EngSci 331 Lab3 ODE – Daniel Clark – 343733502

Anything interesting, unexpected, or complex in the implementation:

* The fixed step solver was harder than I expected, and I ran into issues using numpy arrays where doing y\_next = y[:,k], as this seemed to back-assign y\_next to y[:,k] as well, which meant I ended up adding my new y-values in both the k and k+1 spots. Eventually I figured this out and used np.copy() to fix it.
* The adaptive step solver was easier than I expected after I understood the general idea. I found a good YouTube video which helped me: <https://www.youtube.com/watch?v=6bCBXvsD7gw>

These are my code comments copied into this ‘report’ (no one answered my piazza question so I’ve decided to do both):

**TASK 1:**

# The Bungy cords that will fully dunk the engineering student are the Reg50 and Reg60  
# cords. I will choose to use the Reg60 cord as it dunks the engineering student for  
# the least amount of time, so is the safer option of the two.

# The jumper first hits the water after 3.35 seconds  
# with a velocity of 9.98 m/s

# If the scales are misread as 67kg when the real weight is 85kg, then there could  
# be very serious consequences for the jumper. I have calculated that the 85kg jumper  
# would hit the water with a speed of over 15.1m/s, which is significantly more than  
# the 9.9m/s of an actual 67kg jumper.  
# The jumper also reaches the water faster, in 3.15 seconds as opposed to 3.35 seconds.

**TASK 2:**

# No, I do not expect the system to settle down into a steady state long-term, because it is  
# a chaotic system. Systems that exhibit chaotic behaviour do not tend to converge to steady  
# state solutions, and will instead bounce around in an unexpected manner, never reaching  
# a steady state.

# The plots from step 6 show that a very slight change in starting condition can  
# give drastically different results after a period of time. This means that any  
# numerical error that is introduced by our numerical solvers will become compounded  
# over time. AKA it will be highly sensitive to the numerical accuracy of our  
# solution method. Therefore we cannot expect to accurately predict the exact values  
# for x, y, and z at times that are far into the future, for a chaotic system like  
# this.