58/80

Hw 1

Stefany Fabian Dubón*
Bryn Mawr College
(Dated: February 2, 2023)

[Solve exercises 2.2, 2.6, and 2.10]

1. EXERCISE 2.2

+24

Part a)

A satellite is to be launched into a circular orbit around the earth so that it orbits the planet once every T seconds. Show that the altitude h above the earth's surface that the satellite must have is

$$h = (\frac{GMT^2}{4\pi^2})^{1/3} - R, \tag{1}$$

First we define the constants: $G = 6.67e^{-11}m^3kg^{-1}s^{-2}$ is Newton's gravitational constant M = the mass of the earth R = is the radius T = Time (in seconds) it takes for the satellite to complete one orbit. h = the altitude of the satellite above the Earth's surface

So first, we know that the gravitational force between the satellite and the earth is calculated using Newton's law of gravitation:

$$F = \frac{GMm}{R^2} \tag{2}$$

where in this case m is the mass of the satellite, and r is the distance between the center of the earth and the satellite.

We know that the satellite is in circular orbit, which means that the force of gravity is equal to the centripetal force that keeps it in its orbit:

$$F = \frac{mv^2}{r} \tag{3}$$

If we substitute the gravitational force F into the centripetal force equation, we have:

$$\frac{GMm}{R^2} = \frac{mv^2}{r} \tag{4}$$

Then if we solve for v we get:

$$v = (\frac{GM}{r})^{1/2} \tag{5}$$

and since we know that T can be found using the velocity and radius, we get:

$$T = \frac{2\pi r}{(\frac{GM}{r})^{1/2}} \tag{6}$$

Finally we solve for r,

$$r = (\frac{GMT^2}{4\pi^2})^{1/3} \tag{7}$$

since the altitude h can be found by subtracting the radius of the earth R (h=r-R) we get

$$h = (\frac{GMT^2}{4\pi^2})^{1/3} - R, \tag{8}$$

Good derivation

Part b)

The satellite that orbits the earth the less times a day has the highest the altitude. In other words, it means that the fastest satelite is closer to the earth (has lowest altitude)

Part c)

It doesn't make that much of a difference, but the satellite has a lower altitude for the 23.93 hours

24 hours: 35855910.17617497 meters

23.93 hours: 35773762.329895645 meters

2. EXERCISE 2.6

+17

Part a)

Things we know:

perihelion - closest point to the sun:

$$l_1 v_1 \tag{9}$$

aphelion - most distant point from the sun:

$$l_2 v_2 \tag{10}$$

Kepler's second law tells us:

$$l_2 v_2 = l_1 v_1 \tag{11}$$

Total energy, kinetic plus gravitational is given by

$$E = 1/2mv^2 - G\frac{mM}{r} \tag{12}$$

where m is the planet's mass, $M = 1.9891x10^{30}kg$ is the mass of the Sun, and $G = 6.6738x10^{-11}kg^{-1}s^{-2}$

Now, we can substitute $l_2 = l_1$ and $v_2 = v_1$ into the total energy equation:

$$E = \frac{1}{2}mv_2^2 - G\frac{mM}{l_2} = \frac{1}{2}mv_1^2 - G\frac{mM}{l_1} \tag{13}$$

---

*Electronic address: sfabiandub@brynmawr.edu; URL: Optionalhomepage

If we substitute $l_2 = l_1$ into the equation, we are left with:

$$\frac{1}{2}mv_2^2 - G\frac{mM}{l_1} = \frac{1}{2}mv_1^2 - G\frac{mM}{l_1} \tag{14}$$

Finally after rearranging the equation, we are left with the equation that v_2 is the smaller root of the quadratic equation:

$$v_2^2 - \frac{2GM}{v_1 l_1}v_2 - [v_1^2 - \frac{2GM}{l_1}] = 0 \tag{15}$$

Your writeup should continue with a discussion of your results

3. SURVEY QUESTIONS

+5

The homework did take me some time, mainly because I kept getting stuck on small errors on python, like on question 2.10 a, when I kept trying to use an equation, I kept getting an error saying that the "float" object was not callable. I finally figured out after office hours. I learned on how to put complex equations on latex, I had to google how to put subscripts and small stuff like that, but now I feel way more comfortable. The most interesting problem to me was 2.10, because I got to use the if statement, which I found to me kinda confusing sometimes. I do think the homework is kind of long,(I wasn't able to get 2.10.c :( ) even though there are only 3 exercises, the are multiple parts of each and we still have to do the Latex and pseudo code so i do think it might be a little too long.

Problem 2.10 +12

You need to have a writeup for all problems

Problem 2.2

Fabian Dubon +24

Computational Physics/Astrophysics, Winter 2023:

Grading Rubrics [1]

Haverford College, Prof. Daniel Grin

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

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

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

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

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

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

. 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

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

Please put your name at the top of the code

. 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?

+10 7. LaTeX writeup (up to 10 points)

- 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).

Problem 2.6

Fabian Dubon +17

Computational Physics/Astrophysics, Winter 2023:

Grading Rubrics [1]

Haverford College, Prof. Daniel Grin

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

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

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

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

+0 4. Is the answer correct? Up to+4points

The answer is very far off from what it should be. One thing that I can see is that your equation for "c" is off by a factor of -1. There may be other discrepancies as well though

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

. 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

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

Please put your name at the top of your code

- 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 7. LaTeX writeup (up to 10 points)

- 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)

You should be discussing your results in your writeup as well. What did this question teach you about the Earth's orbit or the orbit of Haley's comet?

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

Problem 2.10

Fabian Dubon +12

Computational Physics/Astrophysics, Winter 2023:

Grading Rubrics [1]

Haverford College, Prof. Daniel Grin

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

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

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

+2 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

Parts a and b are right. For part c you are really just calculating B/A for A=3Z-1 instead of finding the highest B/A. I did not see code for part d

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

. 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

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

Please put your name at the top of your code

. 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?

+0 7. LaTeX writeup (up to 10 points)

. 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)

You do not have anything for this problem in your writeup

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

52/80

Hw 2

Stefany Fabian Dubón*
Bryn Mawr College
(Dated: February 10,2020)

[Solve exercises 2.13 a), 3.1, and 3.3]

1. INTRODUCTION

For this homework assignment, I was able to practice recursion to calculate the Catalan numbers C_n. Then, ways to plot experimental data using an online file was introduced. Finally, I was able to learn how to use a grid of values to create a density plot.

2. EXERCISE 3.6 +23

To create a program that uses recursion, which is the ability of a function to call itself to calculate the C_n. I used the definition of Catalan numbers C_n given in the form:

$$C_n = \begin{cases} 1 if n = 0, \begin{cases} \frac{4n-2}{n+1} C_{n-1} if n > 0 \end{cases} \end{cases}$$

(1)

I first define the function that calculates Catalan (n) using and if statement. I started the if statement saying that id n == O, then i would get 1 as the return, else I would used the formula $\frac{4n-2}{n+1}C_{n-1}$ to get the return number. Next, to use the function to calculate and print C_{100} I just call the function and input the number 100.

3. EXERCISE 3.1 +24

For this exercise, I started by getting the sunspots.txt from the on-line resources, which contains the observed number of sunspots on the Sun for each month since January 1749. I then proceeded to write the starting code that reads in the data and makes a graph of sunspots as a function of time. I did this by defining x and ya values that would read in the specific columns in the file. I then also input the value 1000, so that the code would only display up until the 1000th data point on the graph. I then created a for loops to calculate the running average of the data defined by:

$$Y_k = \frac{1}{2r+1} \sum Y_{k+m} \quad (2)$$

Finally, i plotted both the original data, and the running average on the same graph over the range covered by the first 1000 points

You actually did 3.3

4. EXERCISE 3.2 +22.5

For this exercise, I also used a on-line resourced called stm.txt , which contained a grid of values from scanning tunneling microscope measurement of the (111) surface of silicon. For a little background, a scanning tunneling microscope, is used ti measure the shape of a surface at the atomic level. This results in a grid of values that represent the height of the surface. To compute the density plot of the values found in the file, I created a a program that read the file, and using the imshow argument to create the image.

The assignment was to do either 3.1 or 3.3 and in addition to that do 3.2. What I am going to do is give you the higher of the two scores out of 3.1 or 3.3 (which will be the 24 on 3.1) but then you will get no points for 3.2.

*Electronic address: sfabiandub@brynmawr.edu; URL: Optionalhomepage

5. RESULTS

For exercise 3.1, I used an online file, which obtained the observed number if spots on the sun for each month since January 1749. After modifying the data so that it only displays the first 1000 data points on the graph, as well as to calculate and plot the running average of the data, I obtained the following graph:

Great plot!

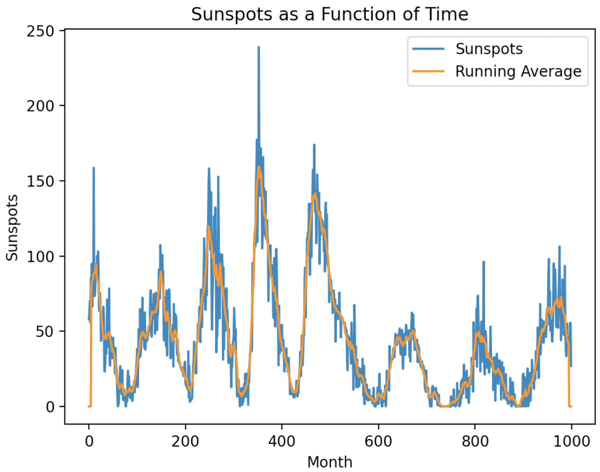

FIG. 1: Sunspots plotted as a function of time

For exercise 3.2, I used a grid of values from scanning tunneling microscope measurements to create a density plot. As a result, I obtained the following image to show the structure of the sillicon surface

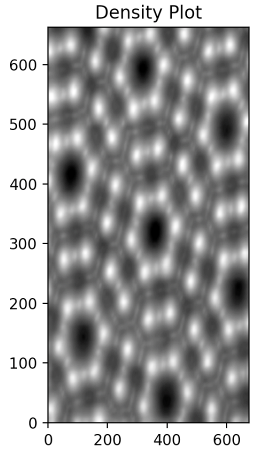

FIG. 2: Image of the Density plot using the values that represent the height of the surface.

+5

6. CONCLUSION

Overall, this homework set did take me some time, since I had some issues with getting my vscode to work with pylab. The problem was that it kept saying that I didn't have the module, and when trying to install it from the terminal, it said that Anaconda didn't have it, in the end I was able to do the graphs using matplotlib. I did learned how to make a legend and how to make a density plot, which even thought I've seen before, I've never really thought deeply on how they are produced.

Problem 2.13

Fabian Dubon +23

Computational Physics/Astrophysics, Winter 2023:

Grading Rubrics [1]

Haverford College, Prof. Daniel Grin

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

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

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

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

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

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

+1 6. Is the code well documented? +3points

. 6.1. Is the code author named?

Please put your name at the top of your code

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

Please include comments to describe your code

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

+10 7. LaTeX writeup (up to 10 points)

. 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).

Problem 3.1

Fabian Dubon +24

Computational Physics/Astrophysics, Winter 2023:

Grading Rubrics [1]

Haverford College, Prof. Daniel Grin

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

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

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

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

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

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

. 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

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

Please put your name at the top of your code

- 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?

+10 7. LaTeX writeup (up to 10 points)

- 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).

Problem 3.3

Fabian Dubon +22.5

Computational Physics/Astrophysics, Winter 2023:

Grading Rubrics [1]

Haverford College, Prof. Daniel Grin

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

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

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

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

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

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

. 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

+1 6. Is the code well documented? +3points

Please put your name at the top of your code

- 6.1. Is the code author named?
- 6.2. Are the functions described and ambiguous variables defined?

Please include comments to describe your code

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

+9.5 7. LaTeX writeup (up to 10 points)

- 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)

Include labels on all axes

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