PHYS304 HW2

Intouch Srijumnnong (Indy), worked with JT
Haverford College
(Dated: February 15, 2023)

1. EXERCISE 2.13A: RECURSION

In this problem, we are asked to write a Python function, using recursion, that calculates the Catalan numbers C_n, and use it to calculate and print C_{100}. The definition given in the problem can be written in the form:

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

Using the conditional Catalan equation above, I write the program using recursion for the function to call and repeat itself with the condition given with if and else function, and then calculate the term. For the C_{100}, the number calculated is 8.965199470901317e+56.

2. EXERCISE 3.3: DENSITY PLOTS OF EXPERIMENTAL DATA,

We are asked to write a program that reads the data contained in the file stm.txt, which contains a grid of values of the height of the surface from scanning tunneling microscope (STM) measurements of the (111) surface of silicon. I generate a density plot of the values using gray scale running from black to white which I think best illustrate the structure of the silicon surface, as shown in the image (1) below.

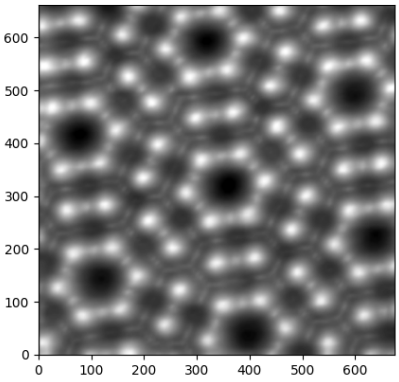

FIG. 1: The density plot of the grid values of the surface of silicon from stm.txt

3. EXERCISE 3.2: CURVE PLOTTING

In this problem, we are asked to make various kinds of adapted plotting. I use the function linspace from numpy to return evenly spaced numbers over a specified interval. This is similar to arange function but instead of step it uses sample number. Linspace is a better option here since we know the exact values for the start and end points of the interval.

a) We are asked to make a plot of the deltoid curve, which is defined by the equations:

$$x = 2cos\theta + cos2\theta,\ y = 2sin\theta - sin2\theta \tag{2}$$

where $0 \leq \theta < 2\pi$. I generate the plot from x and y for each from the equations above, then plot y as a function of x, taken 1,000 values of θ between 0 and 2π, resulted in the image (2) below.

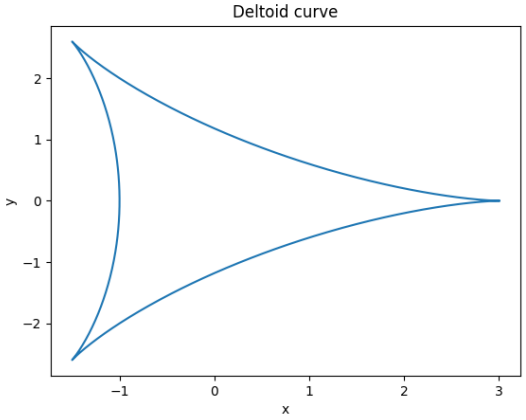

FIG. 2: The deltoid curve plotted with 1,000 values of θ between 0 and 2π

b) Taking the same approach, we are asked to make a plot of the Galilean spiral using a polar plot $r = f(\theta)$, calculating r for a range of values of θ, then converting r and θ to Cartesian coordinates using the standard equations $x = rcos\theta$, $y = rsin\theta$. Using this method, I generate the Galilean spiral plot taken 1,000 values of θ in $r = \theta^2$ for $0 \leq \theta \leq 10\pi$, as shown in the image (3) below.

c) Using the same method above, I make a polar plot of Fey's function taken 10,000 values instead of 1,000 of θ for the more accurate plot, for the equation:

$$r = e^{cos\theta} - 2cos4\theta + sin^5\frac{\theta}{12} \tag{3}$$

in the range of $0 \leq \theta \leq 24\pi$, as shown in the image (4) below.

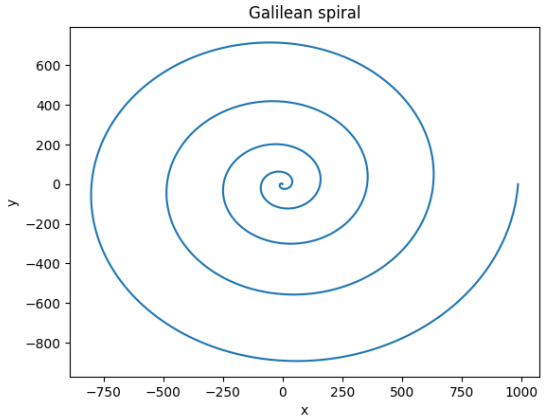

FIG. 3: The Galilean spiral plotted with 1,000 values of θ between 0 and 10π

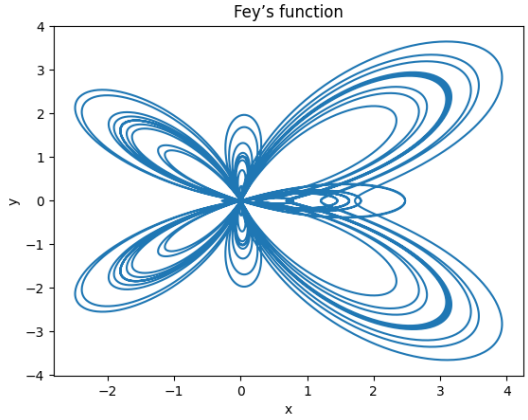

FIG. 4: The Fey's funtion plotted with 10,000 values of θ between 0 and 24π

4. FEEDBACK

I spent about 6-8 hours on this homework including meeting with classmates and attending office hours. Speaking from the perspective that I was doing HW2 and HW3 at the same time, HW2 developed necessary fundamental skills for computational graphics and visualization which is very helpful for HW3 and for further learning materials in the course as well. I think the problems are covered well in the classes and appropriate for the time of the course. I enjoyed doing this homework.

PHY580 HW2

2.13 a) → According to factorial(n):

```
def fact(n):
    if n == 1
        return 1
    else
        return n * fact(n-1)
```

(call it self)

for $n!$ = $\begin{cases} 1 & \text{if } n = 1 \\ n \times (n-1)! & \text{if } n > 1 \end{cases}$

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

```
def C(n): if n == 0
    return 1
else return (4n-2)/(n+1) * C(n-1)
```

$C_{100} = ?$

3.3) → loadtxt from stm.txt

plt.imshow(data)

3.2) a) Deltoid

$x = 2\cos\theta + \cos 2\theta \quad , \quad y = 2\sin\theta - \sin\theta$

$0 \le \theta < 2\pi$

→ linspace or range ? → step

nice thought!

↳ return in sample numbers evenly spaced

→ for θ → linspace $(0, 2\pi)$

→ plt.plot (x, y)

→ axis

→ label

b) Oddities

→ polar $r = \theta^2 \quad , \quad 0 \le \theta \le 10\pi$

for $x = r\cos\theta \; , \; y = r\sin\theta$

θ = linspace $(0, 10\pi)$

→ plt.plot

c) Fey's

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

$0 \le \theta \le 24\pi$

for $x = r\cos\theta \; , \; y = r\sin\theta$

θ = linspace $(0, 24\pi)$

→ plt.plot

2.13

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? -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?
I like how you use an example in describing your code

7. LaTeX writeup (up to 10 points) 8
   - Are key figures and numbers from the problem given? (3 points)
   - Is a brief explanation of physical context given? (2 points) -2 no context about what Catalan #s are
   - 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).

22/25

3.3

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 1

. 6.1. Is the code author named? −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? −1 no documentation

7. LaTeX writeup (up to 10 points) 9

- 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) plot is missing axis labels −1

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

22/25

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 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

. 6.1. Is the code author named? -1
always include name!

. 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).

$\frac{24}{25}$