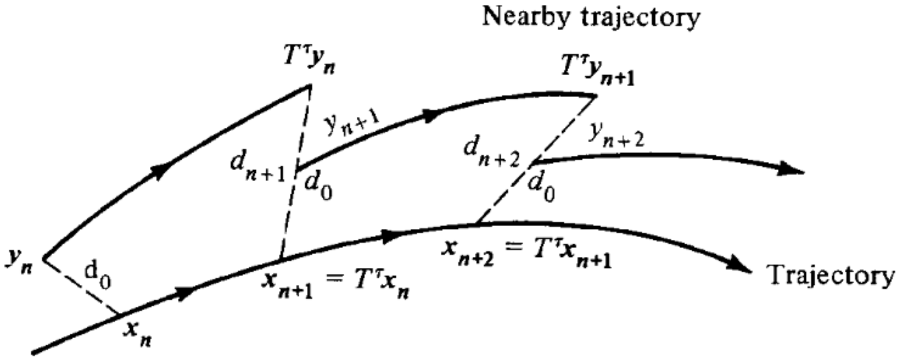


Figure 6: Calculation of Lyapunov exponent[6]

After all these procedures, we can define the maximum Lyapunov exponent by reverse engineering Eq. (2) where τ is the time interval and n is the time step.

$$\lambda_{\max}[n] = \frac{1}{n\tau} \left| \sum_{i=1}^{n} \ln v_i \right| \tag{7}$$

This is a discrete approach to the theoretical definition of the maximum Lyapunov exponent. By this method, we have a $\lambda_{\max}$ value for every time step n and after some sufficiently large n the value we calculate should converge to the theoretical value. For ordered orbits, we expect a zero $\lambda_{\max}$ as t goes to infinity and for chaotic orbits, $\lambda_{\max}$ should stay positive. A more profound technique would be evaluating a small hypersphere around the initial condition to eliminate accidental errors.

Small Alignment Index (SALI)

In 2001, Skokos [11] came up with a new method of checking chaoticity by using a quantity similar to the Lyapunov exponent. This time, we evaluate 2 deviation vectors $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ alongside the trajectory. We further define two quantities: parallel alignment index,

$$d_-(t) \equiv |\boldsymbol{v}_1(t) - \boldsymbol{v}_2(t)|$$

and antiparallel alignment index

$$d_+(t) \equiv |\boldsymbol{v}_1(t) + \boldsymbol{v}_2(t)|$$

They are indirect measures of how the angle between two deviation vectors changes. In his paper, Skokos gives his chaoticity assessing quantity SALI as the following.

$$\text{SALI} \equiv \min\{d_-(t), d_+(t)\} \tag{8}$$

[6] Benettin et al., 1976

SALI stands for smaller alignment index as the smaller of the parallel and antiparallel alignment indices.

Skokos argues that if a trajectory is chaotic, both of the deviation vectors must align with the manifold of the positive Lyapunov exponent. This alignment can occur in two cases, they may align in parallel or in opposite. If they tend to align in parallel

$$d_-(t) \to 0, d_+(t) \to 2, \mathrm{SALI} \to 0$$

or if they align opposite to each other, then

$$d_-(t) \to 2, d_+(t) \to 0, \mathrm{SALI} \to 0$$

If the trajectory is ordered, the deviation vectors converge on the invariant torus but in different directions, therefore, SALI fluctuate around a positive value. Skosos set 10^-8 as the threshold value for the chaotic orbits. Orbits with SALI above 10^-8 is accepted to be ordered and below 10^-8 is accepted to be chaotic. For more detailed information, one can check Skokos et al. [12].

Comparison of Methods

The reason we use different methods is because each method has different decisive points. For instance, the Method of Poincaré Surface indicates that the orbit is chaotic if the piercing points are scattered, yet ordered piercing points cannot guarantee that the orbit is ordered since the chaoticity of the orbit may be too low to start piercing the Poincaré Surface in a scattering manner. Another drawback of the Method of Poincaré Surface is that it requires visual inspection for the loci of the piercing points. However, maximum Lyapunov exponent and SALI produce numerical results. Yet, they are still limited to be decisive only for one thing. Maximum Lyapunov exponent can only decide if the orbit is ordered when $\lambda_{\max}$ gets to zero, and SALI can only tell if the orbit is chaotic when SALI becomes zero. The other way around is not feasible to argue because all of these methods give exact outcomes in the limit $t \to \infty$. In short, decision making and reasoning is dependent on the methods we use as it is summarized in Tab. 2. This situation has both technical and philosophical implications on the reliability of our observations, even our concepts. We will talk about these implications in detail in the conclusion section.

Method	Poincaré Surface	Max. Lyapunov Exponent	SALI
decisive point	chaotic if scattered piercing	ordered if $\lim_{t\to\infty} \lambda_{\max} = 0$	chaotic if $\lim_{t\to\infty} \mathrm{SALI} = 0$

Table 2: Comparison of Methods for Chaoticity Assessment

Numerical Setup

The primary tool of this project is the numerical simulations, which was conducted mainly using the Python programming language regarding its convenience, strength, and popularity. In our simulations, we used the popular time evolution algorithm, 4th order Runge-Kutta. In our procedure, we fixed an energy value first, then, determined 3 of the 4 initial conditions since the last one is predetermined by the energy. Then we evaluated our scheme for 10^4 time units with 10^{-3} time unit steps. Then we repeated the same simulation for different energies and initial conditions for each energy.

At first, we evolved four deviation variables in addition to 4 phase variables. These 4 deviation variables are to calculate how the distance to a nearby point in phase space changes. In other words, the deviation variables are used in calculating Lyapunov characteristic exponents via the variational analysis. Since the phase space needs to be constrained, we normalize the deviation vector to the initial norm at every iteration of Runge-Kutta, as explained before. We take the norm of this deviation vector at every time slice and compute the cumulative sum of the natural logarithm of this norm. This cumulative sum stands for the numerical integration up to every time point and gives us the Lyapunov characteristic exponents. Then for SALI, we repeated the same thing by adding one more deviation vector, hence we evaluated 12 variables in total.

We checked for Poincaré surfaces and used the Poincaré surface as the plane spanned by y and p_y, i.e., $x = 0$. However, one cannot guarantee whether or not the limited precision of a numerical simulation trajectory sequence obtained in the simulation coincides with the Poincaré surface. Therefore, we found the uttermost close points to the Poincaré surface from each side and interpolate to find where the trajectory would pierce the Poincaré surface. These interpolated points constitute the Poincaré plot.

At this point, it is meaningful to mention some limitations of our numerical setup and its implication to our computation of the maximum Lyapunov characteristic exponent. Piercing points of Poincaré surface, LCEs and SALI are meaningful in the limit of time going to infinity and in the limit of time step going to 0. The size of a typical file containing a single trajectory computed with the final time and time step values mentioned above is around 1.2-1.5 GB. This defines a trade-off between these two parameters of our simulation. The longer we evaluate, we have to pick wider time steps to keep the file sizes in a viable range to be processed. We tried different final times and time steps for our simulation and concluded that above some value of time step, the Runge-Kutta algorithm could not work as it should and instead, it produces nonsensical results. This puts an upper limit on the time step. The methods we use are suggestive only as time goes to infinity. In the earlier times of the simulation, it gives no answer to the question regarding the characteristic of the trajectory. This puts a lower limit on the total duration of the simulation. Considering all these, we tried to find an optimal value for both the time step and the duration, yet, in that optimal value, the results are not as good as one can get using better computing equipment. This being mentioned, our analysis has been successful and adequate to make some conclusions.

4 Results

In this section, evident theoretical results about the Hénon-Heiles system are compared to the results of the numerical simulations. We expected to see a gradual transition from ordered phase space to chaotic phase space as energy increases, considering the previous studies on the chaotic nature of the Hénon-Heiles system. In the hypothetical phase space animation of infinitesimally increasing energy values, the regions of ordered orbits should fade out in the growing chaotic sea, only leaving small islands of stability behind. We expect to find the maximum Lyapunov exponent, which is positive for the trajectories starting from the chaotic sea, and the maximum Lyapunov exponent converging to 0 for the trajectories starting from the islands of stability. On the other hand, the trajectories belonging to the chaotic sea regions in the Poincaré surface should have a zero SALI, while trajectories belonging to the islands of stability should fluctuate around a positive value. The concepts of the chaotic sea and islands of stability will get more apparent as we present the Poincaré surface results.

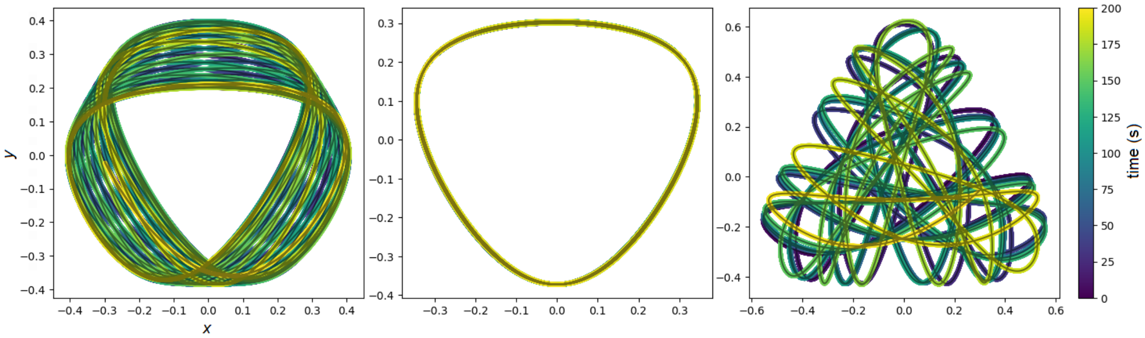


Figure 7: Different trajectories corresponding to y_0 and energies: 0.2-0.15, 0.3-0.125, 0.3-0.15. Quasi-periodic, quasi-periodic, and chaotic trajectory, from left to right. Axes show the spatial coordinates, and the colour bar represents the time.

The numeric precision we chose for our simulations, i.e. $dt = 10^{-3}$, seems to satisfy our expectations as the simulations form a continuous and meaningful sequence of points we can easily name as a trajectory. In Fig. 7, we can see different trajectories starting from different initial conditions. Note that all of the initial conditions chosen in our work are in the form of $(x, y, p_x, p_y) = (0, y_0, p_x(E), 0)$. y_0 is given by us as an external input and p_x is automatically determined by the fixed energy values via the Hamiltonian formula. The topological shape of the trajectories has already revealed greatly. The left one seems like a quasi-period orbit. We can still observe that it is actually a higher-period closed orbit if we run the simulation more. We tried to investigate this by lowering the time step by 10 and increasing the simulation duration by 10. We saw that the trajectory does not close on itself. In much much longer run simulations, the trajectory may show that it is periodic; however, the precision limitations may be hiding this fact. Hence, one needs to be very cautious when making a conclusion.

The middle trajectory is again a quasi-periodic orbit. Despite its closed-path form, we know closeness in 2D does not mean it is a periodic orbit in 4D. We will discuss why it is quasi-periodic and how strong our argumentation is. We will see that

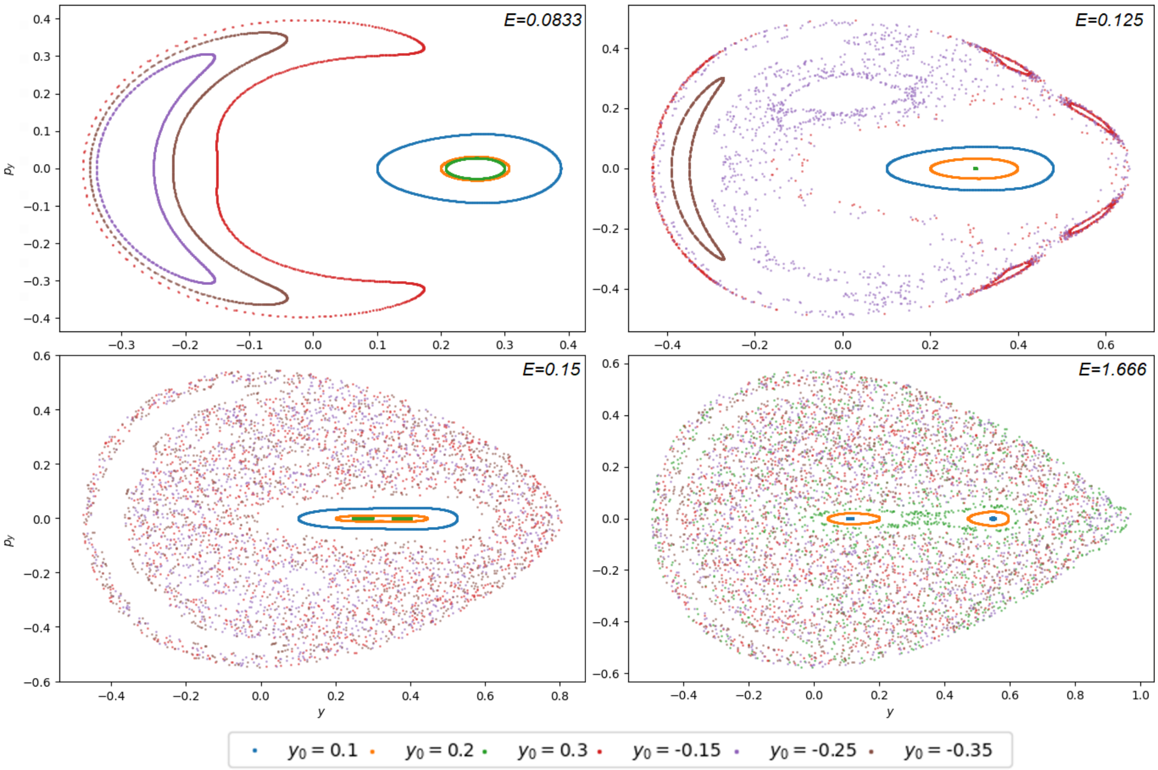


Figure 8: Poincaré surface plots for different energies starting from the same 6 initial conditions in each energy level. 0.0833, 0.125, 0.15, 0.166; top left, top right, bottom left, bottom right. As energy increases, the distribution of piercing points in the Poincaré surface gets scattered, showing that trajectories get more and more chaotic.

limitation of the precision electronic computers again stands as an obstacle to making more confident statements at this point. However, the fact that this model has been studied intensively before and that our expectations from the theoretical conclusions and others' work coincide with the results we get from our own simulations provides a sense of correctness. Judging from that, it is sufficient so far for us to proceed with the assumption of that, indeed the left one is quasi-periodic as well as the middle one.

The right plot shows a chaotic orbit while it is easy to reckon that its complexity and non-order are what renders it a chaotic trajectory. However, one still needs to check this hypothesis more rigorously and scientifically. In the bottom left panel of Fig. 8 is Poincaré surface plot for 6 different initial conditions, including the ones provided in Fig. 7 at energy 0.15. One should notice that fixing the energy already reduces the dimension to 3, giving us constant energy manifolds. By further choosing the Poincaré surface as the $p_y - y$ plane, we get 2D graphs where we can observe the trend of the trajectories through their geometrical indications.

Fig. 8 demonstrates how the characteristics of the trajectories starting from the same (x, y, p_y) change as the energy increases. In the more comprehensive perspective of full-dimensionality of our model, what we look into is that trajectories in different parts of the 4D phase space exhibit different properties, and a physically intuitive quantity, energy, is directly linked to the classification of the trajectories and their positions in the phase space. In systems with dissipative chaos, the variation of model

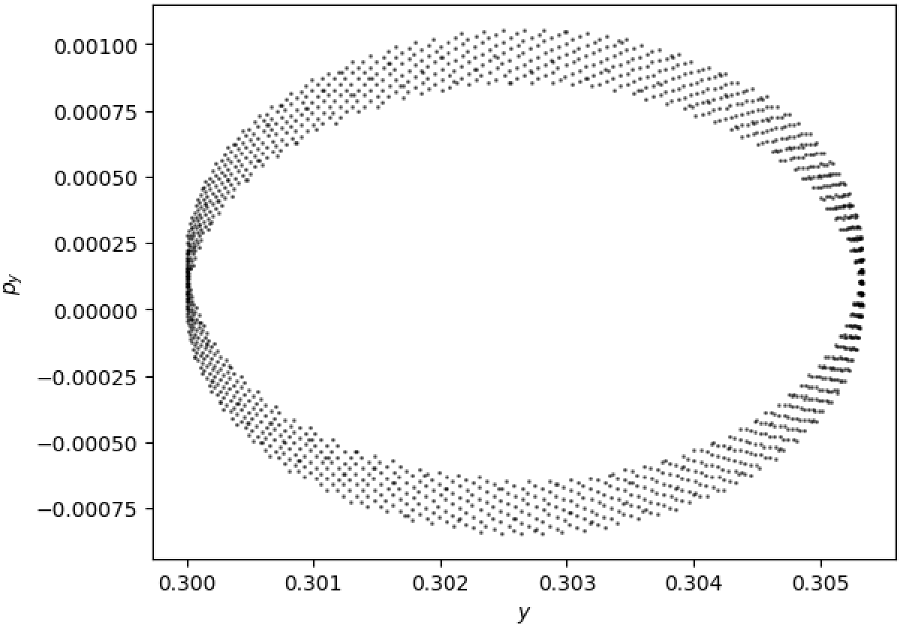


Figure 9: Zoomed-in Poincaré surface plot of the trajectory starting with the energy of 0.125 and the initial conditions $(0, 0.3, p_x(E = 0.125), 0)$.

parameters, bifurcation, defines the trajectory properties. However, in conservative systems, chaos is inherently a part of the system showing itself only some part of the phase space. The generic procedure to study these parts is to slice the phase space by the energy levels and span the 4D space with lower-dimension hypersurfaces.

Fig. 8 describes 4 different hypersurfaces. We give the Poincaré surface plot of the trajectory shown in the middle panel in Fig. 7. It is shown by the green curve in the Poincaré surface plot. The overall look of the graph including different trajectories is not detailed enough to show that it is quasi-periodic. Fig. 9 is the zoomed-in graph and one can see it has an ellipse shape rather than a single point. We would expect it to form a one-point-thick ellipse rather than an annulus-like shape. We think that this is due to the discreteness of our simulations based on the fact that the length of the ellipse from one side to another is in the order of our precision.

A broad analysis of Fig. 8 tells the whole story about how the trajectory behaviours depend on the energy. In the lowest energy given, all trajectories produce an ordered Poincaré plot. As the energy increases, the distribution of the piercing points on the Poincaré surface gets more scattered and *chaotic*. The stability islands we expected to observe are more visible in intermediate energy values, yet at the highest energies, ordered trajectories almost vanish away. The energy of the bottom right panel, 0.166, is very much close to the escape energy from the Hénon-Heiles potential, $1/6 = 0.1\bar{6}$. We cannot run simulations with energies higher than this escape energy. Therefore, we cannot see if all ordered trajectories turn into chaotic ones. It is possible to get infinitesimally close to the escape energy $0.1\bar{6}$ and investigate how does the trajectories transform at every small towards the escape energy, yet this is not in the scope of our study. Still, one can hope to observe that, indeed, all of the regular orbits become chaotic.

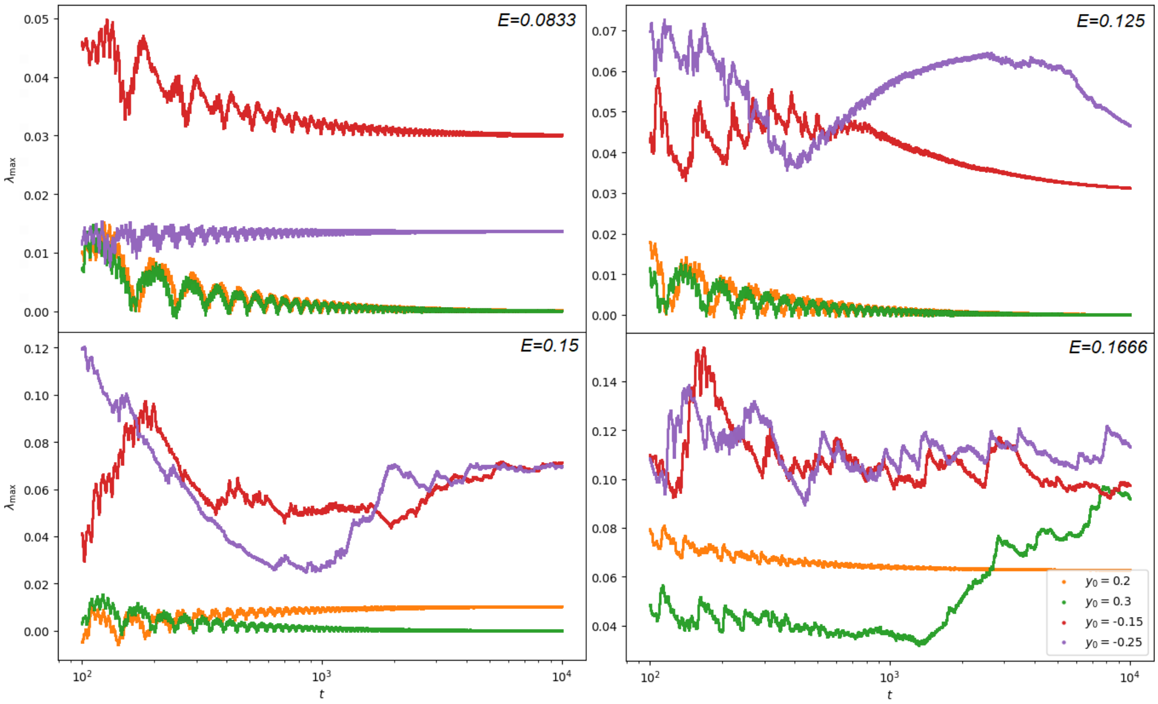


Figure 10: Lyapunov exponents for the same energies in the Poincaré surface plots. The $\lambda_{\max}$ range increases as the energy increases as a consequence of the non-order behaviour of higher energy trajectories.

The most popular measure of the disorder of the trajectory is its maximum Lyapunov characteristic exponent. The results we get after the calculation of max. LCE, as explained in the Methods section, are presented in Fig. 10. We expect to find the max. LCE to converge to 0 for ordered orbits. For chaotic trajectories, we expect the max. LCE to stay in some positive range. In the limit $t \to \infty$, we should get a specific value for the max. LCE even for chaotic orbits but we are constrained by the duration of our simulations.

In Fig. 10, as energy increases, for the trajectories whose max. LCE converges to 0, max. LCE deviates from the neighbourhood of 0. Gradually all of the max. LCEs of the trajectories stay in a positive range. Another observation is that the top value of this positive range where max. LCE is bounded increases in higher energies. This simply tells that at higher energies, the trajectories are somewhat more disordered. One important point to be made here is that the max. LCE of some of the orbits that we titled as quasi-periodic in our Poincaré surface analysis does not converge to 0. One of the reasons for this problem is, of course, the finite-time nature of our simulations. We cannot calculate the LCE in the temporal infinity. Another one is the finite-time step of our nature. The continuity of our trajectory can be bettered with smaller step sizes. It may also be due to the fact that storage units in our usual computers have a precision limit on the data they can store. We must also consider that there are methods better than the ones we used to calculate the max. LCE. Improved approaches can give better/more reliable outcomes. On the other hand, LCEs may not be converging to 0 simply because they do not belong to an ordered trajectory. The transition from ordered trajectories to non-ordered/chaotic

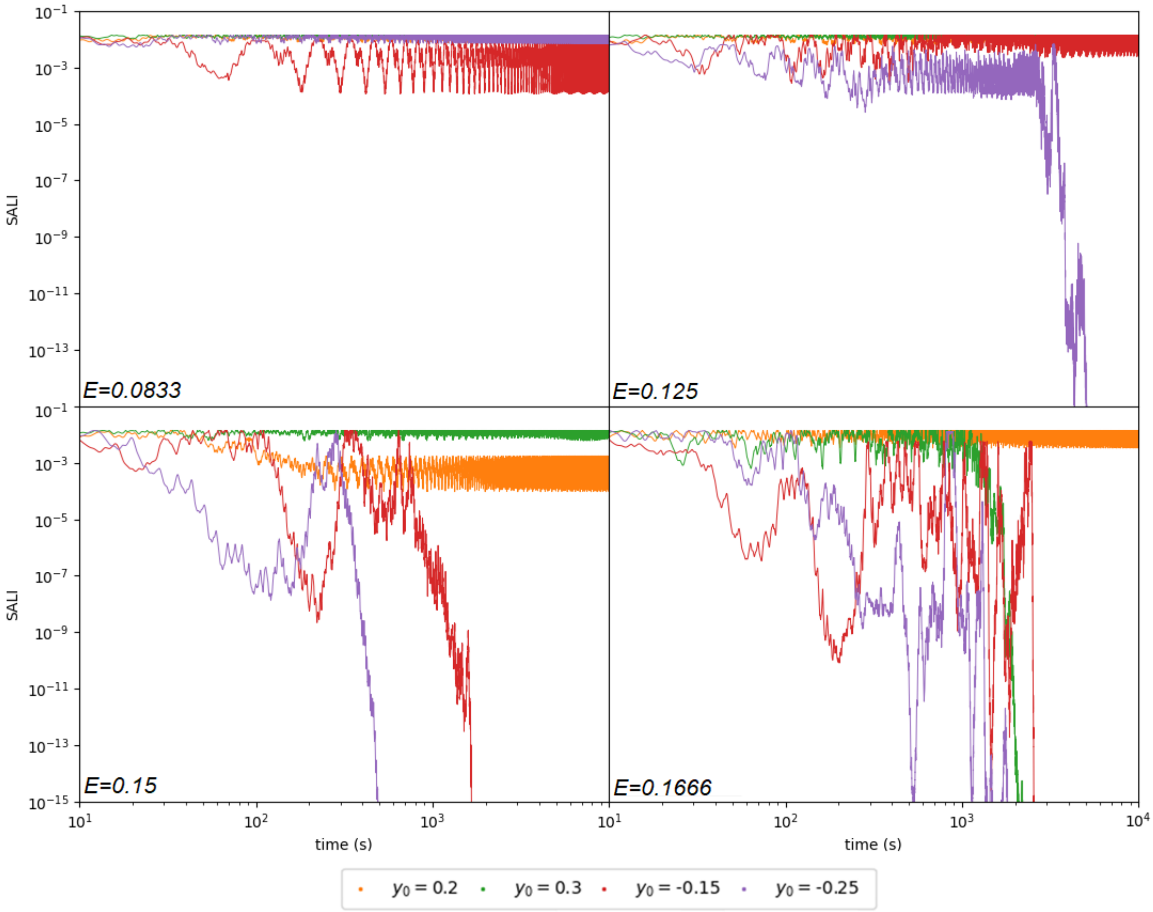


Figure 11: SALI plots for the same energies in the Poincaré surface and max. LCE plots.

trajectories may not be in an on/off switch fashion. Rather this transition may be gradual so that while an ordered orbit still seems like an ordered one as the energy increase, it may have already become a disordered trajectory, yet it requires more time for its effects to be observed.

We can use our analysis by the method of Smaller Alignment Index to approach the problem from the opposite side since SALI is successful on deciding whether or not the orbit is chaotic whereas max. LCE can only say if it is ordered. In Fig. 11, we present SALI plots for the same initial conditions and the same energies presented for max. LCE plots. For $E = 0.833$, none of the SALIs of the orbits fall below the threshold Skosos deteremined, i.e., 10^{-8}, in our simulation time. However, the red orbit gives the signals of turning disorder. It may happen if we wait enough or if we perturb it by means of slightlt increasing its energy. However, when we check the $E = 0.125$ plot, we see that it is the purple orbit that becomes chaotic. Notice that in the Poincaré surface plot of the red orbit for this energy piercing points already get scattered and the max. LCE plot stay positive, all indicating that the orbit is disordered whereas SALI value of the same orbit stays above the chaoticity threshold. This again brings the idea that the transition from ordered to disordered may be happening smoothly. When we check the plot for $E = 0.15$, we see that the results from all of our methods are consistent with each other. Finally, for $E = 0.166$, again, all results are coherent except SALI eliminates the dubious situation about the orange orbit. Poincaré surface analysis shows that it is ordered, but this judgement is made by only visual inspection and cannot be trusted easily. Max. LCE analysis, on the other hand, reveals a confusing fact. Orange orbit's max. LCE converges in a highly non-zero, positive value. Convergence may be an indication of ordered orbit, yet it is very suspicious that the convergence value is not zero. SALI analysis shows that the orange orbit is still in the ordered zone, above the threshold. Even though almost all of our results are consistent, and we easily observe the chaotic nature of a Hamiltonian system as the energy increases, the methods we use may be misleading under some circumstances. The impacts of this problem are to be discussed in the conclusion section.

5 Conclusion

In our project, we tried to implement different methods to analyze a benchmark model for the study of the conservative chaos phenomenon. The choice of a commonly studied system has been the most helpful since we obtained the opportunity of focusing on the implications of different methods without getting suspicious about the results of our simulations. One of the important conclusions we can make is that the analysis method means everything, especially in the survey of chaos since it introduces an unpredictability to the analyses by definition. Assume that one uses a method that is decisive for chaotic orbits, such as SALI. If the analysis concludes that the orbit is indeed chaotic, everything is okay. If the analysis cannot make the conclusion that the orbit is chaotic, it does not mean that the orbit is non-chaotic. Even if the analysis gives a non-zero SALI at the end of the simulation/observation, it is merely impossible to decide on the favour of ordered orbits. One can sense that the orbit is unlikely to be chaotic, but in the framework of the scientific method, the analysis must stay inconclusive. This example can be generalized to how we use our concepts. For instance, when we say that the motion of the earth around the sun is ordered without using the analysis methods of system dynamics, it is nothing but a shallow opinion.

This is another main result of our study. The original work of Hénon and Heiles showed that the motion of celestial bodies may have a non-zero maximum Lyapunov exponent. As a matter of fact, we know today that most of the planets, asteroids, and stars we know have chaotic orbits, including our planet. Their max. LCE may be very small to perceive by humans or not effective in changing our daily life; however, it is not the case for the Hyperion, a moon of Saturn. The maximum Lyapunov exponent of its orientation with respect to its axis is measured to be 1/10 days meaning that an initial uncertainty of 1 degree grows exponentially and leads to an uncertainty of 20 degrees in a month [5]. This is surprising because such a close-by body behaves highly chaotic, and such findings are immensely crucial, particularly, in the times we think of harvesting asteroids, moons, and planets.

However, the study of celestial bodies is not the only field we may use our findings. In control studies, dissipative systems are often supplied with a specifically calculated input source via a feedback and it is obtained a system which has a conservative nature as the overall closed loop form. Our project, once again, shows that a conservative system can exhibit chaos regardless of it is a system of celestial bodies or an artificially built conservative feedback system. Control applications on such models/systems definitely requires more studies on the field of conservative chaos.

Realistic Constraints

Since the physical realization of a two-body system governed by the Hénon-Heiles model is not feasible, this project will have only theoretical and computational approaches. The numerical scheme has the constraint of numerical precision versus runtime trade-off and towards which side of this trade-off we will tend is only dependent on the typical duration of this project, three months. Nevertheless, we don't think our time limitation will not lead to any disadvantages regarding the computational capabilities of computers in 21$^{\text{st}}$ century.

a. Social, Environmental and Economic Impact

The importance of conservative chaos may not be well appreciated, given that chaos manifests itself mainly in the form of dissipative chaos due to restoring forces such as friction. However, considering the recent advancements in space technologies, humankind will encounter engineering challenges on systems exhibiting conservative chaos in outer space more and more. Some examples are space transportation, popularized lately by the leap of commercial space technology companies, and the understanding of the nature of celestial bodies like satellite missions. This fact automatically requires more research on the essence of chaos that will provide information for chaos control. The way things stand, our engineering problems in the future will step into the territory of chaos more and more.

b. Cost Analysis

1 personal computer with 16 GB RAM and 3.2 GHz processing rate: 17.700 TL

Hypothetical salary for the student working 15 hours per week: 3120 TL

c. Standards

No applicable standards, except for the general engineering code of conduct.

6 References

[1] Michel Hénon and Carl Heiles. “The applicability of the third integral of motion: Some numerical experiments”. In: *The Astronomical Journal* 69 (Feb. 1964), p. 73. DOI: 10.1086/109234.

[2] Y. F. Chang, M. Tabor, and J. Weiss. “Analytic structure of the Henon–Heiles Hamiltonian in integrable and nonintegrable regimes”. In: *Journal of Mathematical Physics* 23.4 (Apr. 1982), pp. 531–538. DOI: 10.1063/1.525389.

[3] Euaggelos E. Zotos. “Classifying orbits in the classical Henon-Heiles Hamiltonian system”. In: *Nonlinear Dynamics* 79.3 (Feb. 2015), pp. 1665–1677. DOI: 10.1007/s11071-014-1766-6. arXiv: 1502.02510[nlin].

[4] Iaroslav Ispolatov et al. “Chaos in high-dimensional dissipative dynamical systems”. In: *Scientific Reports* 5.1 (July 30, 2015). Number: 1 Publisher: Nature Publishing Group, p. 12506. DOI: 10.1038/srep12506.

[5] “Applications of chaos”. In: *Chaotic Dynamics: An Introduction Based on Classical Mechanics*. Ed. by Márton Gruiz and Tamás Tél. Cambridge: Cambridge University Press, 2006, pp. 279–317. DOI: 10.1017/CBO9780511803277.011.

[6] Xiaojuan Zhang et al. “Dynamic Analysis and Degenerate Hopf Bifurcation-Based Feedback Control of a Conservative Chaotic System and Its Circuit Simulation”. In: *Complexity* 2021 (May 29, 2021). Ed. by Shijian Cang, pp. 1–15. DOI: 10.1155/2021/5576353.

[7] Jianhong Hao, Jieqing Fan, and Ru Duan. “Controlling chaos in conservative systems by the modified linear feedback method”. In: *2013 10th IEEE International Conference on Control and Automation (ICCA)*. 2013 10th IEEE International Conference on Control and Automation (ICCA). ISSN: 1948-3457. June 2013, pp. 518–521. DOI: 10.1109/ICCA.2013.6565108.

[8] H. Haken. “At least one Lyapunov exponent vanishes if the trajectory of an attractor does not contain a fixed point”. In: *Physics Letters A* 94.2 (Feb. 1983), pp. 71–72. DOI: 10.1016/0375-9601(83)90209-8.

[9] Jaume Masoliver and Ana Ros. “Integrability and chaos: the classical uncertainty”. In: *European Journal of Physics* 32.2 (Mar. 1, 2011), pp. 431–458. DOI: 10.1088/0143-0807/32/2/016. arXiv: 1012.4384[physics].

[10] Giancarlo Benettin et al. “Lyapunov Characteristic Exponents for smooth dynamical systems and for hamiltonian systems; a method for computing all of them. Part 1: Theory”. In: *Meccanica* 15.1 (Mar. 1980), pp. 9–20. DOI: 10.1007/BF02128236.

[11] Ch Skokos. “Alignment indices: a new, simple method for determining the ordered or chaotic nature of orbits”. In: *Journal of Physics A: Mathematical and General* 34.47 (Nov. 30, 2001), pp. 10029–10043. DOI: 10.1088/0305-4470/34/47/309.

[12] Ch Skokos et al. *Smaller alignment index (SALI): Determining the ordered or chaotic nature of orbits in conservative dynamical systems*. Oct. 23, 2002. arXiv: nlin/0210053.

Appendices

A Simulation Code

```python
from IPython.display import display, Latex
import sympy, os, json
import matplotlib as mpl
import matplotlib.pyplot as plt
import import_ipynb
from datetime import datetime
from mpl_toolkits import mplot3d
import pandas as pd
import numpy as np

class phaseSpace:

    def __init__(self, dim, order = 2):

        self.dim = dim
        self.order = order
        self.psDim = dim*order

    def setVariables(self, variableString):

        self.variables = variables = variableString.split()

        if len(variables) == self.psDim:

            print('Variables set:', variables)

            for variable in variables:

                exec("self.%s = 0" % (variable))
                exec("self.v_%s = 0" % (variable))

            self.setPhase(np.zeros(self.psDim))
            self.setVariationalPhase()

        else:

            print('Variable number does not match with phase space dimension')

    def printSystem(self):

        display(Latex(f'$---system---$'))

        dynamics = list(self.dynamicsDict.values())

        for idx, var in enumerate(self.variables):
            exec("%s = self.%s" % (var, var))

        for idx, val in enumerate(self.variables):

            display(Latex(f'$\dot'+str(val)+'='
            +sympy.latex(sympy.sympify(dynamics[idx]))+'$'))

        display(Latex(f'$-----------$'))

    def setDynamics(self, exprDict):

        if len(exprDict) == self.psDim:

            self.dynamicsDict = exprDict
            # self.printSystem()

        else:

            print('Variable number does not match with phase space dimension')

    def setVariationalDynamics(self, exprDict):

        if len(exprDict) == self.psDim:

            self.variationalDynamicsDict = exprDict
            # self.printSystem()

        else:

            print('Variable number does not match with phase space dimension')

    def setPhase(self, phase):

        self.phase = phase
        for idx, var in enumerate(self.variables):
            exec("self.%s = %f" % (var, phase[idx]))

    def setVariationalPhase(self, norm=None, phase=None):

        if phase==None: phase = [0.5,0.5,0.5,0.5]
        if norm==None: norm = 1

        curNorm = np.sqrt(np.sum(np.power(phase,2)))
        phase = np.multiply(phase, norm/curNorm)

        self.variationalPhase = phase
        for idx, var in enumerate(self.variables):
            exec("self.v_%s = %f" % (var, phase[idx]))

        self.vpNorm = norm

    def resetPhase(self, phase):

        self.phase = np.zeros(len(self.phase))

    def printPhase(self):

        display(Latex(f'$---phase---$'))

        for idx, var in enumerate(self.variables):

            display(Latex(f'$'+str(var)+'='+str(eval("self.%s" %
            ↪   (var)))+'$'))

        display(Latex(f'$-----------$'))

    def calcDerivatives(self, phase):

        derivatives = np.zeros(self.psDim)
        for idx, var in enumerate(self.variables):
            exec("%s = %f" % (var, phase[idx]))

        for idx, var in enumerate(self.variables):
            exec("dot%s = %s" % (var, self.dynamicsDict[var]))
            derivatives[idx]=eval("dot%s" % (var))

        return derivatives

    def calcVariationalDerivatives(self, phase):

        derivatives = np.zeros(self.psDim)
        for idx, var in enumerate(self.variables):
            exec("%s = %f" % (var, phase[idx]))
            exec("v_%s = %f" % (var, phase[idx]))

        for idx, var in enumerate(self.variables):
            exec("dot_v_%s = %s" % (var,
            ↪   self.variationalDynamicsDict[var]))
            derivatives[idx]=eval("dot_v_%s" % (var))

        return derivatives

    def calcDerivativesHH(self, phase):

        derivatives = [phase[2],
                       phase[3],
                       -phase[0]-2*phase[0]*phase[1],
                       -phase[1]-phase[0]**2+phase[1]**2]
        return np.array(derivatives)

    def calcDerivativesHH_variational(self, phase):

        p, vp = phase[0:4], phase[4:8]

        derivatives1 = [p[2],
                       p[3],
                       -p[0]-2*p[0]*p[1],
                       -p[1]-p[0]**2+p[1]**2]

        derivatives2 = [vp[2],
                       vp[3],
                       -vp[0]-2*p[0]*vp[1]-2*p[1]*vp[0],
                       -vp[1]-2*p[0]*vp[0]-2*p[1]*vp[1]]

        return np.concatenate((derivatives1, derivatives2))

    def setHamiltonian(self, expr):

        exec("self.H = 0")
        self.Hexpr = expr
        display(Latex(f'$H='+sympy.latex(sympy.sympify(expr))+'$'))

    def calcEnergy(self, phase=None):

        if phase == None: phase = self.phase

        for idx, var in enumerate(self.variables):
            exec("%s = %f" % (var, phase[idx]))

        exec("E = %s" % (self.Hexpr))
        rslt=eval("E")
        return float(rslt)

    def setPhaseWithEnergy(self, phase, energy):

        self.strIC = str(phase) + '_' + str(energy)

        varId = phase.index(None)

        def energyCost(a):

            newPhase = phase
            newPhase[varId] = a[0]
```

```python
            return self.calcEnergy(newPhase)-energy

        from scipy.optimize import fsolve
        res = fsolve(energyCost, 1, xtol=1e-5)
        newPhase = phase
        newPhase[varId] = res[0]

        if np.abs(self.calcEnergy(newPhase)-energy)<=1e-5:

            self.setPhase(newPhase)

        else: print('phase not found')

class timeEvolution():

    def __init__(self, system):

        self.system = system

    def RK4Step(self, phase, dt):

        k1 = self.system.calcDerivativesHH(phase)
        k2 = self.system.calcDerivativesHH(phase + 0.5*dt*k1)
        k3 = self.system.calcDerivativesHH(phase + 0.5*dt*k2)
        k4 = self.system.calcDerivativesHH(phase + dt*k3)

        y_nn = phase + (1/6)*(k1+2*k2+2*k3+k4)*dt

        return y_nn

    def RK4StepWithVariational(self, phase, dt):

        k1 = self.system.calcDerivativesHH_variational(phase)
        k2 = self.system.calcDerivativesHH_variational(phase + 0.5*dt*k1)
        k3 = self.system.calcDerivativesHH_variational(phase + 0.5*dt*k2)
        k4 = self.system.calcDerivativesHH_variational(phase + dt*k3)

        y_nn = phase + (1/6)*(k1+2*k2+2*k3+k4)*dt

        return y_nn

    def evolveRK4(self, t, dt, IC = None):

        self.t = t
        self.dt = dt
        strIC = self.system.strIC
        vpNorm = self.system.vpNorm

        self.fileName = 'henonHeilesSim_t:'+'{:.6g}'.format(t)+'_dt:'+'{:.6g}
        '.format(dt)+'_IC:'+strIC

        import time

        start = time.time()

        if IC == None: IC = self.system.phase

        timeAxis = np.linspace(0,t,int((t+dt)/dt))
        phaseAxis = np.zeros((len(timeAxis), len(IC)))
        variationalPhaseAxis = np.zeros((len(timeAxis), len(IC)))

        phaseAxis[0] = IC
        variationalPhaseAxis[0] = self.system.variationalPhase

        for i in range(len(timeAxis)-1):

            temp = np.concatenate((phaseAxis[i], variationalPhaseAxis[i] * (vpNorm / np.linalg.norm(variationalPhaseAxis[i]))))
            newV = variationalPhaseAxis[i] * (vpNorm / np.linalg.norm(variationalPhaseAxis[i]))
            RK4 = self.RK4StepWithVariational(temp, dt)
            phaseAxis[i+1] = RK4[0:4]
            variationalPhaseAxis[i+1] = RK4[4:8]

        self.system.phase = phaseAxis[-1]
        self.system.variationalPhase = variationalPhaseAxis[-1]

        print('time elapsed:', time.time()-start, 's')

        self.timeAxis = timeAxis
        self.phaseAxis = phaseAxis
        self.variationalPhaseAxis = variationalPhaseAxis

    def saveTimeEvolution(self, folder):

        now = datetime.now()
        dt_string = now.strftime("_%d_%m_%Y__%H_%M_%S")
        if not os.path.exists(folder): os.makedirs(folder)

        fileName = './'+folder+'/'+self.fileName+dt_string+'.csv'

        array = np.column_stack((self.timeAxis, self.phaseAxis, self.variationalPhaseAxis))
        column_values = ['time', 'x', 'y', 'p_x', 'p_y', 'v_x', 'v_y', 'v_p_x', 'v_p_y']

        df = pd.DataFrame(data = array, columns = column_values)

        with open(fileName, 'w') as fout:
                df.to_csv(fout)

    def loadTimeEvolution(self, t, dt, IC, energy, folder = '/simOuts'):

        path = os.getcwd()+folder
        if(os.path.exists(path)):

            files = os.listdir(path)

            for fileName in files:

                fn = fileName.split('_')

                t_file = float((fn[1].split(':'))[1])
                dt_file = float((fn[2].split(':'))[1])

                IC_file = (fn[3].split(':'))[1][1:-1]
                IC_file= list(np.array(IC_file.split(', ')))
                for idx, a in enumerate(IC_file):
                    if a == "None": IC_file[idx] = None
                    else: IC_file[idx] = float(a)

                energy_file = float(fn[4])

                if (t == t_file and dt == dt_file and IC == IC_file and energy == energy_file):

                    fileName = '.' + folder + '/' + fileName

                    df = None
                    with open(fileName, 'r') as fout:
                        df =  pd.read_csv(fout)

                        timeAxis = df['time'].to_numpy()
                        x = df['x'].to_numpy()
                        y = df['y'].to_numpy()
                        p_x = df['p_x'].to_numpy()
                        p_y = df['p_y'].to_numpy()

                        v_x = df['v_x'].to_numpy()
                        v_y = df['v_y'].to_numpy()
                        v_p_x = df['v_p_x'].to_numpy()
                        v_p_y = df['v_p_y'].to_numpy()

                        phaseAxis = np.transpose([x,y,p_x,p_y])
                        variationalPhaseAxis = np.transpose([v_x,v_y,v_p_x,v_p_y])

                        self.t = t
                        self.dt = dt
                        self.IC = IC
                        self.energy = energy

                        self.timeAxis = np.array(timeAxis)
                        self.phaseAxis = np.array(phaseAxis)
                        self.variationalPhaseAxis = np.array(variationalPhaseAxis)

                        return

            else: print('file not found')

    def setPlotTime(self, plotTime):

        self.plotTime = plotTime
        self.timeIdx = np.argmin(np.abs(np.subtract(self.timeAxis, plotTime)))+1

    def plot(self, var, colored = False):

        plt.figure()

        varId = self.system.variables.index(var)
        varAxis = self.phaseAxis[:self.timeIdx,varId]
        c = time = self.timeAxis[:self.timeIdx]

        plt.plot(time, varAxis, c='black', alpha=0.5)
        plt.xlabel(r'$'+var+'$')
        plt.ylabel('time (s)')

        if colored:
            plt.scatter(time, varAxis, c=c, s=10)
            plt.colorbar(label='time (s)')

    def plot2D(self, var1, var2, colored = False):

        plt.figure()

        varId1 = self.system.variables.index(var1)
        varId2 = self.system.variables.index(var2)
        varAxis1 = self.phaseAxis[:self.timeIdx,varId1]
        varAxis2 = self.phaseAxis[:self.timeIdx,varId2]

        plt.plot(varAxis1, varAxis2, c='black', alpha=0.5)
        plt.xlabel(r'$'+var1+'$')
        plt.ylabel(r'$'+var2+'$')

        if colored:
            c = self.timeAxis[:self.timeIdx]
            plt.scatter(varAxis1, varAxis2, c=c, s=10)
            plt.colorbar(label='time (s)')

    def plot3D(self, var1, var2, var3, colored = False):

        fig = plt.figure(figsize = (10, 7))
        ax = plt.axes(projection ="3d")

        varId1 = self.system.variables.index(var1)
        varId2 = self.system.variables.index(var2)
        varId3 = self.system.variables.index(var3)
        varAxis1 = self.phaseAxis[:self.timeIdx,varId1]
        varAxis2 = self.phaseAxis[:self.timeIdx,varId2]
        varAxis3 = self.phaseAxis[:self.timeIdx,varId3]

        ax.plot3D(varAxis1, varAxis2, varAxis3, c='black', alpha=0.5)

        ax.set_xlabel(r'$'+var1+'$')
        ax.set_ylabel(r'$'+var2+'$')
        ax.set_zlabel(r'$'+var3+'$')

        if colored:

            c = self.timeAxis[:self.timeIdx]
            p=ax.scatter3D(varAxis1, varAxis2, varAxis3, c=c, s=10)
            fig.colorbar(p, label='time (s)')

    def poincareSurface(self, var1, var2, poincareVar, clr='black'):

        varId1 = self.system.variables.index(var1)
        varId2 = self.system.variables.index(var2)
        varId3 = self.system.variables.index(poincareVar)
        varAxis1 = self.phaseAxis[:self.timeIdx,varId1]
        varAxis2 = self.phaseAxis[:self.timeIdx,varId2]
        varAxis3 = self.phaseAxis[:self.timeIdx,varId3]

        a = varAxis3
        mult_a = np.sign(a[1:] * a[:-1])
        diff_a = np.sign(a[1:] - a[:-1])
        isPiercing = ((mult_a==-1)+(mult_a==0))*(diff_a==1)
        ratios = -a[:-1]/diff_a

        b = varAxis1
        new_b = np.add((b[1:] - b[:-1])*ratios, b[:-1])
        b = varAxis2
        new_c = np.add((b[1:] - b[:-1])*ratios, b[:-1])

        varAxis1 = []
        varAxis2 = []
        for idx, a in enumerate(isPiercing):

            if a:

                varAxis1.append(new_b[idx])
                varAxis2.append(new_c[idx])

        plt.plot(varAxis1, varAxis2, '.', alpha=0.5, markersize=2,
        ↪   color=clr)
        plt.xlabel(r'$'+var1+'$')
        plt.ylabel(r'$'+var2+'$')

##################

henonHeiles = ps(2, 2)
henonHeiles.setVariables('x y p_x p_y')
henonHeiles.setDynamics({'x': 'p_x',
                     'y': 'p_y',
                     'p_x': '-x-2*x*y',
                     'p_y': '-y-x**2+y**2'})
henonHeiles.setHamiltonian("(1/2)*(p_x**2+p_y**2)+(1/2)*(x**2+y**2)
+x**2*y-(1/3)*y**3")

energy = 0.125
t, dt = 1e4, 1e-3

sim1 = timeEvolution(henonHeiles)
sim1.loadTimeEvolution(t, dt, [0,0.1,None,0], energy)

sim2 = timeEvolution(henonHeiles)
sim2.loadTimeEvolution(t, dt, [0,0.2,None,0], energy)

sim3 = timeEvolution(henonHeiles)
sim3.loadTimeEvolution(t, dt, [0,0.3,None,0], energy)

sim4 = timeEvolution(henonHeiles)
sim4.loadTimeEvolution(t, dt, [0,-0.15,None,0], energy)

sim5 = timeEvolution(henonHeiles)
sim5.loadTimeEvolution(t, dt, [0,-0.25,None,0], energy)

```

```python
sim6 = timeEvolution(henonHeiles)
sim6.loadTimeEvolution(t, dt, [0,-0.35,None,0], energy)

```