The Dynamics of the Chaotic Pendulum

Anjali Madangarli, Nitin Jha, Spandan Pandya

May 14, 2022

Abstract

This paper studies the dynamics of the damped driven pendulum to conclude that it demonstrates chaotic behaviour for certain values of the driving force amplitude. The system takes the period doubling route to chaos with occasional windows of stability indicated by the bifurcation diagram.The behaviour of this system were studied by the use of phase portrait diagrams that indicated global mixing and exponential sensitivity to initial conditions. This feature of chaos was further corroborated by the positive value of the Lyapunov exponent for certain values of the driving frequencies that point to diverging phase portrait trajectories. The Poincaré sections that take the snapshot of the system at the interval of the driving frequency show the existence of tongue like structures that self replicate: indicating the possibility of the system being a strange attractor that is chaotic.

Keywords:Chaos, Pendulum, Non-Linear Dynamics, Period doubling.

1 Introduction

Consider the case of the simple pendulum that consists of a bob, an inextensible string connected to a mechanical pivot. The mechanical pivot rotates providing the system a sinusoidal driving force $F_0 \cos \omega_d t$.

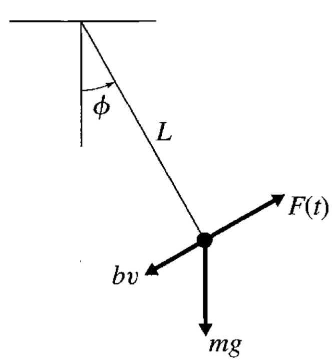


Figure 1: The freebody diagram for the damped driven pendulum

Comparing the torques and moment of inertia $I\ddot{\theta} = \Gamma$, where I is the moment of inertia and Γ is the net torque about the pivot yields the equation of motion for this case. In this case $I = mL^2$, since it is a point mass. The torque arises from the three forces shown in the figure above. The resistive force has magnitude bv and hence exerts a torque $-Lbv = -bL^2\dot{\theta}$. The torque of the weight is $-mgL\sin\theta$, and that of the driving force is $LF_0\cos\omega_d t$. This yields the equation of motion to take the form:

$$mL^2\ddot{\theta} = -bL^2\dot{\theta} - mgL\sin\theta + LF_0\cos\omega_d t \tag{1}$$

Rearranging this equation by dividing everything by mL^2 gives:

$$\ddot{\theta} + \frac{b}{m}\dot{\theta} + \frac{g}{L}\sin\theta = \frac{F_{\text{o}}}{mL}\cos\omega_d t \tag{2}$$

The above system is nonlinear as a result of the $\sin\theta$ term and may or may not show characteristics of chaos, something this paper will discuss over its course. Chaos in the dynamical sense of the word is characterized by three primary features, each of which is to be discussed in the course of this paper: non-linearity, exponential sensitivity to initial conditions and global mixing in its phase space. Consider the equation for a damped and driven simple harmonic oscillator with a small angle approximate to linearise the system.

$$\ddot{\theta} + \frac{b}{m}\dot{\theta} + \frac{g}{L}\theta = \frac{F_{\text{o}}}{mL}\cos\omega_d t \tag{3}$$

In the equation above, the small angle approximation of $\sin\theta \approx \theta$ allows for analytic solutions for the differential equation. However, for the case that such an approximation is avoided, an analytical solution becomes quite difficult to obtain. In the equation above two terms allow for the possibility of chaos: the periodic driving frequency term ωt is a non-autonomous term, allowing for an extra degree of freedom in the system. However, a problem with this system still is that it is linear. Linear systems with finite degrees of freedom do not show chaotic behaviour, and hence in order to explore the possibility of the chaos, the equation must be used in its non linear form:

$$\ddot{\theta} + \frac{b}{m}\dot{\theta} + \frac{g}{L}\sin\theta = \frac{F_{\text{o}}}{mL}\cos\omega_d t \tag{4}$$

This equation may have some potential to exhibit chaotic behaviour since it is no longer a two dimensional linear system but a three dimensional continuous non-linear system. The differential equation above consists of three parameters that can be varied. These parameters consist of the driving frequency ω_d, the damping coefficient $\frac{b}{m}$ which this paper will take to be 2β, and the driving force amplitude: $\frac{F_{\text{o}}}{mL}$. The term $\frac{g}{l}$ is the square of the natural frequency of the pendulum which can be denoted by ω_0^2. Now if we divide the forcing term by the square of the natural frequency we obtain:

$$\frac{F_{\text{o}}}{mL\omega_0^2} = \frac{F_0}{mg} \tag{5}$$

This quantity is nothing but the ratio of the driving force amplitude to the weight of the body. This parameter indicates the strength of the driving force and can be denoted by γ. Hence the equation of motion takes the form:

$$\boxed{\ddot{\theta} + 2\beta\dot{\theta} + \omega_0^2 \sin\theta = \gamma\omega_0^2 \cos\omega_d t} \tag{6}$$

This parameter γ is dimensionless and it measures the strength of the driving force. When $\gamma < 1$, it indicates that the driving force is lesser than the weight of the body, and hence, we would never expect any form of toppling over. The motion will have small amplitudes. However for values of $\gamma > 1$, the driving force is much greater than the pendulum's weight and hence it may even include the case when $\theta = \pi$ or the case where the pendulum is pushed to the top. The equation above will be utilised across the paper in order to analyse chaotic behaviour.

1.1 The Poincaré Bendixson theorem

The Poincaré - Bendixon theorem predicts the long-term behaviour of continuous dynamical systems. It is used to determine whether the bounded solution of a differential equation converges to either a stable limit cycle or a fixed point. However, most importantly it predicts whether the system can exhibit "chaotic" behaviour or not. The Poincaré – Bendixson theorem can be delineated as follows:

1. **Chaos and Dimensionality**: The Poincaré – Bendixson theorem states that chaos cannot be observed in a two-dimensional continuous system. This is because in a two-dimensional plane, trajectories have the property to converge towards a fixed point or attain stable limit cycles. Hence, local divergence, which is a condition for chaos, is not possible. However, in the case of three dimensions, trajectories can move around the phase space endlessly, allowing for possibility of randomness and, thereby, of chaos. Hence, chaotic behaviour is only observed in continuous dynamical systems of dimensions three of more. In the case of simple pendulum, the system did not show chaos when there was no damping or forcing term. This is because in those cases it remained a linear deterministic system. However, chaotic behaviour arises as soon as the forcing term is added, which was sinusoidal and had an explicit dependency on time, making the differential equation non-linear. Mathematically, the differential equation for a damped, forced pendulum can be expressed as follows:

$$\ddot{\theta} + 2\beta\dot{\theta} + \omega_0^2 \sin\theta = \gamma\omega_0^2 \cos\omega_d t \tag{7}$$

Now, if one were to assume $\dot{\theta} = y$, then the above system can be re-described as follows,

$$y = \dot{\theta} \tag{8}$$

$$\dot{y} = -2\beta y - \omega_0^2 \sin\theta + \gamma\omega_0^2 \cos\omega_d t \tag{9}$$

These equations seem to describe a two-dimensional phase plane. However, in accordance to the Poincaré – Bendixson theorem, chaos is not allowed in the two dimensional phase plane. However, this two-dimensional non-autonomous system can be converted into a three-dimensional autonomous system by substituting $z = \omega_d t$, whose time derivative gives $\dot{z} = \omega_d$. Using these substitutions in the previous system of differentials to get the following:

$$y = \dot{\theta} \tag{10}$$

$$\dot{y} = -2\beta y - \omega_0^2 \sin\theta + \gamma\omega_0^2 \cos z \tag{11}$$

$$\dot{z} = \omega_d \tag{12}$$

Hence, Poincaré – Bendixson theorem is no longer applicable here as it is now a three dimentional system. Therefore, it is safe to say that chaos may or may not be exhibited by the system.

2. **Global Mixing**: Another important characteristic of a system capable of showing chaos is global mixing. Due to the uniqueness theorem, it is known that trajectories cannot intersect with each other in the phase space. However, in a three dimensional chaotic system, the trajectories are free to move around in the phase space endlessly in a random fashion. Since, they cannot intersect with each other, they eventually end up covering the entirety of the phase space and this phenomenon is known as global mixing. This also means that the evolution of the system over time takes place in such a way that any region of its phase space eventually overlaps with any other given region.

3. **Exponential sensitivity to initial conditions**: The important characteristic of a system capable of displaying chaotic behaviour is its exponential sensitivity towards initial conditions. This means that even an arbitrarily small change in the beginning can lead to a significantly different trajectory in the future, as these slight changes are persistently magnified over time. This could also mean that every point in a chaotic system is closely approximated by other points that have significantly different trajectories in the future. This sensitivity to initial conditions is measured using Lyapunov exponents.

2 Methodology:

This section will go over the explanation for the integration method used for our model, i.e., RK-4. Further, the algorithms used for producing some of required graphs, such as phase portrait for damped-driven, damped, and undamped pendulum, bifurcation plots, Poincaré sections, and Lyapunov exponents will also be discussed.

2.1 Numerical Integration method

When using numerical methods to integrate an ODE, two main errors are encountered. The main aim is to use to method which minimizes these errors, and thus increases the overall accuracy of the final results.

2.1.1 Numerical Errors

The major sources of error in the final result, while integrating an ODE, is the *round-off* and *truncation* errors. **Round-off error** refers to the lack of precision in the final result as a limitation due to the programming treating some values to their nearest rounded-off values. For example, if in our code, we want to compare if something is equal to zero, the computer might treat values very close to zero as zero; thus leading to some errors in the final output.

2.1.2 Numerical Instability

Numerical instability refers to the overall error induced due to the value of step-length. Euler method is very prone to numerical instability, as with the increase in step-length the truncation error increases and dominates over the round-off errors, and thus producing not so accurate results.

2.1.3 Fourth Order Runge-Kutta Method

In our project, the numerical method used to find the solution of the second order differential equation, i.e., the required equation of motion, was *Fourth order Runge-Kutta Method* (or **RK-4**). RK-4 was used instead of the Euler method since it has lesser truncation errors, and it is much more numerically stable. In Euler method the integration is only done at the beginning of the step, which is very asymmetric in nature. This error can be reduced by creating some sort of symmetry in the process of integrating. (Fitzpatrick) The algorithm for applying the fourth order Runge-Kutta integrator is as follows,

$$y_{i+1} = y_i + \frac{1}{6}k_1 + 2k_2 + 2k_3 + k_4 \tag{13}$$

In the above equation, the values of k are calculated at the mid-point of each step. This is done as follows,

$$k_1 = h\mathrm{f}(t_i, x_i) \tag{14}$$

$$k_2 = h\mathrm{f}(t_i + \frac{h}{2}, x_i + \frac{k_1}{2}) \tag{15}$$

$$k_3 = h\mathrm{f}(t_i\frac{h}{2}, x_i + \frac{k_2}{2}) \tag{16}$$

$$k_4 = h\mathrm{f}(t_i + h, x_i + k_3) \tag{17}$$

In the above equations, h is the step-length defined by the user. The error for an n^{th} order RK-4 method can be approximated as,

$$\epsilon = \frac{n}{h} + h^n$$

In the above equation, the first term represents the round-off error, and second term is the truncation error. The minimum step length and the associated error can be approximated as follows,

$$h_0 \sim \eta^{\frac{1}{(n+1)}}$$

$$\epsilon_0 \sim \eta^{\frac{n}{(n+1)}}$$

From the above method, the errors in the intermediate steps gets cancelled out and the error that remains is of the fourth order of the original error. For example, if the truncation error was 0.1 in the Euler method, then by applying RK-4 method, the error would be reduced to 0.0001. (Fitzpatrick)

2.2 The Lyapunov exponent

Mathematically, Lyapunov exponent gives the rate at which trajectories of two infinitesimally different dynamical system diverge over time. For two systems in phase space with initial separation vector, $\delta\theta_{\mathbf{0}}$, then the rate at which the trajectories diverge can be written as,

$$\delta\theta \approx e^{\lambda t}\delta\theta_{\mathbf{0}} \tag{18}$$

In the above equation, λ is known as the Lyapunov exponent. We can see that this factor is associated with an exponent, and hence, it shows the sensitivity to initial condition that's often being talked about with chaotic systems. It's described that λ can be positive, negative, or zero. The different cases described says the following about our system,

- $\lambda < 0$ or $\lambda = 0$: system is stable, i.e., attains a stable limit cycle.
- $\lambda > 0$: system shows chaos.

Taking ln() on both sides of the Lyapunov exponent yields the following expression:

$$\ln\left(\frac{\delta\theta}{\delta\theta_{\mathbf{0}}}\right) = \lambda t \tag{19}$$

For our project we explored the graphs plotted using the above relation to determine if a system exhibits chaos.

Algorithm Used:

- Two systems with similar initial conditions (i.e., $\theta_1 = 0.0$ and $\theta_2 = 0.001$) were taken to study their nature over time.
- Using the *solve()* function (refer to appendix), we calculated the values of theta for different driving amplitude, each of which belonged in different regimes of stability.
- Now, two values of θ arrays for each initital condition, and each value of γ was calculated. For each gamma, the difference between the θ_1 and θ_2 was found, and the absolute values was taken to ensure only positive values were stored.
- Using eq(19), ln() was taken on the above found value, and each of them were plotted against the time array.

2.3 Bifurcation Diagram

Mathematically, bifurcation diagrams represent the value approached asymptotically (i.e., *fixed points*, *periodic orbits*, or *chaotic attractors*) of a system as a function of a varying parameter. For our case, the value approached a chaotic attractor, and the parameter that was varied was the driving amplitude. The algorithm used for obtaining the bifurcation plots is as follows,

Algorithm Used

- An array was created for a range of values of the driving amplitude (γ).
- A loop was made, so each of the values of driving amplitude was used to find their corresponding θ, and ω values.
- *solve()* function was used to find the value of θ and ω for the system, for one of value of γ, and the loop was restarted with different value of γ
- Only the last 250 (time period) values for each θ and ω were considered for the plot to account for the steady state behavior of the system. These values were plotted against the forcing amplitude (γ) to find the required bifurcation plot.
- For the next plot, the limits of the graph were change to focus only in one of the areas where formation of fractal structures are clear.

2.4 Poincaré Sections

A Poincaré Section or map gives us an idea about the periodicity of a quantity of dynamic system. By constructing this map, one can see if the system is well behaved or have some chaotic nature. To determine a Poincaré Section, one needs to choose a quantity of interest (our case it's θ), and then one needs to sample the rate of change of this quantity periodically, and thus can plot this to get a Poincaré Section. The following diagram illustrates the same,

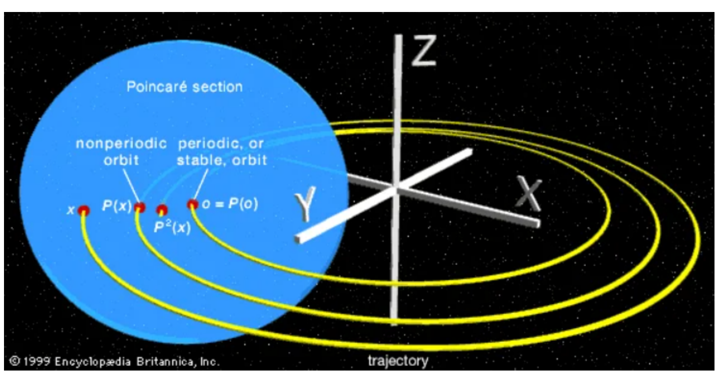


Figure 2: A visual description of a Poincaré Section (Britannica) The intersection of the object of interest with the blue disks will give points in Poincaré space.

For plotting Poincaré space, we choose our points according to the following relation, (Fitzpatrick)

$$\omega t = \phi + k2\pi \tag{20}$$

In the eq(20), ω is the angular velocity, θ is the angular displacement, ϕ is the *Poincaré Phase*, and k is some integer.

Algorithm Used

- An array was defined with the range of driving amplitudes (γ) that was implemented for our model.
- For each value of the above defined driving amplitude, the corresponding value of θ and ω were calculated using the *solve() function*.
- A nested loop was made, and using the definition of Poincaré space, particular value of θ and ω were sampled out. The sampling out was done for every 10^{th} element. Every 10^{th} element was chosen because our time-period is one and step-length is 0.1.
- The sampled values were stored in two different arrays, one containing sampled θ and other contained sampled ω values.
- The θ values had a domain of $(-\infty, \infty)$. These were converted to a restricted domain of $[-\pi, \pi]$. These sampled values were the required Poincaré theta values.
- These two arrays were plotted against each other, and thus a sampled out phase portrait was found, which defined *Poincaré section* for our required system.

3 Results

3.1 Constant Parameters

The following parameters were used across the paper in order to construct diagrams unless the value for a particular parameter is specified otherwise.

Parameter	Value
Natural Frequency (ω_0)	3π
Damping Factor (β)	$\frac{3\pi}{8}$
Driving Frequency (ω_d)	2π
Steady-State Time Period (τ)	1
Initial Time (T_0) (RK-4)	0
Final Time (T_f) (RK-4)	10^5
Step-Length (h)	0.1
Initial Angular Displacement (θ_0)	0
Initial Angular Velocity (ω)	0

Table 1: Constant Parameters used in the model

3.2 Non-Chaotic case

3.2.1 No Forcing or Damping

To begin with, the RK-4 integrator was used to solve the differential equation governing the motion of a simple pendulum, i.e., without the forcing or driving term. Hence, the following differential equation was solved:

$$\ddot{\theta} + {\omega_o}^2 sin\theta = 0$$

Here, ${\omega_o}^2 = \frac{g}{l}$ The phase portrait and position vs. time plot for the same was obtained as follows:

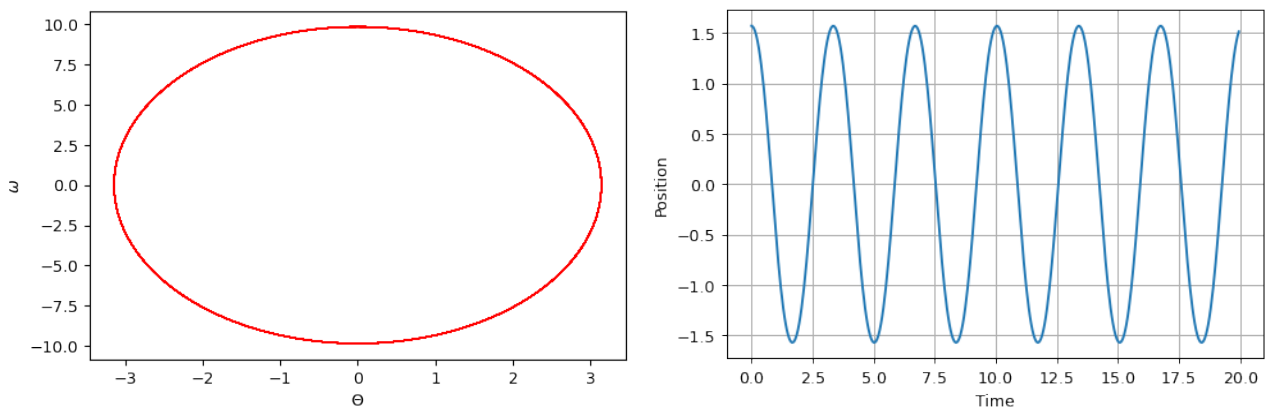


Figure 3: Phase portrait for a simple pendulum and its displacement plot

Hence, it is observed that, after an angular displacement of 2π radians, the phase portrait wraps unto itself, thereby forming the elliptical shape. Whereas, the position vs. time plot described its oscillatory nature with a fixed amplitude. Again, the amplitude of oscillation is fixed due to the absence of any forcing or damping terms.

3.2.2 Energy Plot for a Simple Pendulum

In the case of a simple pendulum, when there is no damping or driving term, there is no dissipation of energy in the system. Hence, the total mechanical energy remains constant. It is known that mechanical energy is the sum total of kinetic and potential energy of the system. The kinetic energy K of the system is given as follows,

$$K = \frac{1}{2} ml^2\omega^2 \tag{21}$$

Here, m is the mass of the bob, l is the length of the string, and ω is the angular velocity of the bob. Whereas, potential energy U of the system at any angular displacement θ is given as follows,

$$U = mgl(1 - cos\theta) \tag{22}$$

Hence, the total energy E of the system, which is the sum of the kinetic and potential energy and is a constant quantity. That is, it does not vary with time. Hence, a total energy vs.

time plot will be a straight line parallel to the X-axis. This can be seen in the following plot:

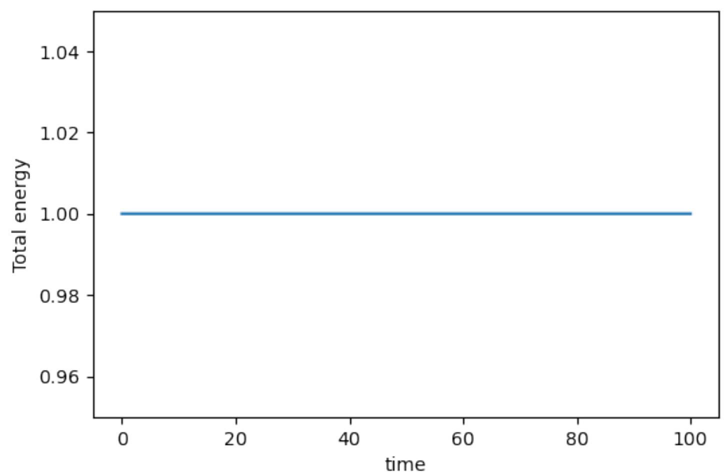


Figure 4: Energy vs. time plot for simple pendulum

This plot was originally created to check the accuracy of the RK-4 Integration method. As one can observe in the graph, the plot is a straight line parallel to the X-axis, at a constant Y-value, i.e, at a constant value of total energy. This is in accordance with what was expected theoretically.

3.2.3 Damped Pendulum

When damping is present, there is dissipation of energy throughout the system. In this case, the equation is re-written as follows,

$$\ddot{\theta} + 2\beta\dot{\theta} + {\omega_o}^2 sin\theta = 0 \tag{23}$$

Here, β is the damping constant. The phase portrait and position vs. time plot for a damped system is obtained as follows:

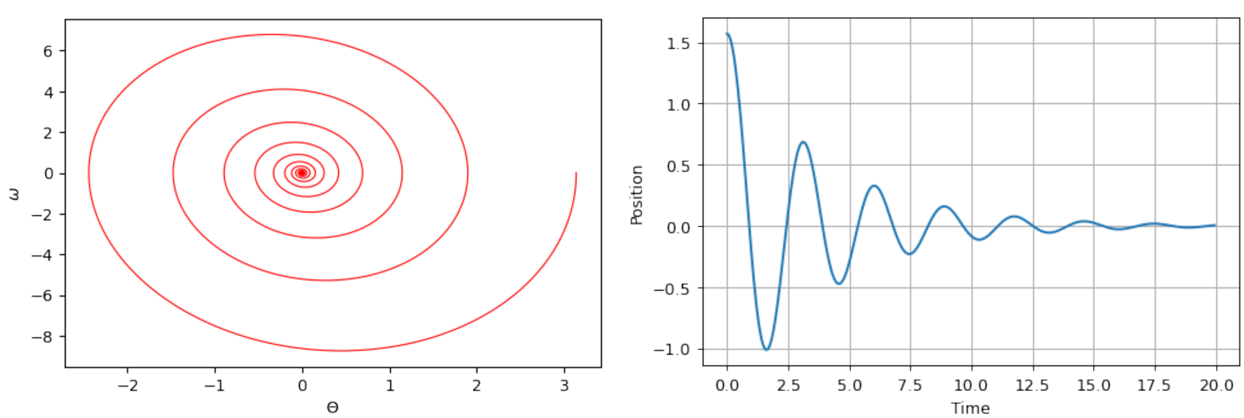


Figure 5: Phase portrait for a damped pendulum and its displacement chart

Therefore, from the position vs. time plot, one can observe that the amplitude of oscillations reduces over time till all the energy in the system has been dissipated and eventually the motion of the pendulum ceases. The period of time that the pendulum takes to come to rest is dependent on the value of the damping constant β. Similarly, from the phase portrait of the damped pendulum, we can observe that the pendulum swings with a decreasing amplitude till it stops at the fixed point $\theta = 0$ and $\omega = 0$. This translates to an inward-closing spiral, as seen in Figure 3. This corresponds to a stable fixed point at the origin.

3.2.4 Damped, Driven Pendulum

In this case, a forcing term has been introduced along with the damping term. Now, the differential equation governing the motion of the damped driven pendulum can be expressed as follows:

$$\ddot{\theta} + 2\beta\dot{\theta} + \omega_0^2 \sin\theta = \gamma\omega_0^2 \cos\omega_d t \tag{24}$$

Here, β is the damping constant, γ is the strength of the driving force, and ω_d is the driving frequency. The phase portrait and position vs. time graph for the same was obtained as follows:

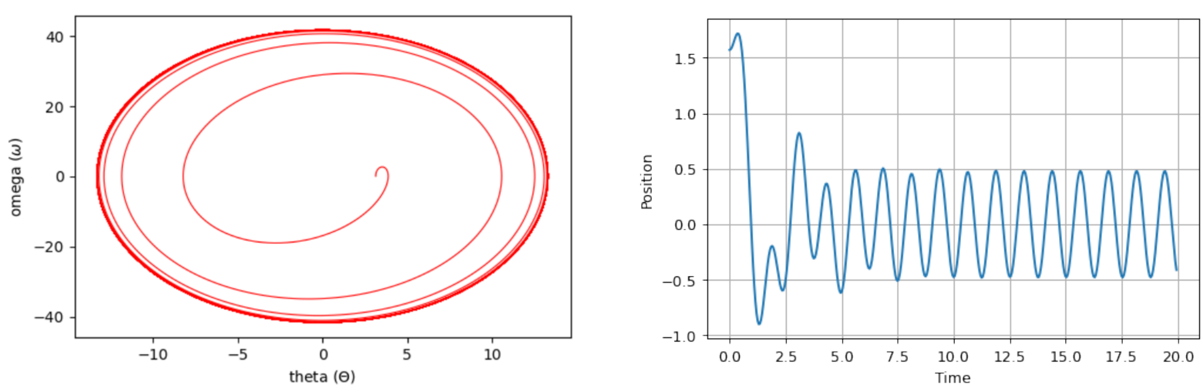


Figure 6: Phase portrait for a damped driven pendulum and its position chart

From the phase portrait it can be observed that due to the presence of the forcing term, the phase trajectory eventually attains a stable limit cycle. They are usually a characteristic of non-linear systems. Similarly, from the position vs. time plot, one can observe that the transients die out, they give rise to steady state behavior. Once the system has attained steady state, it has no memory of the initial conditions, and it starts oscillates with a fixed amplitude as seen in Figure(6). As discussed earlier, a damped driven pendulum system is capable of showcasing chaotic behavior due to the presence of the non-linear non-autonomous forcing term. However, it is important to note that it showcases chaotic behaviour only for certain regimes. These regimes can be determined by varying the values of the amplitude of the forcing term γ.

3.3 Phase Portraits and exponential Sensitivity

One feature of knowing that a system may be chaotic for a particular value of a parameter is to check for its exponential sensitivity to initial conditions. Consider the case for the pendulum when its driving frequency amplitude is $\gamma = 0.1$. For one case let the system have no angular displacement or in other words $\theta_0 = 0$ while the other system have an initial angular displacement of just $\theta_0 = 0.001$. For this we obtain the following result,

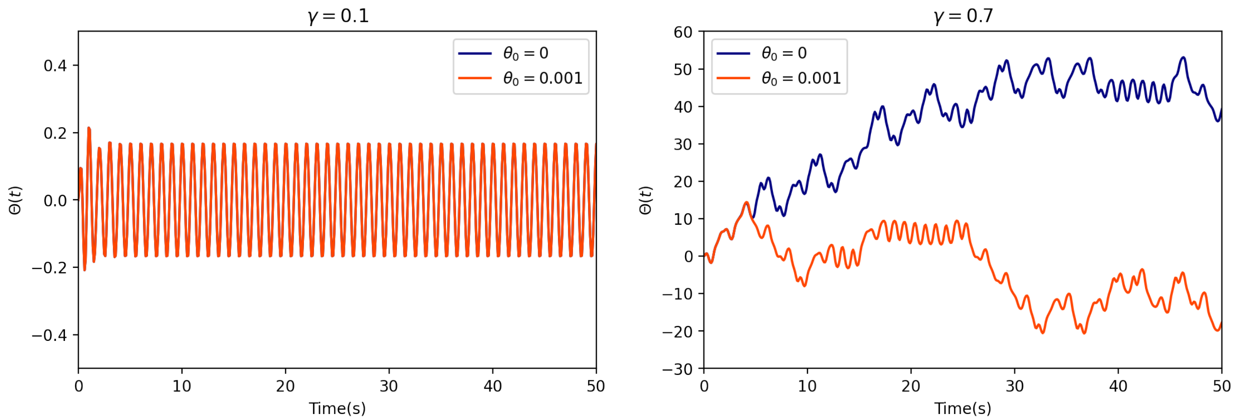


Figure 7: Non chaotic behaviour for $\gamma = 0.1$ and chaotic behaviour for $\gamma = 0.7$

The trajectories for both systems with a marginal change in their angular displacement lead to no significant change in their position time charts. They are both essentially superimposed on one another. This indicates that when the driving amplitude is $\gamma = 0.1$ the system does not show chaos. However, when the driving amplitude is changed to $\gamma = 0.7$ the two pendula seem to take two completely different paths. This indicates that the system is very sensitive to initial conditions. Even though the pendula show similar behaviour from 0 to 10 seconds, the long term behaviour of these two systems are completely different. This difference in the two pendula hint at the possibility of chaos in the system. The divergence of these trajectories can be further analysed by observing their phase portraits.

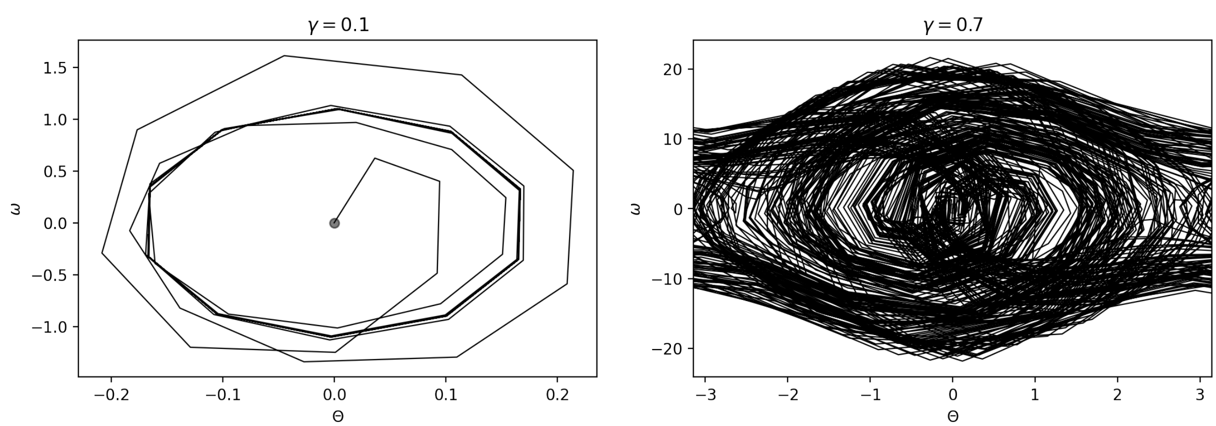


Figure 8: Stability for $\gamma = 0.1$ and Global mixing for $\gamma = 0.7$

From Figure(8) above, when $\gamma = 0.1$, the system reaches a stable limit cycle, and its motion does not cover the entire phase plane. However, the motion for the chaotic case covers a larger area on the phase plane for the same ammount of time. The phase portrait for the $\gamma = 0.7$ case alludes to global mixing- the covering of the entire area in the phase space as time tends to infinity. When a system shows global mixing on the phase plane, it means that the system takes all values for angular displacements and angular velocities, what this means is that the motion never repeats itself over time, signifying chaos.

3.4 The Lyapunov exponent

A dynamical system can have two type of behavior over long term, stable or chaotic. To determine if a dynamical system exhibits chaos, one can study the nature of the Lyapunov exponent associated with that particular system. Since the Lyapunov exponent is related to the exponential factor ($e^{\lambda t}$), when we look at the trajectories of two dynamical systems, with almost identical initial conditions, the sign of the exponent determines whether the trajectories will converge, or diverge at later time periods. For the positive Lyapunov exponent, the factor $e^{\lambda t} \to \infty$, as $t \to \infty$, hence, the trajectories will diverge, indicating that the system had sensitivity to initial conditions. Now, if the Lyapunov constant is negative, then $e^{\lambda t} \to 0$, as $t \to \infty$, hence the trajectories converge.Thus, if the Lyapunov exponent is zero, or negative, the difference in trajectories converges, hence the system does not exhibit chaos. It can also generalised that a negative Lyapunov exponent characterises a periodic attractor, and a positive exponent is a characteristic of a chaotic attractor.

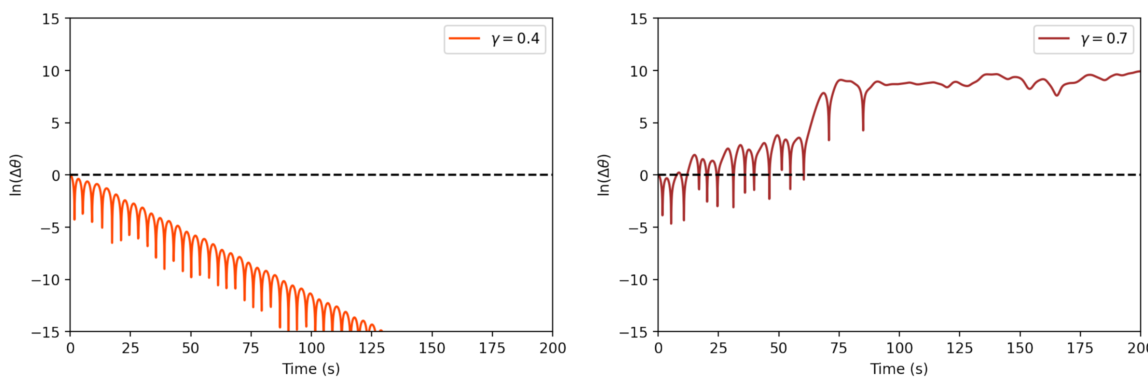


Figure 9: Plot showing deviation of trajectories with respect to time, to study the nature of Lyapunov Exponent. ($\ln \Delta\theta$ Vs Time(s))

The plot for $\gamma = 0.4$ shows that the quanitity $\ln \Delta\theta$ is always less than zero, indicating a periodic attractor, i.e., no chaos. Whereas, the plot for $\gamma = 0.7$ shows that the quanitity $\ln \Delta\theta$ is gradually grows positive, indicating the presence of a chaotic attractor, and thus indicates that the system goes chaotic for the particular driving frequency.

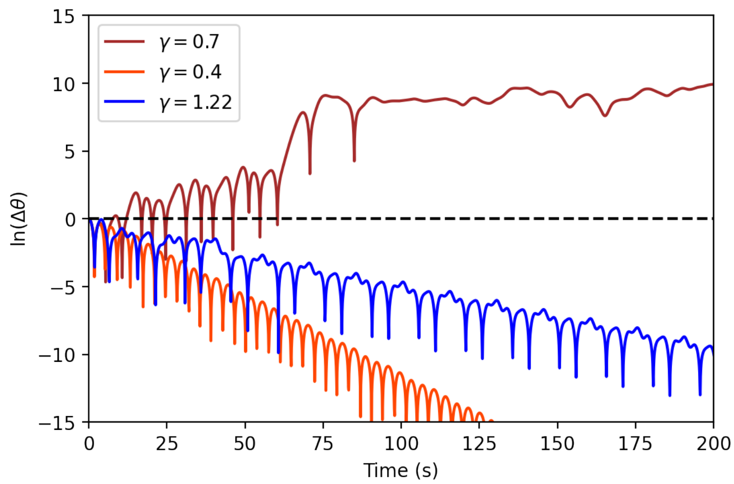


Figure 10: Plot showing deviation of trajectories with respect to time, to study the nature of Lyapunov Exponent for the driving frequency in different regimes, i.e., chaotic and stable regime.

The above figure shows the graph between $\ln \Delta\theta$ and Time for three different driving amplitude, γ. It can be seen that the graphs for $\gamma = 0.4$ and $\gamma = 1.22$ are always below $x = 0$, showing stability over all time. Whereas, for $\gamma = 0.7$, the value on y-axis grows positive after some time, and keeps on growing more positive, thus indicating chaotic nature for our dynamical system. Using eq(18), the value of Lyapunov exponent, λ, were calculated for two different values of γ, one in stable regime and other in chaotic regime. The development of λ over time for the two different driving frequencies can be seen from the plots below,

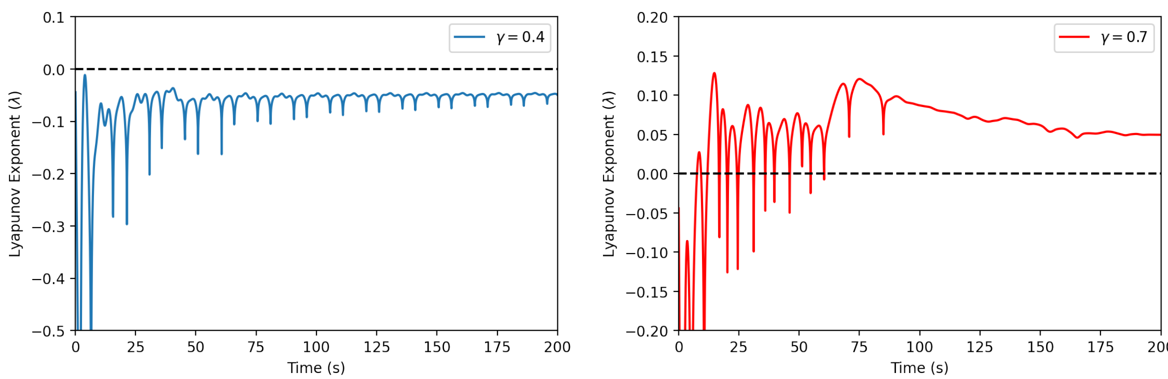


Figure 11: Plot showing the change in Lyapunov exponent with time for two different driving amplitude, $\gamma = 0.4$ and $\gamma = 0.7$

From the above two graphs, it can be seen that for the system having $\gamma = 0.4$, the Lyapunov exponent always remains negative, thus showing stability in the system. Whereas, for $\gamma = 0.7$, it can be seen that the value of Lyapunov exponent gradually become positive, thus indicating the presence of chaos in the system. It can also be seen that for first graph, the value of Lyapunov exponent stabilises which can indicate the presence of steady state for the system.

3.5 Bifurcation Diagram

A system can take various approaches to show chaotic behaviour. One such approach is known as the period doubling cascade. Consider a non chaotic system that is driven at a driving frequency ω_d. After some $t_0 \to \infty$ time has passed, the system reaches some steady state. At this time that we define as steady state, the entire system oscillates at the driving frequency, possessing no memory of its initial conditions. Therefore, the time period of oscillation is given by:

$$\tau = \frac{2\pi}{\omega_d} \tag{25}$$

Now let us consider taking snapshots of the system at this interval τ. For a non chaotic system in steady state with a period τ, we would expect the following:

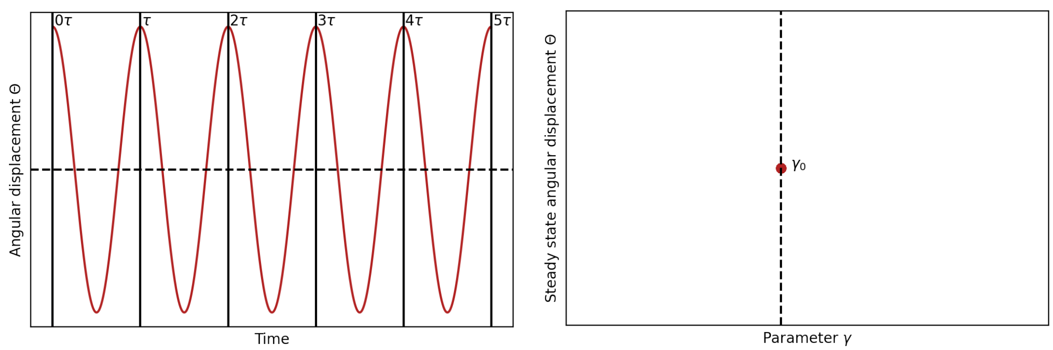


Figure 12: Looking at the long term behaviour of a system that is non chaotic

For a steady state system without chaos, each snapshot at the time period interval corresponds to the same value. This means that the system repeats its motion after each period. These values that correspond to each time period are known as **Poincaré** values. Hence if we were to plot the values of angular displacement for a few hundred steady state values against the driving frequency amplitude, they would all correspond to a single point on the diagram as seen from the plot on the right. Now let's say the driving force amplitude were to change to some parameter γ_1 where the steady state of the system looks something like this the following diagrams,

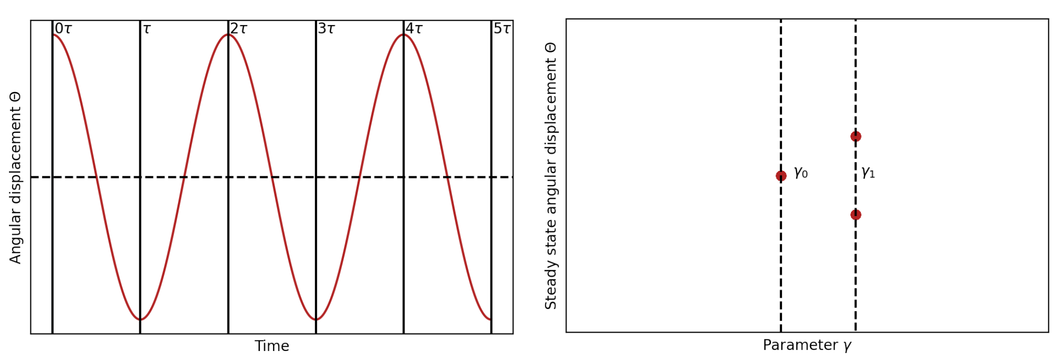


Figure 13: Long term behaviour for γ_1

In this diagram, rather than the system being solely at one value, the values for the angular displacement vary between two values. The angular displacement is the same for $0\tau, 2\tau, 4\tau$. The system also has another value for angular displacement at $\tau, 3\tau, 5\tau$. This means that if we were to plot the steady state values of angular displacement against the driving parameter, we would expect there to be two values. This is seen from the right side diagram (13). Such a situation is known as period doubling. This indicates that the system can be found in any of the two states in the long run. Such a system is still predictable. If we were to apply this algorithm for a range of driving frequency amplitudes and record the steady state values of the function at each time interval τ determined by the driving frequency, we obtain the following bifurcation diagram:

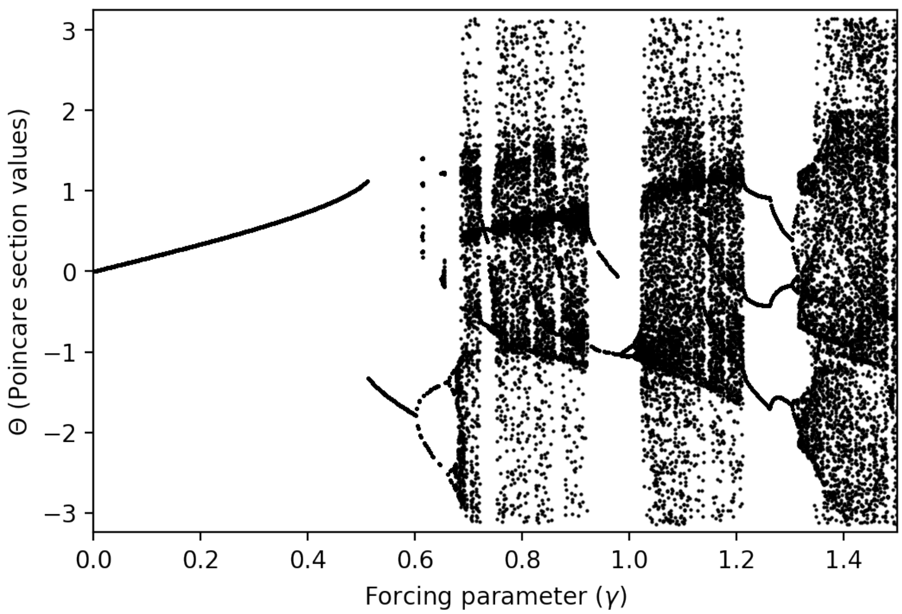


Figure 14: The bifurcation diagram for the chaotic pendulum was created using the following initial conditions: $\omega_d = 2\pi, \tau = \frac{2\pi}{\omega_d} = 1, \omega_0 = 3\pi, \beta = \frac{\omega_0}{8}$

The bifurcation has two types of distinct regions. There are regions that resemble vertical lines. These are regions where there are points that are almost continuously scattered vertically, for instance, the region $0.4 < \gamma < 0.71$. In contrast to this, there are regions where points are much more ordered and discrete on the y-axis.For instance the region from $0 < \gamma < 0.5$ has horizontal line. A similar condition can also be seen for the region $1.2 < \gamma < 1.3$ where there are three lines stacked on top of one another, signifying period trippling. The spread states signify chaos while the discrete packets of points mark the idea of order. This can be seen better by focusing on a certain region in the bifurcation diagram,

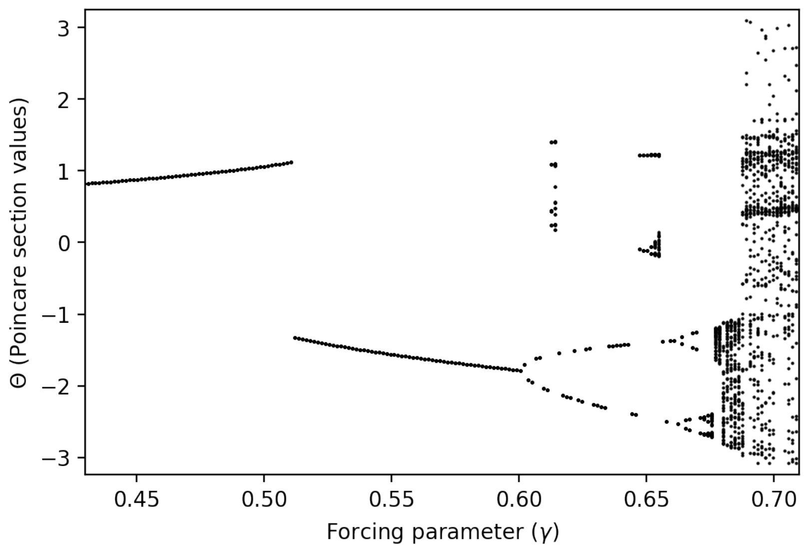


Figure 15: Focusing on the region $0.0 < \gamma < 0.7$ to see bifurcation

Initially, until the system reaches somewhere close to $\gamma = 0.6$ each forcing parameter corresponds to one angle on the y axis. This indicates that the motion repeats every period, as the value for all snapshots at τ time period give the same value. Hence, the system is non chaotic for this regime. At somewhere near $\gamma = 0.6$ the system splits into two segments, indicating a period doubling, where the long term behaviour of the system can be in one of two states. However as we move further ahead, the two branches split into four. Post $\gamma = 0.65$ the system can be found in either of the states indicated by the vertical spread. This indicates that for certain forcing parameter values, the system **DOES NOT** repeat its motion. i.e. the system shows chaotic behaviour. Since the chaos in the system occurs as a result of period doubling, the chaotic pendulum can be classified to be an example of the period doubling cascade. Going back to the bifurcation diagram presented in Figure(14), the following conclusions can be made,

- The system is guaranteed to be non-chaotic for the ranges,

$$0.0 < \gamma < 0.6, \quad 0.9 < \gamma < 1.05, \quad 1.2 < \gamma < 1.3$$

- The system is chaotic for the ranges,

$$0.6 < \gamma < 0.9, \quad 1.05 < \gamma < 1.2, \quad 1.3 < \gamma < 1.5$$

Another hint of chaos that can be obtained through the bifurcation diagram is its self-replicating structure. If one looks at figure 12 closely, one realises that the same bifurcation diagram replicates itself between values of the forcing parameter: $1.2 < \gamma < 1.4$. This self replicating behaviour suggests that the bifurcation diagram may indeed by a fractal structure. Fractals, without going too much into its mathematical formulation are curves whose complexity changes with measurement scale and the length of a fractal curve always diverges to infinity. This implies that the system that shows fractal structures have infinitely many states the system can be in for a given driving parameter. Hence, the damped driven pendulum is a system that exhibits chaos.

3.6 Poincaré Sections

Using phase diagrams to study chaos is quite limiting as a tool since extracting any form of a pattern from such a data becomes quite difficult. A more efficient way to look at the system is to study its long term behaviour. One such way of analysing the system is to look at the system at discrete time intervals. If we apply the same process that we applied for obtaining the steady state angular displacement values in the bifurcation diagram to obtaining the values for angular velocity, and then plot a phase diagram consisting only of the snapshots (at τ), we obtain the Poincaré section for a given value of γ. Consider the case where a system is under going a period two cycle. This corresponds to two concentric circles in the phase plane as follows,

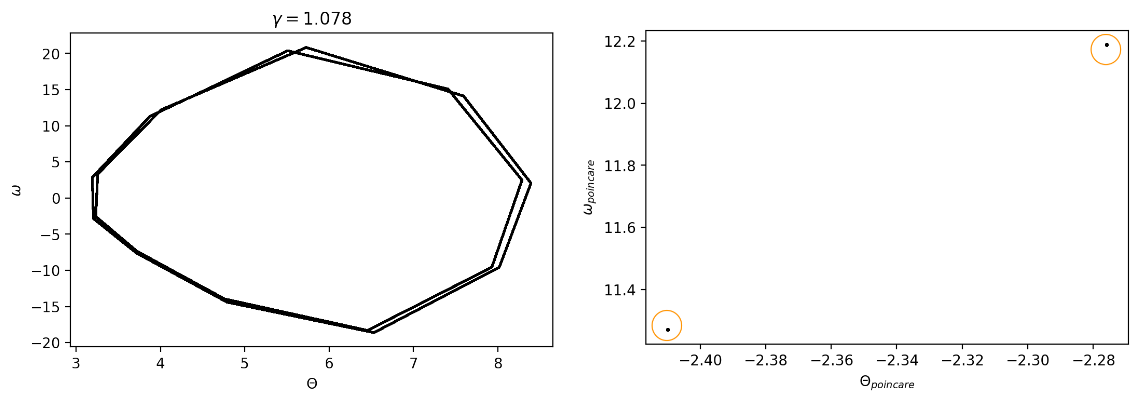


Figure 16: Phase portrait for a two cycle: $\gamma = 1.078, \beta = \omega_0/4$ and its Poincaré section

For the following diagram, if we look at the system, it can be found in either of the two cycles when the transients have died out. Hence when we only look at the values of the angular displacements at our time period τ we would expect there to be two values. Corresponding to this, we would again expect two values for the angular velocity. Hence, the two cycle phase diagram can be encapsulated by just two points on the Poincaré section. However, for chaotic motion that does not repeat itself over time, i.e., the case for $\gamma = 1.5, \beta = \omega_0/8$, the Poincaré section looks as follows,

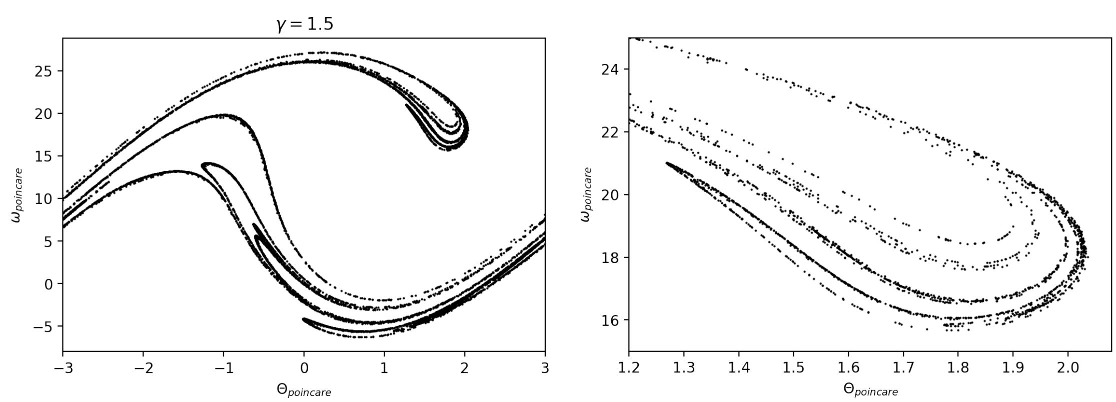


Figure 17: Poincaré section and zoomed portion for $\gamma = 1.5$

The Poincaré section is further evidence that the damped driven pendulum can exhibit chaos. Even though the curve looks simple, it is not a curve at all, but a fractal. Fractals, apart from their mathematically rigorous definition, can be considered self replicating figures. The Poincaré section for the damped-driven pendulum is composed of various tongue like structures that are in turn composed of smaller tongues. These tongues are more evident in the zoomed in portion of the Poincaré section. The fractal nature again corroborates the idea that the system's motion does not repeat itself, and hence consists of a shape that is infinite: like a fractal. When the Poincaré section of the motion of a chaotic system is a fractal, the long-term motion is said to be a strange attractor. This paper does not study the specifics of the strange attractor.

4 Conclusion

This paper studied the non-linear dynamics of the damped-driven pendulum without its linear approximation using an RK4 integrator. This system showcases characteristics of chaos for some values of the driving force parameter γ. Chaos in the system was exhibited through exponential sensitivity to initial conditions, diverging phase portrait trajectories, and the positive Lyapunov exponent. The bifurcation diagram shows a period doubling bifurcation followed by a period doubling cascade, followed by a period three stability region, followed by chaos again. The fractal structures present in the bifurcation diagram and the Poincare sections help classify the damped driven pendulum as a Strange attractor. The fractal structures further solidify the presence of chaos in the system.

5 Discussion

5.1 Future scope of the study

- Study other non-linear oscillators such as the Duffing Oscillator.
- Consider the possibility of chaos by changing β, i.e., damping factor.
- Experimentally verify the above findings with real life situations.
- Consider a non-linear damping dependence, i.e., $F_{\text{friction}} \propto v^2$
- Mathematically verify the fractal nature found in the Bifurcation and Poincaré diagrams.

5.2 Limitations

There are certain limitations to this computational model that need to be addressed. In order to ensure that the system is in steady state, the system must be studied over a considerably large period of time. However, practically this was not possible due to the constraints of time and computational power. However, this limitation could have been resolved to a certain extent by making use of a better processor. This would have significantly shortened the computational time required to run, which otherwise is a time-consuming process, thereby improving the efficiency. Further, while studying the chaotic regimes, smaller time-steps could have been used to improve the accuracy of results obtained. However, as discussed, this was not done due to the high computational cost accompanying the same. Another way of reducing the computational time required to run the code was through the usage of Numpy arrays instead of lists. The aforementioned allows for faster and more efficient operations on arrays of homogeneous data as compared to the latter.

6 References:

1. Strogatz, Steven H. Nonlinear dynamics and chaos: with applications to physics, biology, chemistry, and engineering. CRC press, 2018.

2. Taylor, John R. "Classical Mechanics University Science." (2005)

3. Fitzpatrick, Richard. "Computational physics." Lecture notes, University of Texas at Austin (2006).

4. Landau, Rubin H., and Manuel José Páez. Computational Problems for Physics: With Guided Solutions Using Python. CRC Press, 2018.

5. Bevivino, Josh. "The path from the simple pendulum to chaos." Dynamics at the Horsetooth 1.1 (2009): 1-24.

6. Baker, Gregory L., and Jerry P. Gollub. Chaotic dynamics: an introduction. Cambridge university press, 1996.

7. Fowler, Michael. "Driven Damped Pendulum: Period Doubling, Chaos, Strange Attractors". Galileoandeinstein.Phys.Virginia.Edu, 2022, http://galileoandeinstein.phys.virginia.edu/7010/CM_22a_Period_Doubling_Chaos.html

8. "Angle Transformation To $[-\pi, \pi]$". Mika'S Tech Blog, 2022, https://mika-s.github.io/python/control-theory/kinematics/2017/12/18/transformation-to-pipi.html.

9. Britannica. Mathematics, Poincaré Sections. https://www.britannica.com/science/Poincare-section

Addendum-The Chaotic Pendulum Code

May 14, 2022

1 CODE FOR THE CHAOTIC PENDULUM

[1]:
```
import numpy as np #Main library for computation
import matplotlib.pyplot as plt #To create plots
import scipy as sp #Just in case
plt.rcParams["figure.dpi"]=200 #Size of the plots
```

2 Phase Portrait Code for the Simple Pendulum

Here, the solve function was created for a general case and used to create a phase portrait

[2]:
```
def␣
↪solve(gamma,F_0=0,ANGLE=10,color='darkgoldenrod',label=None,return_plot=True):
↪#initial parameters the code takes

    g = 9.81 #gravitation
    l = 1   #length
    omega_d=5 #Driving frequency

    def f(r, t): #vectorising the code
        theta = r[0]
        omega = r[1]
        dxdt = omega # to convert one second order ODE to 2 first order ODEs
        dvdt = - (g / l)*np.sin(theta)-gamma*dxdt+F_0*np.cos(omega_d*t)␣
↪#equation of motion (unmodified)

        return np.array([dxdt, dvdt], dtype=float)


    a = 0.0 #initial time for integration
    b = 100.0 #final time for integration
    h = 0.1 #step length

    time = np.arange(a, b, h) #array for time
    theta = [] #empty list for appending solutions of ODE
    omega = [] #empty list for appending solutions for angular velocity
```

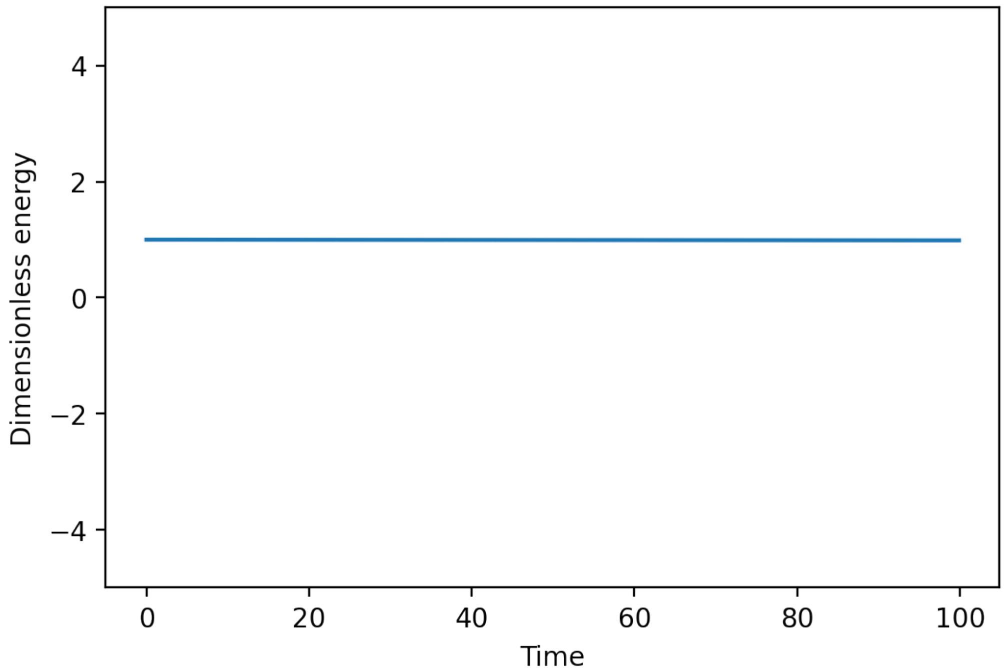
```
    theta_0 = (np.pi / 180) * ANGLE #converting degrees to radians, initial␣
↪displacement
    omega_0 = 0.0 #Initial velocity
    r = np.array([theta_0, omega_0], dtype=float) #Rk4 integrator
    for t in time:
        theta.append(r[0])
        omega.append(r[1])
        k1 = h * f(r, t)
        k2 = h * f(r + 0.5 * k1, t + 0.5 * h)
        k3 = h * f(r + 0.5 * k2, t + 0.5 * h)
        k4 = h * f(r + k3, t + h)
        r  += (k1 + 2 * k2 + 2 * k3 + k4) / 6

    if (return_plot): #additional functionality


        plt.scatter(theta[0],omega[0],color=color,alpha=0.5)
        if (label==None):
            label=r'$\gamma=$'+str(gamma)
        plt.plot(theta, omega, color=color, lw=0.8,label=label)
        plt.xlabel('Angular displacement')
        plt.ylabel('Angular velocity')


    return (np.array(theta),np.array(omega)) #Returns a phase diagram for the␣
↪given conditions
```

3 Verifying RK4

For the case when the pendulum is not damped and not driven, the energy should stay constant. If this were true, then it would indicate that the integrator works well with minimal truncation errors

[3]:
```
time=np.arange(0,100,0.1) #array for time
l=1 #length
g=9.8 #gravity
theta,omega=solve(gamma=0,return_plot=False) #obtaining two arrays for angle␣
↪and velocity
energy=0.5*l**2*omega**2+g*l*(1-np.cos(theta)) #calculating total energy
E_dimensionless=energy/energy[0] #non dimensionalising the energy by initial␣
↪value
fig, ax = plt.subplots()
ax.plot(time,E_dimensionless)
ax.ticklabel_format(useOffset=False)
plt.ylabel('Dimensionless energy')
plt.xlabel('Time')
```

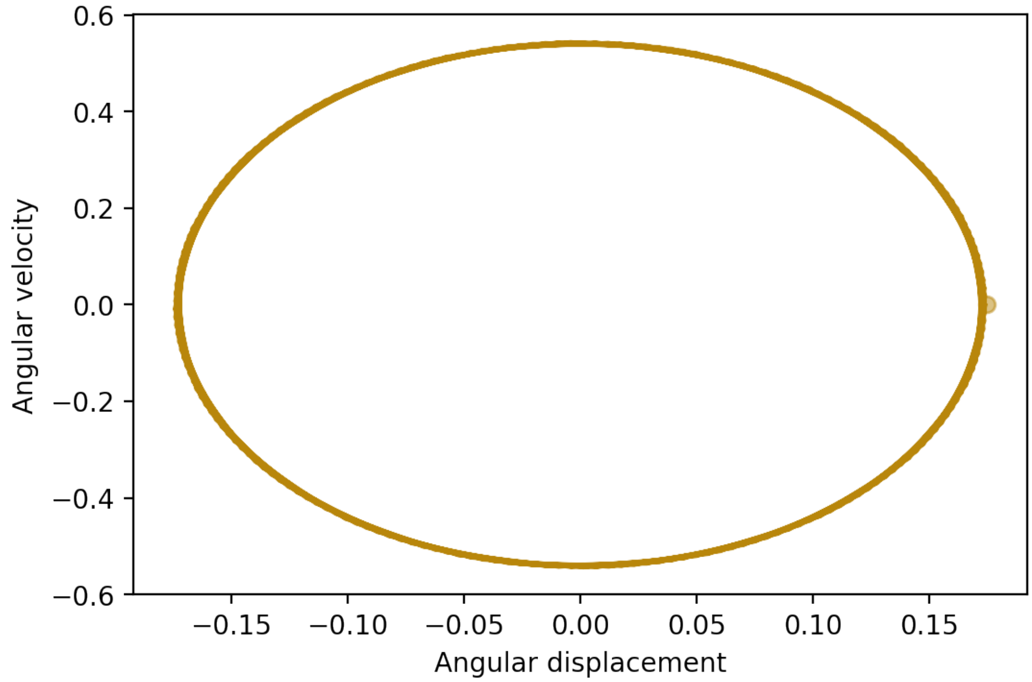
```
plt.ylim(-5,5)
plt.show()
```



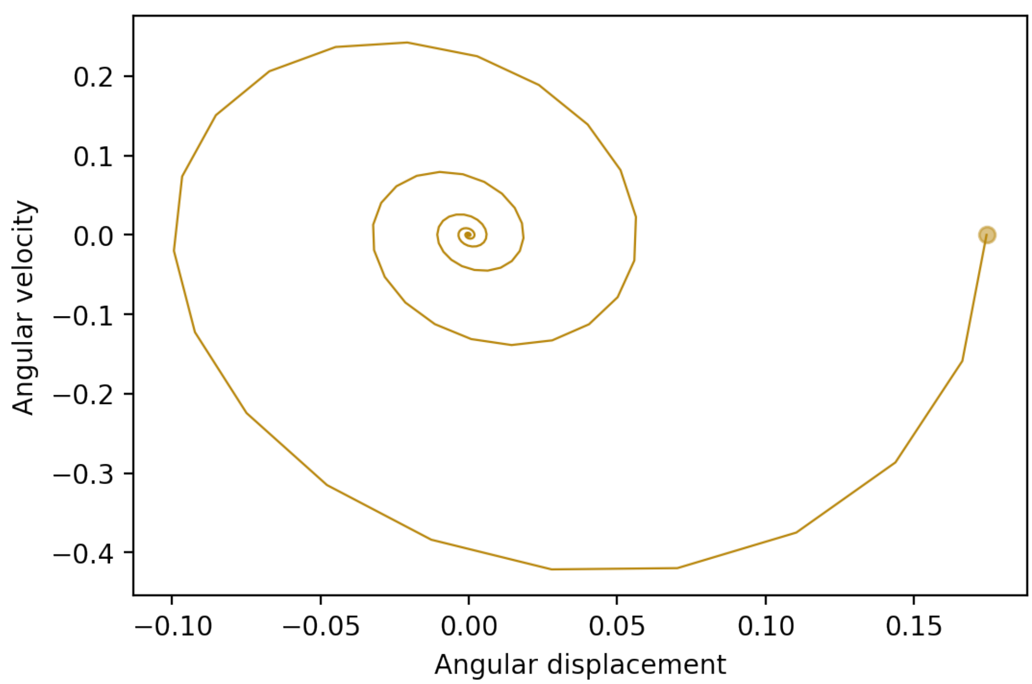
The above diagram shows that energy is conserved and hence the RK4 integrator works. Now we can look at phase diagrams.

4 Phase portraits:

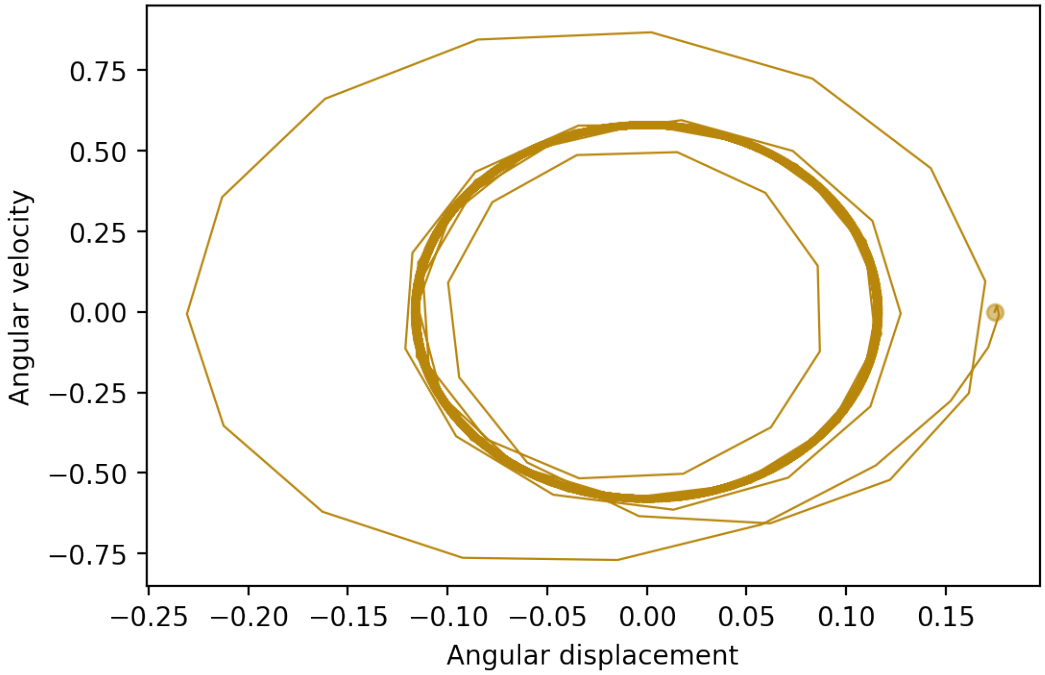
[4]:
```
solve(0,0) #Phase portrait for undamped and undriven case (see linear center)
plt.show()
```



[5]:
```
solve(1.1,0) #Phase portrait for damped but not driven (see stable spiral)
plt.show()
```

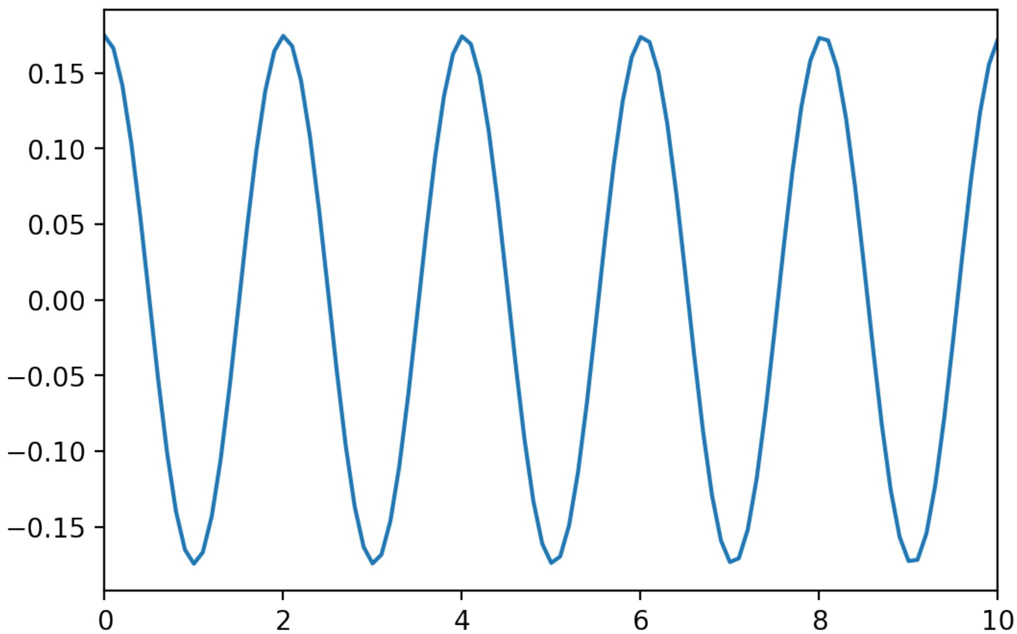


[6]:
```
solve(1.5,2) #Phase portrait for damped driven (see limit cycle)
#To see global mixing or chaos, tune the driving force parameter accordingly
plt.show()
```



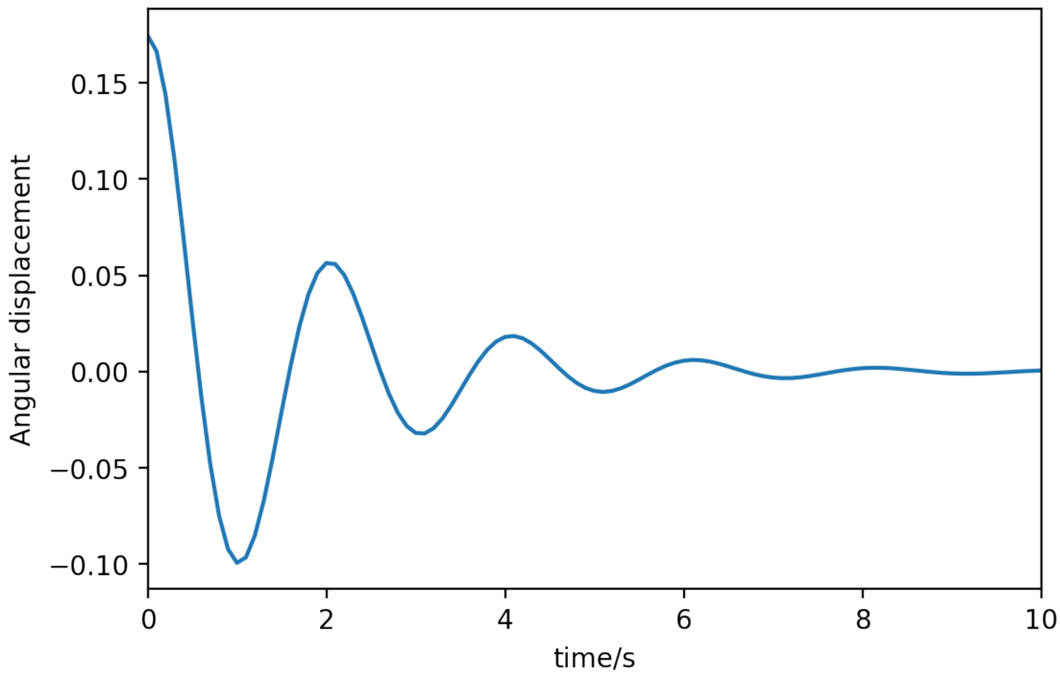
[7]:
```
position,_=solve(0,0,return_plot=False) #obtaining position values from solve
plt.plot(time,position) #for displacement time graph
plt.xlim(0,10)
plt.xlabel('time/s')
plt.ylabel('Angular displacement')
```

[7]: (0.0, 10.0)



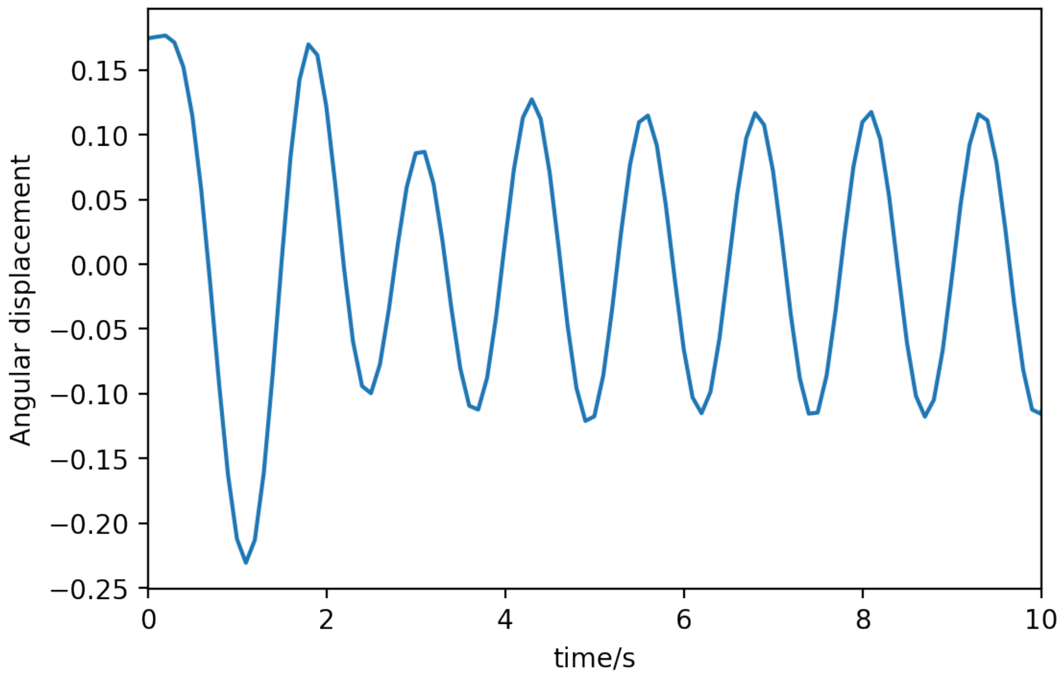
[8]:
```
#displacement time graph for damped oscillator
position,_=solve(1.1,0,return_plot=False)
plt.plot(time,position)
plt.xlim(0,10)
plt.xlabel('time/s')
plt.ylabel('Angular displacement')
```

[8]: Text(0, 0.5, 'Angular displacement')



[9]:
```
#displacement time for damped driven oscillator reaching  a steady state.
position,_=solve(1.5,2,return_plot=False)
plt.plot(time,position)
plt.xlim(0,10)
plt.xlabel('time/s')
plt.ylabel('Angular displacement')
```

[9]: Text(0, 0.5, 'Angular displacement')



5 Non Linear Modification to the equation to study chaos:

[82]:
```
#The ode was changed to include the damping factor beta and driving amplitude␣
↪gamma according to the paper
def solve(gamma=1.05,x_0=0,color='darkgoldenrod',label=None,return_plot=True):

    w_0=3*np.pi #this was the natural frequency
    beta=w_0/8 #this was the damping coefficient
    omega_d=2*np.pi #driving frequency to simplify the poincare sections (see␣
↪later)

    def f(r, t):
        theta = r[0]
        omega = r[1]
        dxdt = omega
        dvdt = - w_0**2*np.sin(theta)-2*beta*dxdt+gamma*w_0**2*np.cos(omega_d*t)

        return np.array([dxdt, dvdt], dtype=float)


    a=0
    b=1000 #use 10^5 for sufficient data points in graph
    h=0.01
```

```
    time = np.arange(a, b, h)
    theta = []
    omega = []

    theta_0 = x_0
    omega_0 = 0.0
    r = np.array([theta_0, omega_0], dtype=float) #rk4 integrator
    for t in time:
        theta.append(r[0])
        omega.append(r[1])
        k1 = h * f(r, t)
        k2 = h * f(r + 0.5 * k1, t + 0.5 * h)
        k3 = h * f(r + 0.5 * k2, t + 0.5 * h)
        k4 = h * f(r + k3, t + h)
        r  += (k1 + 2 * k2 + 2 * k3 + k4) / 6

    return (np.array(theta),np.array(omega))
```

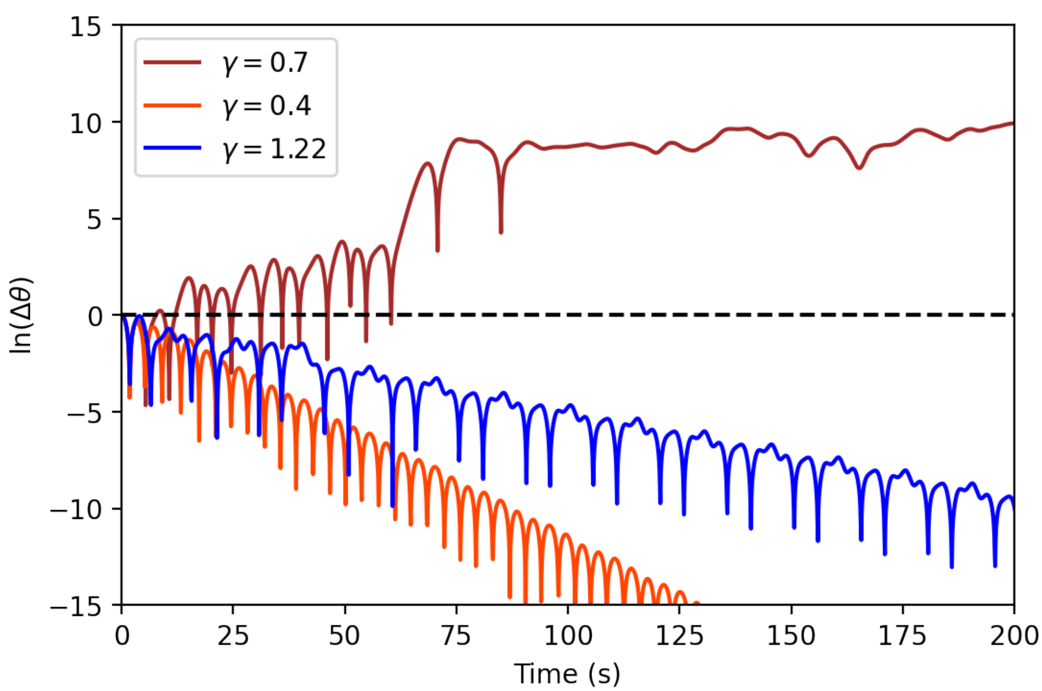
[83]:
```
a=0
b=10000
h=0.1
time = np.arange(a, b, h)

#Gamma=0.7 for studying lyapunov behaviour when it is chaotic
X,_=solve(0.7,0,return_plot=False) #theta  for initial value (where theta_0=0)
Y,_=solve(0.7,0.001,return_plot=False) #theta-2 for initial value (where␣
↪theta_0=0.001)
X_0=X[0] #theta-0 The initial value of the parameter
Y_0=Y[0] #theta-20 The initial value of the parameter
C_0=abs(X_0-Y_0) #Initial seperation vector
Z=np.array(np.log((abs(X-Y)/C_0)))
#Calculating the absolute log value of seperation vector scaled by initial␣
↪values

#Gamma=0.4 for studying lyapunov behaviour when it is non chaotic
X1,_=solve(0.4,0,return_plot=False) #theta
Y1,_=solve(0.4,0.001,return_plot=False) #theta-2
X_01=X1[0] #theta-0
Y_01=Y1[0] #theta-20
C_01=abs(X_01-Y_01)
Z1=np.array(np.log((abs(X1-Y1)/C_01)))

#Gamma=1.22 for studying lyapunov behaviour when it is non chaotic
X2,_=solve(1.22,0,return_plot=False) #theta
Y2,_=solve(1.22,0.001,return_plot=False) #theta-2
X_02=X2[0] #theta-0
Y_02=Y2[0] #theta-20
```

```
C_02=abs(X_02-Y_02)
Z2=np.array(np.log((abs(X2-Y2)/C_02)))
```

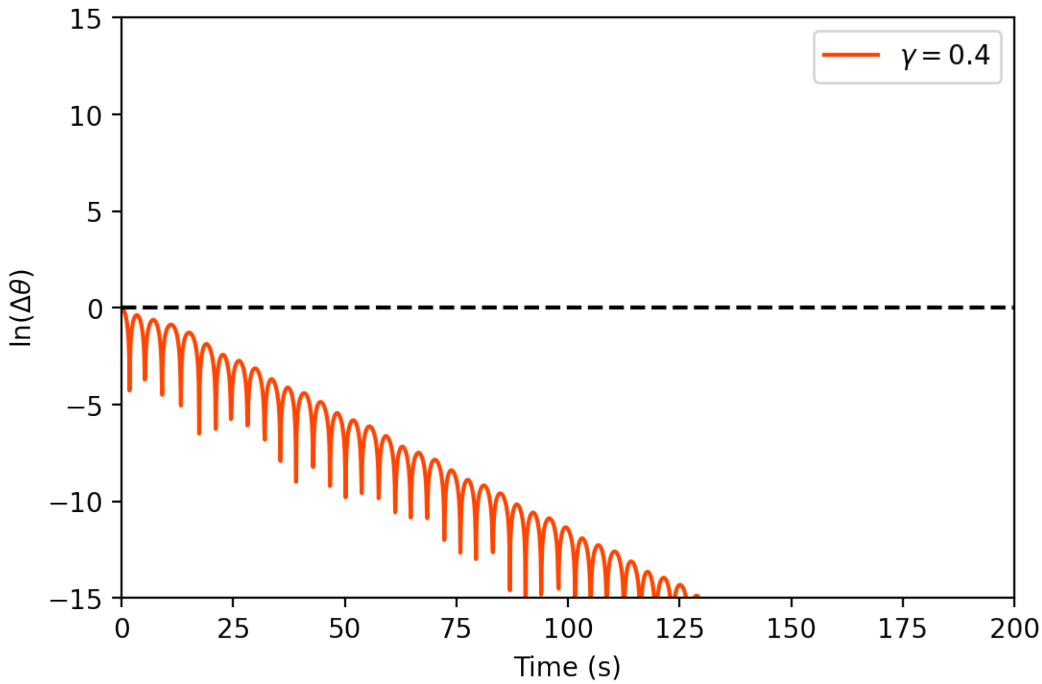
```
C:\Users\HP\AppData\Local\Temp/ipykernel_6656/3341240796.py:15: RuntimeWarning:
divide by zero encountered in log
  Z1=np.array(np.log((abs(X1-Y1)/C_01)))
C:\Users\HP\AppData\Local\Temp/ipykernel_6656/3341240796.py:29: RuntimeWarning:
divide by zero encountered in log
  Z2=np.array(np.log((abs(X2-Y2)/C_02)))
```

5.1 Required plots for Lyapunov exponent

[33]:
```
plt.plot(time, Z, label='$\gamma = 0.7$', c='brown')
plt.plot(time, Z1, label='$\gamma = 0.4$', c='orangered')
plt.plot(time, Z2, label='$\gamma = 1.22$', c='blue')
plt.xlabel('Time (s)')
plt.ylabel('ln($\Delta$$\\theta$)')
plt.legend()
plt.xlim(0,200)
plt.axhline(0,color='k',linestyle='--')
plt.ylim(-15,15)
#put that x=0 line plis
plt.show()
```



[39]:
```
plt.plot(time, Z1, label='$\gamma = 0.4$', c='orangered')
plt.xlabel('Time (s)')
plt.ylabel('ln($\Delta$$\\theta$)')
plt.legend()
plt.xlim(0,200)
plt.axhline(0,color='k',linestyle='--')
plt.ylim(-15,15)
#put that x=0 line plis
plt.show()
```



[64]:
```
lamda = [] #empty array to store lyapunov exponent
for i in range(len(Z1)):
    lamda.append(Z1[i]/time[i]) #value of lyapunov exponent for gamma=0.4
```

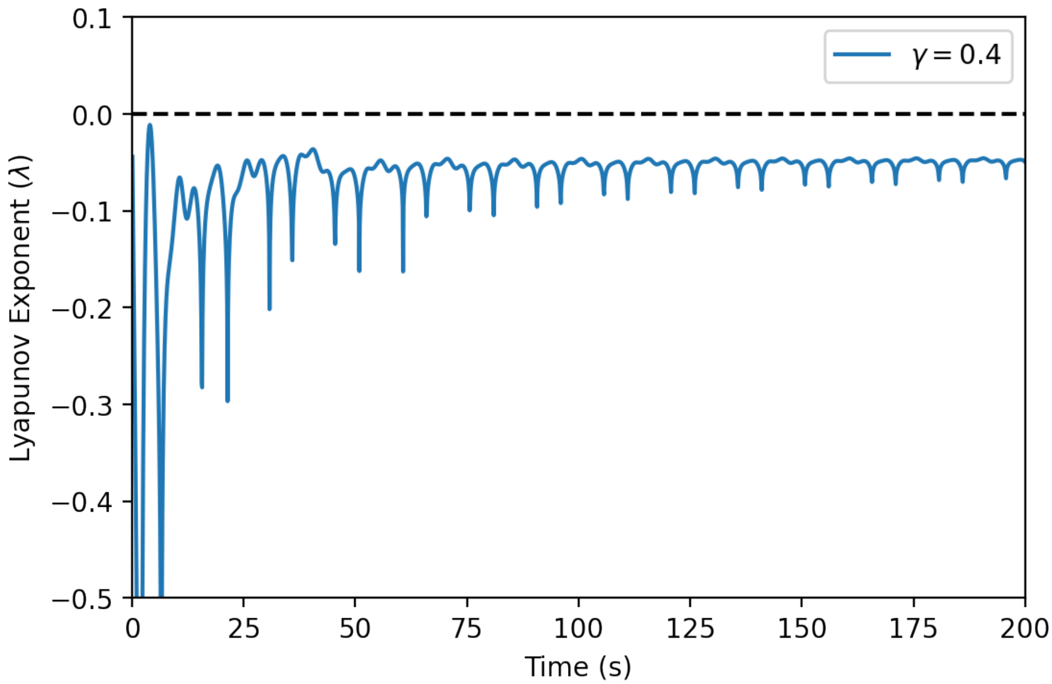
```
C:\Users\HP\AppData\Local\Temp/ipykernel_6656/3731619160.py:4: RuntimeWarning:
invalid value encountered in double_scalars
  lamda.append(Z1[i]/time[i])
```

5.2 Plotting Lyapunov exponent versus time:

[80]:
```
plt.plot(time, lamda, label='$\gamma = 0.4$') #Shows non Chaotic Behaviour
plt.xlim(0,200)
plt.ylim(-0.5,0.1)
plt.xlabel('Time (s)')
```

```
plt.ylabel('Lyapunov Exponent ($\lambda$)')
plt.legend()
plt.axhline(0,color='k',linestyle='--')
```

[80]: <matplotlib.lines.Line2D at 0x1acb0416b80>

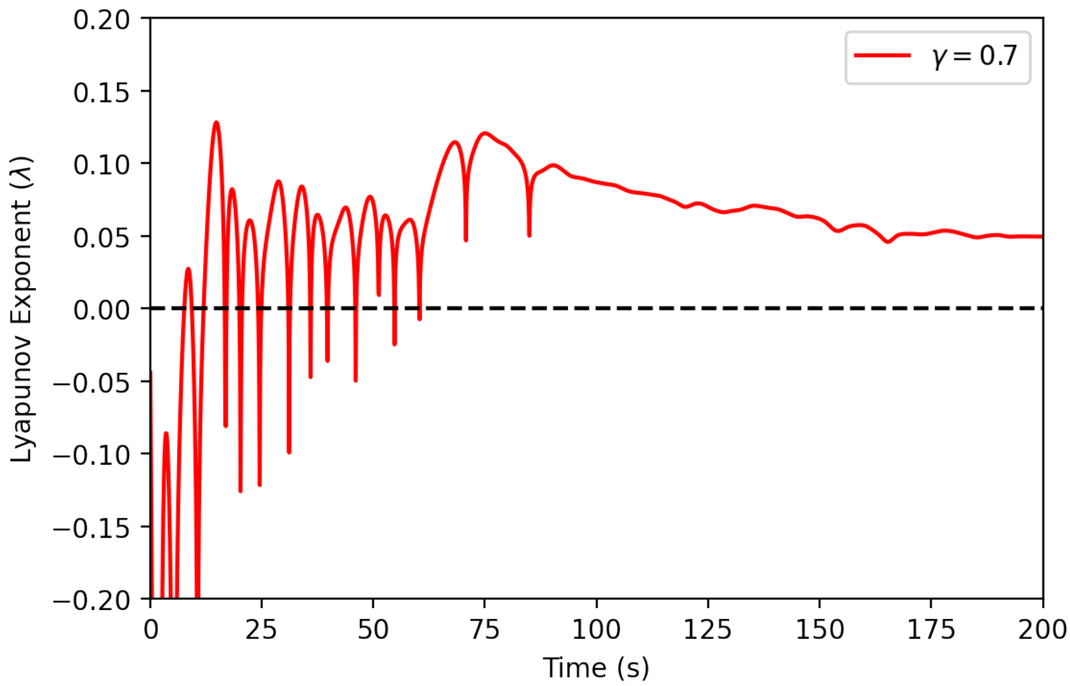


[75]:
```
lamda2 = [] #Same as above for gamma=0.7

for i in range(len(Z)):
    lamda2.append(Z[i]/time[i])
```

```
C:\Users\HP\AppData\Local\Temp/ipykernel_6656/3290217158.py:4: RuntimeWarning:
invalid value encountered in double_scalars
  lamda2.append(Z[i]/time[i])
```

[84]:
```
plt.plot(time, lamda2, label='$\gamma = 0.7 $', color='red')
#Shows chaos as exponent value reaches positive quadrant
plt.xlim(0,200)
plt.ylim(-0.2,0.2)
plt.xlabel('Time (s)')
plt.ylabel('Lyapunov Exponent ($\lambda$)')
plt.axhline(0,color='k',linestyle='--')
plt.legend()
```

[84]: <matplotlib.legend.Legend at 0x1acad84eca0>



6 Bifurcation diagrams

[2]:
```
#Taken from science blog (check citation) to convert the domain of the angle␣
↪from -inf,inf to -pi,pi
def truncated_remainder(dividend, divisor):
    divided_number = dividend / divisor
    divided_number = \
        -int(-divided_number) if divided_number < 0 else int(divided_number)

    remainder = dividend - divisor * divided_number

    return remainder
```

[3]:
```
#Taken from science blog (check citation) to convert the domain of the angle␣
↪from -inf,inf to -pi,pi
def transform_to_pipi(input_angle):
    revolutions = int((input_angle + np.sign(input_angle) * np.pi) / (2 * np.
↪pi))

    p1 = truncated_remainder(input_angle + np.sign(input_angle) * np.pi, 2 *np.
↪pi)
```

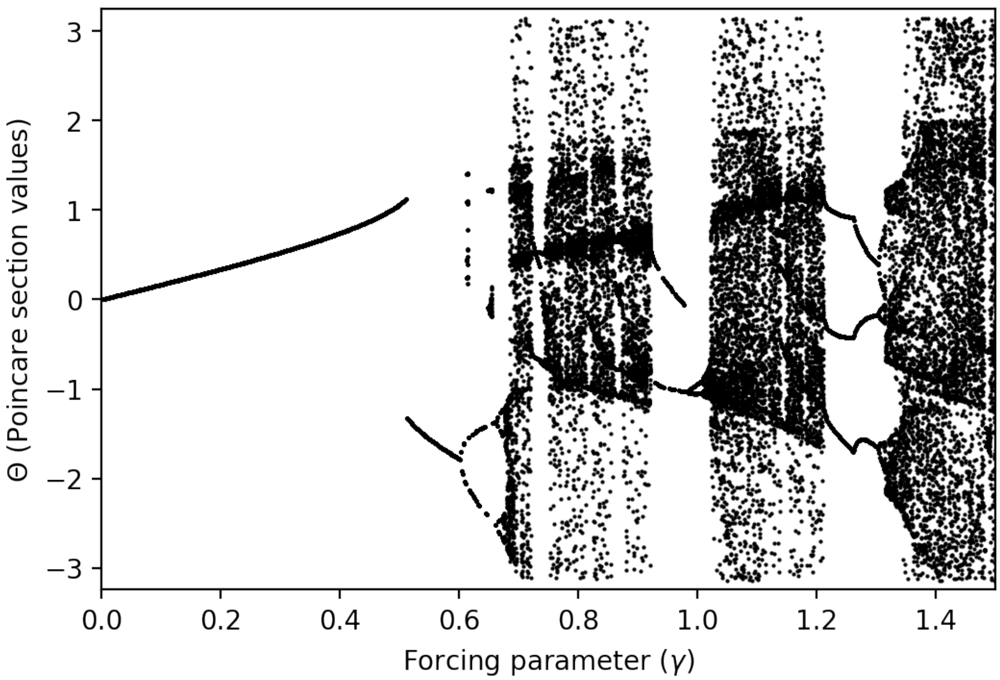
```
    p2 = (np.sign(np.sign(input_angle)
                  + 2 * (np.sign(np.fabs((truncated_remainder(input_angle + np.
↪pi, 2 * np.pi))
                                      / (2 *np.pi))) - 1))) *np.pi

    output_angle = p1 - p2

    return output_angle
```

[4]:
```
def solve(gamma=1.05,color='darkgoldenrod',label=None,return_plot=True):␣
↪#redefining solve for bifurcations

    w_0=3*np.pi
    beta=w_0/8
    omega_d=2*np.pi

    def f(r, t):
        theta = r[0]
        omega = r[1]
        dxdt = omega
        dvdt = - w_0**2*np.sin(theta)-2*beta*dxdt+gamma*w_0**2*np.cos(omega_d*t)

        return np.array([dxdt, dvdt], dtype=float)


    a=0
    b=5000 #keep in a suitable range to reduce computational time
    h=0.1 #0.01 takes a significant ammount of time.
    time = np.arange(a, b, h)
    theta = []
    omega = []

    theta_0 = 0
    omega_0 = 0.0
    r = np.array([theta_0, omega_0], dtype=float)
    for t in time:
        theta.append(transform_to_pipi(r[0]))
        omega.append(r[1])
        k1 = h * f(r, t)
        k2 = h * f(r + 0.5 * k1, t + 0.5 * h)
        k3 = h * f(r + 0.5 * k2, t + 0.5 * h)
        k4 = h * f(r + k3, t + h)
        r  += (k1 + 2 * k2 + 2 * k3 + k4) / 6

    if (return_plot):

        plt.scatter(theta[0],omega[0],color=color,alpha=0.5)
```

```
        if (label==None):
            label=r'$\gamma=$'+str(gamma)
        plt.plot(theta, omega, color=color, lw=0.8,label=label)

    return (np.array(theta),np.array(omega))
```

[5]:
```
a = 0.0
b = 5000.0
h = 0.1
time = np.arange(a, b, h)
F_0=np.linspace(0,1.5,1000) #The values of the parameter as an array to␣
↪calculate bifurcations (1000 values)
Theta_new=[]
```

[9]:
```
for i in F_0:
    Theta, _ =solve(i,return_plot=False) #gives the theta value for a given F_0
    Theta_new=Theta[-500:] #cuts the array only taking last 500 values, assumed␣
↪to be steady state
    for j in range(len(Theta_new)):
        if j%10==0: #j is a time index ranging from 0.1,0.2... divisibility by␣
↪10 gives 1 second 2 second values
            plt.plot(i,Theta_new[j], 'ko',ms=0.5) #plotting bifurcation value␣
↪for the steady state
            plt.ylim(-np.pi-0.1,np.pi+0.1)
            plt.xlim(0,1.5)
            plt.xlabel('Forcing parameter ($\gamma$)')
            plt.ylabel('$\Theta$ (Poincare section values)')
#divisibility by 10 occurs because the time period is 2pi/driving=1. Hence, 1␣
↪time unit= 1 tau
```



7 Poincare Maps

[2]:
```
def solve(gamma=1.
↪5,theta_0=0,color='darkgoldenrod',label=None,return_plot=False): #solve␣
↪redefined for a new value

    w_0=3*np.pi
    beta=w_0/8
    omega_d=2*np.pi


    def f(r, t):
        theta = r[0]
        omega = r[1]
        dxdt = omega
        dvdt = - w_0**2*np.sin(theta)-2*beta*dxdt+gamma*w_0**2*np.cos(omega_d*t)

        return np.array([dxdt, dvdt], dtype=float)


    a = 0.0
    b = 100000.0 #keep in sufficient range. This will give more points on the␣
↪diagram
```

```python
    h = 0.1

    time = np.arange(a, b, h)
    theta = []
    omega = []

    omega_0 = 0.0
    r = np.array([theta_0, omega_0], dtype=float)
    for t in time:
        theta.append(r[0])
        omega.append(r[1])
        k1 = h * f(r, t)
        k2 = h * f(r + 0.5 * k1, t + 0.5 * h)
        k3 = h * f(r + 0.5 * k2, t + 0.5 * h)
        k4 = h * f(r + k3, t + h)
        r  += (k1 + 2 * k2 + 2 * k3 + k4) / 6

    if (return_plot):
        plt.scatter(theta[0],omega[0],color=color,alpha=0.5)
        if (label==None):
            label=r'$\gamma=$'+str(gamma)
        plt.plot(theta, omega, color=color, lw=0.8,label=label)

    return (np.array(theta),np.array(omega))
```

[3]:

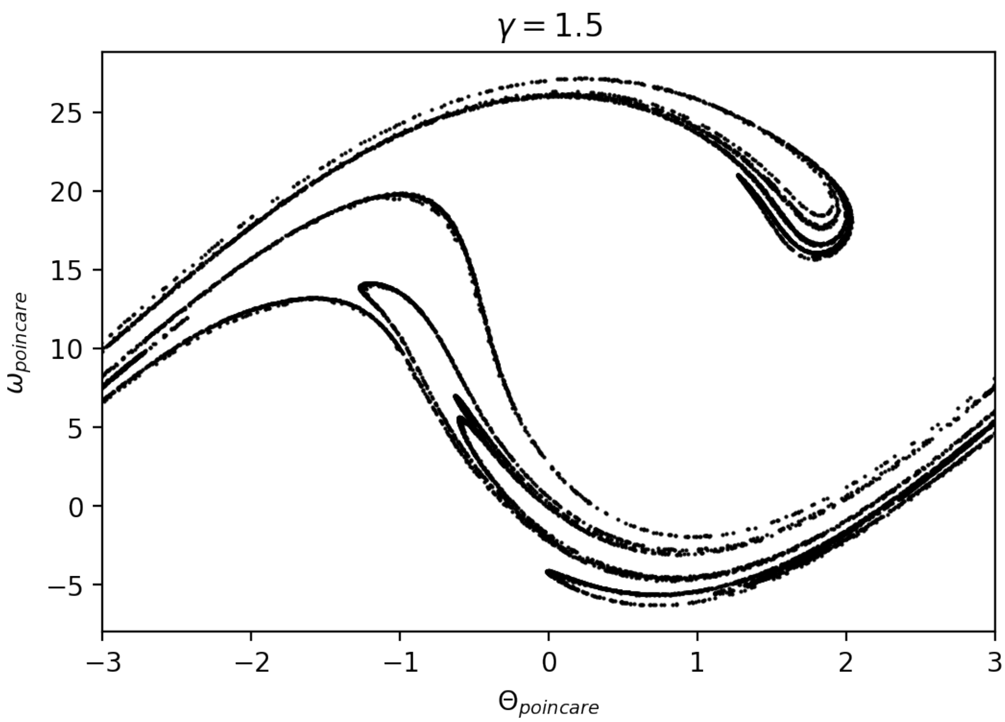
```python
x_poincare=[] #empty list to store poincare section values for angle
y_poincare=[] #empty list to store poincare section values for angular velocity
theta,omega=solve(1.5) #solving for gamma=1.5 which is known for its chaotic␣
↪behaviour
for j in range(len(theta)):
    if j%10==0: #again dividing by 10 to obtain values of angle and velocity at␣
↪tau interval.
        #we are dividing by 10 because h=0.1. Divide by 100 if h=0.01. w_d must␣
↪be 2pi for this to work
        x_poincare.append(theta[j])
        y_poincare.append(omega[j])
```

[6]:

```python
# The problem with the above poincare angle value is that its domain is from␣
↪-inf to inf
#this domain for angles needs to be changed to -pi to pi, this conversion can␣
↪be done by transform function.
XX=[] #blank list for storing transformed values
for i in range(len(x_poincare)):
    XX.append(transform_to_pipi(x_poincare[i])) #transformed values
```

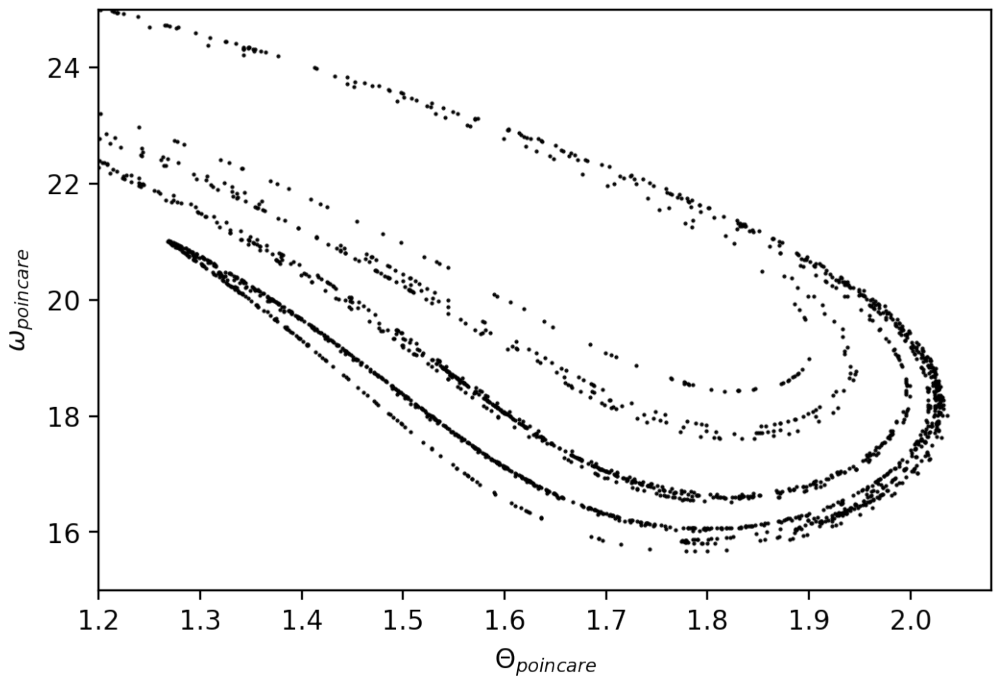
[7]:
```
plt.plot(XX,y_poincare,'ko',ms=0.5) #gives poincare section for gamma=1.5.␣
↪Shows fractal patterns.
#change the value of gamma above to get graphs for the non chaotic value of␣
↪gamma.
plt.xlim(-3,3)
plt.xlabel('$\Theta_{{poincare}}$')
plt.ylabel('$\omega_{{poincare}}$')
plt.title('$\gamma=1.5$')
```

[7]:
```
Text(0.5, 1.0, '$\\gamma=1.5$')
```



[8]:
```
#Zooming in on the tongues to further discuss fractal patterns by
plt.plot(XX,y_poincare,'ko',ms=0.5)
plt.xlim(1.2,2.08)
plt.ylim(15,25)
plt.xlabel('$\Theta_{{poincare}}$')
plt.ylabel('$\omega_{{poincare}}$')
```

[8]:
```
Text(0, 0.5, '$\\omega_{{poincare}}$')
```



24
22
20
18
16
1.2
1.3
1.4
1.5
1.6
1.7
1.8
1.9
2.0
$\omega_{poincare}$
$\Theta_{poincare}$