PHYS 304 HW 2

Sophia Lanava*
Bryn Mawr College
(Dated: February 8, 2023)

1. EXERCISE 2.13

Catalan numbers are a series of integers that play an important role in various systems. The equation below 1 is used to calculate the value of the nth term of the series.

$$C_n = \begin{cases} 1 & \text{if } n = 0 \\ \frac{4n-2}{n+1} C_{n-1} & \text{if } n > 0 \end{cases} \quad (1)$$

The goal of this exercise was to create a program that would calculate the 100th Catalan number. Using recursion, I created a program that would output 1 if n=0, and would output the calculation if n¿0. The 100th Catalan number is $8.965 \cdot 10^{56}$.

2. EXERCISE 3.1

The goal of this exercise was to create a plot of the number of sunspots on the Sun as a function of time. Sunspots are small, dark patches on the Sun's surface caused by reduced temperature. I began by importing the data provided by Mark Newman, which was formatted as an array with two columns, the first being time (in months since January 1749) and the second being the number of sunspots. I assigned the first column to the x-axis and the second to the y-axis. After adding labels, I obtained the plot shown in figure 1 1.

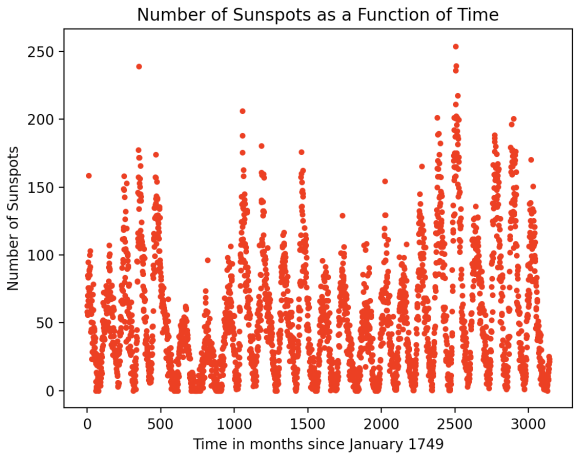

FIG. 1: A graph of sunspots as a function of time.

I adjusted the graph to only show the first 1000 data points 2 by changing the limits of the x-axis.

Finally, I added a line showing the running average by utilizing the given equation 2, with an r value of 5. This graph 3 was also restricted to the first 1000 data points.

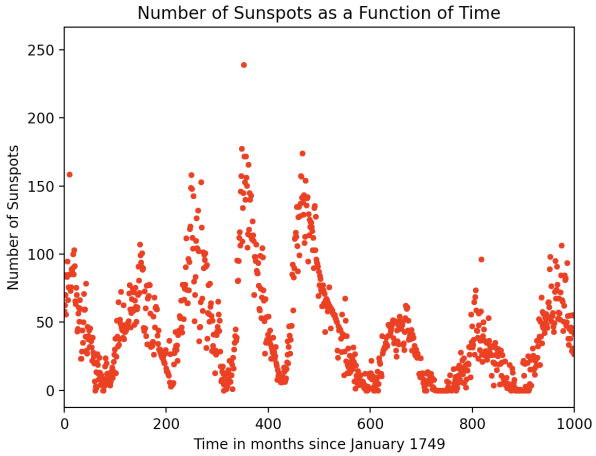

FIG. 2: A graph of sunspots as a function of time, adjusted to only show the first 1000 data points.

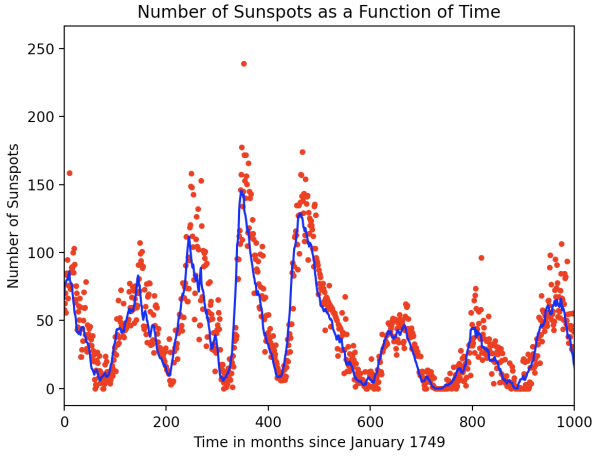

FIG. 3: A graph of sunspots as a function of time, adjusted to only show the first 1000 data points. The running average is shown in blue.

*Electronic address: slanava@brynmawr.edu

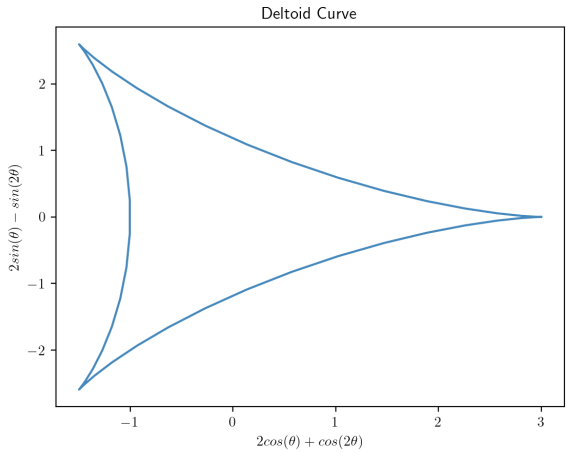

FIG. 4: The deltoid curve.

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

3. EXERCISE 3.2

The goal of this exercise was to create a plot of the deltoid curve. I defined the x and y axes using equations 3 and 4, and I restricted the value of θ to all values between 0 and 2π. This resulted in the plot shown in figure 4

Can use /begin{align} eq 1 // eq 2 /end{align}

$$x = 2\cos\theta + \cos 2\theta \tag{3}$$

$$y = 2\sin\theta - \sin 2\theta \tag{4}$$

use /cos /sin

I used a similar method to create a plot of the Galilean spiral 5. Before plotting, I had to convert θ and r into Cartesian coordinates. The equations used here were $r = \theta^2$, $x = rcos\theta$, and $y = rsin\theta$. θ is limited to the values between 0 and 10π. For this plot, and the following plot, I utilized LaTeX formatting so that θ would be legible in the labels.

Finally, I used this same method yet again to create a plot of Fey's function. The radius component of the polar coordinates is given by equation 5. θ is limited between 0 and 24π. The graph of this function came out strangely, and I am unsure as to why.

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

when you use np.linspace, default gives 50 values in the range, plot looks better if you use linspace(0, 24*pi, 1000) so you have 1000 values in the range.

In the future you can import numpy as np and then write np.loadtext, np.pi, np.array, etc. instead of having to import each seperately.

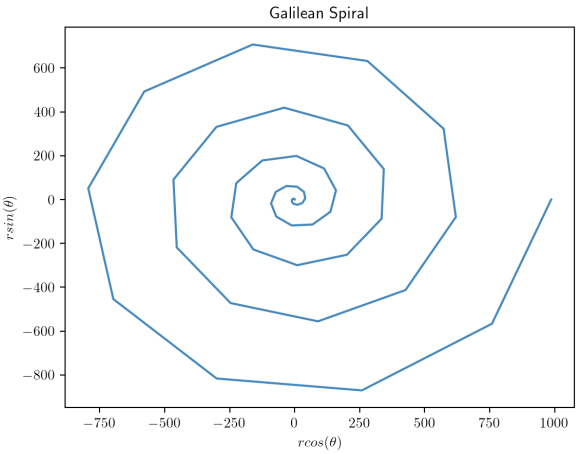

FIG. 5: The Galilean spiral.

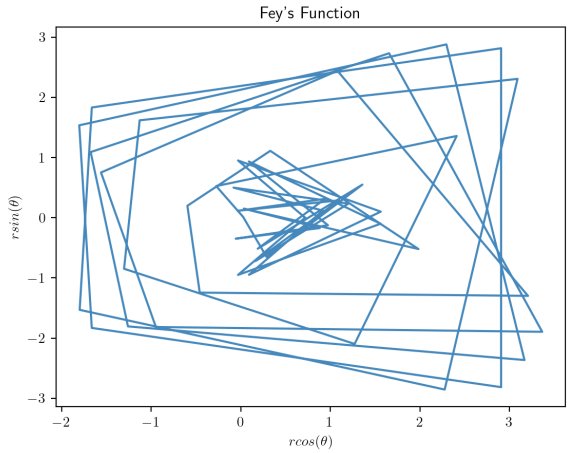

FIG. 6: Fey's function.

2.13) catalan numbers via recursion

- n = 100
- define function Cn
  - check value of n
    - if n=0 → outcome
    - if n>0 → outcome
- print value of Cn

3.1) plotting experimental data

a) - import needed packages
- load data sunspots.txt
- define axes
  - x = first column (column 0), time
  - y = second column (column 1), # sunspots
- plot (adjust markers/color)
- add labels + title

b) same as part a, but add a limit to the x axis

c) same as above, but add
- r = 5
- define range of values
- calculate avg
- plot avg

3.2) curve plotting

a) - define the range of θ values
- calculate x and y
- define axes
- label axes + title
- plot

b) - import needed packages
- define x, y as empty functions
- define range of θ values
  - for these values, define r, x, y
- plot (x, y)

c) same as part b, but change r and θ range

Computational Physics/Astrophysics, Winter 2023:

Grading Rubrics [1]

Haverford College, Prof. Daniel Grin

For coding assignments, roughly 25 points will be available per problem.

1. Does the program complete without crashing in a reasonable time frame? If yes, up to +3 points. 3

2. Does the program use the exact program files given (if given), and produce an answer in the specified format? If yes, +1 points 1

3. Does the code follow the problem specifications (i.e numerical method; output requested etc.) Up to +2 points 2

4. Is the answer correct? Up to+4points 4

5. Is the code readable? Up to+2points 2

. 5.1. Are variables named reasonably?

. 5.2. Are the user-functions and imports used?

. 5.3. Are units explained (if necessary)?

. 5.4. Are algorithms found on the internet/book/etc. properly attributed?

[1] Inspired by rubric of D. Narayanan, U. Florida, and C. Cooksey, U. Hawaii

6. Is the code well documented? +3points 2

. 6.1. Is the code author named?

. 6.2. Are the functions described and ambiguous variables defined?

. 6.3. Is the code functionality (i.e. can I run it easily enough?) documented? -1 no documentation

7. LaTeX writeup (up to 10 points)
   - Are key figures and numbers from the problem given? (3 points) 3
   - Is a brief explanation of physical context given? (2 points) 2
   - If relevant, are helpful analytic scalings or known solutions given? (1 point) 1
   - Are 3-4 key equations listed (preferably the ones solved in the programming assignment) and algorithms named? (2 points) 2
   - Are collaborators clearly acknowledged? (1 point) 1
   - Are any outside references appropriately cited? (1 point) 1

Note, even if (1), (2), (3), or (4) are not correct, one can still obtain many points via (5), (6), and (7).

$\frac{24}{25}$

3.1

Computational Physics/Astrophysics, Winter 2023:

Grading Rubrics [1]

Haverford College, Prof. Daniel Grin

For coding assignments, roughly 25 points will be available per problem.

1. Does the program complete without crashing in a reasonable time frame? If yes, up to +3 points. 3

2. Does the program use the exact program files given (if given), and produce an answer in the specified format? If yes, +1 points 1

3. Does the code follow the problem specifications (i.e numerical method; output requested etc.) Up to +2 points 2

4. Is the answer correct? Up to+4points 4

5. Is the code readable? Up to+2points 2

. 5.1. Are variables named reasonably?

. 5.2. Are the user-functions and imports used?

. 5.3. Are units explained (if necessary)?

. 5.4. Are algorithms found on the internet/book/etc. properly attributed?

[1] Inspired by rubric of D. Narayanan, U. Florida, and C. Cooksey, U. Hawaii

6. Is the code well documented? +3points 3

. 6.1. Is the code author named?

. 6.2. Are the functions described and ambiguous variables defined?

. 6.3. Is the code functionality (i.e. can I run it easily enough?) documented?

7. LaTeX writeup (up to 10 points) 10

Great explanation of context

- Are key figures and numbers from the problem given? (3 points)
- Is a brief explanation of physical context given? (2 points)
- If relevant, are helpful analytic scalings or known solutions given? (1 point)
- Are 3-4 key equations listed (preferably the ones solved in the programming assignment) and algorithms named? (2 points)
- Are collaborators clearly acknowledged? (1 point)
- Are any outside references appropriately cited? (1 point)

Note, even if (1), (2), (3), or (4) are not correct, one can still obtain many points via (5), (6), and (7).

$$\frac{25}{25}$$

3.2

Computational Physics/Astrophysics, Winter 2023:

Grading Rubrics [1]

Haverford College, Prof. Daniel Grin

For coding assignments, roughly 25 points will be available per problem.

1. Does the program complete without crashing in a reasonable time frame? If yes, up to +3 points. 3

2. Does the program use the exact program files given (if given), and produce an answer in the specified format? If yes, +1 points 1

3. Does the code follow the problem specifications (i.e numerical method; output requested etc.) Up to +2 points 2

4. Is the answer correct? Up to+4points 3
   -1 for last plot
5. Is the code readable? Up to+2points 2

   . 5.1. Are variables named reasonably?

   . 5.2. Are the user-functions and imports used?

   . 5.3. Are units explained (if necessary)?

   . 5.4. Are algorithms found on the internet/book/etc. properly attributed?

[1] Inspired by rubric of D. Narayanan, U. Florida, and C. Cooksey, U. Hawaii

6. Is the code well documented? +3points 3

. 6.1. Is the code author named?

. 6.2. Are the functions described and ambiguous variables defined?

. 6.3. Is the code functionality (i.e. can I run it easily enough?) documented?

7. LaTeX writeup (up to 10 points) 10

- Are key figures and numbers from the problem given? (3 points)
- Is a brief explanation of physical context given? (2 points)
- If relevant, are helpful analytic scalings or known solutions given? (1 point)
- Are 3-4 key equations listed (preferably the ones solved in the programming assignment) and algorithms named? (2 points)
- Are collaborators clearly acknowledged? (1 point)
- Are any outside references appropriately cited? (1 point)

Note, even if (1), (2), (3), or (4) are not correct, one can still obtain many points via (5), (6), and (7).

$\frac{24}{25}$