

Homework 8, ordinary differential equations

I have done all parts

I use my generic list class from one of the previous homeworks to store the output of every substep.

PART A

Output: The only output are the images pendulum.png sin.png showing my ODE results.

In part A I (1) implemented the Dormand Prince 5-4 embedded stepper, and (2) an adaptive stepsize driver.

As an example, I have solved (3) a simple problem, in this case $y' = -y$ (should be $\sin(x)$) (4) the requested problem, pendulum with friction. The demonstration that this is working are the two figures with the sine-curve and the recreated graph of the pendulum problem.

I did, originally, attempt to implement a generic ode stepper and driver, but C# –unlike C++ – can not accept statements such as `Data* 0.5` where `Data` is a generic type, likewise, C# does not allow addition of generic types. So the Dormand Prince stepper is not as optimized as it could have been in C++, yet another reason I will likely not use C# again after this course unless specifically told to do so.

For this reason, my ode uses a custom variable length list of doubles class, which combines features from the vec and genlist exercises.

I was supposed to first implement the output of the entire path in part B, but I really don't see how I could reproduce the plot from Scipy without doing that in A already.

Do note that I implement a max step size; this is merely done for graphical purposes (we might want to plot the path of the particle, and hence we need somewhat short steps). This can be turned off.

PART B

Output: Again, the only output are the images pendulum.png sin.png showing my ODE results. They are identical to what I got in A, though the underlying method for calculating them is slightly differently

In part B I store all the points each step (actually, I did that in A as well) and I check the error on a per-point basis (Actually that was the way I originally made it) I recreate the same plots to demonstrate that this works as well

PART C

Output: Again, the only output is an image showing the path the planets take. I did not see any reason to output any text files.

In part C I reproduce the stable gravitational system, do note I use 2D vectors, rather than 3D, as it all happens in a plane. I use the conditions in the paper and get the same infinity-symbol loop as they got.