PHYS H304 Homework 2

Petra Mengistu, Collaborators: Emma Martignoni*

(Dated: February 9, 2023)

This is the lab report write-up of the approach and methods used for the second problems set in PHYS H304.

1. INTRODUCTION

This is a summary of the methods and major equations used for this problem set. In this report, we discuss the recursive algorithm we develop to explore the Catalan series. In addition, we show the plots we produced involving polar and Cartesian coordinate translation as well as sunspot data and boxcar smoothing. The major equations used are included where appropriate and a summary of the approach used to develop the code is included as well. All equations are taken from Computational Physics by Mark Newman [1].

2. METHODS

2.1. Exercise 2.13a

The Catalan series is described as the set of numbers following the sequence 1, 1, 2, 5, 14, 132..., with useful applications to representing states of disorder in quantum mechanics. To determine the nth Catalan number in this series, we use the recursive formula given in [1]. From the sequence, we can observe that the first term in the Catalan series (corresponding to $\mathrm{C_0}$) is 1. The next consecutive term is a factor of $\frac{4n-2}{n+1}\cdot$ multiplying the preceding term, C_{n-1}. This relationship is represented in Eq. 2.1.

$$C_n = \begin{cases} 1 & \text{if } n = 0; \\ \frac{4n-2}{n+1} \cdot \mathrm{C_{n-1}} & \text{if } n \geq 1. \end{cases}$$

In Python, we coded this in a similar manner to that of a factorial recursive function, where we defined a catalan function with an if statement to check the value of the index. If it was the first Catalan number, we directed the output to return a value of 1; otherwise, we implemented the second line of the piece-wise function.

2.2. Exercise 3.1

Sunspots are generally regions of decreased magnetic activity on the sun's surface, appearing less luminous than surrounding areas. [2] Using sunspotters, we can determine the number of sunspots at a given time of observation. As part of the data available on [1], there have been measurements of the number of sunspots observed in a given month since January 1749. As this text file contains both the number of months and the number of sunspots observed in each month, we can plot the number of sunspots as a function of time. To do so, we extract the number of months as an individual array that we set as the independent variable and similarly extract the number of observed sunspots as another array set as the dependent variable. We do so using the "np.array" function in Python to copy the first and second columns of the original file as arrays of time (in months) and number of sunspots respectively.

For part 'b', we limit the arrays to only include the first thousand entries by restricting the size of the arrays we extract to range only from the first to the thousandth entry for both the number of months and sunspots.

For part 'c', we implement the box car smoothing method to only plot the running averages of each entry. We take the average number of sunspots ranging over the span of five months prior and after the month of interest. Then, we do the same for the sunspot number of the adjacent month, and for every consecutive month after that. The averages are determined by the expression given in Eq. 1, where r=5 and Y_k refers to the number of sunspots for the given number of months, k.

$$Y_k = \frac{1}{2r}\sum_{m=-r}^{m=r} Y_{k+m} \tag{1}$$

Hence, each array contains ten measurements of the number of sunspots for ten months.

2.3. Exercise 3.2

In this exercise, we focus on developing translations between different coordinate systems and expressions of variables. For part a, we use parametric expressions of variables to generate the well-known deltoid curve. Parametric equations constitute expressing two variables in terms of a common ("parametric") variable. For the specific parametric equations given in [1], shown in Eq. 2 and 3, the resulting plot of y as a function of x yields the deltoid curve.

$$x = 2cos(\theta) + cos(2\theta) \tag{2}$$

*Electronic address: pmengistu@haverford.edu

$$x = 2\sin\theta - \sin 2\theta \tag{3}$$

Assigning the parametric equations to the variables x and y in Python, we use the matplotlib library features to plot y as a function of x as the parametric variable θ ranges from 0 to 2π, in radians.
For the remaining two parts of this exercise, we make transitions between polar and Cartesian coordinate systems, which are particularly useful when a given system has circular symmetry. The Galilean spiral, for instance, is a great indicator of a system with such symmetry. The polar form of this function is given in Eq. 4, where θ ranges from 0 to 10π, in radians.

$$r = \theta^2 \tag{4}$$

As θ increases, r increases with a square proportionality. Since the Cartesian coordinate expression of polar coordinates involve the projection of the radius onto the x and y axes, the increase in radius with increasing theta results in a spiral shape with ever increasing concentric circular shapes. The conversion from polar coordinates to Cartesian coordinates are given in Eq. 5. We substitute Eq. 4 into Eq. 5 to fully determine the range of values of x and y.

$$\begin{aligned} x &= r\cos\theta \\ y &= r\sin\theta. \end{aligned}$$

Similarly, for the last portion of this exercise, we determine the form of the plot for Fey's function, given in Eq. 5, where θ ranges from 0 to 24π radians.

$$r = e^{\cos\theta} - 4\cos 2\theta + \sin^5 \frac{\theta}{12} \tag{5}$$

Substituting the expression for the radius r given in Eq. 5 into Eq. 5, we can determine the plot of Fey's function with Cartesian coordinates expressed in polar form.

3. RESULTS AND CONCLUSION

For the Catalan series exercise, we determined that the 100th Catalan number has a value of $8.965x10^{56}$. As this is a number representing a term in a series, it is unitless. For Exercise 3.2, we determined the plots for a, b, c, in Figures 1, 2, and 3. Figures 3 and 2 show the differences in curves when the polar function "r" varies.

For Exercise 3.1, the results are shown in Figure 4, 5, and 6. We observe a notable difference in the plots between 4 and 5, as there are significantly less data points in the latter. Inspecting the plot in Figure 6 shows that the overplotted average curve shown in orange represents a much smoother fit to the data as it flattens out the sharp peaks appropriately. For further description, see the codes attached in the folders. The outputs are printed in each code file.

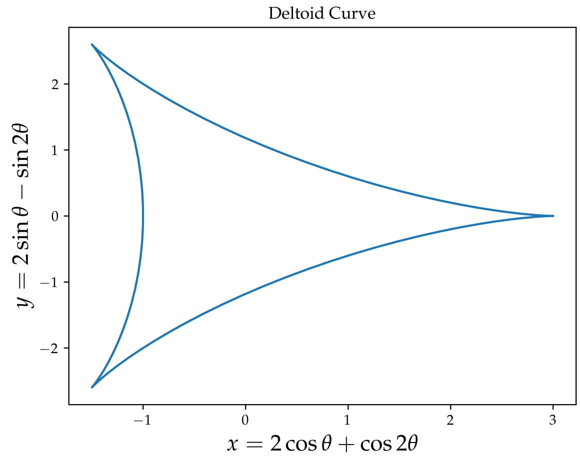

FIG. 1: A plot of the deltoid curve using the appropriate parametric equations for x and y shown on the respective axes.

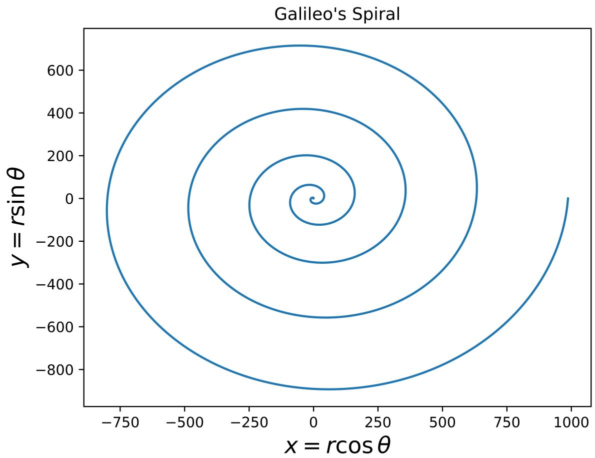

FIG. 2: A plot of Galileo's spiral (r=f(θ)=θ^2) for the polar forms of the Cartesian representations of x and y shown on the respective axes.

3.1. Survey Question

The most interesting problem was 3.1c. The homework took me about 7 hours to complete. I learnt how to strategically code using recursive functions and explore different plotting styles with LaTeX in matplotlib. I think the problem set length and difficulty were just right.

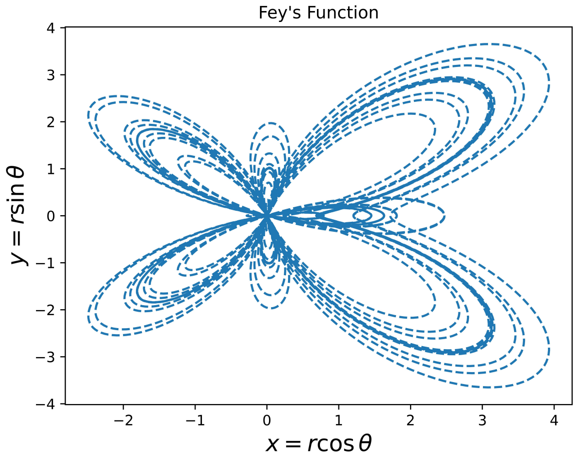

FIG. 3: A plot of Fey's function (r=f(θ)= $e^{\cos\theta} - 4\cos 2\theta + \sin^5\frac{\theta}{12}$) for the polar forms of the Cartesian representations of x and y shown on the respective axes.

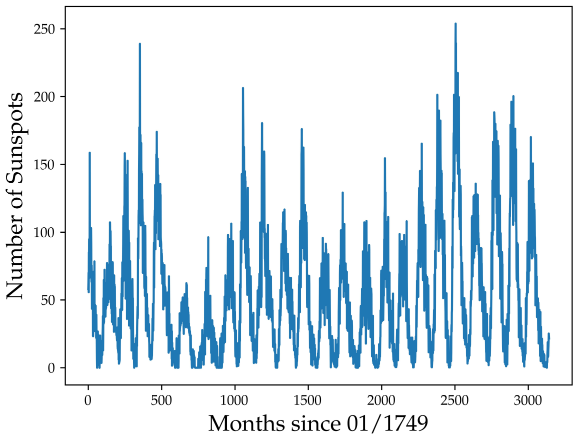

FIG. 4: A plot of the number of sunspots observed as a function of time since January 1749.

[1] M. Newman, *Computational Physics* ([Createspace], 2012), URL http://www-personal.umich.edu/~mejn/cp/index.html.

[2] B. Ryden and B. M. Peterson, *Foundations of Astrophysics* (Cambridge University Press, 2020).

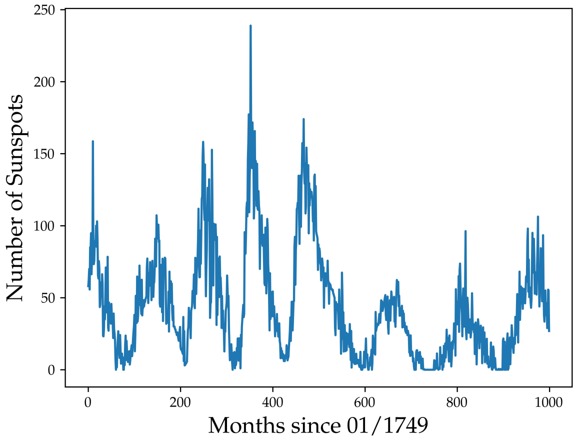

FIG. 5: A plot of the number of sunspots observed as a function of time for the first 1000 months since January 1749.

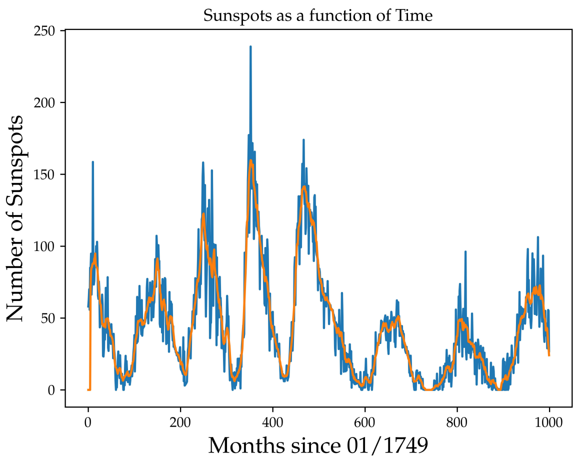

FIG. 6: The superposition of the plot shown in Figure 5 and the running averages of the sunpsots determined using Subsection 2.2 against time