47+46+5=98

98/117

PHYS 304 HW2 Xiyue Shen

Xiyue Shen

February 2024

Required (graded) Exercises
All Python scripts are located in the same folder. Check the script for specific lines.

HW1

1. **Exercise 2.2: Altitude of a satellite**

   (a) For a satellite with mass m orbiting around the Earth with mass M, the motion can be described as a circular motion. The centripetal force is,

$$F_c = \frac{mv^2}{r} \tag{1}$$

where v is the tangential velocity, and r is the radius. We can write r as $R+h$, where R is the earth's radius, and h is the satellite altitude. The source of the centripetal force comes from the gravity of the earth. As Newton described, the gravitational force is,

$$F_g = G\frac{Mm}{r^2} \tag{2}$$

where the r is the same distance in equation 1.G is the gravitational constant, $6.674 \times 10^{-11} m^3 kg^{-1} s^{-2}$.
Let $F_g = F_c$, then we have,

$$G\frac{Mm}{r^2} = \frac{mv^2}{r}$$
$$v = \sqrt{\frac{GM}{r}} \tag{3}$$

We have $v = \frac{2\pi r}{T}$ as defined. Then, we can substitute the velocity term in equation 3 so that we can derive a relation in between period and altitude.

$$
\begin{aligned}
\frac{2\pi r}{T} &= \sqrt{\frac{GM}{r}} \\
r &= (\frac{GMT^2}{4\pi^2})^{1/3} \\
h + R &= (\frac{GMT^2}{4\pi^2})^{1/3} \\
h &= (\frac{GMT^2}{4\pi^2})^{1/3} - R \qquad (4)
\end{aligned}
$$

From the second line, r was substituted by $h + R$. Equation 4 is what's given in part a.

(b) Figure 1 shows my script. Here, I import all the necessary packages. Then, I assign several constants. "height" is the equation part a gives. Through "rcParams", we import LaTex font for plotting.

```
#Exercise 2.2 part b
import matplotlib.pyplot as plt
import numpy as np
plt.rcParams['text.usetex'] = True
plt.rcParams['text.latex.preamble'] = r'\usepackage{bm}'
plt.rcParams['pgf.texsystem'] = 'pdflatex'  # or 'latex'

T=input("what period T you would like to?")
T=float(T)
G=6.67e-11
M=5.97e24
R=6.371e6
pi=np.pi
height=(G*M*T**2/(4*pi**2))**(1/3)-R
print("the altitude of the satellite is", height)
```

Figure 1: Programming for calculating the height given a period

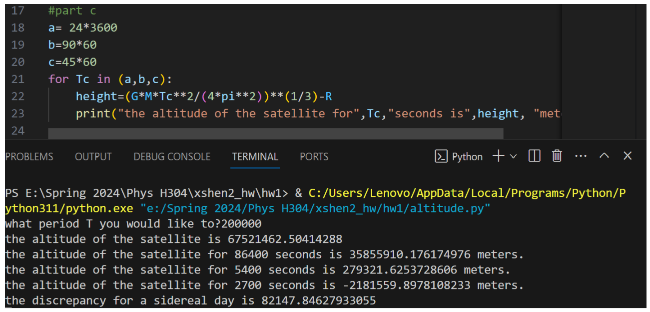
(c) For one day (86400 seconds), the altitude is 35855910.176174976 meters; for 90 minutes (5400 seconds), the altitude is 279321.6253728606 meters; for 45 minutes (2700 seconds), the altitude is -2181559.8978108233 meters, as indicated by figure 2.

(d) For a sidereal day, as the code indicates, I calculated the heights for $Ta = 24$ hours and $Tb = 23.93$ hours. Then, I subtracted the heights ha and hb to get the difference. The discrepancy is 82147.84627933055 meters, as shown by figure 3

2. **Exercise 2.5: Quantum potential step**

HW1

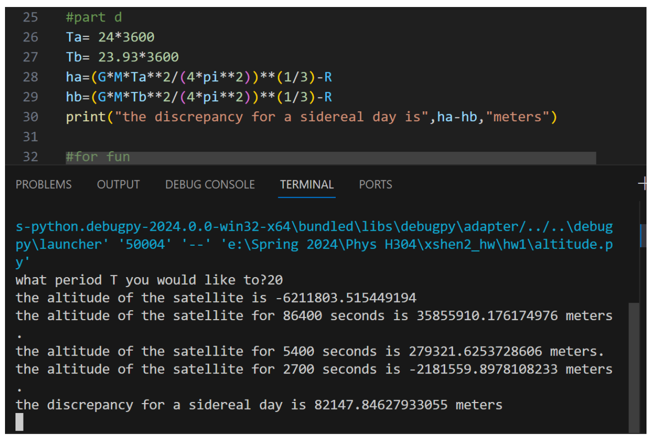
Firstly, I input several parameters used as constants, such as electron mass m, Planck constant h, and the joules and electron-volt converting constant j.



```
#part c
a= 24*3600
b=90*60
c=45*60
for Tc in (a,b,c):
    height=(G*M*Tc**2/(4*pi**2))**(1/3)-R
    print("the altitude of the satellite for",Tc,"seconds is",height, "met
```

```
PS E:\Spring 2024\Phys H304\xshen2_hw\hw1> & C:/Users/Lenovo/AppData/Local/Programs/Python/P
ython311/python.exe "e:/Spring 2024/Phys H304/xshen2_hw/hw1/altitude.py"
what period T you would like to?200000
the altitude of the satellite is 67521462.50414288
the altitude of the satellite for 86400 seconds is 35855910.176174976 meters.
the altitude of the satellite for 5400 seconds is 279321.6253728606 meters.
the altitude of the satellite for 2700 seconds is -2181559.8978108233 meters.
the discrepancy for a sidereal day is 82147.84627933055
```

Figure 2: Attached is my code and result after running.



```
#part d
Ta= 24*3600
Tb= 23.93*3600
ha=(G*M*Ta**2/(4*pi**2))**(1/3)-R
hb=(G*M*Tb**2/(4*pi**2))**(1/3)-R
print("the discrepancy for a sidereal day is",ha-hb,"meters")

#for fun
```

```
s-python.debugpy-2024.0.0-win32-x64\bundled\libs\debugpy\adapter/../..\debug
py\launcher' '50004' '--' 'e:\Spring 2024\Phys H304\xshen2_hw\hw1\altitude.p
y'
what period T you would like to?20
the altitude of the satellite is -6211803.515449194
the altitude of the satellite for 86400 seconds is 35855910.176174976 meters
.
the altitude of the satellite for 5400 seconds is 279321.6253728606 meters.
the altitude of the satellite for 2700 seconds is -2181559.8978108233 meters
.
the discrepancy for a sidereal day is 82147.84627933055 meters
```

Figure 3: Height difference of a sidereal day.

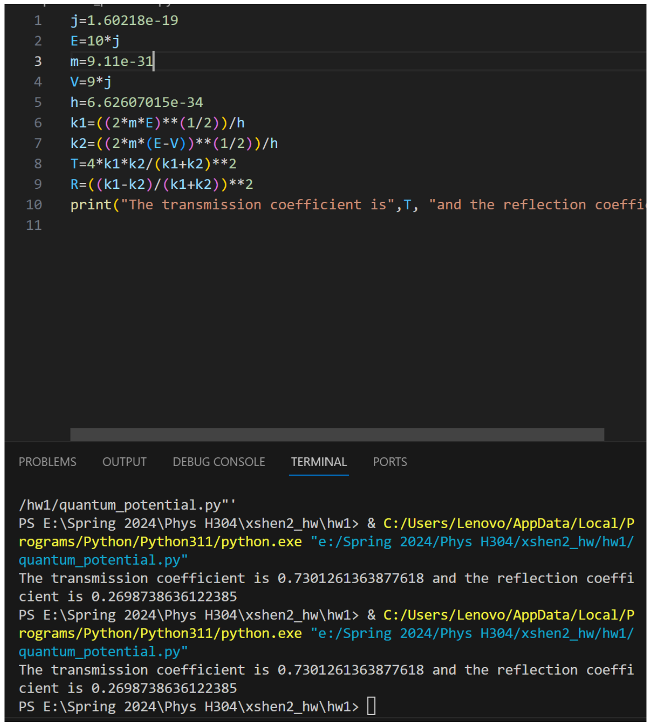
Then, I assign values to energy E and potential V. k_1 and k_2 are equations given for wavevectors. T and R are the probability for transmission and reflection.

After all equations and constants are set, I print out the probabilities. For transmission probability, I get 0.730; for reflection probability, I get 0.270 as shown in figure 4.

3. **Exercise 3.3: STM density plot**

   Coding structure: Firstly, I import all packages that I need, such as "numpy", "matplotlib", and some LaTex rendering. After downloading the data and saving it in the same folder as my python script, I use "loadtxt" to read and get the data. Then, I use inshow to plot the data. I got three different plots with gray, rainbow, and Viridis color bars. To make the color contrast more obvious, I set vmin and vmax values, which employ the extremes in the data to make the color distribution.

   Figure 5d is a $3D$ plot, which shows the height distribution more obviously.



```
j=1.60218e-19
E=10*j
m=9.11e-31
V=9*j
h=6.62607015e-34
k1=((2*m*E)**(1/2))/h
k2=((2*m*(E-V))**(1/2))/h
T=4*k1*k2/(k1+k2)**2
R=((k1-k2)/(k1+k2))**2
print("The transmission coefficient is",T, "and the reflection coeffi
```

```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

/hw1/quantum_potential.py"'
PS E:\Spring 2024\Phys H304\xshen2_hw\hw1> & C:/Users/Lenovo/AppData/Local/P
rograms/Python/Python311/python.exe "e:/Spring 2024/Phys H304/xshen2_hw/hw1/
quantum_potential.py"
The transmission coefficient is 0.7301261363877618 and the reflection coeffi
cient is 0.2698738636122385
PS E:\Spring 2024\Phys H304\xshen2_hw\hw1> & C:/Users/Lenovo/AppData/Local/P
rograms/Python/Python311/python.exe "e:/Spring 2024/Phys H304/xshen2_hw/hw1/
quantum_potential.py"
The transmission coefficient is 0.7301261363877618 and the reflection coeffi
cient is 0.2698738636122385
PS E:\Spring 2024\Phys H304\xshen2_hw\hw1>
```

Figure 4: Quantum potential probability code

you do a really good job explaining your code, next time explain the problem/ your results too

4. **Exercise 3.2: Curve plotting**

   I assigned θ with evenly distributed values using "linspace". Then, set x and y as functions of θ given in the problem.

   (a) We have $x = 2\cos(\theta) + \cos(2\theta)$ and $y = 2\sin(\theta)\sin(2\theta)$

   (b) We have $r = \theta^2$; then, $x = \theta^2\cos(\theta)$ and $y = \theta^2\sin(\theta)$. In the coding, I used ϕ instead of θ to avoid confusion with part a.

   (c) We have $r = e^{cos\theta} - 2\cos(4\theta) + \sin^5(\frac{\theta}{12})$; then, $x = [e^{cos\theta} - 2\cos(4\theta) + \sin^5(\frac{\theta}{12})]\cos(\theta)$ and $y = [e^{cos\theta} - 2\cos(4\theta) + \sin^5(\frac{\theta}{12})]\sin(\theta)$. I used α instead of θ to avoid confusion with part a in the coding.

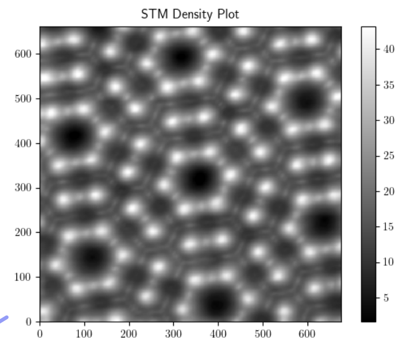
psuedocode?

Survey Questions

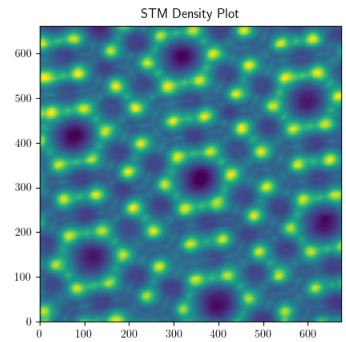
I spent roughly 3 hours on this week's homework, but I spent another 3 hours figuring out how to apply LaTex rendering and adding the path to the terminal. I learned to make plots, define equations, arrange, linspace, etc. I think the problem set is about the right length.

+5

Required (ungraded) work

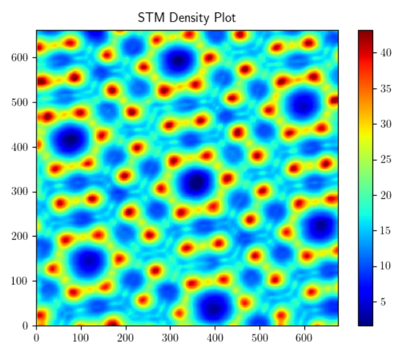


(a) STM plot in gray colorbar

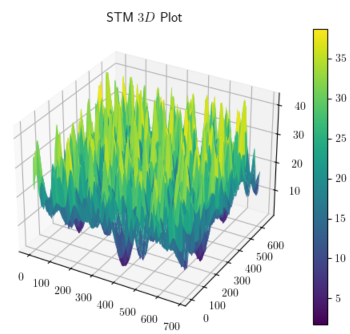


(b) STM plot in Viridis colorbar

Nice Plots!



(c) STM plot in Rainbow colorbar



(d) 3D STM Plot

Figure 5: STM Density Plot

- I have downloaded the repository
- I followed along with class notes and understood every code
- I tried some examples.

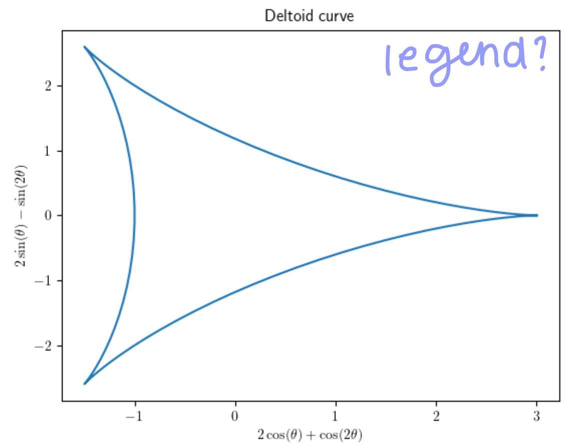


Figure 6: Deltoid curve

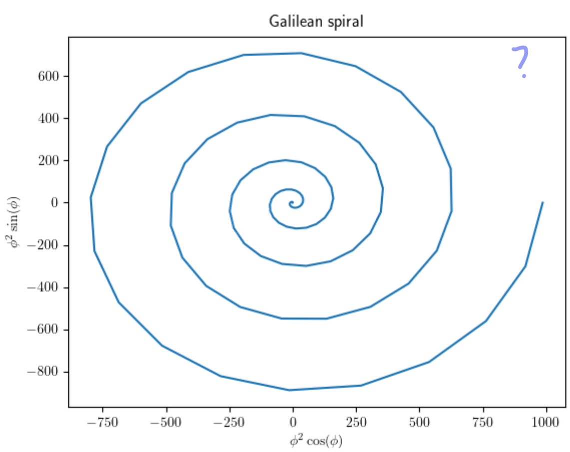


Figure 7: Galilean spiral

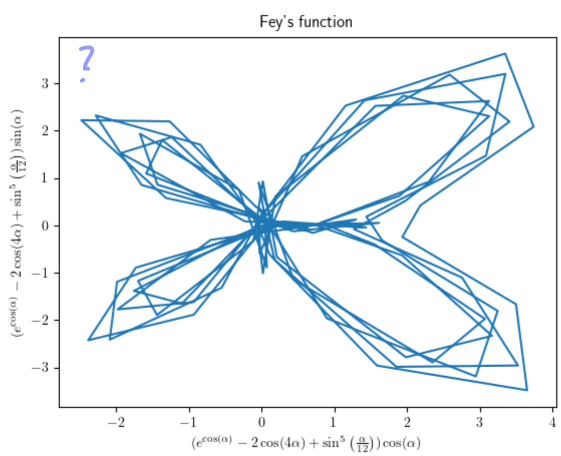


Figure 8: Fey's function

Ex. 3.3 47/56

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)

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

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 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? Need more comments explaining how your code works -1

9. Write-up (up to 28 points)

0 . Is the problem-solving approach clearly indicated through a flow-chart, pseudo-code, or other appropriate schematic? (+5 points) please submit psuedocode with your write-up -5

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

1 . Are key figures and numbers from the problem given? (+ 3 points) Explain the problem more, -2 e.g. what is STM?

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)

Ex. 3.2 46/56

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)

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

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?

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)

0 . Is the problem-solving approach clearly indicated through a flow-chart, pseudo-code, or other appropriate schematic? (+5 points) Please include psuedocode with your write-up -5

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

1 . Are key figures and numbers from the problem given? (+ 3 points) Explain the problem more, e.g. what kinds of graphs are you making? -2

3 . Do figures and or tables have captions/legends/units clearly indicated. (+ 4 points) Figures need legends -1

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) some explation is there, need more of a description of your figures and eqs. -1

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)