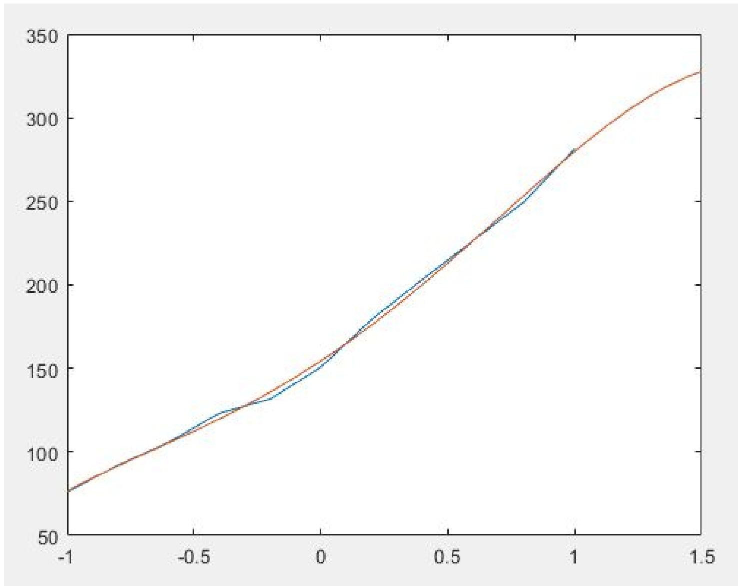
Jakob Horvath
u1092049



Least Squares Residuals:

1	2	3	4	5
29.5906	14.3406	10.4128	9.0487	7.8438

```
 degree = 4
Solution of the system is 154.504103 100.365654 36.101155 1.263295 -12.432195
 norm residual 9.048683 in 327 iterations
 Estimate for 2010 pop: 303.332126
 Estimate for 2010 pop: 319.991445
>>
```

After scaling the values, the least squares approximation for the U.S. Census data was successfully computed. Using a tolerance of 1e-5, polynomial degrees of 2, 3, 4, and 5 were tested to evaluate the accuracy in predicting the U.S. population in 2010 and 2019. A polynomial degree of 4 produced the best result when judging based on their respective residuals as well as the accuracy of their population estimates. As can be seen above, the least squares residual was smallest for a polynomial degree of 5, however its estimates were significantly farther from the actual data. I found that an alpha value of 0.69 to be the best in terms of reducing the work.

```
Power iteration 1:
Power iteration 2:
Power iteration 3:
Power iteration 4:
Power iteration 5:
Power iteration 6:
Power iteration 7:
Power method converged in 7 iterations.

 Estimated largest eigenvalue of A: 16.116844
 Actual largest eigenvalue of A: 16.116844
```

Using the PowerMethod and a starting vector of [1; 2; 1] resulted in an accurate estimate for the largest eigenvalue of matrix A. I used MATLAB's eig function to produce the actual value and confirm my esimate's accuracy.

```
Power iteration 96:
Power iteration 97:
Power iteration 98:
Power iteration 99:
Power iteration 100:
Power method failed to converge in 100 iterations.

 Estimated largest eigenvalue of B: NaN
 Actual largest eigenvalue of B: 3.000000
```

An initial guess of [2; 3; 2] resulted in a NaN estimate for matrix B's largest eigenvalue. This can be attributed to the fact that the PowerMethod function quickly computes a vector with values too small to be understood. After the 36th iteration, MATLAB chooses to represent the vector as a list of NaNs.

```
Power iteration 1:
Power method converged in 1 iterations.

 Estimated largest eigenvalue of B: 1.000000
 Actual largest eigenvalue of B: 3.000000
```

Meanwhile, a starting vector of [1; -1; 1] meant that the PowerMethod converged to an answer after only 1 iteration. The reasoning for this is that the method ends up computing the same vector as the one it was given. It cannot update its value, and it therefore passes the tolerance check after 1 iteration.

```
Power method converged in 2 iterations.
Warning: Matrix is close to singular or badly scaled.
> In assign5_2 (line 34)


 Estimated smallest eigenvalue of A: 2.819614e-16
 Actual smallest eigenvalue of A: -9.759185e-16
```

Using MATLAB's linsolve inside the PowerMethodInv function as well as immediately following the function call was found to be the best way to obtain the smallest eigenvalue of matrix A. Although the numerical value does not seem to be similar, the estimated and actual values were both in the range of e-16.

```
Power iteration 14:
Power iteration 15:
Power iteration 16:
Power iteration 17:
Power iteration 18:
Power iteration 19:
Power iteration 20:
Power iteration 21:
Power iteration 22:
Power method converged in 22 iterations.

 Estimated largest eigenvalue of M: 0.910582
```

When considering the biology example, the largest eigenvalue of the resulting matrix turned out to be 0.910582. Since the value is less than 1.0, it is safe to assume that the population will eventually die out.

1
3.0495e-39
3.0140e-39
2.6480e-39
1.6334e-39

A starting vector of [100; 200; 150; 75] reduced to essentially 0, as seen above, when the original matrix was raised to a power of 1000. Therefore, the population would be considered 0 at the year 1000, and this aligns with my previous findings.

```
Power iteration 32:
Power iteration 33:
Power iteration 34:
Power iteration 35:
Power iteration 36:
Power iteration 37:
Power iteration 38:
Power iteration 39:
Power iteration 40:
Power iteration 41:
Power method converged in 41 iterations.

 Estimated largest eigenvalue of Mnew: 1.079689
```

After updating the death rate of the final group in the population to be 0.01, the largest eigenvalue was changed to 1.079689. Now that the value is larger than 1.0, the hypothetical population can be expected to grow in an unbounded manner.