PHYS 123, 2023, Group Homework:

This homework assignment will be a **group assignment.** Group exercises like this better emulate many aspects of the business or research environments you will find yourselves in in the future; we also believe that it will permit you to do work faster, pool your resources, and tackle tougher computer problems than we could otherwise ask of you. It will certainly **help some of you catch up in computing skills.** It should provide an opportunity for greater learning and creativity on your part, and, if past experience is any guide, more fun.

The groups will consist of three or four students.

A couple of guidelines for group problems:

1. Each group will "solve" the problems together and assist each other in understanding the material.

2. Each group will submit ONE SINGLE set of solutions, or "report." THE SIZE OF THE REPORT SHOULD BE NO DIFFERENT THAN THE HOMEWORK YOU HAVE BEEN HANDING IN INDIVIDUALLY. Don't make too big a deal out of it!

3. You will decide among yourselves how to divide the work. Make sure that everyone is involved in some way in formulating and solving the problem and in producing the HW solutions.

4. To ensure there is a sharing of the responsibility, every group member will indicate that s/he agrees with the group’s answers and WHAT S/HE HAS CONTRIBUTED TO THE WORK BY INDICATING WITH A SHORT SENTENCE WHAT S/HE CONTRIBUTED TO THE WORK AND THEN SIGN THE REPORT!

Your group can use whatever software package you would like. I am recommending Python as a good choice. Python has become more popular here on campus and more generally in the engineering and scientific community. It has the advantage of being free with many different packages available for use with: graphics, numerical integration, matrix calculations, artificial intelligence, etc.

Personally, I like to use Jupyter Notebook to organize my Python Code and save plots. I install Anaconda (www.anaconda.com) and launch the Jupyter Notebook from Anaconda.

HW G-1

Replicate the example program to calculate and plot the x_n for the logistics equation $x_{n+1} = Rx_n(1 - x_n)$

The example program in Jupyter format can be found in LogisticsMapChatGPT.ipynb and a PDF printout in LogisticsMapChatGPT.pdf

HW G-2

Now that you have been able to calculate and plot the points of the populations, find the values of R_n where the logistics equation switches from

- R_1 is when the logistics equation switches from period 1 to period 2
- R_2, period 2 to period 4
- R_3, period 4 to period 8
- R_4, period 8 to period 16
- R_5, period 16 to period 32

HW G-3

Repeat the exercise in **HW G-2** but instead of using the logistics equation use the "laser"mapping $x_{n+1} = Gx_n(1 - \tanh(x_n))$. Find the values of G_n where the logistics equation switches from

- G_1 is when the logistics equation switches from period 1 to period 2
- G_2, period 2 to period 4
- G_3, period 4 to period 8
- G_4, period 8 to period 16
- G_5, period 16 to period 32

HW G-4

Define the ratio

$$\delta_n = \frac{R_{n-1} - R_{n-2}}{R_n - R_{n-1}}$$

The Feigenbaum constant is defined as $\delta = \lim_{n\to\infty} \delta_n = R_{n-1} - R_{n-1}$ and is $\delta = 4.669201...$

Find the first few terms of δ_n using the results of **HW G-2** and similarly the first few terms of δ_n for the "laser" equation. You should find that they both approach the Feigenbaum constant for increasing values of n. The fact that the Feigenbaum constant appears in both the logistics equation and the "laser" equation is an example of the universality of the Feigenbaum constant in chaotic systems.

HW G-5

Sensitivity to Initial Conditions
Edward Lorenz (https://en.wikipedia.org/wiki/Edward_Norton_Lorenz) is recognized as a founder of modern chaos theory which he discovered and explored as part of his work on predicting weather. To make a long story short, he found that computer model predictions of the weather were very sensitive the initial conditions that were fed into the computer model. This explains why it is difficult to accurately predict the weather for more than a few days to a week. In this HW you will explore this concept but using the much simpler logistics equation rather than trying to predict the weather.

Consider the logistics equation $x_{n+1} = Rx_n(1-x_n)$ with the growth parameter set to $R = 3.6$. Define a sequence x_{true} to be set of the x_n calculated with $R = 3.6$ and $x_0 = 0.5$ Define a sequence x_{pred} to be set of the x_n calculated with $R = 3.6$ and $x_0 = 0.5 + \varepsilon$. The idea is that even a small error ε in the initial condition will lead to predicted values that vary wildly from the "true" values.

Make a plot that includes both x_{true} and x_{pred} versus the iteration number with

a) The error set to $\varepsilon = 0.05$
b) The error set to $\varepsilon = 0.005$
c) The error set to $\varepsilon = 0.0005$

These plots will demonstrate the idea that predicted values will start to diverge from the true values, but with a smaller error it will take more iterations before the divergence becomes noticeable.

HW G-6

Use the same logistics map, $R = 3.6$, and $x_o = 0.5$ as in the previous HW problem.

Define the iteration number at which the two sequences x_{true} and x_{pred} have diverged to be the first iteration number $n = n_{\text{diverge}}$ that results in an error between the predicted and true value of $\left|x_{\text{n, pred}} - x_{\text{n, true}}\right| > 0.1$

a) Calculate the value of n_{diverge} for a range of errors $0.1 \geq \varepsilon \geq 10^{-8}$ and plot n_{diverge} versus $-\log(\varepsilon)$.

b) Use this plot to estimate, for each factor of 10 reduction in the initial error ε , the additional number of iterations that can be predicted before diverging.

Roughly, the number of additional number of iterations is the same for each factor of 10 reduction in the initial error. Thus, you can begin to see the limits placed on the ability to predict the weather.
