1 Comments

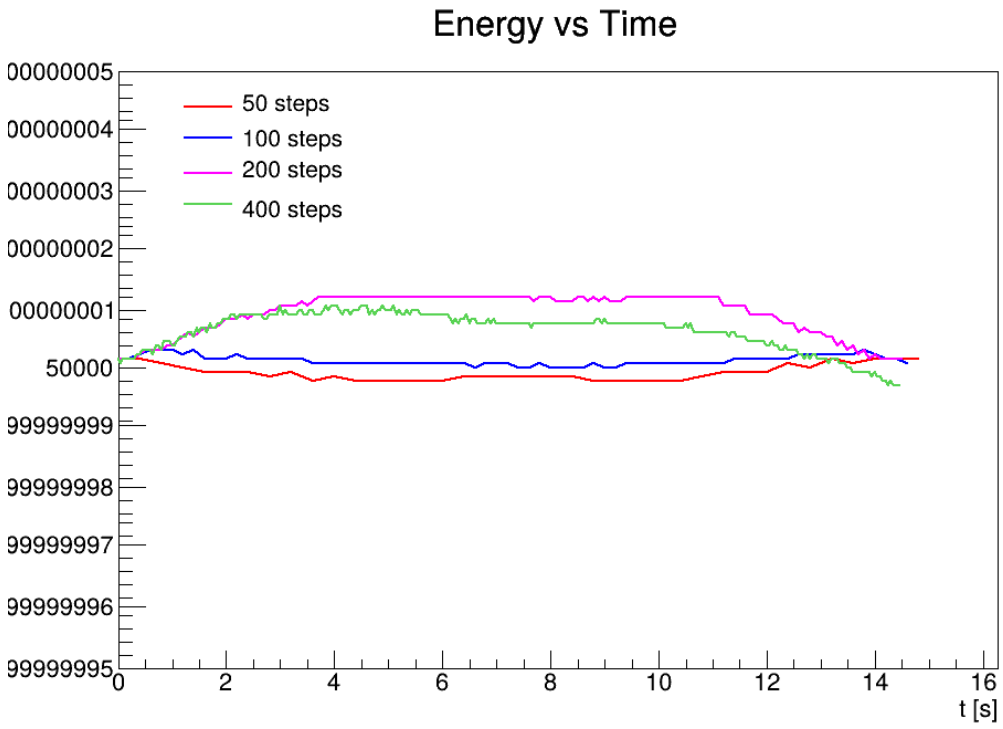

Figure 1: Energy vs Time for different steps

So I calculate energy as $\frac{m}{2}(x^2+y^2)+mgy$ and changed step size by changing the number of steps. In RKnDemo.cpp, the step size is variable (" // example of variable step algorithm, here the estimate accuracy is limited to 1e-4 // in the plot you will see the change in step size thoughout the time interval"), but increasing the total number of steps should always decrease the step size, so changing N should be enough to look at how step size influences total energy. For all the step sizes I tested, they clearly center around 50,000 J and they have end behaviors of increasing and decreasing for small and large time. N=100 seems to be the most steady, but they all are pretty flat. This is good because we could see a non-physical effect of energy not being conserved due to numerically solving the ODE. However our algorithm is effective and suppresses that potential effect.

RKnDemo.cpp already has terminal velocity implemented for the default parameters, so executing that code gives the output "Final velocity = 27.6553", so a terminal velocity of 27.6553 m/s. The next step is to plot terminal velocity as a function of mass, which I do in vterm2.cpp. Terminal velocity implies that $\frac{dv}{dt} = -g + \frac{k}{m}v^2 = 0$ so we get that the terminal velocity is proportional to the square root of m, and that is the shape we see in the plot. The analytical

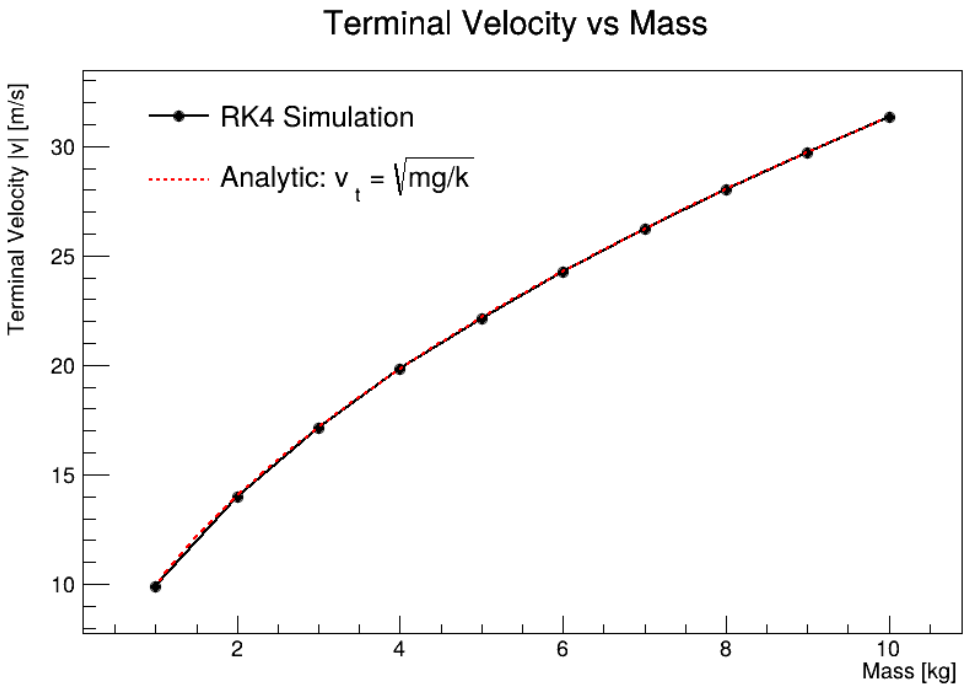

Figure 2: Terminal velocity vs mass with air resistance

solution is also plotted, and we see that they match nearly perfectly.

My plots seem well behaved, but the air resistance model could be validated with some additional plots. For example, explicitly checking terminal velocity with drag vs without drag can show if air resistance behaves as we expect (slowing it down).