**Problem 1**

Solve the population DE:



With alpha = 1.444 x 10-5 and M = 1000. Go from t = 0 to t=25. Change in t = 0.0001. Use both the forward Euler and the Rugge-Kutta methods to solve. Make a plot of the solutions.

Python 3.9 was used for this program. Additionally, I used several libraries to help solve. This program can be run by running Test 2.1.py. I used Visual Studio Code to code and run.

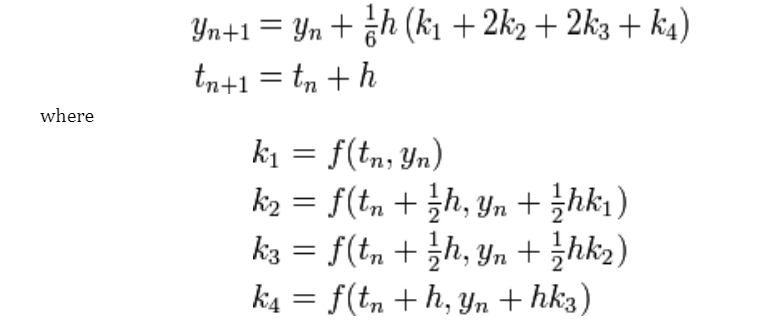
**Algorithm Description**

I followed the algorithm showed in the notes.

This is the formula I used for forward Euler was yn+1 = yn + hf(t2, yn). I substituted in the values, and this was my formula:

y.append(y[i] + h \* (-1.444E-5)\*y[i]\*(1000 - y[i]))

This was the formula I used for RG:



Plugging in the values, this is was my formula:

k1 = (-1.444E-5)\*(y1[i])\*(1000 - y1[i])

    k2 = (-1.444E-5)\*(y1[i] + 1/2 \* h \* k1)\*(1000 - (y1[i] + 1/2 \* h \* k1))

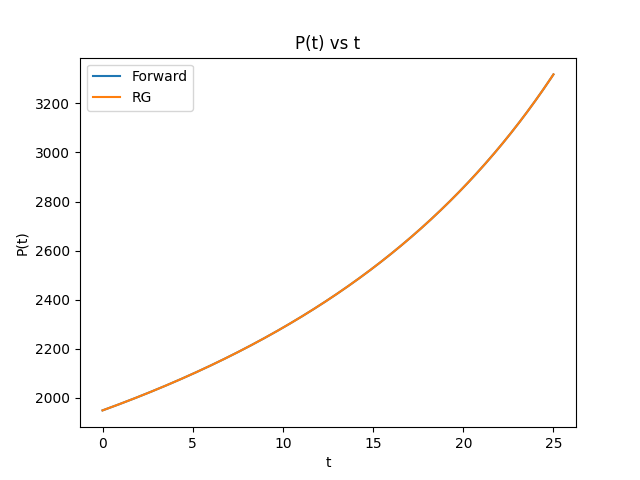
    k3 = (-1.444E-5)\*(y1[i] + 1/2 \* h \* k2)\*(1000 - (y1[i] + 1/2 \* h \* k2))

    k4 = (-1.444E-5)\*(y1[i] + h \* k3)\*(1000 - (y1[i] + h \* k3))

    y1.append(y1[i]+1/6\*h \* (k1 + 2\*k2 + 2\*k3 + k4))

**Results**

This is the graph formed:



You cannot see the Forward Euler line, because it is so close to the RG line.

At t = 25:

Forward = 3317.88

RG = 3317.96

**Performance**

This program takes around 10 seconds to run. Although I could have avoided lists for a faster run time, this is not really a problem in this case.