

Attached written notes are my pseudo code for the problems.

8.10

a) $\frac{d^2x}{dt^2} = \frac{dv_x}{dt} = -GM\frac{x}{r^3}$

$$\frac{d^2y}{dt^2} = \frac{dv_y}{dt} = -GM\frac{y}{r^3}$$

$$\begin{aligned}\frac{dx}{dt} &= v_x \\ \frac{dv_x}{dt} &= -GM\frac{x}{r^3} \\ \frac{dy}{dt} &= v_y \\ \frac{dv_y}{dt} &= -GM\frac{y}{r^3}\end{aligned}$$

b). Use 4th order Runge-Kutta method
pseudo code :

```
import everything
G = 6.67 E-11
m = 1.989 E 30.
def f(r,t):
    x = r[0]
    vx = r[1]
    y = r[2]
    vy = r[3]
    fx = vx
    fy = vy
    fx = -GMx/r^3
    fy = -GMy/r^3
    return ( )
```

```
a = 0
b =
N =
h = (b-a)/N
```

```
tpoints = np.arange(a, b, h)
x_values = []
y = []
vx = []
vy = []
```

```
r = np.array([4e12, 0, 0, 500], float)
```

```
for t in tpoints:
    x_values.append(r[0])
    y ...
    vx ...
    vy ...
    k1 =
    k2 =
```

$$k_3 =$$

$$k_4 =$$

$$r_t = \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4).$$

plt.plot().

c). We change the "else" statement in the "if" loop and add the r_1 and r_2 into the while loop.

while $t < b$:

$$r_1 = np.copy(r)$$

$$r_2 = np.copy(r)$$

...

if :

...

else:

$$h = h \cdot e^{1/4}$$

$$r_2 = np.copy(r)$$

$$k_1 = h \times f(r_2, t)$$

$$k_2 = h \times f(r_2 + 0.5 \cdot k_1, t + 0.5h)$$

$$k_3 = h \times f(r_2 + 0.5 \cdot k_2, t + 0.5h)$$

$$k_4 = h \times f(r_2 + k_3, t + h)$$

$$r_2 + = (k_1 + 2k_2 + 2k_3 + k_4) / 6.$$

$$k_1 = h \times f(r_2, t)$$

$$k_2 = h \times f(r_2 + 0.5 \cdot k_1, t + 0.5h)$$

$$k_3 = h \times f(r_2 + 0.5 \cdot k_2, t + 0.5h)$$

$$k_4 = h \times f(r_2 + k_3, t + h)$$

$$r_2 + = (k_1 + 2k_2 + 2k_3 + k_4) / 6$$

plt.plot().

d).

plt.scatter().

8.14

as.

$$\hat{H} |4\rangle = E |4\rangle$$

$$i\hbar \frac{\partial}{\partial t} \psi(x) = E \psi(x)$$

$$-\frac{\hbar^2}{2m} \frac{d^2\psi}{dx^2} + V(x) \psi(x) = E \psi(x)$$

$$-\frac{\hbar^2}{2m} \frac{d^2\psi}{dx^2} = (E - V(x)) \psi(x)$$

$$\frac{d^2\psi}{dx^2} = (V(x) - E) \cdot \frac{2m}{\hbar^2} \psi(x)$$

$$\frac{d\psi}{dx} = \phi$$

$$\frac{d\phi}{dx} = (V(x) - E) \cdot \frac{2m}{\hbar^2} \psi(x)$$

Pseudo code:

```
import all libraries
import all constants
```

```
def V(x):
```

$$\text{return } V_0 x^2 / a^2$$

```
def f(r, x, E):
```

$$\psi = r[0]$$

$$\phi = r[1]$$

$$f\psi = \phi$$

$$f\phi = (V(x) - E) \cdot \frac{2m}{\hbar^2} \psi(x)$$

```
return array([f\psi, f\phi], float)
```

```
from
squarewell.py.
```

```
def solve(E):
```

$$\psi = 0$$

$$\phi = 1$$

```
r = array([psi, phi], float)
```

```
for x in (-10a, 10a, h):
```

$$k_1 =$$

$$k_2 =$$

$$k_3 =$$

$$k_4 =$$

$$r += (k_1 + 2k_2 + 2k_3 + k_4) / 6$$

```
return r[0]
```

$E = []$

```
for n in range(3): # Set the loop for each energy state.
```

$E_1 =$ # initial guess

$E_2 =$ # initial guess

$\psi_2 = \text{solve}(E_2)$.

$\text{target} = e / 1000$

while $(E_1 - E_2) > \text{target}$:

$$\psi_1, \psi_2 = \psi_2, \text{ solve } (E_2)$$

$$E_1, E_2 = E_2, E_2 - \psi_2 (E_2 - E_1) / (\psi_2 - \psi_1)$$

E.append(E_2)

print()

b> def V(x):

$$\text{return } V_0 x^4 / a^4$$

All the other parts are the same.

c>

for n in range(len(E)):

$$E = E[n]$$

$$\psi = 0$$

$$\phi = 1$$

$$\psi_values = []$$

$$\phi_values = []$$

$$r = \text{array}([\psi, \phi])$$

Inside the loop

for x in range(-5a, 5a, h):

ψ_values.append(r[0])

φ_values.append(r[1])

$$k_1 =$$

$$k_2 =$$

$$k_3 =$$

$$k_4 =$$

$$r_t =$$

transform the list to an array to make calculation easier

trapezoid
method.

$$t = \frac{1}{2}h \psi_values[0] + \frac{1}{2}h \psi_values[-1]$$

for k in range(1, len(ψ_values)):

$$t += \psi_values[k] \cdot h$$

$$\text{Normalization} = 1/t$$

plt.plot(t)

$$54.5 + 52.5 = 111.5$$

111.5 / 117

PHYS 304 AS5

Xiyue Shen*

Haverford College Department of Physics

(Dated: April 12, 2024)

I asked for an extension on this homework. Yang and Melanie helped me to figure out some problems. I have three python files.

PHYS_304_AS9_8.10a&b_Xiyue_Shen and *PHYS_304_AS9_8.10c&d_Xiyue_Shen* files are the comet trajectory problem, numbered as 8.10 in the textbook.

PHYS_304_AS9_8.14_Xiyue_Shen is the file for a quantum harmonic oscillator labeled as 8.14 in the textbook.

1. EXERCISE

1.1. Exercise 8.10: Cometary orbits

1.1.1. part a

For this part, we have two second-order differential equations. Following the trick that we have done for the simple harmonic oscillation, we have split the second-order differential equations into two first-order equations by introducing a new parameter. In this case, I introduced one parameter for each differential equation. Let $\frac{dx}{dt}$ equal to a new parameter. Then, the differential form of this new parameter is equal to the original equations. The form looks like this,

$$\frac{dx}{dt} = v_x \quad (1)$$

$$\frac{dv_x}{dt} = -GM \frac{x}{r^3} \quad (2)$$

(3)

Similarly, for the y direction, we have,

$$\frac{dy}{dt} = v_y \quad (4)$$

$$\frac{dv_y}{dt} = -GM \frac{y}{r^3} \quad (5)$$

(6)

where $r = \sqrt{x^2 + y^2}$

*Electronic address:
URL: [Optional homepage](http://optionalhomepage)

xshen2@brynmawr.edu;

1.1.2. part b

For this part, we will code the four differential equations using the fourth-order Runge-Kutta method. It's very similar to what we have done in the simple harmonic oscillator problem. The difference here would be adding two more parameters in every array, such as the initial condition array. The function must be modified to include all the functions shown in part a by the pseudocode. One thing worth noticing is that we want to do more than two full cycles by correctly setting the time step. The time step I have found is $h = 10^4 s$ since we have a very high initial condition. The plot is shown in figure 1

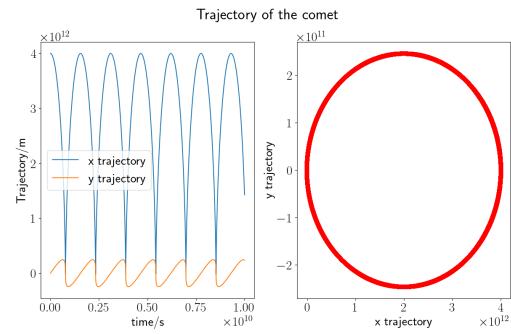


FIG. 1: [The trajectory of the comet with initial conditions $x = 4 \times 10^{12} m$, $y = 0$, $v_x = 0$, and $v_y = 500 ms^{-1}$]

In figure 1, we have two subplots. The left one has a time scale on the x-axis. I made this plot mainly to see how many cycles we go through. Also, we have to tell if the energy is conserved through this plot. Both x and y amplitudes don't decay with time, which indicates a conservation of energy. The right one is a $X - Y$ plot, which shows a circle, as expected by the problem.

1.1.3. part c and d

In this part, we are supposed to use the adaptive size to calculate the comet trajectory with target accuracy is $\delta = 1 \text{ km/day}$, which is $3.17 \times 10^{-5} \text{ m/s}$. I followed the steps in "pendulum_adaptive.py" and adjust the else loop a bit, as shown in the pseudocode attached. The basic idea here is that we have an error value equal to $\sqrt{dx^2 + dy^2}$, where dx and dy are the values difference between the one big step and two small steps methods. Since we have two coordinates, I took the square root of the two combined errors. The new step size is,

1.2. Exercise 8.14: Quantum oscillators

$$h' = h \left(\frac{30h\delta}{\text{error}} \right)^{1/4} = h\rho^{1/4} \quad (7)$$

, where $\rho = \frac{30h\delta}{\text{error}}$

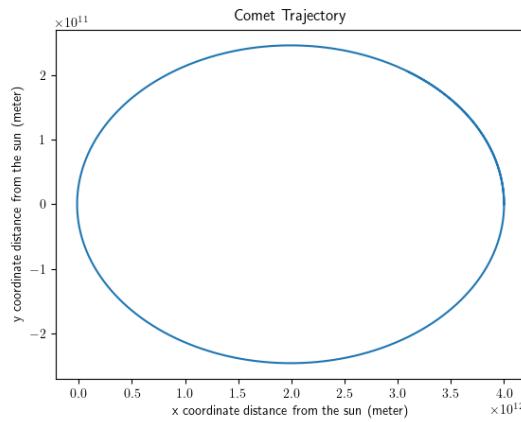


FIG. 2: [Comet trajectory with adaptive step size]

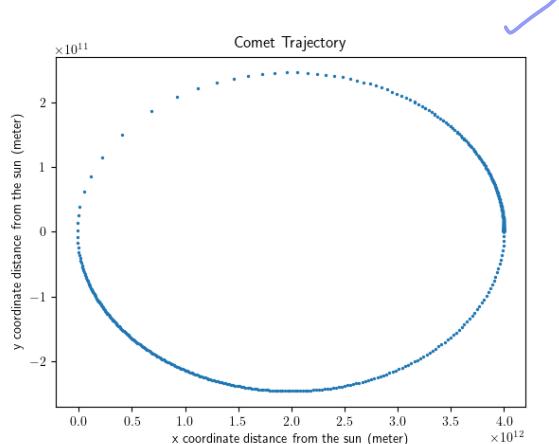


FIG. 3: [Comet trajectory with adaptive stepsize in scatter plot]

Figure 2 and figure 3 are what we have for part c and part d. In figure 3, I plotted every 20 point for a clear view. I expected the scatter plot to show a symmetric behavior, which means the points distribution on the top left and bottom right should exhibit the same density. One possible explanation here is that our initial conditions give a nonzero x coordinate value and a 0 y coordinate value. In this case, the error array dx and dy will have different estimates in the symmetric zone. So, this may break the expected symmetry.

1.2.1. part a

For this part, we are given a harmonic potential $V(x) = V_0 x^2/a^2$. The one-dimensional, time-independent Schroedinger equation is,

$$-\frac{\hbar^2}{2m} \frac{d^2\psi(x)}{dx^2} + V(x)\psi(x) = E\psi(x) \quad (8)$$

In this equation, \hbar is the Planck constant over 2π , $\psi(x)$ is the wave function, m is the atom's mass, and E is the corresponding state energy. We can rewrite the equation to get the second-order differential equation on one side,

$$\frac{d^2\psi(x)}{dx^2} = \frac{2m}{\hbar^2} (V(x) - E)\psi(x) \quad (9)$$

To split it into two first-order differential equations, let $\frac{d\psi}{dx} = \phi$. Then we have,

$$\frac{d\psi}{dx} = \phi \quad (10)$$

$$\frac{d\phi}{dx} = \frac{2m}{\hbar^2} (V(x) - E)\psi \quad (11)$$

To form the code, I followed the *squarewell.py* example using the secant method. The pseudocode is shown in the previous attachment. In the main program, to solve the energy, I put them in a loop for each energy state. The result shows that the ground state energy is 138.03 eV, the first excited energy state is 414.07 eV, and the second excited state energy is 690.12 eV. To see if this result is correct, each state's energy difference is constant, around 276 eV.

1.2.2. part b

For this part, we are given a different potential, $V(x) = V_0 \frac{x^4}{a^4}$. In the code, I changed the potential equation and the initial guess while keeping the rest the same as part a. The energy I got for the first three states are 205.31 eV, 735.69 eV, and 1443.57 eV. I tested this as the correct answer by plotting the wavefunction, which is part c.

1.2.3. part c

For this part, we are gonna solve the wavefunction and make a plot of the wavefunction. I set up a loop to run the solution for each energy state. Then, using the fourth-order Runge-Kutta method, we can solve the wavefunction and store the values. After this, a trapezoidal method is used to integrate for normalization. The normalization factor is $N = 1/\int |\psi|^2 dx$. The plot is shown in figure 4

2. SURVEY

✓ 15

I spent roughly 8 hours on this homework. I'm now more familiar with adaptive stepsize differential equation problems. I like the comet trajectory problem most. It took me a very long time to figure out the break in the code. I practiced my troubleshooting skills. I think this set is about the right length.

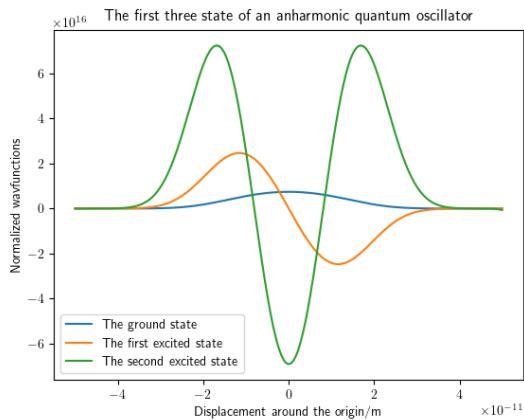


FIG. 4: [The first three wave states of the anharmonic quantum oscillator]

3. UNGRADED PART

I have done all of the required and ungraded work.

8.10

54.5/56

Computational Physics/Astrophysics, Winter 2024:

Grading Rubrics¹

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)

3.5

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

error with adaptive step size, we expect smaller steps on left side of graph -0.5

2

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)? **Units? - 1**

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

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

. 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?

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)

8.14

52/56

Computational Physics/Astrophysics, Winter 2024:

Grading Rubrics¹

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)
- 3 6. Is the output answer correct? (+4 points).
normalization error - 1
- 2 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)? **UNITS? - 1**
- . 5.4. Are algorithms found on the internet/book/etc. properly attributed?

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

- . 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?

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)
- 1** . Is a brief explanation of physical context given? (+2 points) *expand on explanation of anharmonic solution - 1*
- 0** . If relevant, are helpful analytic scalings or known solutions given? (+1 point) *compare to known solution - 1*
- 3** . Is the algorithm used explicitly stated and justified? (+3 points) *- 10N - 1*
- 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)