Problem 1

Solve the population DE:

$$\begin{cases} \dfrac{dP}{dt} = -\alpha P(M - P) \\ P(t = 0) = 1949 \end{cases}$$

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 $y_{n+1} = y_n + hf(t_2, y_n)$. 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:

$$y_{n+1} = y_n + \tfrac{1}{6}h\left(k_1 + 2k_2 + 2k_3 + k_4\right)$$
$$t_{n+1} = t_n + h$$

where

$$k_1 = f(t_n, y_n)$$
$$k_2 = f(t_n + \tfrac{1}{2}h, y_n + \tfrac{1}{2}hk_1)$$
$$k_3 = f(t_n + \tfrac{1}{2}h, y_n + \tfrac{1}{2}hk_2)$$
$$k_4 = f(t_n + h, y_n + hk_3)$$

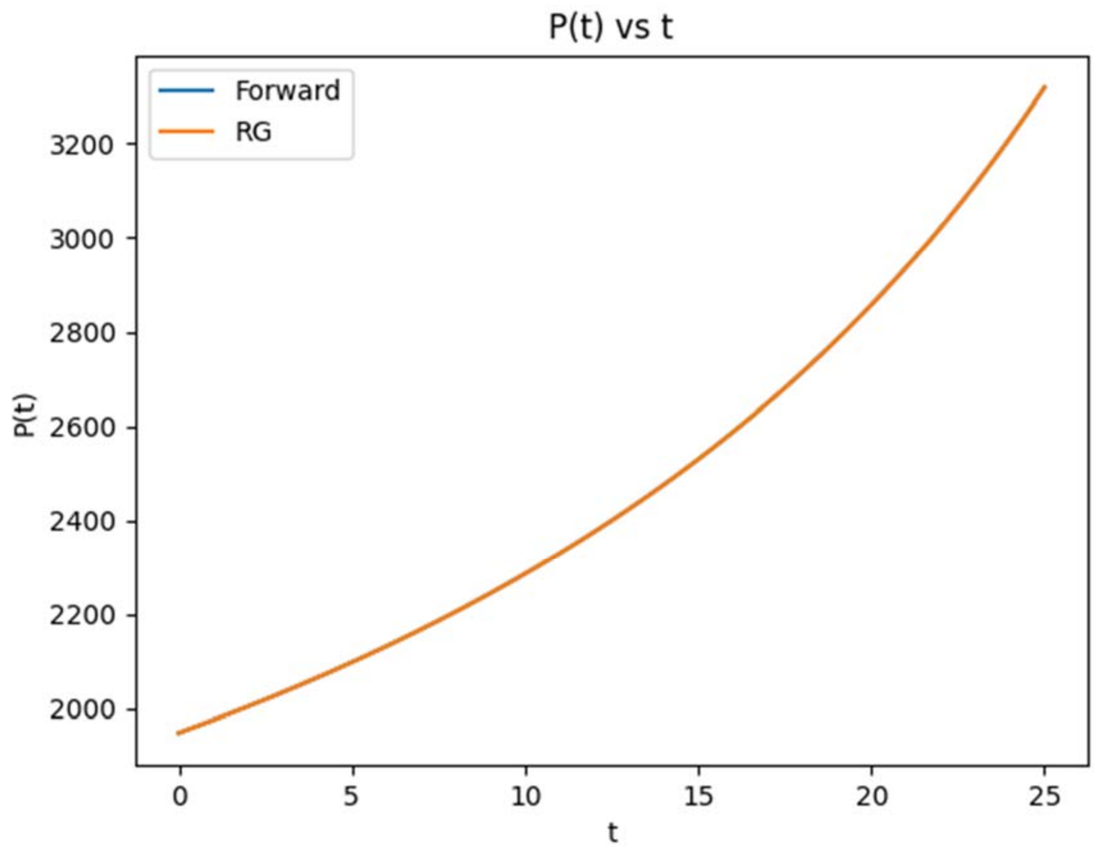
Plugging in the values, this is was my formula:

```
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.