PHY407 Lab-07 Report

Patrick Sandoval[1] and Chin Chong Leong[2]

[1] *Q2 - Hydrogen Atom*
[2] *Q1 - Space Garbage*

1. Q1 - SPACE GARBAGE

For the pseudo-code of all the sub-question please refer to the python script titled `Lab07_Q1.py`. The main functions used in this part are stored in `MyFunctions.py`

1.1. *Q1a - Trajectory from Adaptive Method*

For this part of the exercise, we implemented the adaptive step size method as described in pg.358-359 in the textbook, with initial time step and target accuracy per unit time set to $h = 0.01$ and $\delta = 10^{-6}$ respectively. To summarize, at each time point t we do two calculations on the positions and velocities of the particles: two steps of size h with RK4 as $\mathbf{r}_1$, and one step of size $2h$ with RK4 as $\mathbf{r}_2$ (both at the same starting time point). Then we calculated the quantity:

$$\rho = \frac{h\delta}{\sqrt{\epsilon_x^2 + \epsilon_y^2}} \tag{1}$$

where the errors are given by eq.8.54 in the textbook:

$$\epsilon_x = \frac{1}{30}(x_1 - x_2) \tag{2}$$

$$\epsilon_y = \frac{1}{30}(y_1 - y_2) \tag{3}$$

With this quantity we can then calculate the new time step size with

$$h' = h\rho^{0.25} \tag{4}$$

If our calculated quantity $\rho \geq 1$, then we can use the RK4 to calculate positions and velocities at $(t + h')$ and we are good to move to the next time point given by $(t + 2h')$; however if $\rho < 1$, then we need to go back to the start and re-calculate everything using the new h' until eventually $\rho \geq 1$.

Figure 1 shows the trajectory of the particle with both the non-adaptive RK4 and adaptive RK4 method. As we can see from the figure, for adaptive method the separation between points is large when orbiting far away from center, while the separation between points near center is small (of which corresponds to region with large velocity change).

1.2. *Q1b - Time requires for Adaptive and Non-adaptive method*

We have also implemented a timer while we were doing the calculation in section Q1a to record time it takes to finish the simulation. In order to achieve the same target accuracy per unit time in the non-adaptive method, we now need to use 10,000 time steps as suggested in the question. We timed this version of non-adaptive method and showed the results of both method below.

	Time required [s]
Non-adaptive method	0.66008
Adaptive method	0.20095

Table 1: Time requires to complete simulation for non-adaptive and adaptive method in order to achieve a target error-per-second $\delta = 10^{-6}$

Note that the non-adaptive method requires more time than adaptive method here because the non-adaptive method need to go through 10,000 RK4 calculations, while the adaptive method only need to go through approximately $(3 \times 396 = 1188)$ RK4 calculations

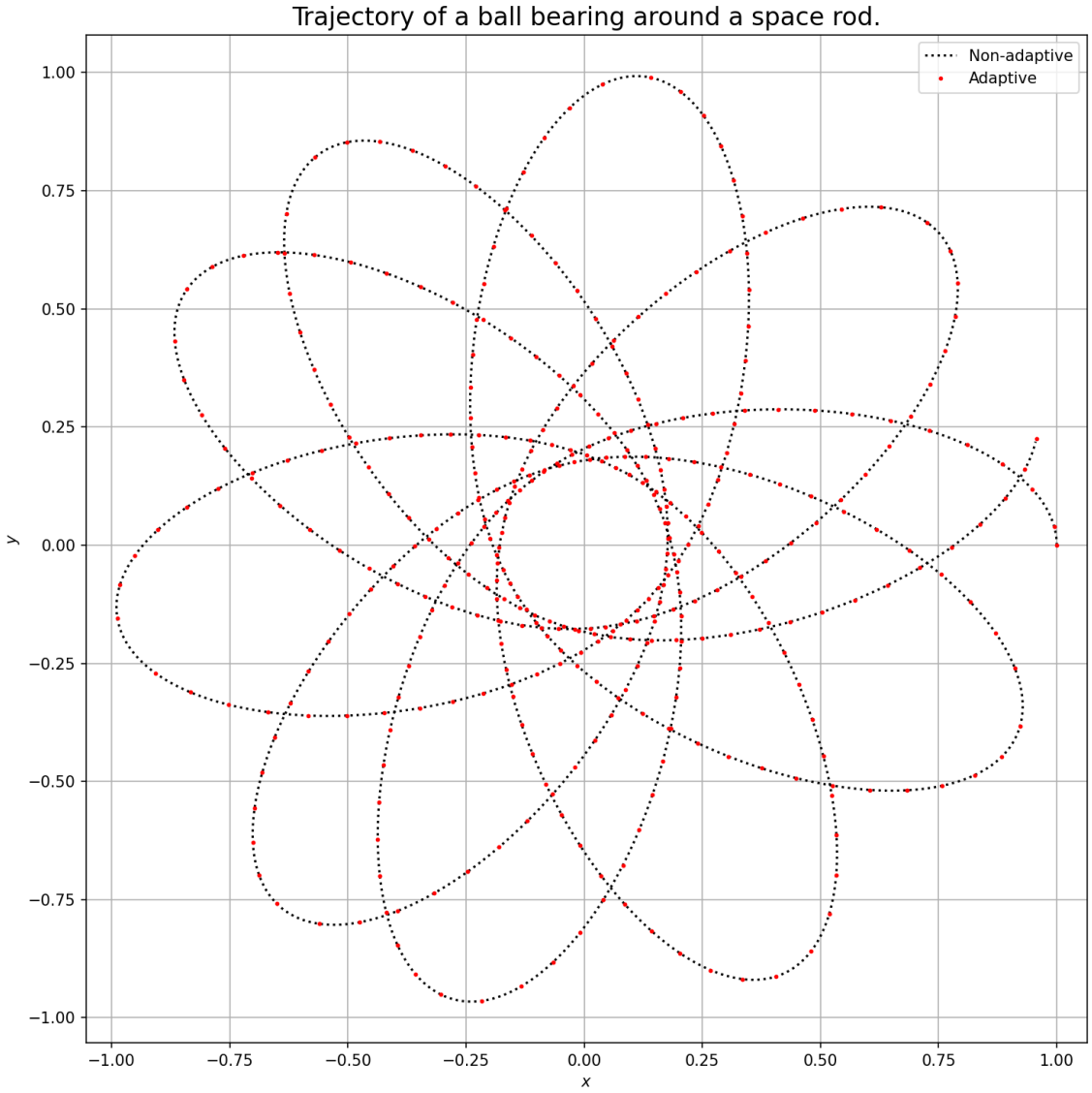

Figure 1: The trajectory of particle near a steel rod in space using non-adaptive RK4 and adaptive RK4 method. The small black dots correspond to positions where non-adaptive method calculation are performed on, which uses 1000 points to complete the simulation; the red dots correspond to positions where adaptive method calculations are performed on, which only uses 396 points to complete.

1.3. *Q1c - Size of Time Step in Adaptive Method*

By book-keeping the size of time step h in the calculation in section Q1a, we plot the size of time steps as a function of time in Figure 2.

As we can see from the figure, the time step gets periodically small throughout the simulation. Combining the observation in the trajectory we had in section Q1a, we can conclude that the size of time step is small in the region closed to the center, of which corresponds to region with large velocity change (based on our knowledge of conservation of angular momentum); and the size of time step is large in the regions far away from the center, of which correspond to region with small velocity change.

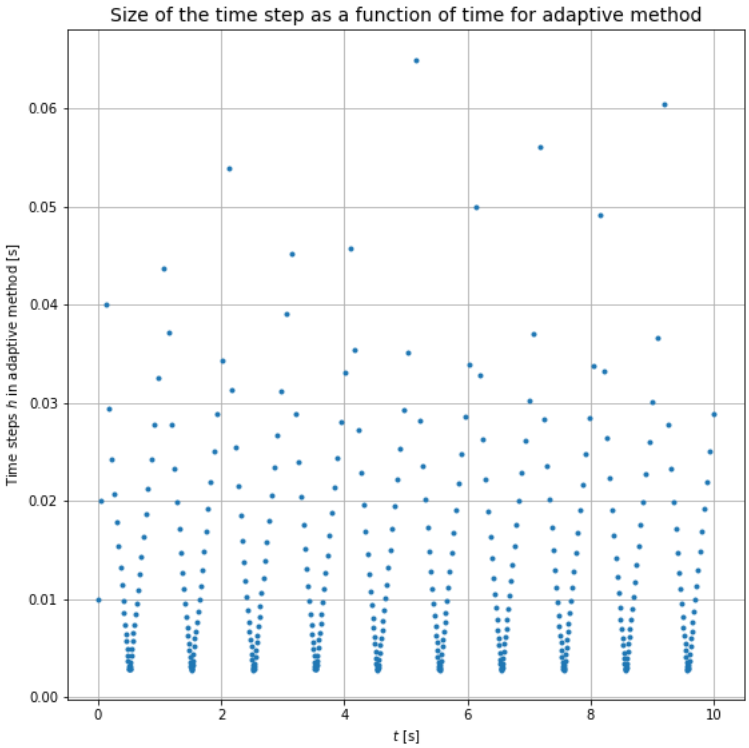

Figure 2: Size of time steps as a function of time in the simulation for adaptive method

2. Q2 - HYDROGEN ATOM

For the code, pseudo-code and plots please refer to the python script `Lab07_Q2.py` and `MyFunctions.py` for the individual functions and RK4 routines.

2.1. *Q2a - Separability of Schrodinger Eq.*

The time independent Schrodinger equation describes the behaviour of stationary states. In the case for a central potential where the only degree of freedom is the radial distance we can separate the wave function into to main functions $\psi(r,\theta,\phi) = R(r)Y_l^m(\theta,\phi)$, where Y_l^m is the spherical harmonic of degree m and order l. In this

$$\frac{d}{dr}\left(r^2\frac{dR}{dr}\right) - \frac{2mr^2}{\hbar^2}[V(r) - E]R(r) = l(l+1)R(r) \tag{5}$$

Where $V(r)$ is the potential for the Hydrogen atom given by

$$V(r) = \frac{-e^2}{4\pi\epsilon_0 r} \tag{6}$$

In order to solve eq.5 numerically through the RK method we need to split into into two coupled first oder ODE's by applying the following substitution

$$S(r) = r^2\frac{dR}{dr} \tag{7}$$

$$\implies \frac{dR}{dr} = \frac{S(r)}{r^2} \tag{8}$$

$$\implies \frac{dS}{dr} = R(r)\left[l(l+1) + \frac{2mr^2}{\hbar^2}(V(r) - E)\right] \tag{9}$$

Please refer the python script titled `MyFunctions,py` where we have a series of functions used to implement the RK4 routine along with the shooting method on the bound state energies. The the roots of the shooting method were found through the secant method by setting a threshold of $e/1000$. Additionally, we set the boundary conditions to zero at $R(h)$ and at $R(20a)$ such that we avoid divergence at the beginning and we integrate for long enough distance to avoid infinite computational time. We note that a is the Bohr radius and $h = 0.002a$ is the step size

2.2. *Q2b - Ground State and 1^{st} Excited State Energy*

Through the implementation described in subsection 2.1 we can apply the shooting method the compute the energy eigenvalues of the Hydrogen atom. However, in order to improve the accuracy of our routine I chose to make the step size smaller to $h = 0.0001a$. The reason to this change is motivated by the accuracy of the eigenfunction on the boundaries. The wavefunction is highly sensitive to the energy therefore any small inaccuracies in the energies can lead to great inaccuracies on the boundaries. Table **??** shows the results of our implementation of shooting method and the number of iterations it took the value to converge to a thousandth of an eV.

We see that our results are consistent with the known energies of stationary states of the Hydrogen atom, where we notice that the energy of $E_2^0 \approx E_2^1$ which is what we expected from our theory.

2.3. *Q2c - Hydrogen Atom Radial Wavefunction*

Now that we have the bound state energies we can use them to compute the radial component of the wave function by using the RK4 method from subsection 2.1 and inputting the solved energies. After solving for these wave-functions we normalize them by integrating their absolute value square and dividing the w.f by the square root of the result of the integration. In the python script for this questions we integrate the normalized wavefunction again to verify it integrates to 1, and we print out the result.

From Figure 3 we see that all our solutions satisfied the imposed boundary conditions and they follow the same general shapes of the solutions for the Hydrogen atom. We see the strict exponential decay for the ground state energy. We also see the $-re^{-r}$ relation for the next excited state for order $l = 0$ and lastly we also see the re^{r} relation in the first excited state for $l = 1$.

Table 2: Energy eigenvalues for radial component of wavefunction of the Hydrogen atom

n: state l: order	E_n^l (eV)	Iteration #
l=0, n=1	-13.601	7
l=0, n=2	-3.400	4
l=1, n=2	-3.401	5

NOTE—For the energy eigenvalue print outs please refer to the python script `Lab07_Q2.py` where we print the energy eigenvalues as they converge.

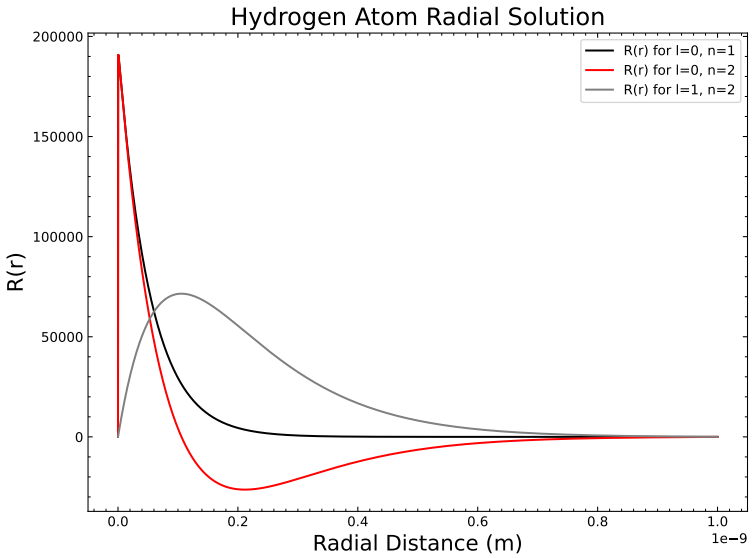

Figure 3: Radial component solution of wavefunction in Hydrogen atom for ground and first excited states for different spherical harmonic order. We see that boundary conditions are met for $r = h$ and $r = 20a$, where a is the Bohr radius.