Lam Comp Class HW3

Mike Ortiz

September 2020

1 Problem 1

In this problem we are using the unit test for our matrix class that we built in homework 2. From here we are supposed to test all the functions we have written for our matrix class against numpy's matrix functions using python's unit testing. I was only able to get the add function after some time so I got rid of the rest because of time constraints. I do understand what the unit test does but I cannot get it on the code for some reason for the rest of functions. I tired, I really did. I am sorry it took forever but I do get why the numpy way is better

2 Problem 2

For problem two I was supposed to solve for the energy states of hydrogen up to 9 states. The way it happens it by the shell either absorbing or emitting an electron. Both take energy to move up or down. I was supposed to show this using the data given in the dat files provided and using a matrix class. We were also solving for $B_l u$, $A_u l$ and J and other equations from the homework. Again I did not finish this problem because of time, but I know I am supposed to build a matrix for J, $A_u l$, $B_l u$. Using the Energy State function we are comparing the matrix's of $B_l u$ and $A_u l$. We will see a difference in these two numbers because one is moving up in shells and the other is moving down in shells i.e it takes energy to move up and down these shells and we are seeing if it is the same amount of energy to absorb or to emit. The conclusion I am sure moving down an energy state would be less energy than moving up and this could have been proven if I completed the script but I got it working till here I think. Maybe when I become even better at python I will be able to show that difference.

3 Problem 3

Problem is testing the three numerical methods of Euler's, Heun, and RK4. We are putting these methods into a function to be able to be called in a different scripts to check their numerical analysis against certain functions. For this problem we will be choosing a function and testing it out. I spent a lot of time on this problem and I got it to work. Well I think. I know I had some issues trying to call for the other problems, but lets talk about the figures.

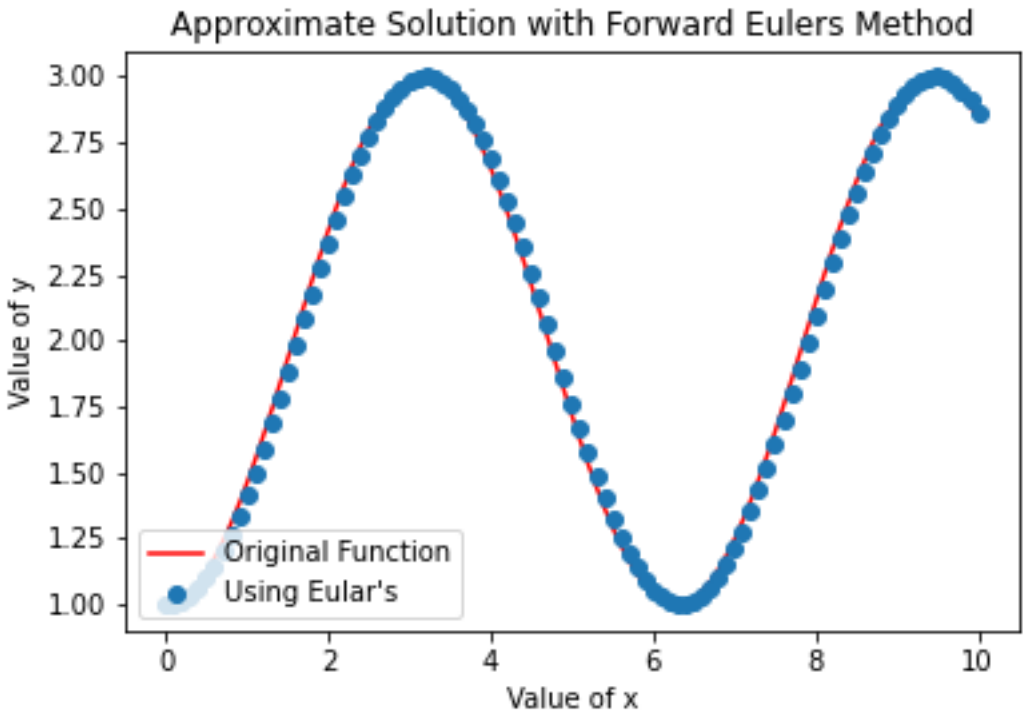

Figure 1: As we can see from the image we are testing a sinusoidal wave. I choose this to broaden my horizons on using different functions here and there. As we can see with 101 steps it is pretty close but the red line that is the basic cos function is not lining up with with out Euler method analytic solution of cos. It is close but I think the other two will be be better estimates of the functions I choose.

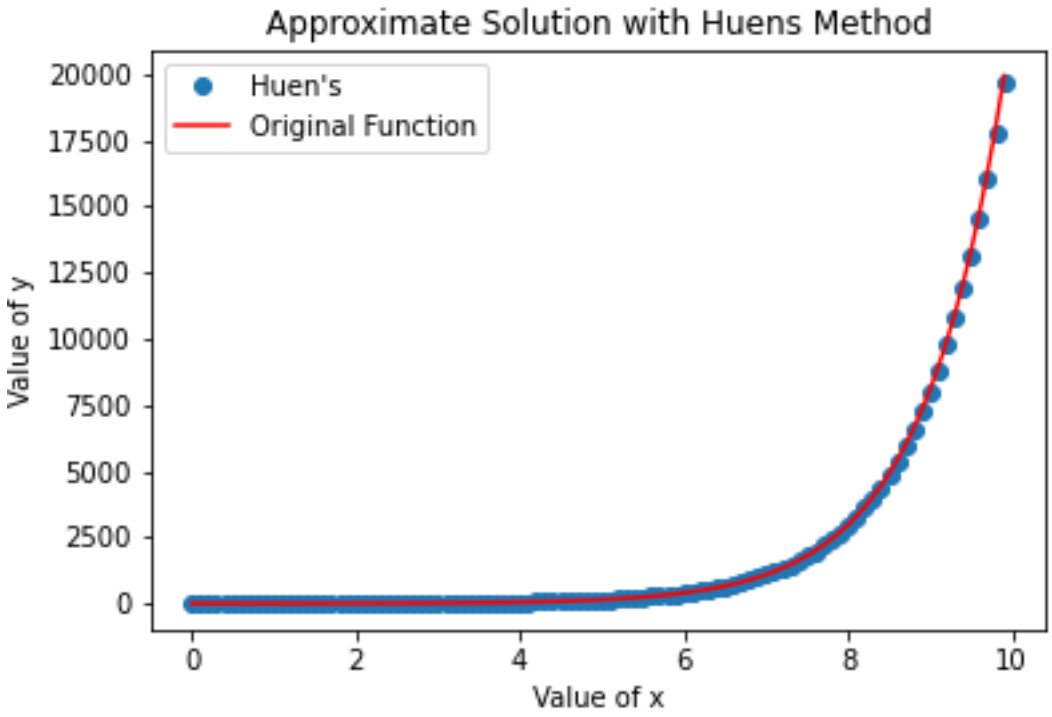

Figure 2: In this figue we are using Heun's method to see the analytic solution compared to a known function. In this case we will be evaluation an exponential e^x to check what make a comparison. So as we can see here it is pretty accurate as far as Heun's method goes. Not until 12500 that we start to see a break and less accurate points. Seems to be over estimating it around that y. Also I found interesting is tha the Heun method is just an RK method but onyl to the 2nd order I belive. If I am not mistaken.

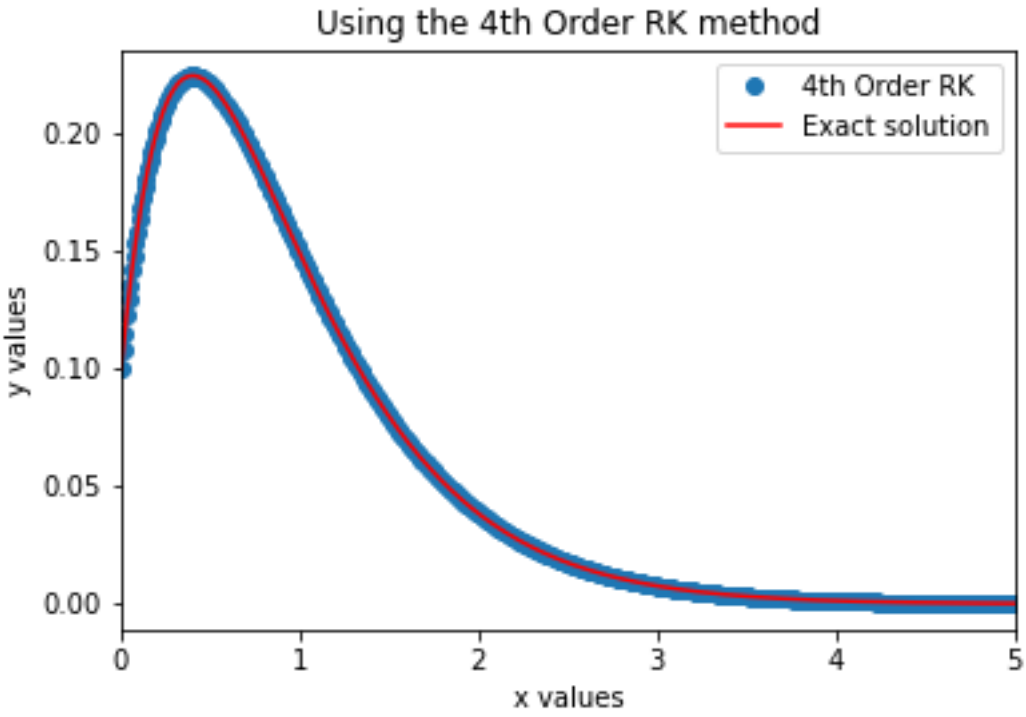

Figure 3: For the Rk4 method we are testing another exponential but I put a little twist so it would not get boring haha. The most important thing to see here is the accuracy. I have trouble seeing the difference so I had to print the difference between the two functions. I got a number of $10^{-}10$ difference. Which is pretty amazing if you ask me. I will be using this method of analyzing numerical solutions for ODE's more often.

4 Problem 4

For problem 4 I am supposed to be using odeint for it's damped pendulum equation and graph. I was supposed to use my analytic solution methods of a damped pendulum equation and compare it to odeint and see what the difference was. I was only able to recreate the odeint graph of omega and theta to see the damping effect omega had on the pendulum viewing it through theta. Which in return my analytic solution would have been damped and we could have seen it's accuracy sadly I was not able to produce my ODE to compare, but I do know well think think that since the RK4 method was the most accurate analytic solution that it would have the best comparison to the odeint follow by the Heun and follow by Euler's. It think where I went wrong is with arguments in my ODE scripts to match the arguments of the odeint script. I could not get them to match but I am sure I am super close but will need more time to dig around the odeint to see if I can see the issue.

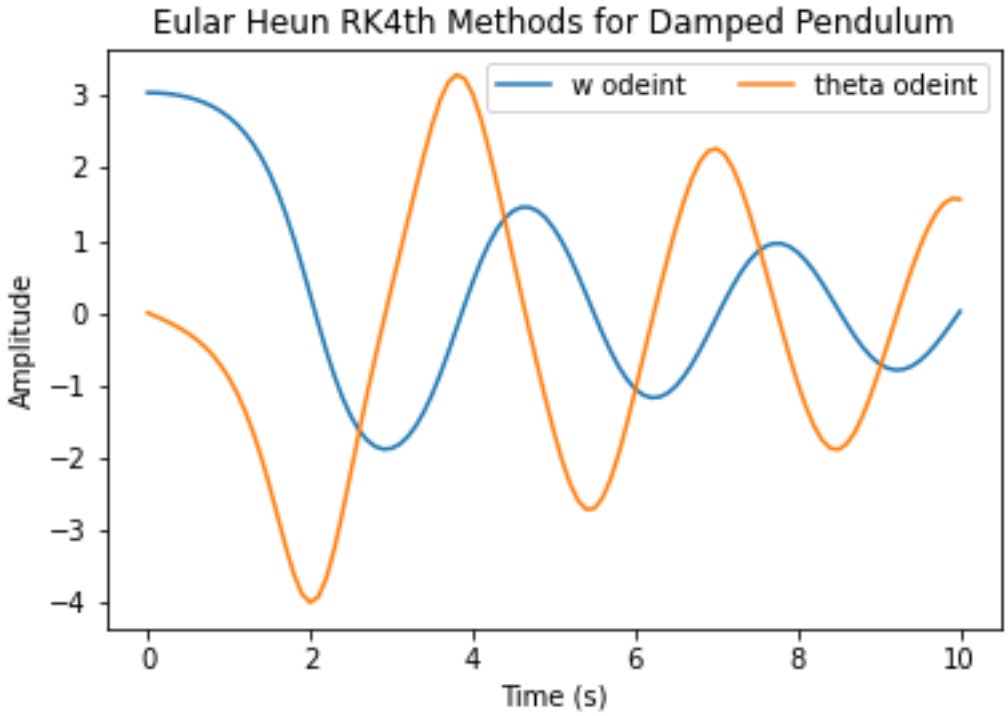

Figure 4: As stated above we just have the odeint graph from scipy with the damping effect on a pendulum. Sad I could not get it to work : (

5 Problem 5

Using the Stiff ODE solver allows us to solve the ODE to a much higher precision based on how how you can go in the interations and the step size to be able to compare ODE45 for example. This is used for I think a way more diffucult ODE to solve by hand or not as accurete as the the other ODE solvers we have been testing. Again my problem is getting the arguments to match but and all that and since I did not get it to work like I did for problem 4 I knew I would run into the same issue. I hope to come back to this problem later when I have increased my skills in python and numerical analyses.

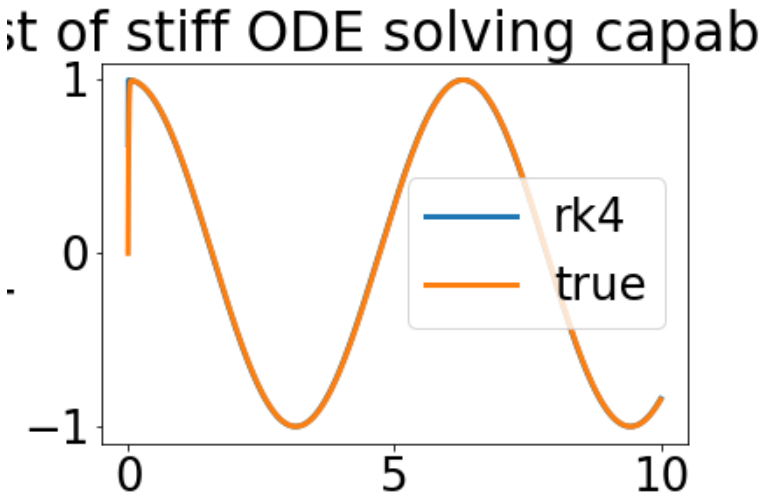

Figure 5: This is where I would be able to plot my things together to check but it did not work out unfortunately.I did how ever produced this graph with just the stiff equations but I had changed my code I did not reverse it back from trying to get it to work. More python sadness.