Homework 3

Ryan Wills

ASTP720

February 29th, 2019

https://github.com/RyanWills16/ASTP720

ODE solvers

Writing the ordinary differential equation solvers was pretty straight forward. Rather than solving for y_i+1 though, I solve for the current time step and use information from the previous step by starting at the second index. I wrote the code so that it would take either a list of times or a start and stop time and a step size. I used all of the same initial conditions as the example on `ode_int` did.

Figure 1 shows the solution for theta and omega of the damped pendulum by the Euler method. I used the r-square value as a sort of test to see how close the solution from my ODE solver is to `ode_int`. As can be seen, the Euler method is pretty close in terms of r-square, but not perfect. Figure 2 shows the solution from Heun's method. It doesn't really appear different, but the r-square value is slightly better than for Euler. Last is Figure 3, which shows the rk4 solution, which turns out to be a perfect match to the `ode_int` solution.

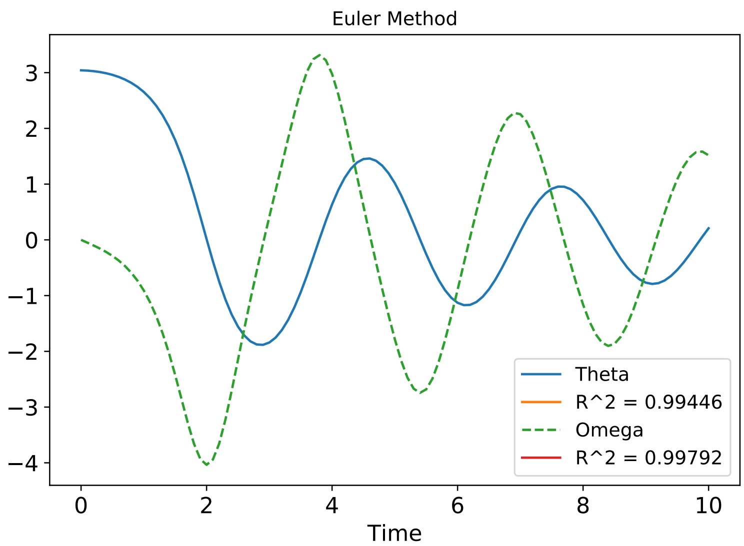

FIG. 1: Euler method for the damped pendulum.

Stiff ODE

For the stiff ODE, I used lambda = 25 and an initial condition of y(0) = 1. I had to edit my ODE solvers to take only one variable, and all of them were able to solve it pretty well I think. Each of Figures 4 to 6 have points, which represent my solvers solution, and a line that represents the points drawn from the solution equation provided in the homework assignment. Each method is a pretty good fit, but it seems as if RK4 did the best by eye.

Stellar Structure

Unfortunately I could not get my ODE solvers to work for the stellar structure equations, even after trying many different methods. I made sure to check all my units with astropy units for the calculations I was making and everything came out okay. The dP/dr equation came out to have units of $dyn/cm^2/cm$, which is the cgs unit for pressure per unit distance, and the dM/dr equation came out to have units of g/cm, cgs units for mass per unit distance. I tried different ways of doing the initial calculation in my solver, but I couldn't get anything to work.

The problem on the part of the pressure is that dP/dr calculated at any radius other than zero is so much smaller

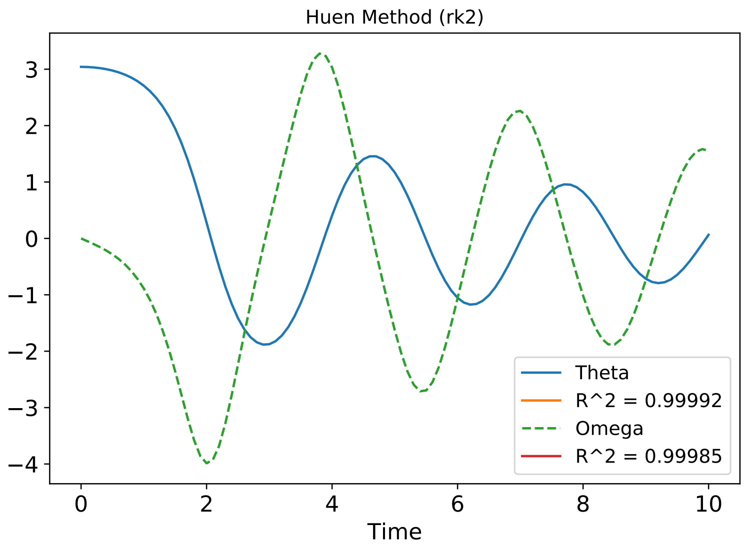

FIG. 2: Heun's method for the damped pendulum.

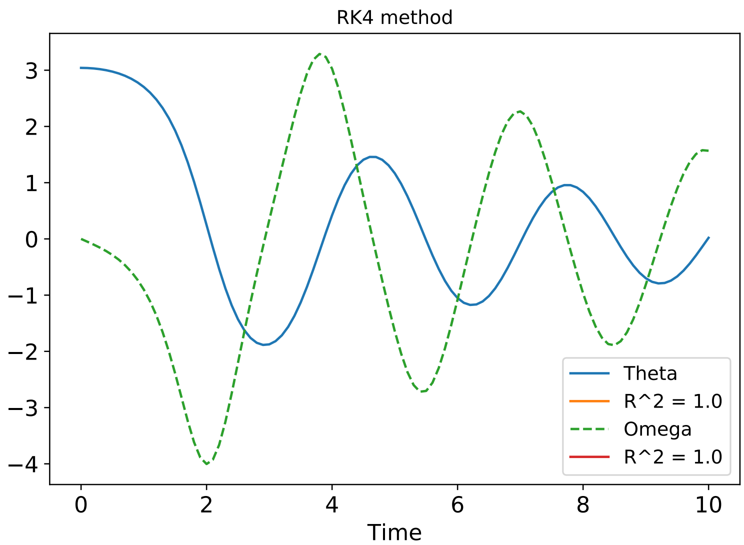

FIG. 3: RK4 method for the damped pendulum.

than the initial pressure, it doesn't change it at all. The initial pressure is on the order of magnitude of 10^{19}, while dP/dr calculated at even very small r is on the order of magnitude 10^5. So when the calculation $y = y_{i-1} + (1/6) * h * (K1 + 2K2 + ...)$ is made, it subtracts (since K1, K2, K3, K4 end up being negative) such a small number that the initial value for pressure doesn't change at all. It could be that I'm just having conceptual issues, I'm not sure. Otherwise, I wasn't able to move on to try solving the equations for the neutron star because of this. Figures 7 and 8 show the plots that I got as a result of these errors. The plot of pressure doubles initially because K1 ends up being just the value of the initial condition. I had to roll back the changes I made to RK4 in order to make these plots because it messed up the solutions to the stiff ODE and the damped pendulum, so if you try to run my code as it is now, you'll get an error for the pressure and mass calculations. This error comes up because it can't evaluate the function at r = 0.

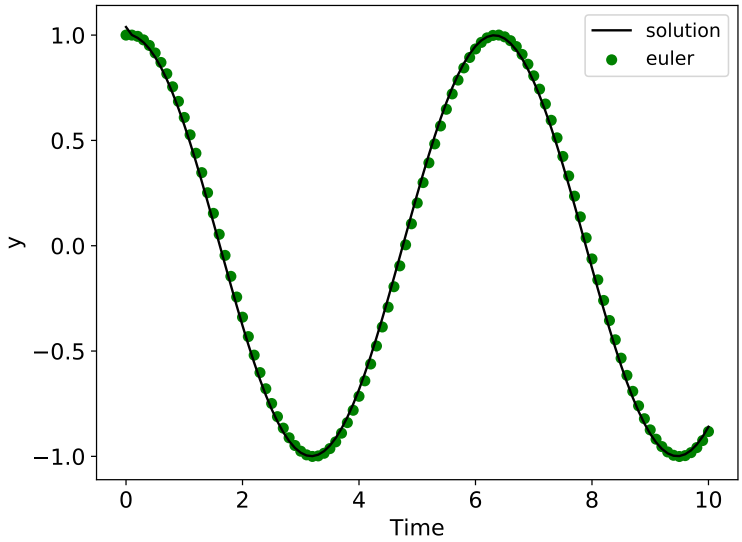

FIG. 4: Euler method for the stiff ODE

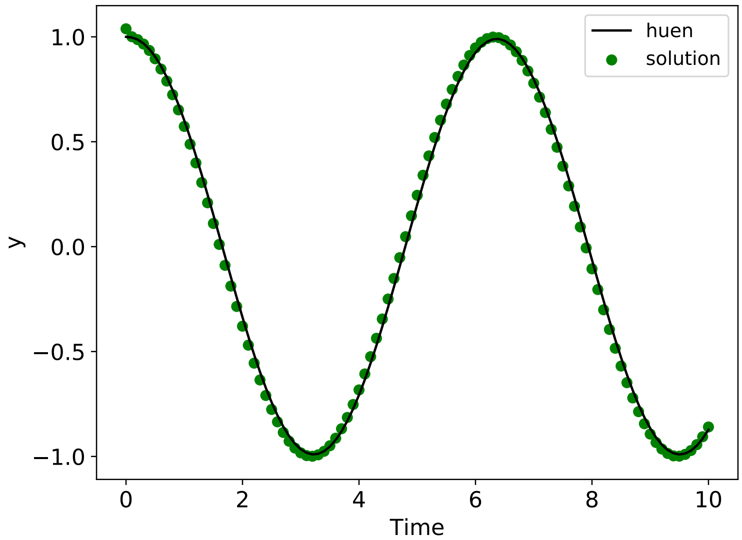

FIG. 5: Heun's method for the stiff ODE

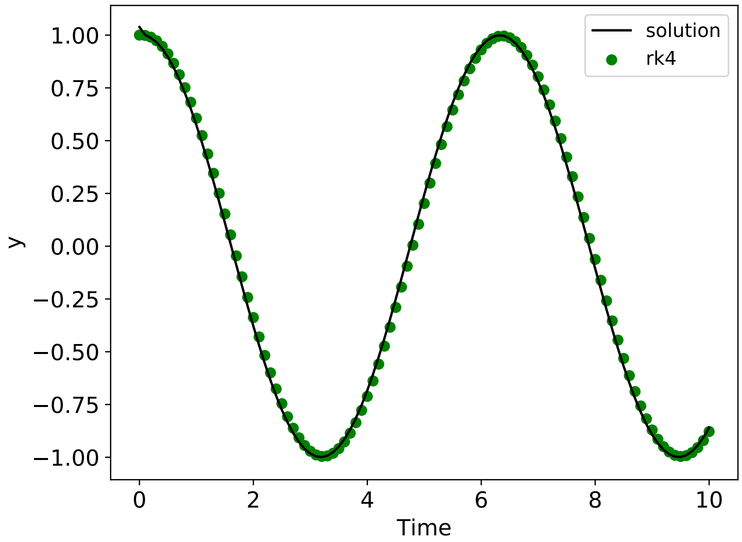

FIG. 6: RK4 method for the stiff ODE

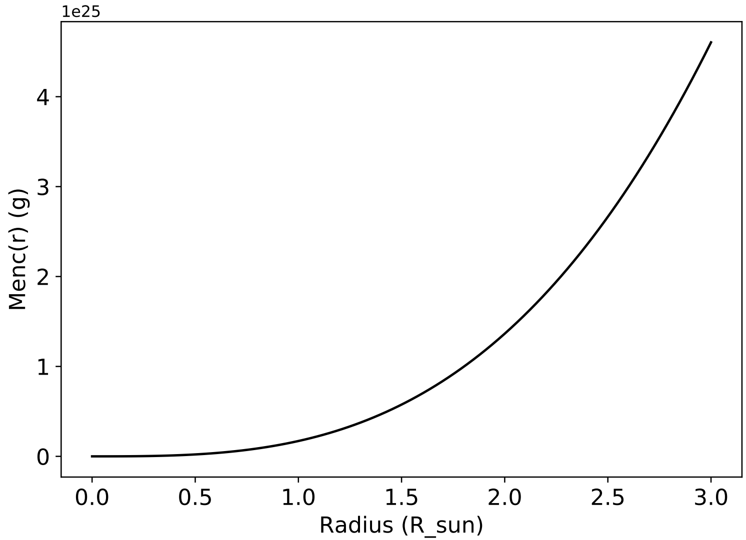

FIG. 7: The label on the x axis should actually be "Radius (`R_Earth`) since it is parameterized in terms of Earth radii.

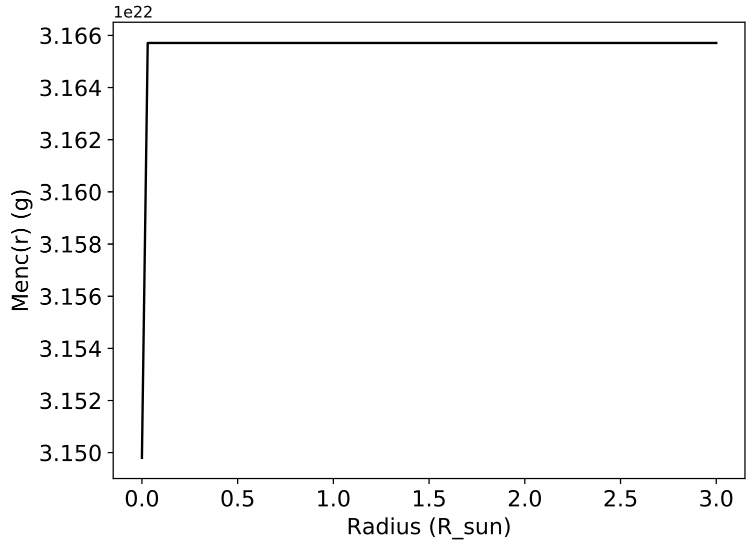

FIG. 8: The label on the x axis should actually be "Radius (`R_Earth`) since it is parameterized in terms of Earth radii. Also the Y axis should be Pressure in terms of g/cm^2, curse copying and pasting axis labels.