Link to HW3 code

1 Task 1

The unit testing for the matrix class can be found in **test_mat.py**. It tests all of the methods of the Matrix class, including the determinant which has been fixed to work generally between now and the last homework submission.

2 Task 2

We start with values for A_{ul} which are loaded in from the provided coefficients file. We relate A_{ul} to B_{ul} and B_{lu} using Equations 2 and 3 from the assignment and with some re-arranging find that

$$B_{ul} = A_{ul}\frac{c^2}{2h\nu^3} \tag{1}$$

$$B_{lu} = B_{ul}\frac{g_u}{g_l}, \tag{2}$$

which, when combined with $\bar{J}$ given by

$$\bar{J} \approx \frac{2h\nu_0^3}{c^2}\frac{1}{e^{\Delta E/kT}}, \tag{3}$$

where $\Delta E = -13.6(\frac{1}{n_1^2} - \frac{1}{n_2^2})$, gives the required components to write the equations dictating equilibrium in the system.

By writing out the first few sets of equations, a pattern emerges which allows us to set up M, the matrix of equation terms:

- if row $>$ col: $M_{ij} = -B_{ij}\bar{J}_{ij}$
- if row $<$ col: $M_{ij} = -(A_{ij} + B_{ij}\bar{J}_{ij})$
- if row $=$ col: $M_{ij} = \sum_{k=j}^{N} B_{ik}\bar{J}_{ik} + \sum_{k=1}^{j} B_{ik}\bar{J}_{ik} + \sum_{k=1}^{j} A_{ik}$.

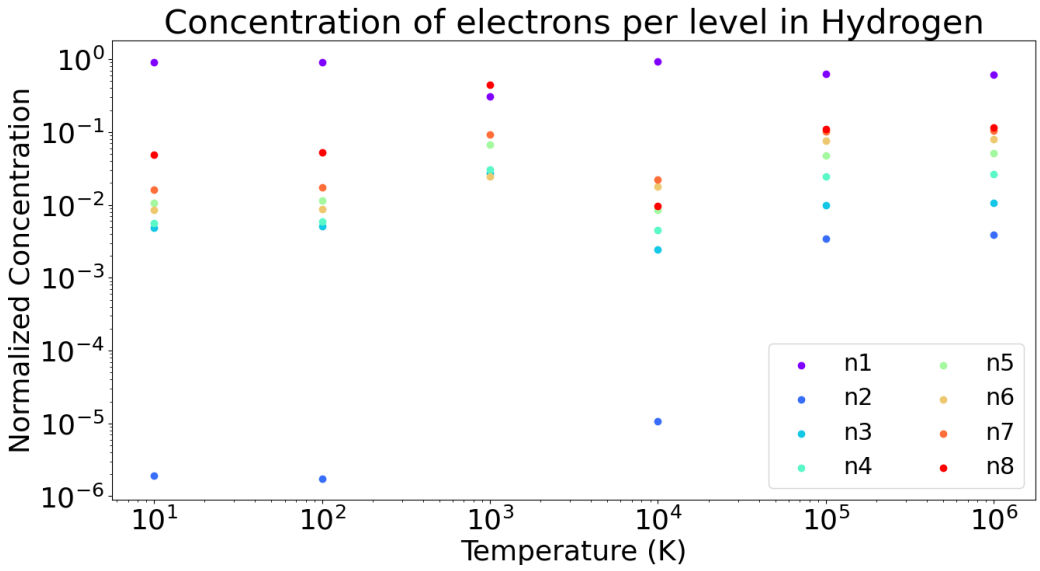

Figure 1: Concentration evolution as a function of temperature.

The caveat here is that (I believe) the $>$ and $<$ are supposed to be flipped, however doing that produces a diagonal matrix which results in a trivial (and incorrect) solution, so I chose to flip the inequalities to get an interesting answer from the problem.

By inverting this equation matrix, we produce the matrix by which we can multiply our **almost** zero matrix (representing the right-hand side of the equation from which all terms have been subtracted) in order to get our concentrations.

By iterating over temperatures spaced logarithmically between 10 and 10^6 Kelvin, we obtain the concentration evolution shown in Figure 1.

The result makes sense in some regard, such as the n1, being the lowest level, being most populated as it is the most energetically favorable level. However, the concentrations at high temperatures above $\sim$ 11000 K are suspicious, as at these temperatures, the average particle energy is above 13.6 eV and so one would expect that the concentrations would all plummet at these levels. Despite this, the increase in concentration of the higher energy levels at higher temperatures follows logically.

3 Task 3

The ODE solvers can be found in **odes.py**. They include Euler's, Heun's, and the Runge-Kutta 4th order method.

4 Task 4

The ODE solvers performed phenomenally compared to scipy's odeint function. Figures 2 and 3 show the overall performance comparison and the slight differences emerging from numerical differences stemming from the choice of solver, respectively.

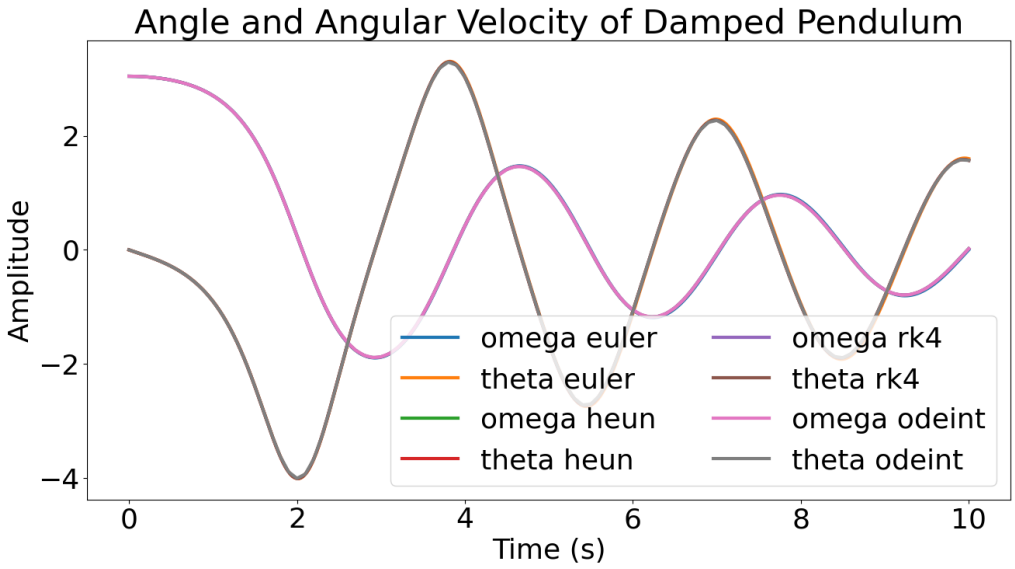

Figure 2: Performance comparison of my Euler's, Heun's, and RK4 ODE solving methods against the scipy odeint solver.

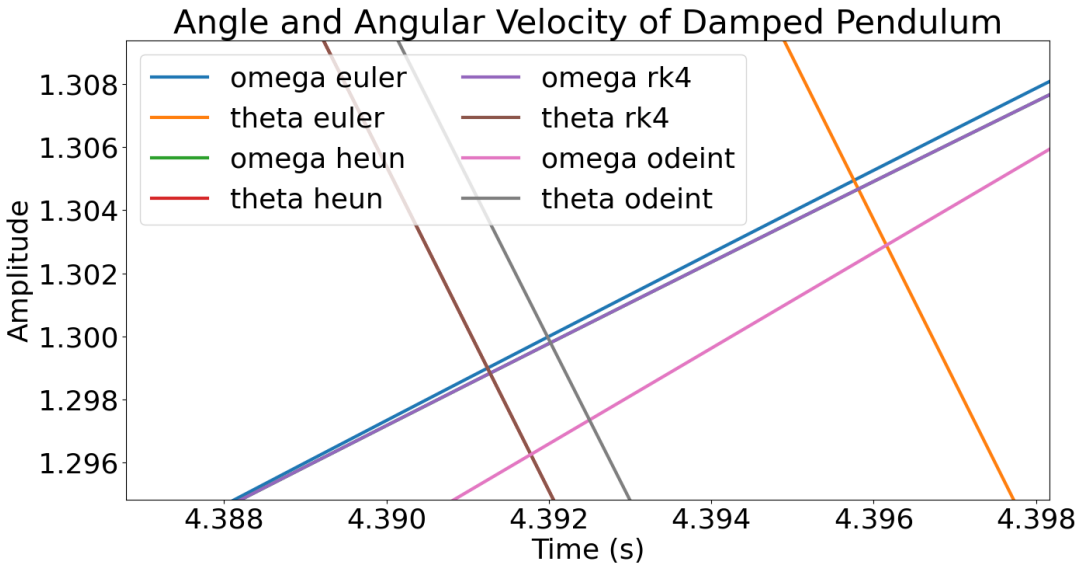

Figure 3: Zoom-in of Figure 2 to show slight numerical differences emerging from choice of method.

5 Task 5

The stiff ODE also provided little trouble to the solvers, with the exception that the starting value had to be something very small instead of 0. If this was not the case, the solvers would return a 0 solution for all times.

A stiffness parameter values of $\lambda = 10$ yielded the plot in Figure 4, where the equation is still not stiff enough to make a sharp peak at the start. This changes in Figure 5 when $\lambda = 1000$ as in the second plot, where the initial peak is very sharp. Zooming in on Figure 5 shows that this sharpness manifests from numerical discontinuities near the start, but is also a good display of how the different solvers handle the problem.

The stability of the methods seems uniform except for Euler's method, which is less smooth and more discontinuous as can be seen in Figure 6. The RK4 method appears to come closest to the solution which is unsurprising considering it uses the highest order of corrective terms.

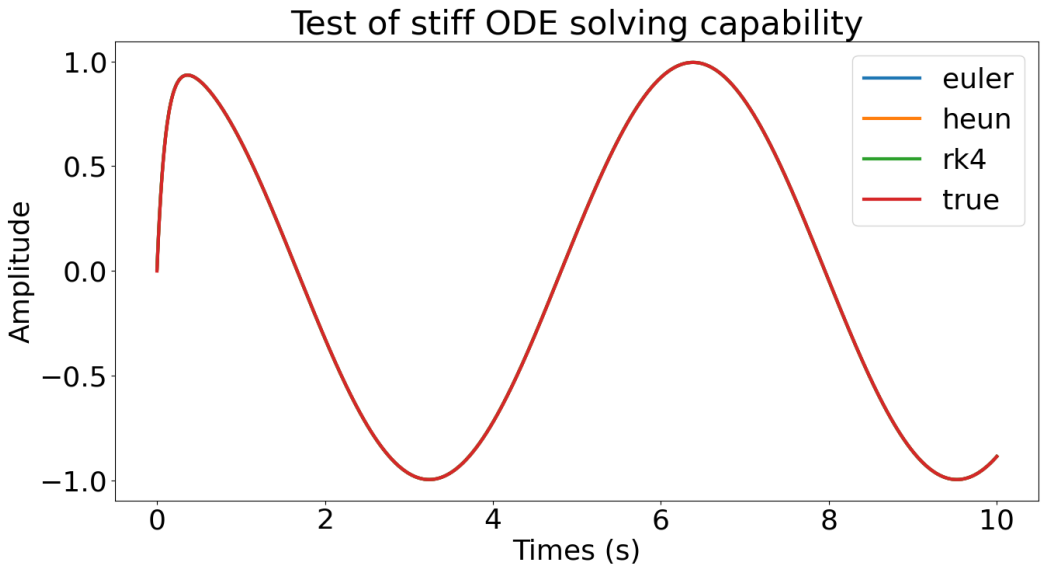

Figure 4: Stiff ODE with stiffness parameter $\lambda = 10$.

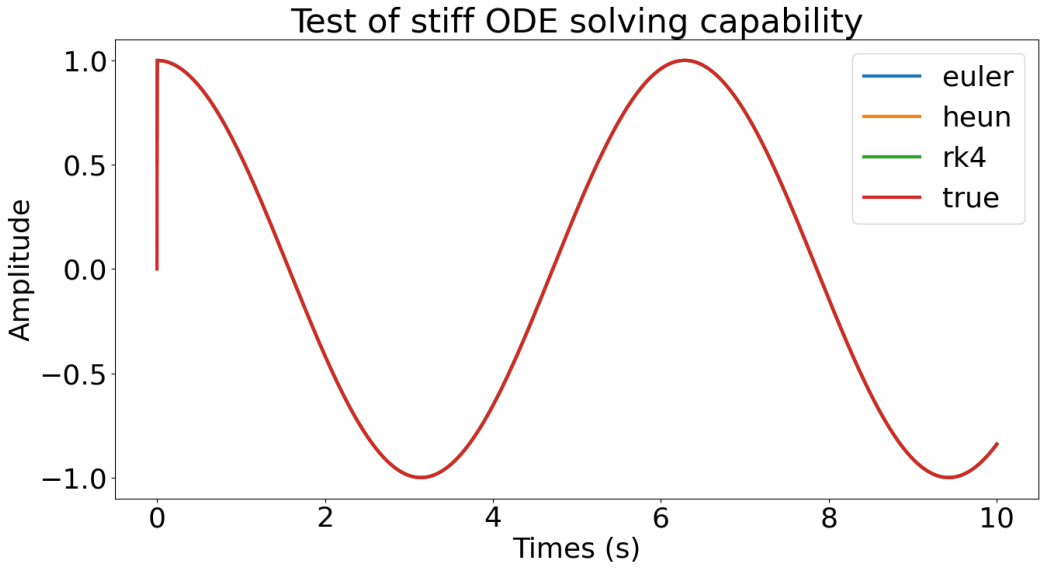

Figure 5: Stiff ODE with stiffness parameter $\lambda = 1000$.

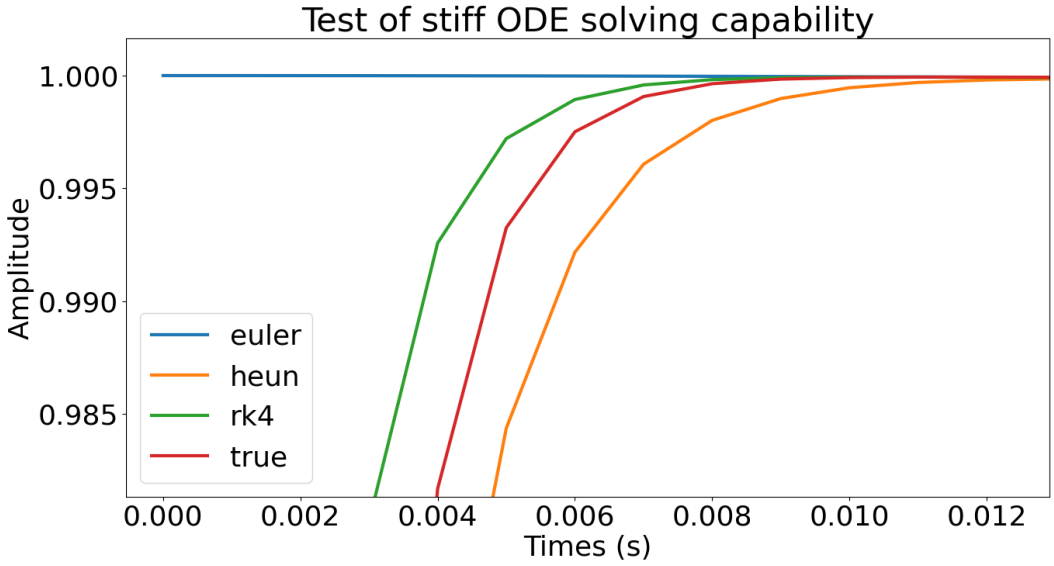

Figure 6: Zoom-in of Figure 5 showing numerical differences between the solvers as well as the reason for the increased sharpness of the peak as λ increases.