Thomas Neyman
MATH 3310
February 4 2022
Homework 1

Question 1.)

a. The change in y_n is measured by $-0.4y_n$ so for every n, y will decrease by 40% of its value.
   If the starting value of $y_0 = 100$, $y_5 = 100 * (0.4^5)$: $y_5 = 1.024$

b. Recursion Formula: $y_{(n+1)} = y_n * 0.4$:
   $y_0 = 100$
   $y_1 = 100 * 0.4 = 40$
   $y_2 = 40 * 0.4 = 16$
   $y_3 = 16 * 0.4 = 6.4$
   $y_4 = 6.4 * 0.4 = 2.56$
   $y_5 = 2.56 * 0.4 = 1.024$
   etc...

Question 2.)

Using - The population growth of a single species

Problem Statement: A population of fruit flies are contained in a closed system and being monitored at a laboratory. Scientists would like to know what the population can be expected to be after 14 days with a starting population of 20 flies.

Variables:

- food given to the flies
- the lifespan of a fruit fly
- temperature of the environment
- The ratio of males to females
- Humidity in the environment
- Light levels flies are exposed to

Relevant Variables:

- The temperature is an important variable because it is known to affect the development rate of the insect, so it needs to be accounted for.
- The starting ratio of males to females is important because 1 male could cause 19 females to lay eggs vs. 10 males and 10 females where only 10 females could lay eggs.

- The lifespan of a fruit fly is the final important factor to consider. If a fruit fly only lives a very short time and can be expected to die before the experiment is over, then fruit flies that are too old need to be subtracted from the end total.

All other variables can be assumed and would make the model more complicated than it needs to be. Food, humidity, and light levels could all be constant factors that don't have a major effect on the end result, or at least a minimal effect.

Question 3.)

Using - Which computer systems offer the most speed?

Problem Statement: A company would like to host their own web server to provide high quality support for their customers at any point in time and expect large amounts of traffic. The systems administrator of the company is tasked with choosing a computer system that will be reliably fast enough to meet these requirements. What specs would be needed for a maximum time-to-first-byte speed of 1 second with an average of 1000 clients using the web server at any given time?

Variables:

- The types of hard drives being used
- RAID configuration
- Number of CPUs, clock speeds, number of cores, available CPU cache
- Type and speed of network adapters
- RAM speed, size, quality, and reliability
- Temperature of the system

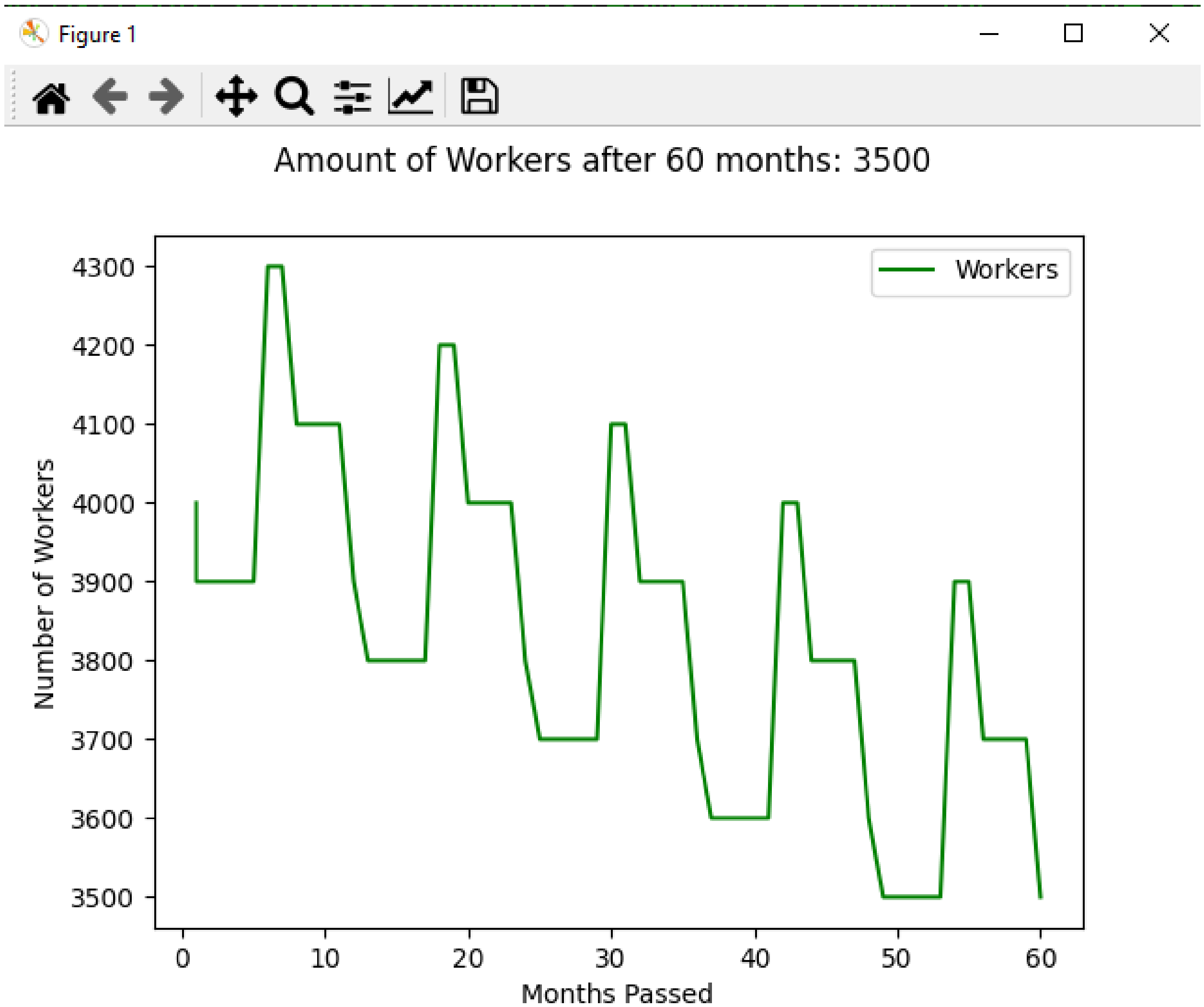
Relevant Variables: When determining the speed of a computer system, almost all factors play an important role in deciding the end result, but we can try to narrow the variables down to some constants to make the model easier.

- The CPU is extremely important and can't be overlooked. How fast the CPU can do calculations determines how fast a client gets a response.
- The amount of available memory and its speed also drastically affects how the server will perform. The quality and reliability of the RAM can be assumed to be constants.
- If traditional SATA hard drives are being used, the disk speeds and port speeds need to be accounted for. If NVMe data is being used, its clock speeds need to be accounted for.

If we simplify to only these 3 somewhat complex variables, it will make the model much more manageable. For constants, we can assume the RAID configuration will be RAID 0 no matter the implementation, the network adapter speeds will be 10Gb/s, and the temperature of the system will be controlled by the room the server is housed in, keeping it constant. All of these affect the speed of the server, but can be made into constants.

Submodel:
CPU performance = $f(\mathit{clock\ speed},\ \mathit{number\ of\ cores},\ \mathit{cache})$
RAM performance = $g(\mathit{clock\ speed},\ \mathit{capacity})$
HDD/SSD performance = $h(\mathit{SATA\ Type},\ \mathit{RPM})$
Model = $k(\mathit{CPU},\ \mathit{RAM},\ \mathit{HDD/SSD},\ \mathit{constants})$

Question 4.)



This graph shows the trend of workers employed at the power plant over a course of 5 years, or 60 months. At the end, we can see that the final result is 3500 workers remaining after 5 years. This means there is a drop of 100 workers every year. This is a simple model and makes many assumptions. We assume that the amount of workers changing is the exact same every year and we don't account for any outside factors that may affect the amount of people changing. We also assume that beside the key times of change throughout the year, the population of workers is constant otherwise and doesn't change.
The Python code used to generate this plot:

```python
def problem4():
    # The number of years to repeat the model
    N = 5
    # starting number of workers working at the power plant
    workers = [4000]
    # start time at 1 just so it starts at 1 on the plot instead of 0
    time = [1]

    for i in range(N):
        for j in range(12):
            if j == 0:
                # 100 workers leave in January
                workers.append(workers[j + (i * 12)] - 100)
            elif j == 5:
                # 400 students arrive in June
                workers.append(workers[j + (i * 12)] + 400)
            elif j == 7:
                # 200 students leave in August
                workers.append(workers[j + (i * 12)] - 200)
            elif j == 11:
                # 200 workers retire in December
                workers.append(workers[j + (i * 12)] - 200)
            else:
                # otherwise the amount of workers stays the same
                workers.append(workers[j + (i * 12)])
            time.append(j + (i * 12) + 1)

    arrayLength = len(workers)
    result = "Amount of Workers after {} months: {}".format(arrayLength - 1,
workers[arrayLength - 1])

    plt.plot(time, workers, color = 'green', label = 'Workers')
    plt.legend(loc = 'best')
    plt.ylabel("Number of Workers")
    plt.xlabel("Months Passed")
    plt.suptitle(result)
    plt.show()
```

Question 5.)

Out of a population of 6000 people, we start with 5090 people who are susceptible to the disease, 10 who are infected, and 0 who have recovered. We will track the progression of the disease in weeks. We need to assume some things for this model.

- The population never changes from 6000
- The average length of the disease will be 4 weeks
- After going into the recovered group, you are unable to go back into infected

Because the average length of the disease is 4 weeks and the model is being tracked for every week, the number of people moved out of the infected group every week will be the reciprocal of 4, so 1/4 or 25% move out of the infected group each week.

The model for the interaction will look like this:

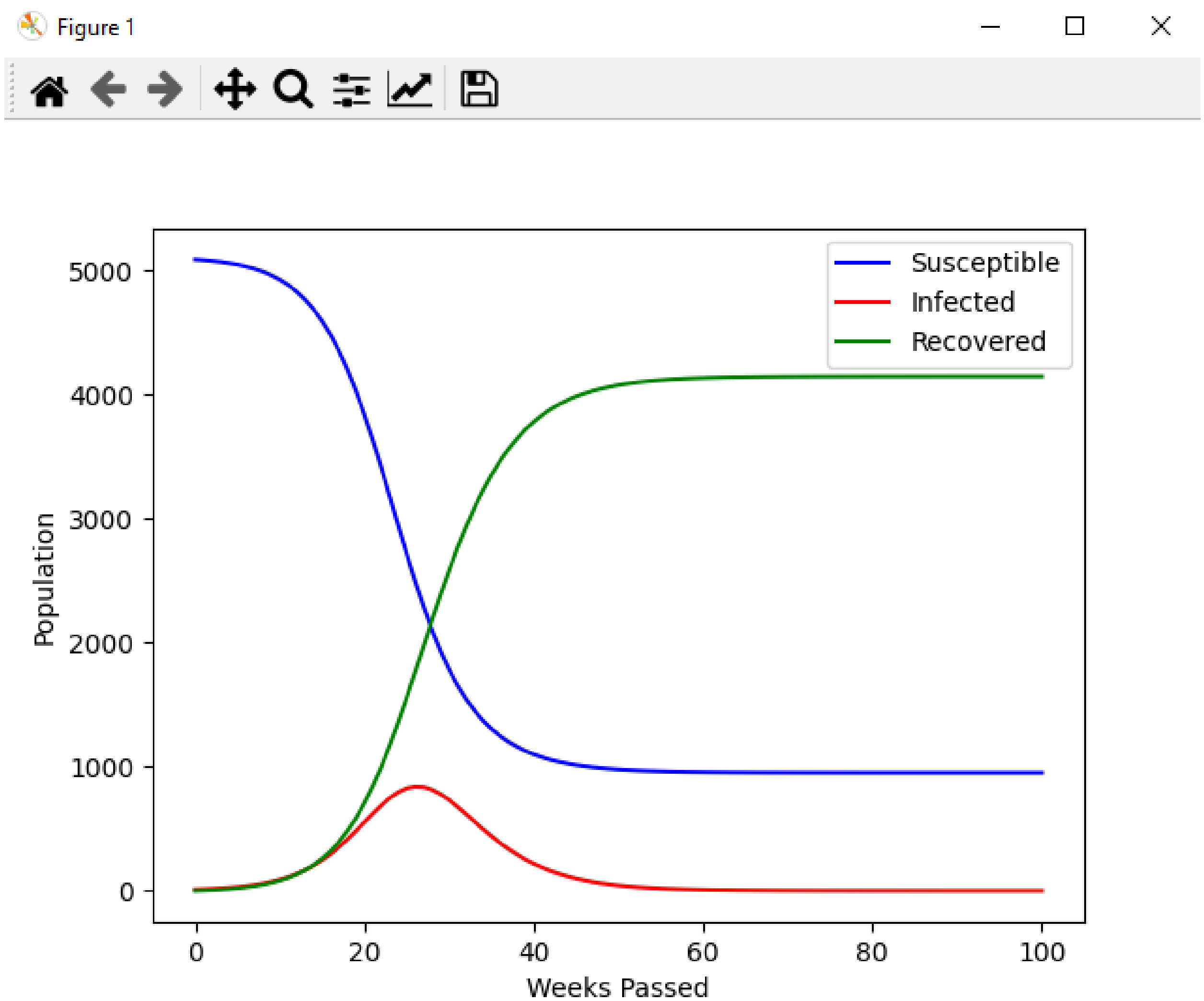
$$S_{t+1} = S_t - bS_tI_t$$
$$I_{t+1} = I_t + bS_tI_t - aI_t$$
$$R_{t+1} = R_t + aI_t$$

Variable *a* will represent the percent of people who will go into the recovered group with each passing week. So *a* = 0.25.

Variable *b* represents the interaction coefficient. So to get our *b* value, we calculate:

$$I_1 = I_0 - 0.5I_0 - bI_0S_0 == 10 - 0.5(10) - b(10)(5090)$$

We end up with *a* = 0.25, and *b* = 5/50,900 ≈ 0.0000098 Our interaction coefficient is very small. When the model is input into Python, we get the following graph for the first 100 weeks:



From what the graph returns, we can see our model is clearly reaching an equilibrium somewhere between 60 - 70 weeks. Upon close inspection, equilibrium is being reached around 64 weeks. This equilibrium represents a point where the sickness will no longer be spread so the amount of recovered and susceptible people remain the same, while the amount of infected people will not grow any larger than 0.

As for the realism of the values, they aren't the most realistic. Being sick for 4 weeks straight is a very long time and most sicknesses do not last that long at all. However, with this model, changing the length of the sickness to something like 3 days causes the model to not behave well. 3 days would be a length of 3/7 which has a reciprocal of 7/3. If a is given a value larger than 1, the model fails to work.

Code:

```
def problem5():
    # for an entire population of 6000 people
    N = 6000
    # number of susceptible people, recovered people, and infected people
    S, R, I = [5090], [0], [10]
    # recovery percent and interaction coefficient
    a, b = 0.25, 5 / 50900
    # the number of weeks
    week = [0]

    j = 0
    while j < 100:
        S.append(S[j] - b * S[j] * I[j])
        I.append(I[j] + b * S[j] * I[j] - a * I[j])
        R.append(R[j] + a * I[j])
        week.append(j + 1)
        j += 1

    plt.plot(week, S, color = 'blue', label = 'Susceptible')
    plt.plot(week, I, color = 'red', label = 'Infected')
    plt.plot(week, R, color = 'green', label = 'Recovered')
    plt.legend(loc = 'best')
    plt.ylabel("Population")
    plt.xlabel("Weeks Passed")
    plt.show()
```