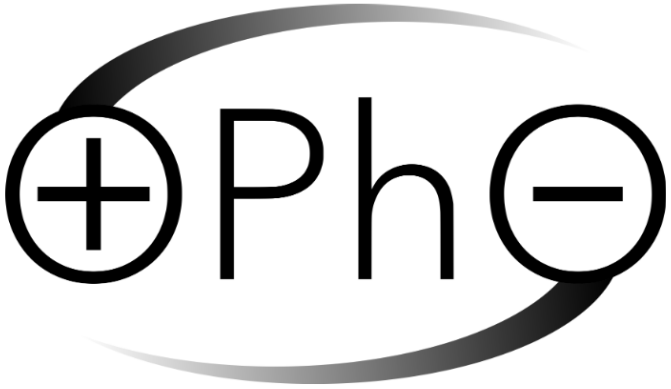
2023 Online Physics Olympiad: Invitational Contest



Experimental Exam

Sponsors

This competition could not be possible without the help of our sponsors, who are all doing great things in physics, math, and education.

General Instructions

The experimental examination consists of 1 long answer question worth 25 points over 1 full day from August 6, 0:01 am GMT.

- **The team leader should submit their final solution document in this google form.**
- If you wish to request a clarification, please use this form. To see all clarifications, view this document.
- Participants are given a google form where they are allowed to submit up-to 100 megabytes of data for their solutions. It is recommended that participants write their solutions in LaTeX. However, handwritten solutions (or a combination of both) are accepted too. If participants have more than one photo of a handwritten solution (`jpg, png`, etc), it is required to organize them in the correct order in a `pdf` before submitting. If you wish a premade LaTeX template, we have made one for you here.

Specific Rules

For any part of this paper, you are allowed to use online tools and resources to help you, as long as you are not requesting help from anyone outside of your team. Allowable resources include Wikipedia, research papers, Wolfram Alpha, Python, Excel, etc.

However, you must document every resource that you use and cite them when applicable. As a general rule of thumb, you should derive any results that cannot be found on Wikipedia. Therefore, solutions along the lines of: "By Wolfram Alpha, this is true." will not be accepted. Be reasonable please.

Every time you are asked to run an experiment, you must provide the input parameters and a screenshot of the output.

Accessing the Program

To access the Python notebook, follow this **link**. You will be able to perform all the code online, without downloading anything. If you cannot access the link, we will also provide the source code on our website.

Background Information

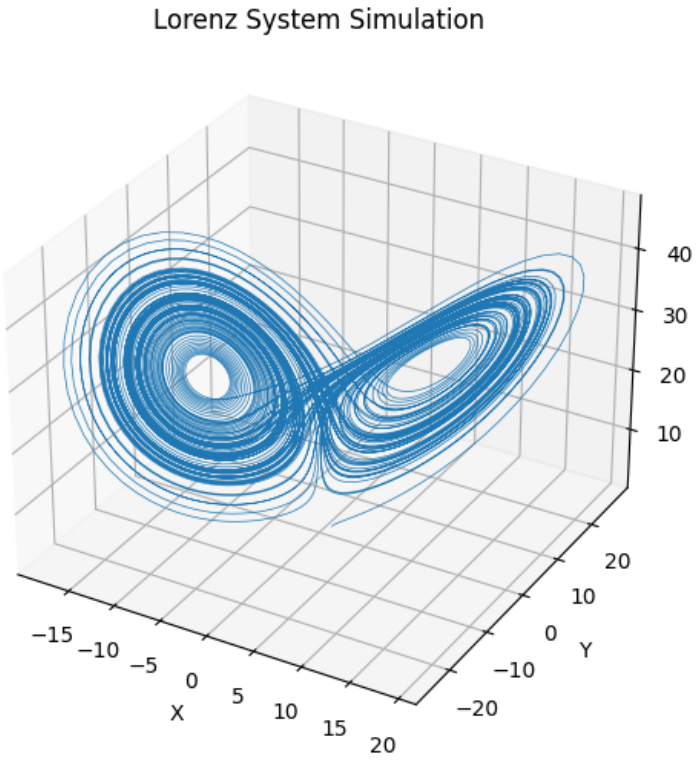
In this problem, you will be running a computer simulation written in *Python* to complete a series of questions relating to the Lorenz system, a system of ordinary differential equations. While you do not need to fully understand how exactly the code operates, it could be beneficial to grasp the process the code follows.

The script employs a simple algorithm to simulate and illustrate the Lorenz system.

1. Define constants for the Lorenz system equations, define the system of equations as a function, and establish the initial state. Set simulation parameters including time-step, maximum time, and the number of steps.
2. Implement Fourth-order and Second-order Runge-Kutta methods as functions. These will be used for solving the Lorenz system equations.
3. Run a loop for each time step where the system's state is updated using the Fourth-order Runge-Kutta method and stored in the state trajectory.
4. Plot the state trajectory of the Lorenz system in a 3D graph to visualize the system's behavior over time.

The Lorenz System

The Lorenz system is a system of ordinary differential equations (ODEs) that was first studied by Edward Lorenz when investigating the physical properties of a convective fluid flow. It is notable for having chaotic solutions for certain parameter values and initial conditions. In particular, the system shows sensitive dependence on initial conditions, which is a key property of chaotic systems. In other words, trajectories starting at slightly different initial conditions can end up at vastly different states. For more information, see this video on the Butterfly effect by Veritasium.



The Lorenz system is defined as a coupled differential equation for the quantities (x, y, z) where x is proportional to the rate of convection, y to the horizontal temperature variation, and z to the vertical temperature variation:

$$\begin{aligned}\dot{x} &= \sigma(y - x)\\ \dot{y} &= x(\rho - z) - y\\ \dot{z} &= xy - \beta z\end{aligned}$$

where σ, ρ, and β are system parameters. The parameter σ is the Prandtl number, ρ is the Rayleigh number, and β is a geometric factor related to the shape of the fluid layer. The dot represents a time derivative.

The Lorenz system exhibits a variety of behaviors as the parameters σ, ρ, and β vary, including fixed points, limit cycles, and chaos. For certain parameter values, the system has chaotic solutions. The most famous example of this is the case where $\sigma = 10$, $\rho = 28$, and $\beta = 8/3$. This set of parameter values gives rise to the Lorenz attractor, a fractal structure that is the path traced by the system in the phase space.

To simulate the Lorenz system, we use a special technique of approximation for differential equations called the Runge-Kutta methods. The basic idea is to compute the derivative at several points within a time step and then combine these derivatives in a weighted average to estimate the state of the system at the next time step. This approach provides a more accurate prediction than simply extrapolating the derivative at the beginning of the time step, as done in the simpler first-order Euler method.

For this problem, you do not have to write much code. We have written most of the needed functions. All you have to do is paste them in the Google Colab.

- `system(state, t)`: This function represents the Lorenz system. It takes in a state vector $[x, y, z]$ and a time t, and outputs the derivatives $[\dot{x}, \dot{y}, \dot{z}]$.
- `rk4_step(state, t, dt, system)`: This function implements the fourth-order Runge-Kutta (RK4) method for numerical integration. It takes in a state vector $[x, y, z]$, a time t, a time step $\mathrm{d}t$, and a system function. It outputs the state at time $t + \mathrm{d}t$.
- `rk2_step(state, t, dt, system)`: Similar to `rk4_step`, this function implements the second-order Runge-Kutta (RK2) method for numerical integration.
- `get_distance(state1, state2)`: This function takes in two state vectors and returns the Cartesian distance between them.

Problem 1

What condition needs to be satisifed for σ, ρ, β in order for the ODE to be stable? (i.e. bounded orbits). Where are the attractors located?

Problem 2

Suppose you change the parameter ρ slightly above its original value, and keep σ and β at their original values. How does the system behavior change? In particular, how does the position of the attractors change, and what happens to the system's stability?

Now suppose you keep ρ at its original value, and change σ and β to values slightly different from their original ones. Again, describe the changes in the system behavior, the position of the attractors, and the system's stability.

To analyze how chaotic a system is, we often use the Lypaunov exponent. Let's consider two trajectories of a dynamical system that start at slightly different initial conditions. We denote the state of the trajectory at time t as $\mathbf{r}(t) = [x(t), y(t), z(t)]$, and the state of the second trajectory as $\mathbf{r}(t) + \delta\mathbf{r}(t)$, where δ is a small perturbation.

The distance between both trajectories can be represented as $d(t) = ||\delta\mathbf{r}(t)||$. If the system is sensitive to initial conditions, then over time, $d(t)$ will grow or shrink at an exponential rate. Therefore, we can rewrite as

$$d(t) \approx d(0)e^{\lambda t}$$

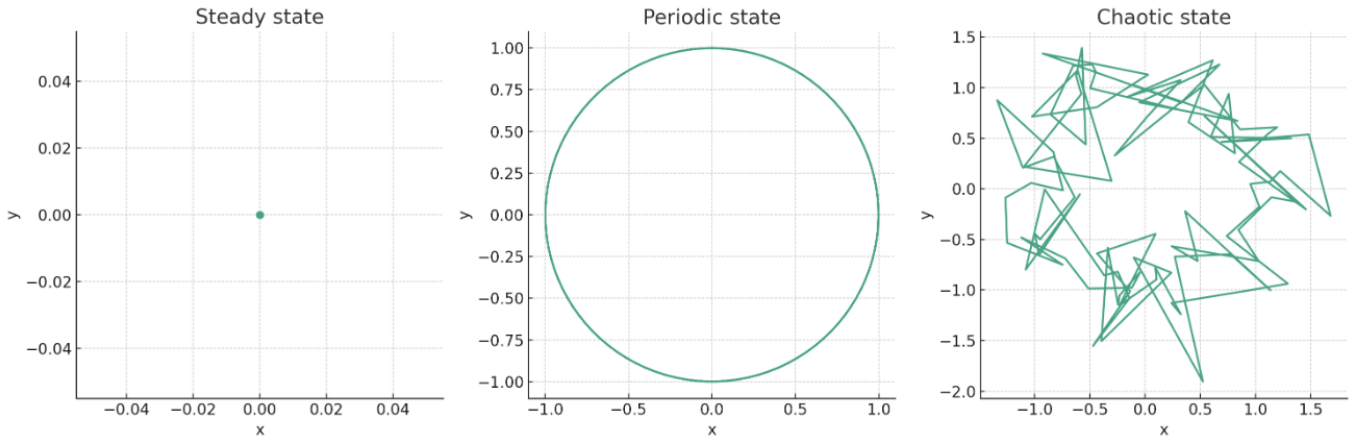
where λ is the Lypaunov exponent. Here, we assume that $d(0)$ is small enough so that an exponential approximation can be taken. Then, by taking natural logarithms, we find that the Lypaunov exponent can be written as

$$\lambda = \lim_{t\to\infty} \frac{1}{t} \ln\left(\frac{d(t)}{d(0)}\right)$$

As $r \in \mathbb{R}^3$ for the Lorenz system, there will be 3 Lypaunov exponents $\{\lambda_1, \lambda_2, \lambda_3\}$ that characterize the system in all 3 directions. We are typically most interested in the Maximum Lypaunov Exponent (MLE) as that tells us a lot about the system itself.

- If the Lypaunov exponent is positive ($\lambda > 0$), the trajectories are diverging on average (as $e^{\lambda t}$ approaches infinity) and the system is chaotic.
- If it is negative ($\lambda < 0$), the trajectories are converging on average (as $e^{-\lambda t}$ approaches 0) and the system is stable.
- If it is zero ($\lambda = 0$), the trajectories neither converge nor diverge on average (as $e^0 = 1$), indicating a neutral or marginally periodic stable system.

Below, we show a plot for an unrelated system and its behavior for corresponding MLE.



Problem 3

Compute the Maximal Lyapunov Exponent (MLE) for a Lorenz system of these parameters:

- $\sigma = 15.6$
- $\rho = 35.4$
- $\beta = 3.13$

What does this imply about the system? See if you can estimate uncertainties!

Problem 4

Call $T_{\max}$ the maximal time for when a simulation is accurate to 99% of reality. A simulation is characterized by the specific numerical solver and the time step $\mathrm{d}t$.

Using `rk4_step`, estimate $T_{\max}$ at various values of $\mathrm{d}t$, given the initial point of (1,1,1). Make a plot.

Then do the same thing with `rk2_step`. What differences do you notice?

Problem 5

Let $\sigma = 10, \rho = 28, \beta = 8/3$. What is the average angular frequency ω_0? Report with uncertainty. How does this change with position?