remember to only submit the number of problems asked for (choose 2), For now all 3 will be graded and highest score chosen

Homework 2 Write-Up

49.5+52.5=106.5

106.5/117

Petra Budavari*
Haverford College Department of Physics
(Dated: February 16, 2024)

1. EXERCISE 3.1 PLOTTING EXPERIMENTAL DATA

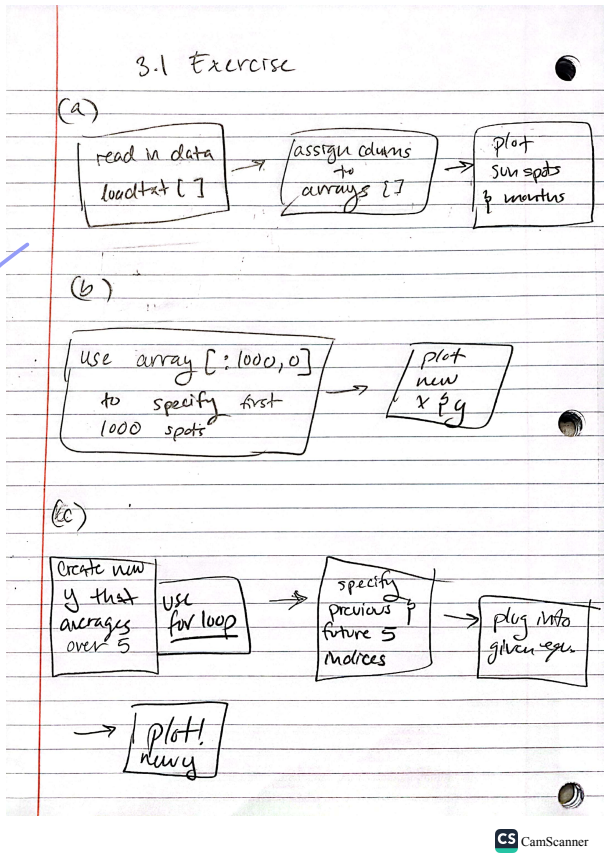

FIG. 1: Flow-chart for Exercise 3.1.

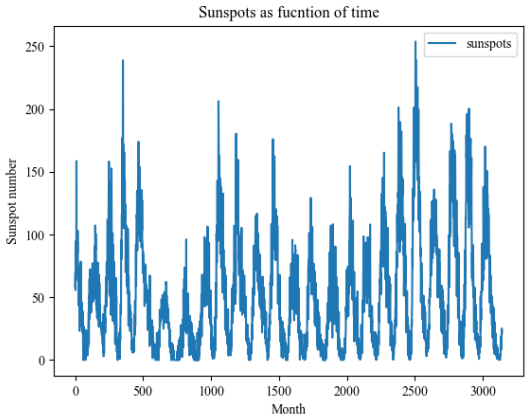

FIG. 2: Number of Sunspots over Months since 1749.

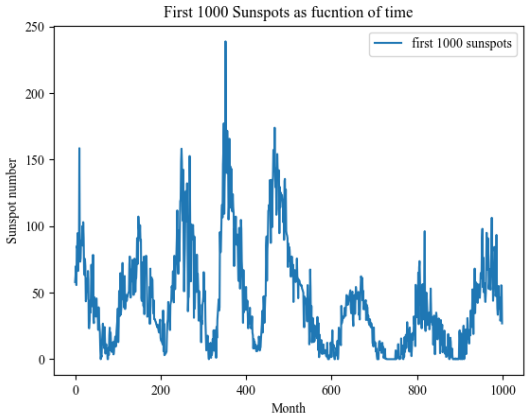

FIG. 3: First 1000 Sunspots over Months since 1749.

This exercise uses data from a file called sunspots.txt. I plotted the number of sunspots on the Sun for each month since Jan. 1749. First, I plotted all of the sunspots 2, then only the ones in the the first thousand months 3. I did this by making an array of the first set of data (the sunspot numbers) and another array for the months, then plotted then as my x and y for Part (a) and (b).

For Part (c), I calculated and plotted the running average where r = 5. I did this by taking the average of the 5 data points of sunspots before and after a certain point defined by an index and using the average to plot that point at the specified index 4. To plot both the running average over the original data I just didn't close the first plot 5. Then again only took the first 1000 points using the array indices 6. To solve the problem of not having enough data for a running average for the first 5 data points, I omitted those first 5 points from the running average calculations.

$$Y_k = (1/(2r+1)) \sum_{m=-r}^{r} y_{k+m} \quad (1)$$

*Electronic address: pbudavari@haverford.edu

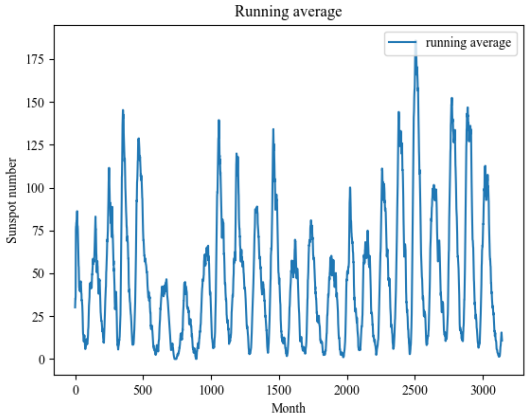

FIG. 4: Running average of sunspots.

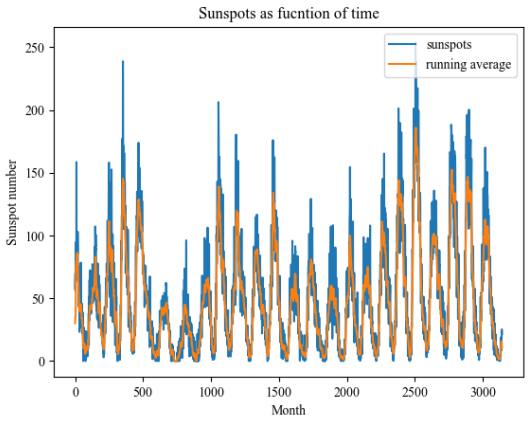

FIG. 5: Running average overlaid on original data.

what does this result mean?

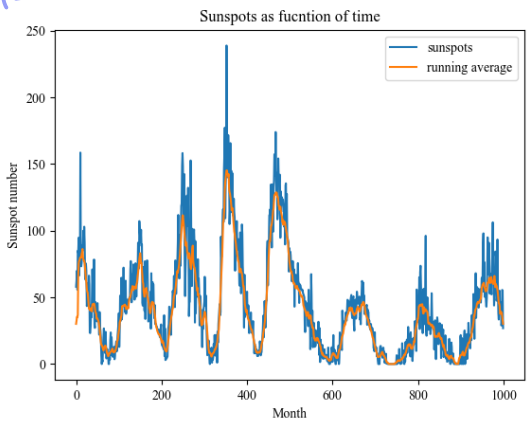

FIG. 6: First 1000 of overlaid data.

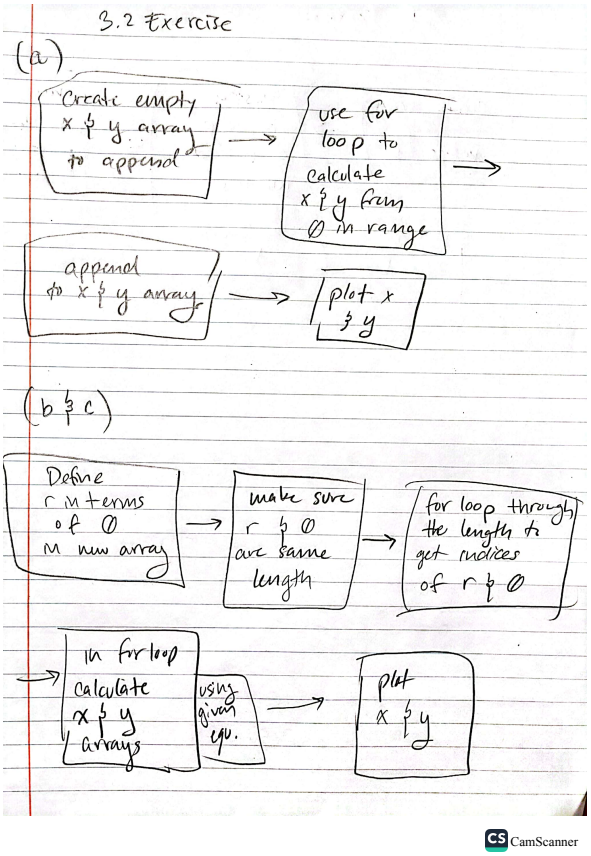

FIG. 7: Flow-chart for Exercise 3.2.

2. EXERCISE 3.2 CURVE PLOTTING

For this exercise, I created a deltoid curve by first creating two empty arrays to which I then appended the outputs of a for loop which calculated x and y from the set ranges of θ and Equations 2 and 3 8.

Figure 8?

$$x = 2cos(\theta) + cos(2\theta) \text{ where } 0 \leq \theta \leq 2\pi \quad (2)$$

$$y = 2sin(\theta) - sin(2\theta) \text{ where } 0 \leq \theta \leq 2\pi \quad (3)$$

For Part (b), I make a plot of a Galilean spiral by converting polar coordinates with the parameters for r defined in Equation (4), to Cartesian coordinates. I did the same thing as in Part (a) with a new set of arrays and Equations (5) and (6) 9.

Figure 9?

$$r = (\theta)^2 \text{ where } 0 \leq \theta \leq 10\pi \quad (4)$$

$$x = rcos(\theta) \quad (5)$$

$$y = 2sin(\theta) \quad (6)$$

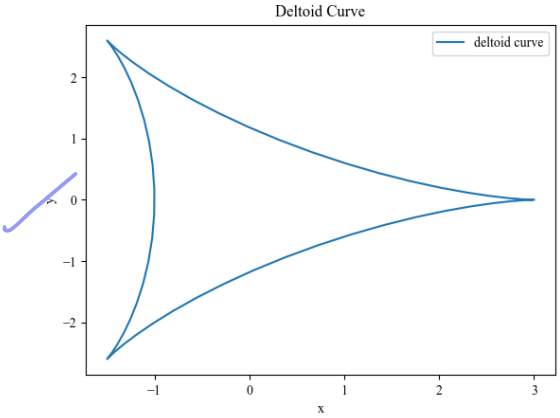

FIG. 8: Deltoid Curve.

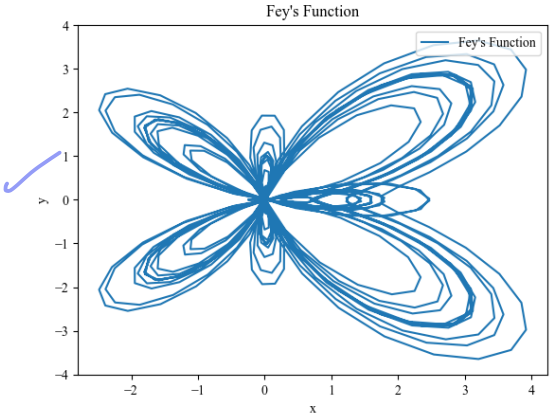

FIG. 10: Fey's Function.

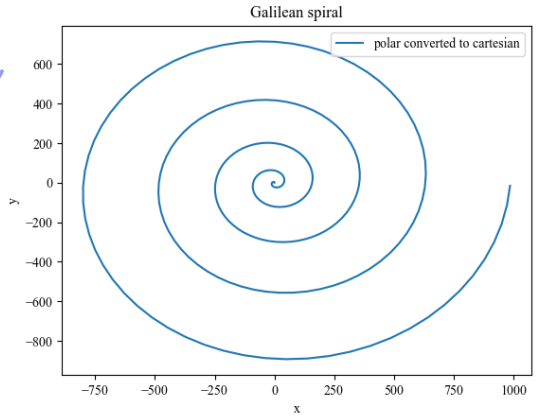

FIG. 9: Galilean Spiral.

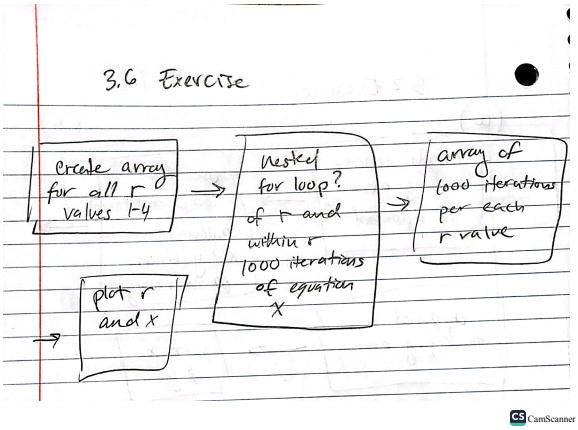

FIG. 11: Flow-chart for Exercise 3.6.

For the last part of this exercise, I used the same method as in Part (b) with Equation (7) defining r, to plot Fey's Function 10.

Figure 10?

$$r = exp(cos(\theta)) - 2cos(4\theta)) + (sin(\theta/12))^5 \text{ where } 0 \leq \theta \leq 24\pi \tag{7}$$

3. EXERCISE 3.6 DETERMINISTIC CHAOS

In this exercise we are asked to plot the Feigenbaum plot 12, an iterative map from Equation (8) to answer some questions about it. I found that from r=1 to approximately r=3, the Feigenbaum plot shows a fixed plot, from r=3 to r=3.5, it settles into a limit cycle, and for r values greater than 3.5 the system moved to chaotic behavior. A fixed point in the iteration can be identified as a line, while a limit cycle jumps between 2 to say 4 values to create orderly bifurcations which look like 'loops'. Chaotic behavior looks random even though it is not.

should just be a single point

$$x' = rx(1 - x) \tag{8}$$

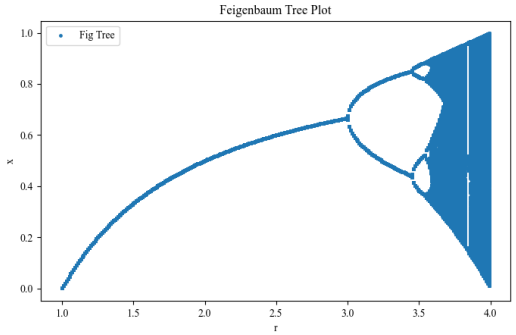

FIG. 12: Feigenbaum Tree Plot.

4. SURVEY QUESTIONS

The homework this week took approximately 10 hours. I learned how to use for loops and how to use matplotlib to create nice(ish) looking plots. I thought the problems were each reasonable and fun, but as someone with less coding experience it took a while.

EX. 3.1 49.5/56

Computational Physics/Astrophysics, Winter 2024:

Grading Rubrics [1]

Haverford College, Prof. Daniel Grin

For coding assignments, roughly 56 points will be available per problem. Partial credit available on all non-1 items.

4 1. Does the program complete without crashing in a reasonable time frame? (+4 points)

[this prevents me from seeing the rest of your plots]

1 2. Does the program use the exact program files given (if given), and produce an answer in the specified format? (+2 points) comment out plt.savefig() before you submit, otherwise code produces an error because that directory does not exist on my computer -1

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

5 4. Is the algorithm appropriate for the problem? If a specific algorithm was requested in the prompt, was it used? (+5 points)

4 5. If relevant, were proper parameters/choices made for a numerically converged answer? (+4 points)

4 6. Is the output answer correct? (+4 points).

3 7. Is the code readable? (+3 points)

. 5.1. Are variables named reasonably?

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

---

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

. 5.3. Are units explained (if necessary)?

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

1.5 8. Is the code well documented? (+3 points)

. 6.1. Is the code author named? please comment your name at the top of your code -1

. 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? some attempt was made, but need more comments describing how your code works -0.5

9. Write-up (up to 28 points)

5. Is the problem-solving approach clearly indicated through a flow-chart, pseudo-code, or other appropriate schematic? (+5 points)

✓. Is a clear, legible LaTeX type-set write up handed in?

1. Are key figures and numbers from the problem given? (+ 3 points) need to explain the problem, eg. define a sunspot, etc. -2

4. Do figures and or tables have captions/legends/units clearly indicated. (+ 4 points)

3. Do figures have a sufficient number of points to infer the claimed/desired trends? (+ 3 points)

0. Is a brief explanation of physical context given? (+2 points) no explanation given, explain the trend we see in your graph -2

1. If relevant, are helpful analytic scalings or known solutions given? (+1 point)

3. Is the algorithm used explicitly stated and justified? (+3 points)

2. When relevant, are numerical errors/convergence justified/shown/explained? (+2 points)

2. Are 3-4 key equations listed (preferably the ones solved in the programming assignment) and algorithms named? (+2 points)
1. Are collaborators clearly acknowledged? (+1 point)
2. Are any outside references appropriately cited? (+2 point)

EX 3.2 49.5/56

Computational Physics/Astrophysics, Winter 2024:

Grading Rubrics [1]

Haverford College, Prof. Daniel Grin

For coding assignments, roughly 56 points will be available per problem. Partial credit available on all non-1 items.

3 1. Does the program complete without crashing in a reasonable time frame? (+4 points)

Don't print out information unless the problem asks for it -1

[this prevents me from seeing the rest of your plots]

1 2. Does the program use the exact program files given (if given), and produce an answer in the specified format? (+2 points) comment out plt.savefig() before you submit, otherwise code produces an error because that directory does not exist on my computer -1

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

Plots don't have titles/axis labels, make sure plt.show() goes at the end -1

5 4. Is the algorithm appropriate for the problem? If a specific algorithm was requested in the prompt, was it used? (+5 points)

4 5. If relevant, were proper parameters/choices made for a numerically converged answer? (+4 points)

4 6. Is the output answer correct? (+4 points).

3 7. Is the code readable? (+3 points)

. 5.1. Are variables named reasonably?

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

---

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

. 5.3. Are units explained (if necessary)?

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

1.5 8. Is the code well documented? (+3 points)

. 6.1. Is the code author named? Please comment your name at the top of your code -1

. 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? Make sure you leave comments on your entire code, even if you repeat something -0.5

9. Write-up (up to 28 points)

5 . Is the problem-solving approach clearly indicated through a flow-chart, pseudo-code, or other appropriate schematic? (+5 points)

✓ . Is a clear, legible LaTeX type-set write up handed in?

3. Are key figures and numbers from the problem given? (+ 3 points)

4. Do figures and or tables have captions/legends/units clearly indicated. (+ 4 points)

3. Do figures have a sufficient number of points to infer the claimed/desired trends? (+ 3 points)

0. Is a brief explanation of physical context given? (+2 points) No explanation given, analyze your final graphs/the form of the eqs. -2

1. If relevant, are helpful analytic scalings or known solutions given? (+1 point)

3. Is the algorithm used explicitly stated and justified? (+3 points)

2. When relevant, are numerical errors/convergence justified/shown/explained? (+2 points)

2. Are 3-4 key equations listed (preferably the ones solved in the programming assignment) and algorithms named? (+2 points)
1. Are collaborators clearly acknowledged? (+1 point)
2. Are any outside references appropriately cited? (+2 point)

EX. 3.6 52/56

Computational Physics/Astrophysics, Winter 2024:

Grading Rubrics [1]

Haverford College, Prof. Daniel Grin

For coding assignments, roughly 56 points will be available per problem. Partial credit available on all non-1 items.

3 1. Does the program complete without crashing in a reasonable time frame? (+4 points)

don't print out information unless the problem asks for it (especially inside a loop!) -1

[this prevents me from seeing the rest of your plots]

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

comment out plt.savefig() before you submit, otherwise code produces an error because that directory does not exist on my computer -1

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

Plots don't have titles/axis labels, make sure plt.show() goes at the end -1

5 4. Is the algorithm appropriate for the problem? If a specific algorithm was requested in the prompt, was it used? (+5 points)

4 5. If relevant, were proper parameters/choices made for a numerically converged answer? (+4 points)

4 6. Is the output answer correct? (+4 points).

3 7. Is the code readable? (+3 points)

. 5.1. Are variables named reasonably?

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

---

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

. 5.3. Are units explained (if necessary)?

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

2 8. Is the code well documented? (+3 points)

. 6.1. Is the code author named? please comment your name at the top of your code -1

. 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?

9. Write-up (up to 28 points)

5 . Is the problem-solving approach clearly indicated through a flow-chart, pseudo-code, or other appropriate schematic? (+5 points)

✓ . Is a clear, legible LaTeX type-set write up handed in?

3 . Are key figures and numbers from the problem given? (+ 3 points)

4 . Do figures and or tables have captions/legends/units clearly indicated. (+ 4 points)

3 . Do figures have a sufficient number of points to infer the claimed/desired trends? (+ 3 points)

2 . Is a brief explanation of physical context given? (+2 points)

1 . If relevant, are helpful analytic scalings or known solutions given? (+1 point)

3 . Is the algorithm used explicitly stated and justified? (+3 points)

2 . When relevant, are numerical errors/convergence justified/shown/explained? (+2 points)

2. Are 3-4 key equations listed (preferably the ones solved in the programming assignment) and algorithms named? (+2 points)
1. Are collaborators clearly acknowledged? (+1 point)
2. Are any outside references appropriately cited? (+2 point)