PHY407 Lab06

(Yonatan Eyob Q3), (Kivanc Aykac Q1,Q2)
Student Numbers: 1004253309, 1004326222

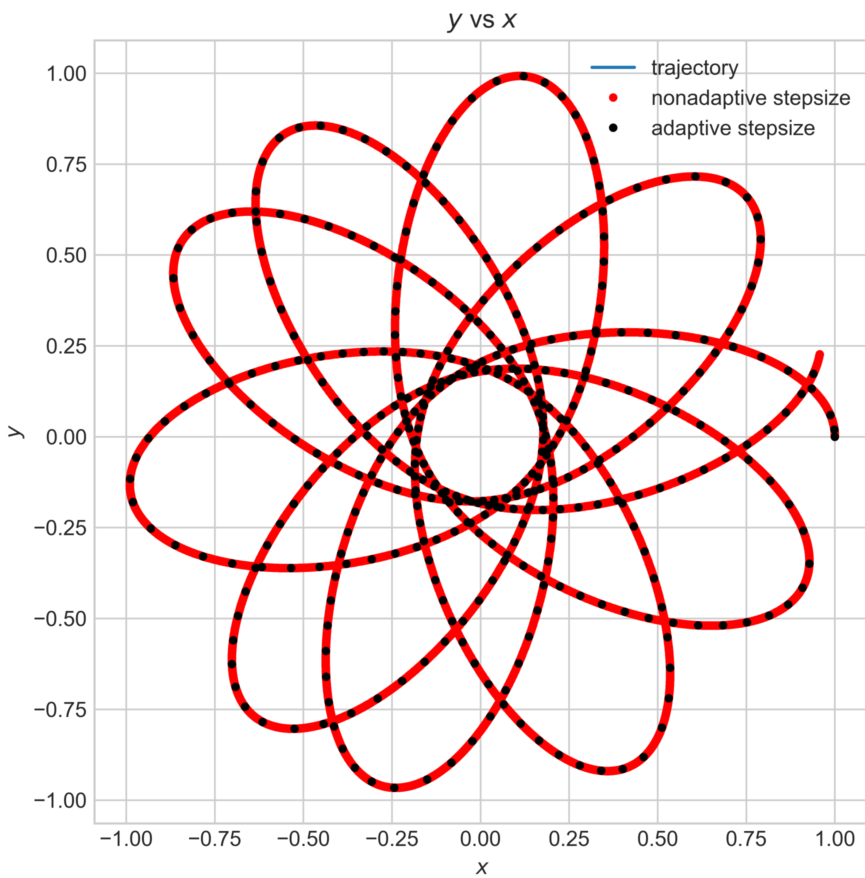
November 1^{st}, 2020

1 Return of the space garbage

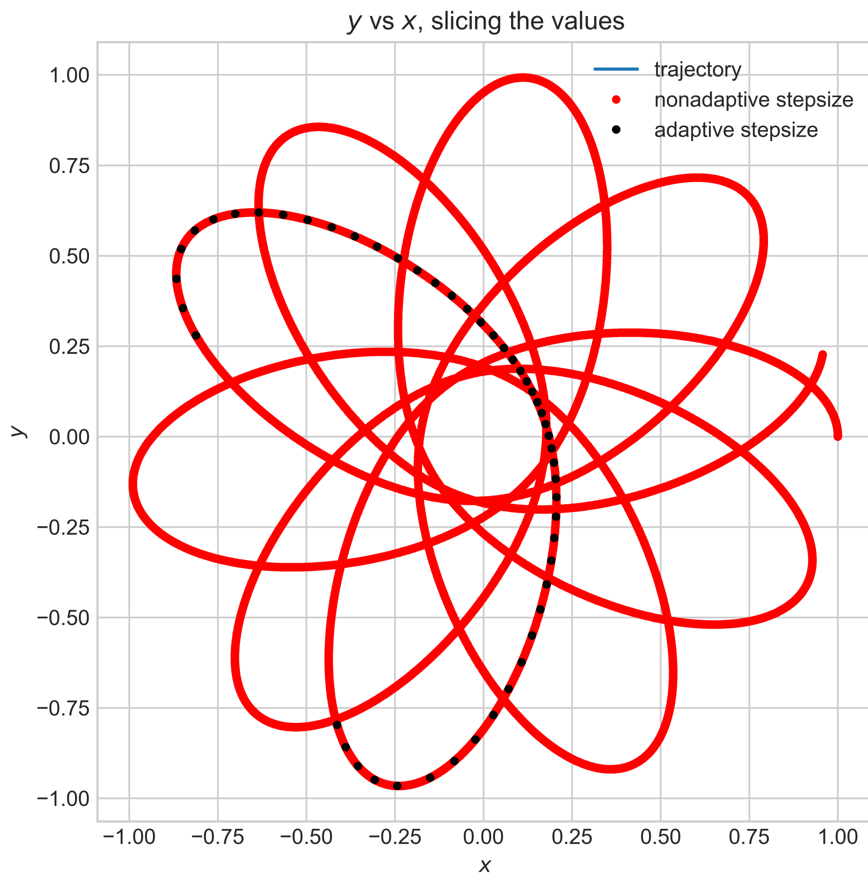
In this exercise, the space garbage question from the last lab is going to be further investigated. The ODE for the trajectory of a spherical ball-bearing around a heavy steel rod is going to be solved using adaptive step sizing and will be compared to the non adaptive method used in Lab 6.

1.a

In this part, goal is to plot individual points of the adaptive step size algorithm at each time step, and overlay it on top of the one from Lab 6, Q1. For the non-adaptive method, $h = 0.001$ should be used to reach the similar level of accuracy. Below is the figure for this part:



As can be seen the blue line that represents the trajectory as found in the previous lab is not visible and the other two plots are overlapping on it. This shows that the plots are finding consistent trajectory results. Also, the red dots that represent the non-adaptive method is covering the entire trajectory with no density changes. On the other hand -as the name suggests- in the adaptive step sizing plot, it is visible to see the areas of the trajectory closer to the centre are getting a much smaller step compared the the other parts of the orbit. To more clearly see this dense regions, below is the figure with sliced adaptive method values:



As can be seen the adaptive step size method is treating each interval with delicacy in order to chase the target error.

1.b

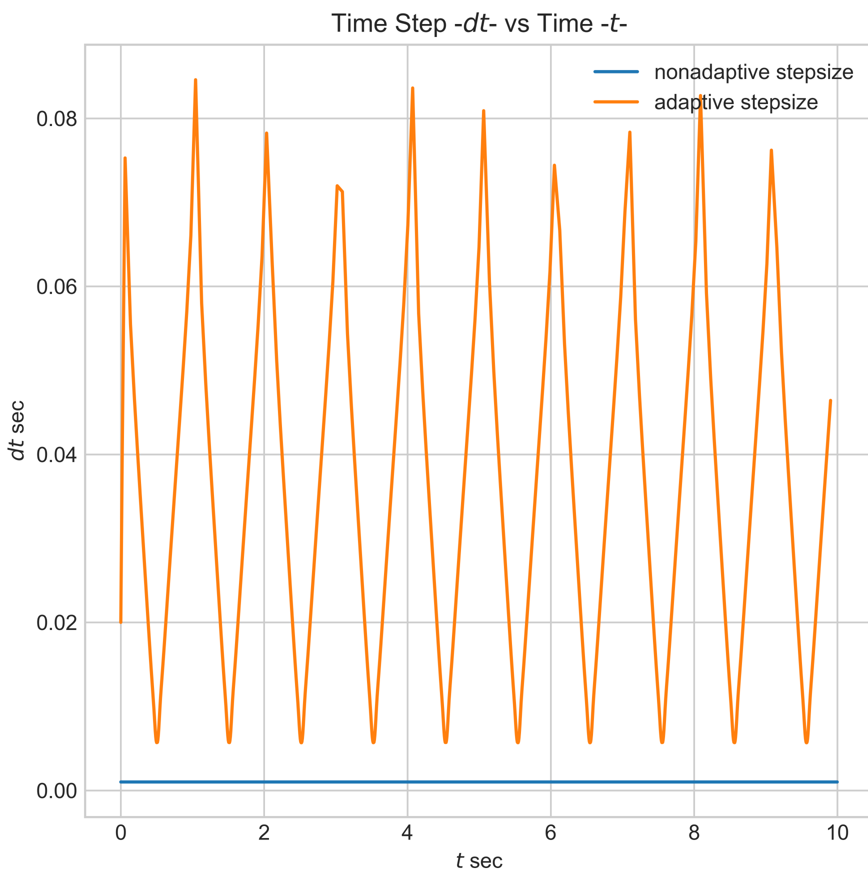
In this part, the time it took for both methods are measured and printed. Below is the output:

```
    Time taken
Nonadaptive Step Size: 0.8414130210876465 sec
Adaptive Step Size: 0.21417713165283203 sec
Difference Between methods: -0.6272358894348145 sec
```

As can be seen, adaptive step size method is much faster, with almost the quarter of the nonadaptive step size method. This is because the method varies its stepping making it to avoid unnecessary calculations while at the same time preserving the desired accuracy.

1.c

In this part, the `tpoints` arrays of the two points is juxtaposed. This array was used in calculating the trajectories. Below is the figure:



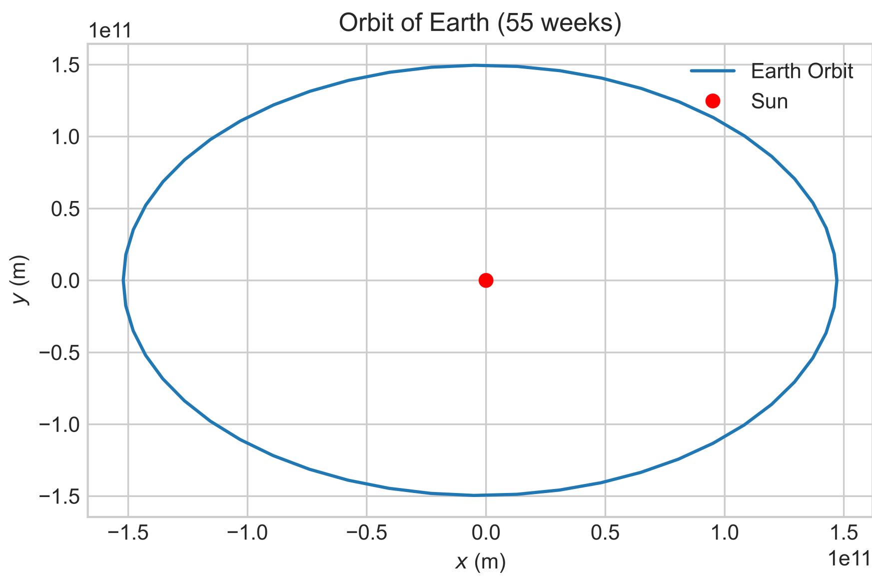
As can be seen, the nonadaptive method used the same dt in each step of the calulation as step size ($h = 0.001$)was constant and a simple `numpy.arange()` function was used in creating the entire `tpoints` array. However, the adaptive method varies its h and thus have to vary its dt values in each step of the calculation. Notice how the number of peaks matches the number of revolutions observed in the trajectory figure. Time steps pretend to widen as h values widen on the away points from the centre.

2 Bulirsch-Stoer method: Earth and Pluto's orbits

In this exercise, goal is to plot and compare the trajectories of Earth and Pluto calculated using Bulirsch–Stoer method.

2.a

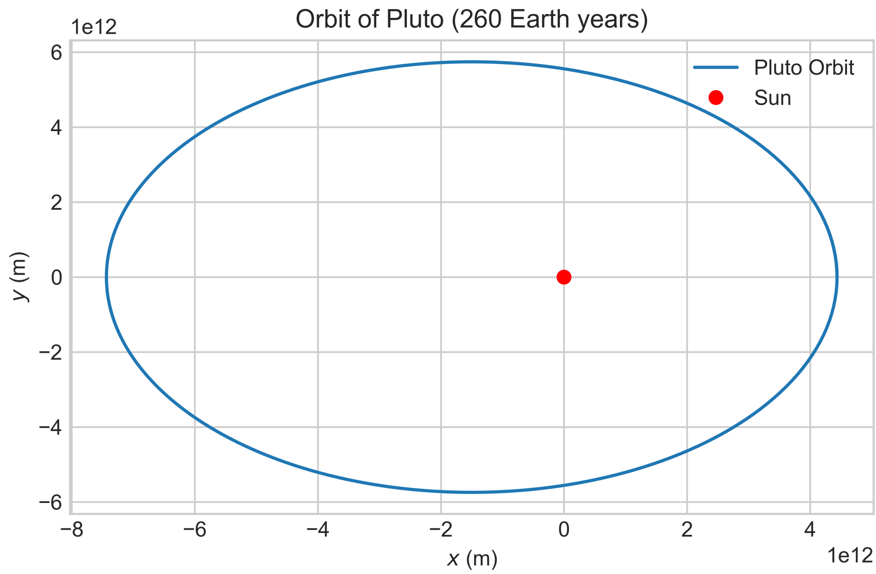
In this part, goal is to plot the orbit of Earth. To do Bulirsch–Stoer method, exercise 8.7 (Newman) is closely observed, and a similar approach is tried in the coding here as well. Units of meter per year was used to overcome the overflowing issues, but it does not matter because only $y vs x$ is plotted here. Below is the graph of Earth's orbit using Bulirsch–Stoer method:



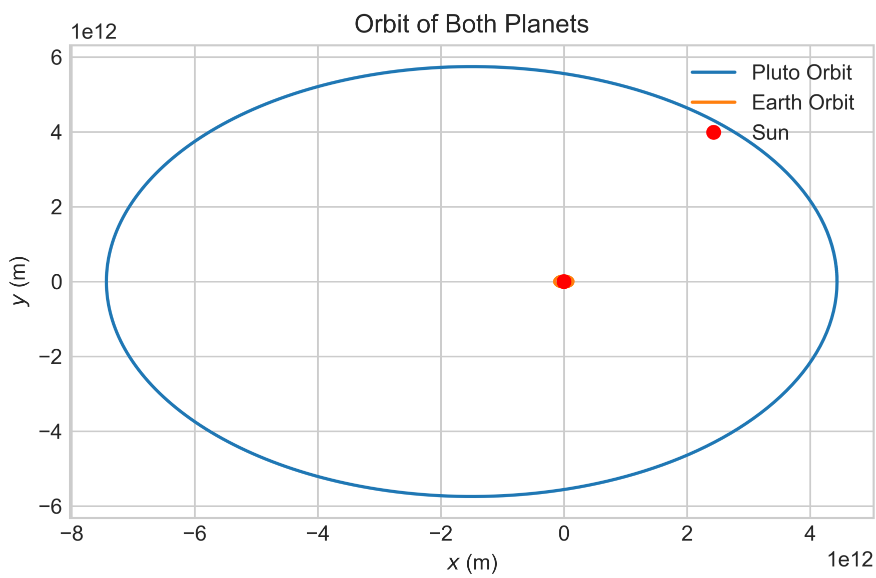
As can be seen the orbit is slightly shifted to the left. Only 55 weeks of revolution is plotted here as it is enough to show the one revolution around the Sun.

2.b

In this part the same method applied to the Pluto. But his time, H value is set to $H = 1$ so that the intervals in the calculation is 1 Earth year. Below is the orbit of Pluto:



As can be seen Pluto's orbit focal point (Sun) is much more distinguishable than the Earth's. To further investigate the difference between the two orbits, below is the graph that plots both of them:



As can be seen, Pluto's orbit is sweepingly more wide than the Earth's. So much that the Earth's orbit is hardly even visible.

3 The hydrogen atom

in this section we will find the energy eigenvalues for cases n=1 l=0, n=2 l=0, and n=2 l=1, and plot the corresponding eigen functions for the radial part of the Schrödinger equation using the RK4 and shooting method, and then we will compare the our numerically calculated solutions to the explicit solutions given to us on the handout.

3.a

[Nothing to submit]

3.b

Using the shooting method and RK4, we numerically calculated the Energy eigen values for three different cases: n=1 l=0, n=2 l=0, and n=2 l=1. our results are listed below:

Energy values for 3 different cases		
n	l	Energy
–	–	———————————
1	0	-13.600180806415297eV
2	0	-3.400369298579445eV
2	1	-3.4012498253523304eV

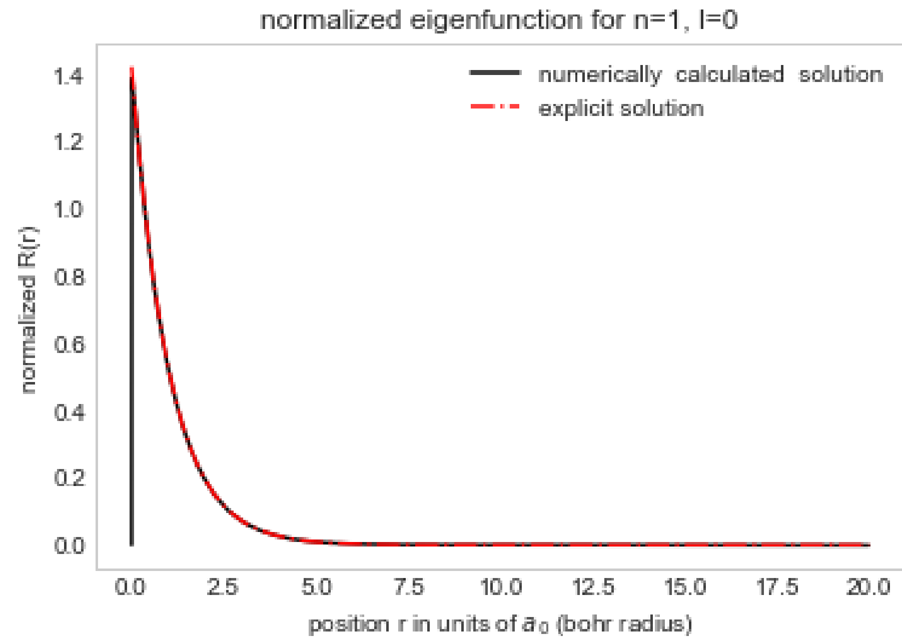
While coding, we noticed that decreasing the target and step size increases the run time of the code but also increases the accuracy of the calculation. To get the results listed above, we set the step size to 0.0001a and the target to e/1000, where 'e' is equal to the charge of an electron and 'a' is bohr's radius, both in SI units. When the width of the well was set low (below 10a) or set high (above 40a) the energy calculation became less accurate so there is a sweet spot between 10a and 40a to get the most accurate energy calculations. We settled with 20a for our calculations. Increasing the width of the well also increases the run time of the code.

3.c

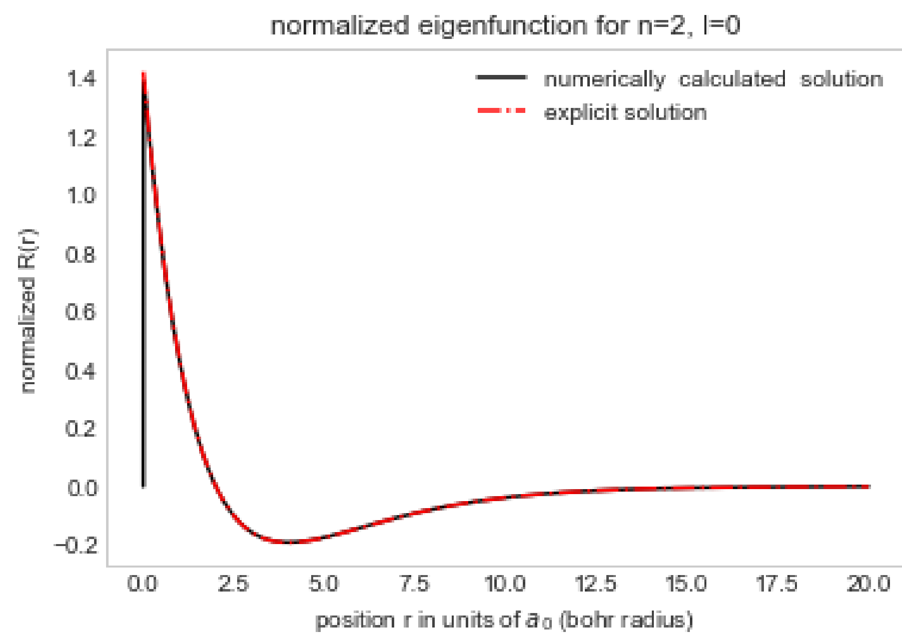
[Nothing to submit]

3.d

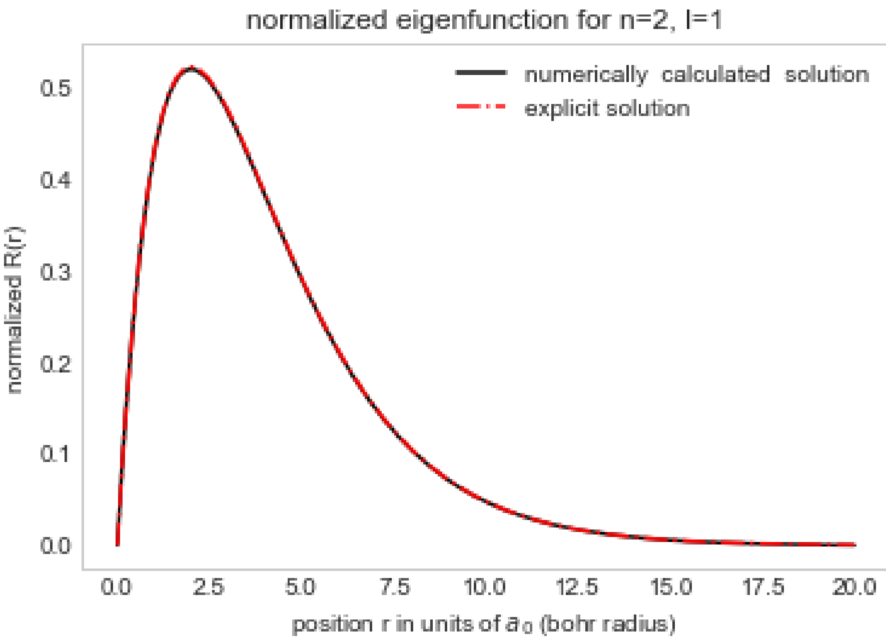
Now we will compare our numerically calculated solutions with the explicit solutions by plotting the eigen-functions on the same graph for each of the three cases.



The numerically calculated solution seems to perfectly match the explicit solution for n=1 l=0, except for the very beginning.



The numerically calculated solution seems to perfectly match the explicit solution for n=2 l=0, except for the very beginning. similar to the n=1 l=0 case.



The numerically calculated solution seems to perfectly match the explicit solution for n=2 l=1.

All of the explicit solutions perfectly match the numerically calculated solutions except for the beginning of the n=1 l=0 and n=2 l=0 cases because the numerically calculated solutions started at R(h)=0. Both solutions have the same shape and number of zero crossings if we don't include the beginning of the numerical solutions for n=1 l=0 and n=2 l=0. When playing around with the code, we noticed that increasing the width of the well too much causes the end of the numerically calculated n=1 l=0 plot to diverge and making the width too small makes the numerically calculated plots stray away from the explicit solution plots. A width of 20a is a good choice for comparing the graphs!