Simulation of Chaotic Operation Of A Damped Driven Pendulum Using Python

JOAN JANI
Department of Engineering Physics
Polytechnic University of Tirana
Rr. S. Delvina, 1001
ALBANIA

Abstract: In this paper, we are presenting a new pedagogical method for the introduction of the study of non-linear systems. Our approach is based on the use of open-source software which is publicly available. In response to this motivation, we are using the Python programming language which offers a holistic approach to scientific research. We will present the analysis of the pendulum motion under the influence of an external force. The differential equation governing the system will be presented and solved using numerical methods. Moreover, the phase diagram of the system will be presented for various system parameters. We will describe the transition to a chaotic operation and the key factors of this procedure. The chaotic behaviour is verified by calculating the maximum Lyapunov exponent. This pedagogical approach emerging is based on the physical properties of the system and not on the numerical methods used so that the student can understand the dynamics of the system more comprehensively.

Key-Words: Chaos, simple harmonic oscillator, damped driven pendulum, phase space, Lyapunov exponent

Received: October 19, 2021. Revised: October 23, 2022. Accepted: November 25, 2022. Published: January 25, 2023.

Introduction

Computer modeling is an essential part of engineering education, as it provides students with the opportunity to develop and apply their knowledge in a practical setting. By utilizing computer models, engineers can gain a better understanding of how different components interact within complex systems, [1]. For this purpose, the knowledge of computer programming should be an integral part of the education of engineers and scientists. In the past, languages such as Fortran, C, C++, etc. have been used for expressing the mathematical models which are describing the system, [2]. Using these languages requires experience and a deep understanding of how computers work which should not be a prerequisite for a new scientist to get an introduction to the field of dynamical system study. On the other hand, the Python programming language offers a user-friendly working environment, offers fairly good documentation, and can be used even by the new scientist, [3].

Python is an ideal language for physics simulation and modeling due to its ease of use, flexibility, and availability of a wide range of libraries for physics. Python allows for fast prototyping of simulations and models and can be used to quickly develop user interfaces for visualization and control. Additionally, Python can be used to integrate data from different sources, such as sensor data, to develop complex models and simulations. Python is also well-suited for developing physics-based models and simulations due to its powerful graphics capabilities. Finally, Python is a great choice for physics research and engineering, as it can be used to automate data analysis and the development of models and simulations. The Python programming language has been generally accepted as a scientific tool in the field of the study of dynamical systems, [4].

There are several publicly available packages for the Python programming language that offer ready-to-use numerical methods, [5]. The use of these methods focuses students' interest in understanding the model and the main parameters and constants that govern it. As an example of the application of these methods, we will present the use of the function `odeint()` for solving differential the system of differential equations which describe the behavior of the system could be found in the library `scipy` which is a scientific computation library that apart from ODE (Ordinary Differential Equations) solvers offers modules for optimization, integration, interpolation, linear algebra, FFT (Fast Fourier transform), signal, and image processing, [6]. The use of Python for teaching purposes in the context of courses such as Computational Physics and Dynamical Systems is constantly gaining ground over other languages, [7].

Chaos is the appearance of a lack of order in a system that nevertheless follows laws or rules. [8] Chaos is present in systems with non-linear behavior, the non-linearity in a system means that the measured values in the output of the system are not proportional to the values of the input. The presence of non-linearity in a system does not mean that the system will behave chaotically but a form of non-linearity if required to achieve chaotic behavior, [9]. A system showing chaotic behavior undergoes transitions between non-chaotic and chaotic states in general, [10]. A similar approach could be found in [11] where the transition to chaos is made by changing the angular frequency of external force, in our approach the transition to chaos is made by changing the amplitude of the external force applied to the system.

1 Simple oscillator

We begin our pedagogical method by using the integration method on a well-known problem such as that of the oscillator.

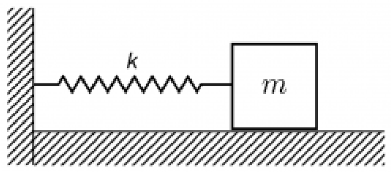

Fig. 1: Simple Harmonic Oscillator with a spring mass system

In figure 1 a simple harmonic oscillator with a spring mass system is presented. Since the constant of elasticity of the spring is denoted with k and the mass of the object with m, we can express the differential equation of the motion of the system as follows:

$$m\frac{d^2x}{dt^2} = -kx \tag{1}$$

the system is governed by a second-order linear differential equation, [12]. To apply a numerical method of integration for equation (1) we rewrite it as a pair of first-order differential equations, which are presented at equation (2).

$$\frac{dv}{dt} = -\frac{k}{m}x \quad \text{and} \quad \frac{dx}{dt} = v \tag{2}$$

We suppose the system with parameters $k = 9\,N/m$ and $m = 1\,kg$ and with initial conditions $x = 5\,m$ and $v = 0\,m/s$. The simulation will calculate the position and velocity for the time interval of the oscillator from $t = 0\,s$ to $t = 10\,s$. Will be calculated $N = 1000$ points of the trajectory. Increasing the sampling (N>1000) of the trajectory would require more computing power to achieve the solution, while reducing it would result in an increase in the computational error of the numerical method.

For our purpose, we will use the `numpy` which is a Python library used for working with arrays, and the `matplotlib` which is a plotting library for the Python programming language.

```
# Oscilator
from scipy.integrate import odeint
import numpy as np
import matplotlib.pyplot as plt
# System Parameters
[k, m] = [9, 1]
# Initial Conditions
x = [5,0]
# Simulation parameters
N = 1000
t = np.linspace(0,10,N)
def f(x,t):
    dvdt = -(k/m)*x[0]
    dxdt = x[1]
    return [dxdt,dvdt]
solution = odeint(f,x,t)
```

The above code implements the solution. The code can be easily understood by someone who has no programming knowledge. In the beginning, the necessary packages are loaded into the program. The commands for the system parameters and the initial conditions are given following with the function `f(x,t)` with the differential equations of the system. The function `odeint` solves the system and the results are saved to variable `solution`

```
plt.plot(t,solution[:,0])
plt.xlabel('Time t[s]')
plt.ylabel('x[m]')
plt.show()
```

To plot the solution which we are getting from `odeint`, the above code is used. The presented code uses the library `matplotlib` which has a very similar syntax to `MATLAB`. The result is presented in figure (2) from were the sinusoidal correlation of the position of the oscillator in relation to time can be seen.

The phase space of the system is created with some minor changes to the above code. (fig. 3). The phase diagram gives us an overview of the evolution of the system. The position of the oscillator is shown on the horizontal axis, while its speed is on the vertical axis. Furthermore, from the Fig. 3 we observe that the period of the system is $T = 2.09\,s$ as we expect from the relation $T = 2\pi/\sqrt{k/m}$

2 Driven damped pendulum

Considering a simple pendulum in a constant gravitational field as it is presented in fig. 4. The equation of motion is given by the differential equation 3 where its current angle is denoted by θ and angular velocity

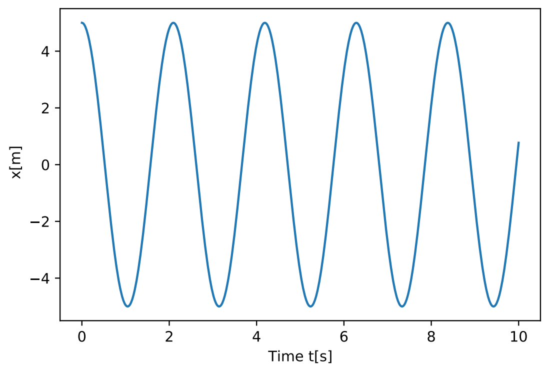

Fig. 2: Position of the oscillator vs time

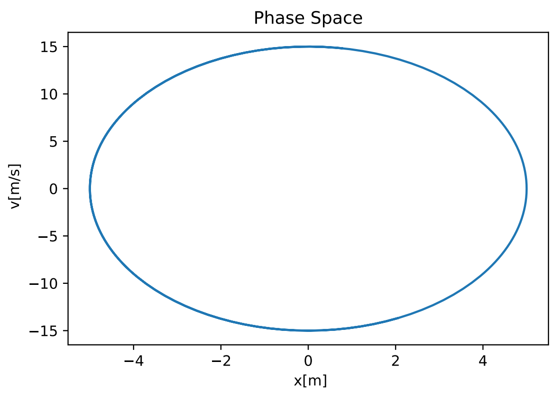

Fig. 3: Phase space of the harmonic oscillator

by $\dot{\theta}$.

$$-ml\ddot{\theta} = mg\sin(\theta) \tag{3}$$

The reason that the small-angle approximation is made is that otherwise the differential equation which describes the system could not be solved using analytical methods. The solution of the equation could easily get using the code below. Regardless of the initial conditions, the system will go to a steady state. [11]

Equation (4) shows the differential equation of a damped-driven pendulum. The motion of the pendulum is damped by a force proportional to its velocity which is expressed in the equation with the term $-\beta\dot{\theta}$. Moreover, equation (4) represents a pendulum with a time-varying external force with amplitude A and frequency ω expressed by the term $A\sin(\omega t)$.

$$\ddot{\theta} = -\frac{g}{L}\sin\theta - \beta\dot{\theta} + A\sin(\omega t) \tag{4}$$

For solving the differential equation (4) we consider a system with parameters, $g = 10\,m/s^2$, $L = 10\,m$, $\beta = 0.5\,Ns/m$, $A = 1\,N$ and $\omega = 2,3\,rad/s$. The simulation starts from initial position $\theta = 0.5\,rad$ and $\dot{\theta} = 0\,rad/s$. The simulation will be performed for $0 < t < 200\,s$ using $N = 20000$ time samples.

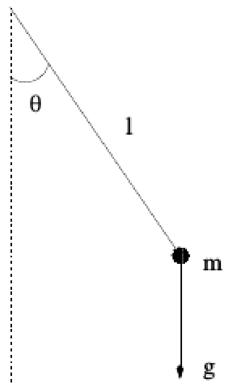

Fig. 4: A simple pendulum

```
# Driven dumped pendulum
from scipy.integrate import odeint
import numpy as np
import matplotlib.pyplot as plt

# System Parameters
[g, L, b]= [10, 10 ,0.5]
[w , A]  =  [2/3, 1]
# Initial Conditions
θ = [0.5,0]
# Simulation parameters
N = 20000
t = np.linspace(0,200,N)
def f(θ,t):
    dvdt = −(g/L)∗np.sin(θ[0])...
           ...−b∗θ[1]+A∗np.sin(w∗t)
    dθdt = θ[1]
    return [dθdt,dvdt]
solution = odeint(f,θ,t)
```

We run the code and the results are saved in the variable "solution". The time needed for the solution is relatively short although the number of calculation points is significantly larger. Then we proceed with the creation of the corresponding graphical representations of the angular displacement with time and phase space.

The graph is shown in Fig. 5 where we see that after the time of 100 seconds the system performs harmonic oscillation of constant amplitude and frequency. At this point we let the students experiment with the various initial conditions of the system. By choosing different initial conditions we see that the system exhibits the same behavior and finally performs harmonic oscillations.

To construct the phase diagram we will use the states of the system after harmonic oscillation is reached. We should also ensure that the value of the

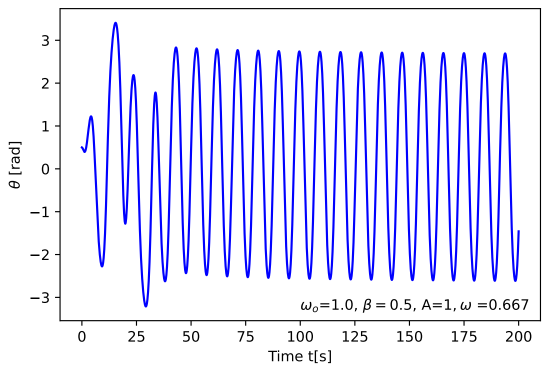

Fig. 5: Position of the oscillator vs time

angle should not exceed the range $-\pi \leq \theta \leq \pi$.

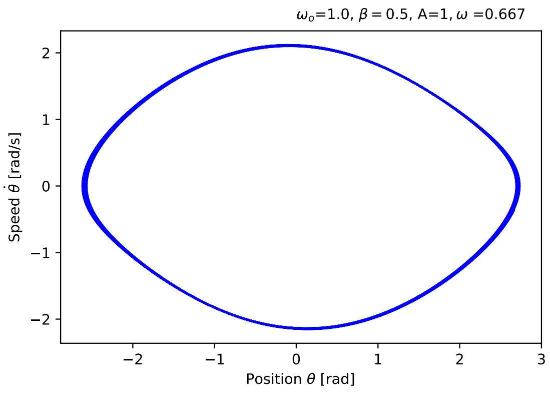

Fig. 6: Phase space of the system with $A = 1\,N$

In order to bring the system described by the equation (4) in chaotic operation we use the method of period doubling by changing the amplitude of the external force. We have changed the angular frequency of the external force to $\omega = 2/3\,rad/s$ and we change the amplitude of the external force to $A = [1.35, 1.45, 1.47,\ \text{and}\ 1.5]$

The transition to chaos is shown in figure 7 where the phase space of the system is presented when the amplitude of the driving force is $A = 1.35$. The trajectory of the system in the phase space diagram is a closed loop as expected. Increasing the amplitude to $A = 1.45$ the trajectory in the phase space is changing, and a second path is appearing denoting the period doubling of the system as it is shown in fig. 8.

By increasing the amplitude to $A = 1.47$ the phase space diagram of the system changes again as it is presented in figure 9. The system doubles its period, but its behavior is still predictable.

By choosing the amplitude of the external force $A = 1.5$ the operation of the system becomes chaotic

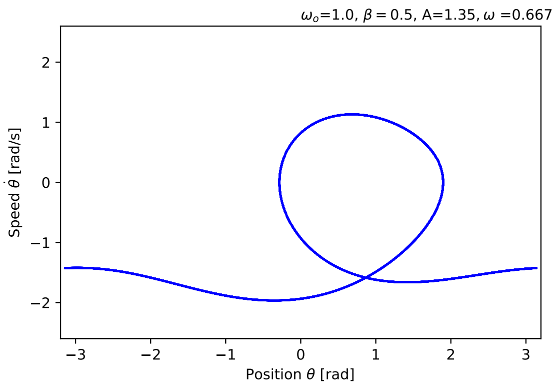

Fig. 7: Phase space of the system with $A = 1.35\,N$

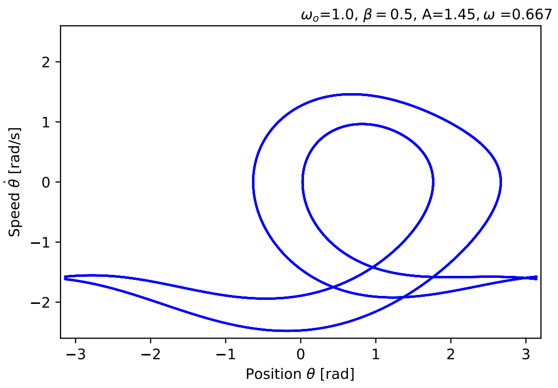

Fig. 8: Phase space of the system with $A = 1.45\,N$

as it shown in Fig. 10. It has to be mentioned that the motion of the system is not random. There is a periodicity but the period is large enough. This makes a small observation of the operation look totally stochastic. In this state, the system is called a strange attractor. In this situation, a small change in the initial conditions of the system will cause a big difference in its motion, regardless that the shape of the phase space will not change.

The chaotic operation could be characterized by the rate of separation of infinitesimally close trajectories. this is done by the calculation of the Lyapunov exponent described below. The presence of chaos is verified also in electronic circuits [13].

3 Lyapunov exponent calculation

The chaotic behavior of a systems is verified by the calculation of the Lyapunov exponent, [14]. The calculation of this exponent is done by equation (5):

$$\delta_t \cong \delta_0 e^{\lambda t} \qquad \lambda = \frac{1}{N}\sum_{j=1}^{N} \log \frac{\delta_{t,j}}{\delta_{0,j}} \tag{5}$$

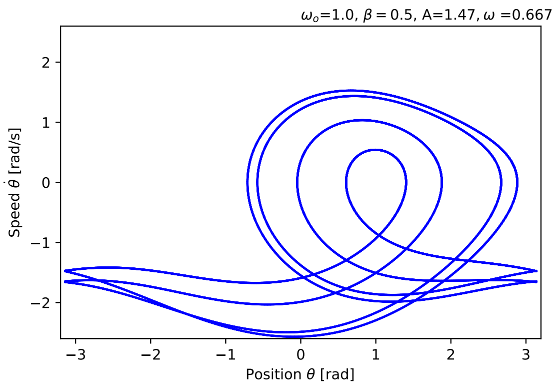

Fig. 9: Phase space of the system with $A = 1.47\,N$

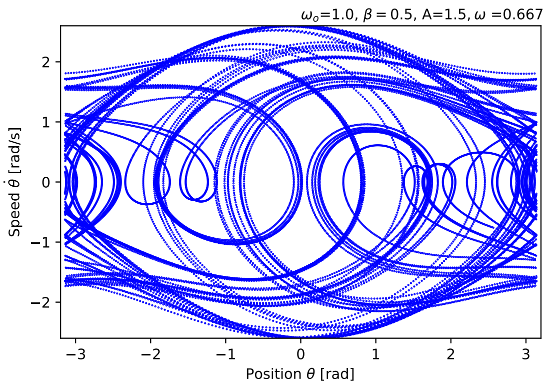

Fig. 10: Phase space of the system with $A = 1.5$

were λ is the Lyapunov exponent δ_t is the separation of trajectories after t time, δ_0 is the initial separation. For discrete time systems, like in our case, N is the time samples needed for the calculation. In order to make the calculation, we simulate two trajectories of the system whose initial conditions differ very little. The initial conditions θ_{1o} and θ_{2o} where $\theta_{2o} = \theta_{1o} + \epsilon$, where ϵ is value very close to the machine epsilon, for our case $\epsilon = 2.220e - 16$. so the initial separation of the trajectories in the phase space is:

$$\delta_0 = |\theta_{2o} - \theta_{1o}|$$

After the calculation of each trajectory for the corresponding initial condition we calculate their difference through the relation.

$$\delta_0 = \Delta\theta = |\theta_2 - \theta_1|$$

the difference is growing exponentially with time as it is shown in fig. 11. The graph is assumed to be linear when we create a log-linear plot. From our calculations, the Lyapunov exponent of this system is $\lambda = 0.06132158503950399$. The positive value verifies the chaotic operation as was expected.

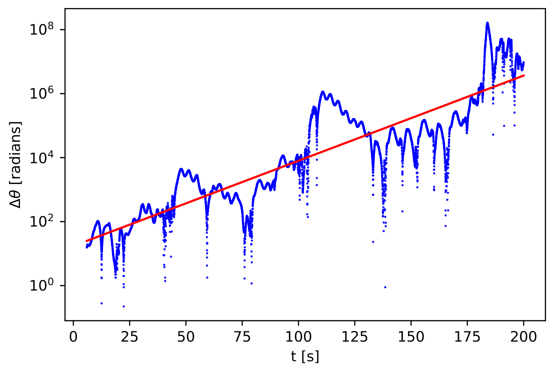

Fig. 11: Exponential increase of the difference between two very close orbits with respect to time. The red line represents the linear regression.

4 Discussion

This approach is applied in the context of the Computational Physics course taught to second-year students in the Applied Physics department of the Polytechnic University of Tirana in Albania. After the explanation of the basic principles governing the phenomenon, the students, with the help of the lecturer, construct the mathematical model that describes it. This model is then run on the computer and the graphs describing its evolution are created.

Our didactic approach presents clearly and comprehensively the transition of a nonlinear system from order to chaos. The demonstration could be presented in the lecture or a laboratory environment giving students a hands-on experience in the field of dynamic systems and chaos.

In the future, we will try to make a create the code that implements the Poincare section of the system.

5 Conclusion

In this paper, we demonstrate the use of Python programming language in simulating a simple linear system. We showed that this methodology can be used in simulations of a non-linear system. This demonstration can be used as a didactic tool for introduction to dynamical system analysis of its elegance and ease of use.

References:

[1] E. A. Gago, C. L. Brstilo, and N. de Brito, “Computational simulation: Multidisciplinary teaching of dynamic models from the linear algebra perspective,” *WSEAS Transactions On Advances In Engineering Education*, 2022.

[2] P. W. Gaffney, “A performance evaluation of some fortran subroutines for the solution of

stiff oscillatory ordinary differential equations," *ACM Transactions on Mathematical Software (TOMS)*, vol. 10, no. 1, pp. 58–72, 1984.

[3] P. Borcherds, "Python: a language for computational physics," *Computer Physics Communications*, vol. 177, no. 1, pp. 199–201, 2007. Proceedings of the Conference on Computational Physics 2006.

[4] B. W. l. Margolis, "Simupy: A python framework for modeling and simulating dynamical systems," *Journal of Open Source Software*, vol. 2, no. 17, p. 396, 2017.

[5] D. R. Gwynllyw, K. L. Henderson, E. G. Guillot, *et al.*, "Using python in the teaching of numerical analysis.," *MSOR Connections*, vol. 18, no. 2, 2020.

[6] J. Nunez-Iglesias, S. Van Der Walt, and H. Dashnow, *Elegant SciPy: The art of scientific python.* " O'Reilly Media, Inc.", 2017.

[7] J. Kusalaas, "Numerical methods in engineering with python 3," 2013.

[8] A. BenSaïda, "A practical test for noisy chaotic dynamics," *SoftwareX*, vol. 3-4, pp. 1–5, 2015.

[9] S. Strogatz, *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry, and Engineering*. CRC Press, 2018.

[10] H. Nagashima, Y. Baba, and M. Nakahara, *Introduction to chaos: physics and mathematics of chaotic phenomena*. CRC Press, 2019.

[11] E. Ayars, "Computational physics with python," *California State University*, 2013.

[12] K. N. Anagnostopoulos, *Computational Physics, Vol I: A Practical Introduction to Computational Physics and Scientific Computing*. Konstantinos Anagnostopoulos, 2014.

[13] M. Hanias, I. Giannis, and G. Tombras, "Chaotic operation by a single transistor circuit in the reverse active region," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 20, no. 1, p. 013105, 2010.

[14] J. Jani and P. Malkaj, "Numerical calculation of lyapunov exponents in various nonlinear chaotic systems," *International Journal of Scientific & Technology Research*, vol. 3, no. 7, pp. 87–90, 2014.

Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)

The author contributed in the present research, at all stages from the formulation of the problem to the final findings and solution.

Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself

No funding was received for conducting this study.

Conflict of Interest

The author has no conflict of interest to declare that is relevant to the content of this article.