Epidemiology Simulation of COVID-19

Understanding the Spread
of a Global Pandemic

Final Report
Spring 2021

Group Members:
Spenser Cheung
Tushar Goel
Christopher Magana
Steven Nguyen

Professor:
Ningfang Mi

Northeastern University
EECE 5643: Simulation and Performance Evaluation

Table of Contents

Abstract

COVID-19 has cemented itself globally as the worst pandemic to occur in recent history. It has impacted the lives of billions of people worldwide. It is essential to understand the implications of a pandemic as well as the spread. Measures to control the spread of the virus have lagged behind contagion progression. Failure to understand the pandemic has resulted in the loss of thousands of lives in the United States and millions worldwide. As such, it is absolutely crucial for data to be quickly compiled and analyzed for better transparency and information in regards to the virus. The problem we aim to solve is to predict the end of the pandemic. Based on current and projected rates of infection and dosage distribution our simulation aims to predict when herd immunity is reached within the United States. The project consists of two components: a Python web scraping script and a MATLAB simulation script. We have developed a MATLAB simulation tool to help predict future trends, specifically herd immunity and COVID-19 remaining duration, based on various parameters such as social distancing, transmissibility, and vaccination rates. These simulation parameters are either taken directly or inferred from data on the CDC website via a Python web scraper. When executed, the program will spawn mobile nodes which will interact with each other. Based on their interactions, we are able to predict determinants such as vaccination herd immunity dates and COVID-19 duration to better understand the impact of vaccination rates and social distancing on a given population.

Introduction

COVID-19 has likely established itself as the worst infectious disease pandemic of a generation. As of April 17 the total number of cases exceeds 30 million. The widespread infection leads to a risk of overwhelming healthcare services. Decreased access to medical care could prove fatal to those at high risk increasing the number of deaths. Economic consequences from world-wide lockdowns have devastated many and the trillions borrowed to stimulate the economy set the dollar itself at risk for inflation. Children born today are likely to be paying off this debt with tax dollars for a pandemic many years into their future. Additionally, the stress and social isolation many experienced during the pandemic is not to be underestimated. Riots and protests erupted across the country amidst elections and the frequency of mass shootings increased by a significant amount.

Early on, as the threat became clear, governments relied on computer simulations to determine how best to reduce its impact. It is essential to understand the development of a global pandemic and due to global efforts humanity seems to be on the path to recovery. Measures to control the spread of the virus's early stages have lagged behind the progression of the disease. Epidemiological modelling of the spread of COVID-19 has been an essential tool in improving the decision making of government officials. The modeling environment provides the ability to test different solutions without experimenting on the real population. Developing these models can lead to better understanding of the nature of the disease's progression. This leads to more efficient implementation of solutions with better results in high pressure situations that are highly time sensitive. Figures 1-4 show the progression of COVID-19 over time.

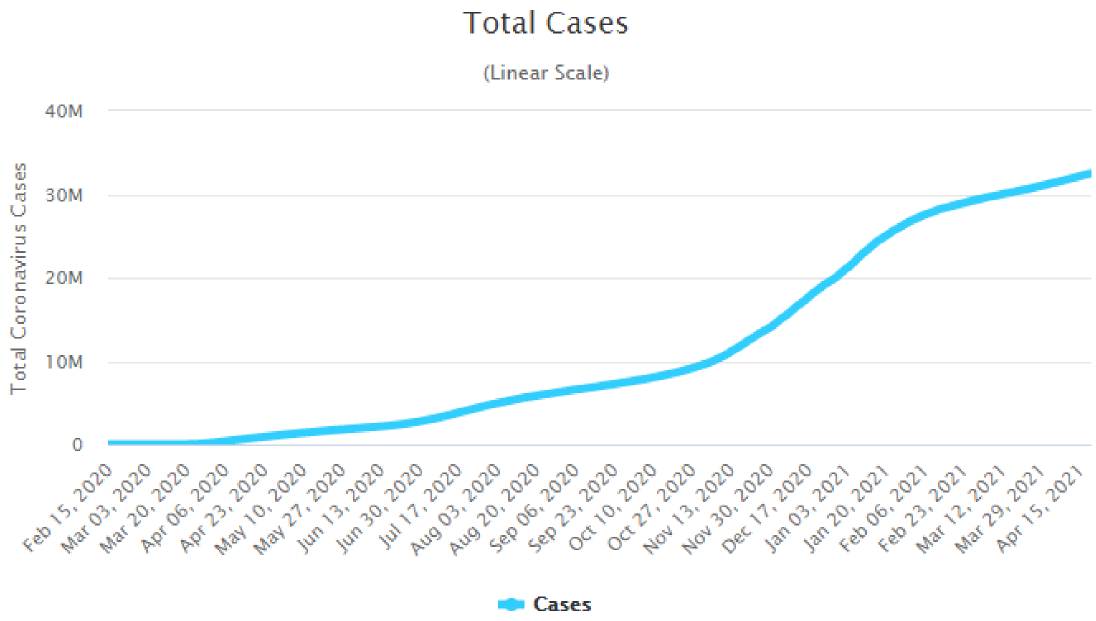


Figure 1: Total Coronavirus Cases, in the United States, from Feb. 15th to April 10th[7]

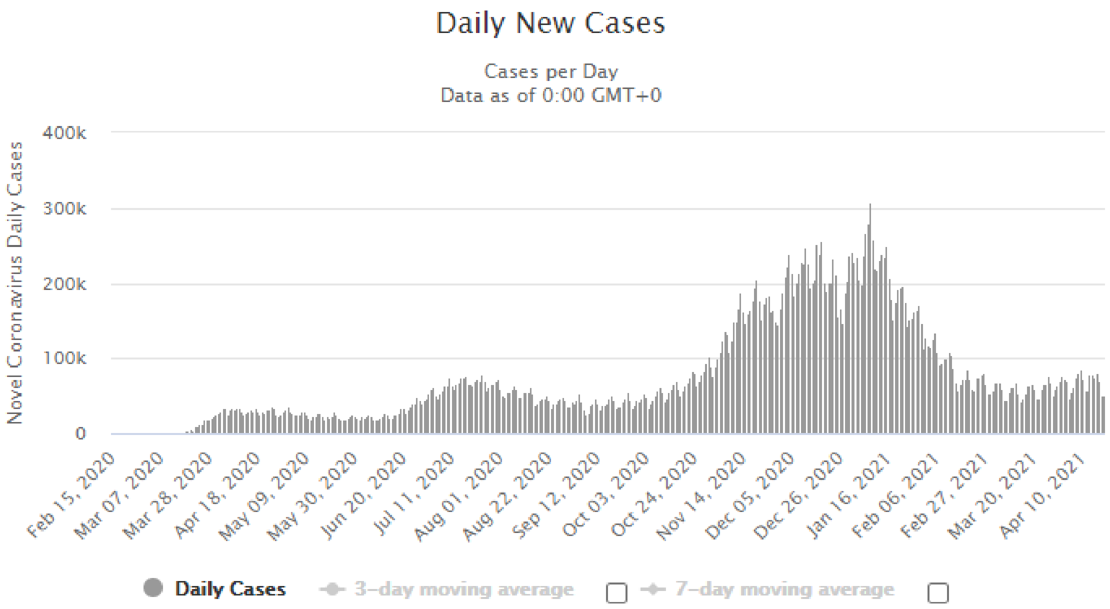


Figure 2: Daily new cases of the Coronavirus, in the United States from Feb 15th to April 10th[7]

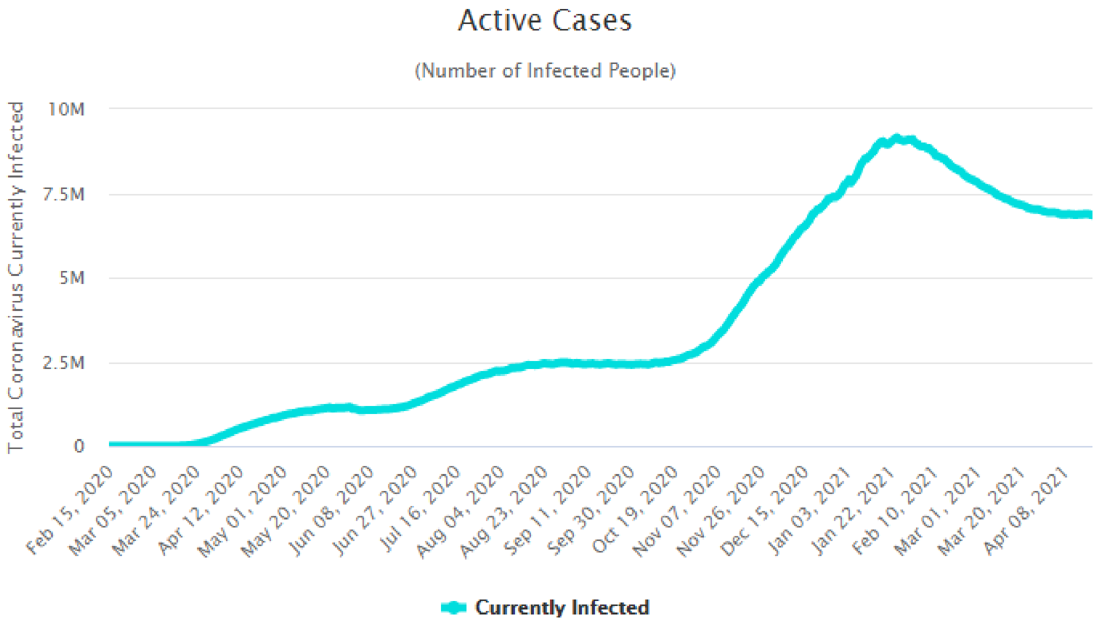


Figure 3: Plot of total number of individuals infected with the Coronavirus in the United States[7]

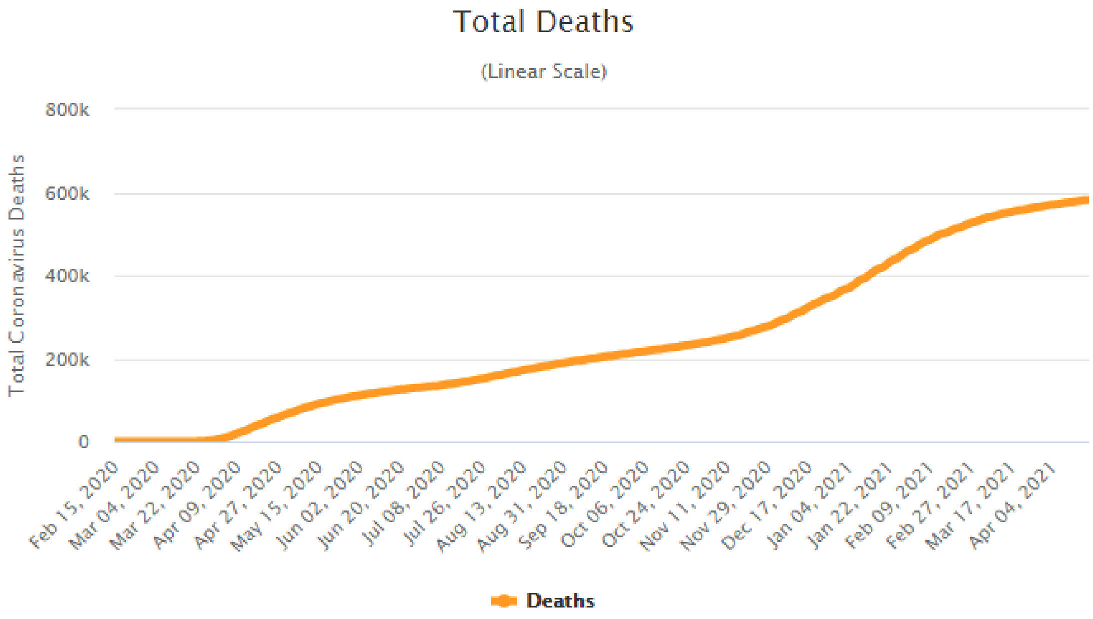


Figure 4: Plot of the total number of deaths caused by the CoronaVirus in the United States[7]

Different decisions may require different models. For example, simulating the disruption of supply chains would require a different model then a model being used to determine how best to move from an isolation period to normal everyday life.

Rapid development of a vaccine has been the hope for many since the early days of the pandemic. About a year later, despite the development of several vaccines, not every person is able to obtain one. The problem we aim to solve is to provide a live prediction of the end of the pandemic based on a live feed of current statistical analysis.

Background

COVID-19 was first reported to the world health organisation in December of 2019 and is a coronavirus disease caused by severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2). The disease has a large list of symptoms the most severe of which being pneumonia requiring respiratory support. The overall fatality estimate is between 0.24%-3.0% depending on available healthcare. While most cases are mild, older adults and those with underlying medical conditions, are at increased risk of severe illness. Lockdowns were put in place as the virus progressed.

Former president Trump imposed travel restrictions against China in order to keep the virus out of the US. There were holes in the travel bans, and it took weeks to ban travel to Europe, another hotbed of outbreak at the time. The main strategy to avoid spread has been lockdown. The US is considered to have taken its time in order to issue a lockdown order relative to other countries. Rather than examining the first day upon a death from COVID-19 a better start, after which countries need to begin acting seriously is when the death rates reach one death per million in population size. For example, Germany, France, and the United Kingdom imposed their lockdowns after reaching this point within a couple of days learning from Italy, which took almost a week and was experiencing spikes in infections across the country. For comparison, the United States, even 45 days later, had yet to impose a lockdown. The US did not act sooner, primarily due to lack of information. On january 21st the first case of COVID-19 was identified in the United States. However, test kits were just starting to be approved. Trump then made the statement that the situation was under control and it was just one person who was coming from China. Quarantine measures were performed however, ideally extensive testing on individuals in the area should have been performed however this was not possible. As the disease progressed, information in the US was always one step behind thus making it very difficult to control the virus. As the tests and information collection improved models were fine tuned and highly accurate. As models provided officials with better insights better decisions were made.

In the current stages of the pandemic, many have grown accustomed to the situation. It is in the best interest of American Citizens to continue, proper protocol, despite the inconvenience, in order to speed up the recovery of the nation. This model seeks to predict how much longer this will be necessary based on current rates and measures implemented. This information can be helpful to give Americans a sense of their performance against COVID-19.

Recent Modeling papers on COVID-19 focus on epidemiology attempting to estimate infection rates and to provide estimates on the effectiveness of different interventions. The SEIR model (Susceptible – Exposed – Infectious- Recovered) is one of the most popular models. Since it can describe the proportion of the population in each state at any given time. Hellewell et al. (2020), on the other hand, use a stochastic transmission model to assess the effectiveness of contact tracing and quarantine. (See the end of the appendix for references to papers examined).

Simulation Modeling methods

Four primary modeling methods include: system dynamics, Agent based modeling, discrete event simulation, and Hybrid Simulation. It is suggested that the choice of modelling method is made based on decision makers' requirements, type of problem and system complexity and its characteristics.

- **System dynamics (SD)**: differential equation based models that represent real world systems in terms of stocks (e.g., of material resources, knowledge, people, money), flows between these stocks, and information that determines the values of the flows (Borshchev & Filippov, 2004). Feedback effects and delays are key SD elements used to explain system behaviour. SD was first developed by Jay Forrester in the late 1950's to help managers better understand industrial problems (Forrester, 1961). The best-known within infectious disease epidemiology is the SIR model (Kermack & McKendrick, 1927). These models are typically used for strategic decisions or decisions affecting a whole population.
- **Agent based modelling (ABM)**: can be used to model the interactions of individuals within a population, allowing a decision-maker to determine how small changes in behaviour and interaction may influence population level outputs. In models of disease spread, the modelling of social networks and spatial movements are also vital for accurately describing transmission and these can be incorporated into ABMs. ABMs are stochastic models, enabling the variability of human behaviour to be incorporated into the model to help understand the variability in the likely effectiveness of proposed interventions. In an opinion piece, Epstein (2009) suggests that ABM is appropriate for modelling pandemics.
- **Discrete event simulation (DES)**: stochastic models that take account of variability in the time taken to carry out activities and the times between arrivals into the system. Goldman et al. (2010) point out that while the roots of DES go back to 1777, DES as we know it today was mainly developed in the years 1945–1970. DES models are typically used to model the operation of systems over time, where entities (people, parts, tasks, messages) flow through a number of queues and activities. They are generally suitable for determining the impact of resource availability (doctors; nurses), on waiting times and the number of entities waiting in the queues or going through the system.
- **Hybrid simulation (HS)**: models that combine two or more of the above modelling techniques (Brailsford et al., 2019; Mustafee et al., 2017). HS is typically used to represent a complex system behaviour where different parts of the system can be better captured by two or more simulation methods.

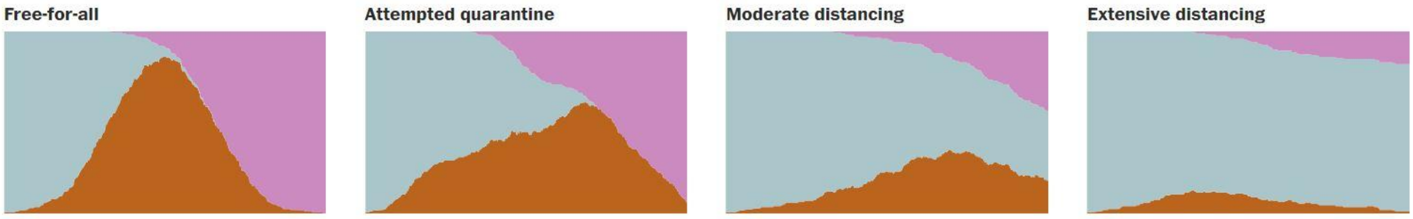


Figure 5: COVID-19 Simulations based on varying degrees of social distancing[2]

Methodology

The specific procedures, and techniques used to identify, select, process and analyze information were based around the goal to predict the end of the COVID-19 Pandemic. A good mark for the end of a pandemic is herd immunity, which occurs once enough people in a community become immune to a disease such that its spread is unlikely. This marks an entire community as protected, even those who are not themselves immune.

When does a community reach herd immunity? It depends on the reproduction number, or R0. The R0 tells you the average number of people that a single person with the virus can infect if those people aren't already immune. The higher the R0, the more people need to be resistant to reach herd immunity.
Researchers think that the R0 for COVID-19 is between 2 and 3. This means that one person can infect two to three other people. It also means 50% to 67% of the population would need to be resistant before herd immunity kicks in and the infection rates start to go down. Due to the emergence of new variants, with varying characteristics, herd immunity is defined to be when 70% of the population becomes resistant.

In order to roughly determine the date herd immunity is reached, agent based modelling is used to create a Susceptible-Exposed-Infectious-Removed (SEIR) model. Individuals are represented by nodes which move freely. Individuals choosing to socially isolate themselves are represented by nodes with no movement. In a real population, interactions do not occur between every individual unless the population is small. Due to node limitations this is compensated for by barriers that slow and inhibit nodes from freely moving across the entire population space.

The SEIR model is a popular approach, however this model seeks to take into account live data. Simulations are largely reliant on information used to create the models. Predictions early in the stages of the virus, likely have larger error margins than even the same models with more up to date statistics. To compensate for this, the simulation contains a data pipeline, which obtains current data from trusted sources such as the Centers for Disease Control and Prevention (CDC), and analyses that data in order to determine the starting points for the simulation. Starting a simulation, at the current state of the event its modeling has the additional benefit of reducing the amount of time that needs to be simulated, and thus the interactions and overall efficiency of the simulation. However, it is worth noting that the introduction of asynchronous HTTP requests for data may appear to increase the time it takes for the program to run. This should not be confused with the runtime of the simulation itself. This is because a simulation that does not update its statistical reference points needs to be updated manually, meaning the simulation itself cannot effectively be run until such updates are completed. The only difference is that we are doing this within the same program.

Implementation

In order to properly propose a data pipeline, determining the simulation approach and programming languages was crucial. For this SEIR model implementation, a method of acquiring live data acquisition and a method of displaying a real time simulation were deemed imperative. To tackle live data acquisition, a Python web-scraper was decided on. Python allows access to multiple packages that enables users access to web scraping (i.e. Beautiful Soup, Selenium) tools or API calling tools (i.e. HTML requests). Once web scraping is complete, the MATLAB simulation script then sets parameters to use based on the most recent data acquired. MATLAB was chosen as the simulation's development language because of how we are modeling our parameters. The simulation uses a vector of vectors to classify different population states, and MATLAB does a very good job of handling the required matrix operations. MATLAB also allows the developers to easily assess variable values, without developers having to print values to the console; this allows the developer to spend less time debugging if parameter values are set correctly. So, once parameters are properly set from the CSV file, the simulation proceeds; displaying both a visual representation of nodes maneuvering around an environment and a graph of the population's status (i.e. percent infected). Upon completion, the MATLAB simulation script outputs captured frames from the simulation run into an AVI video file.

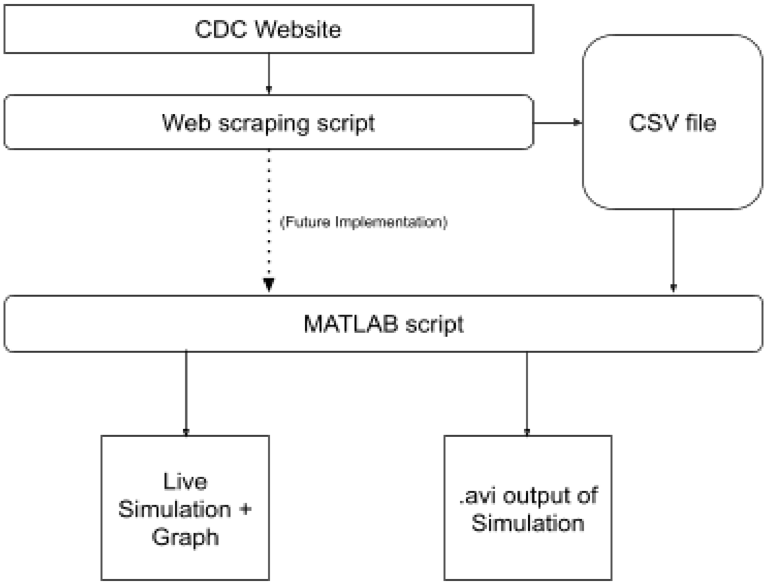


Figure 6: Data Pipeline of the COVID-19 Epidemiology Simulation

Experimental Setup

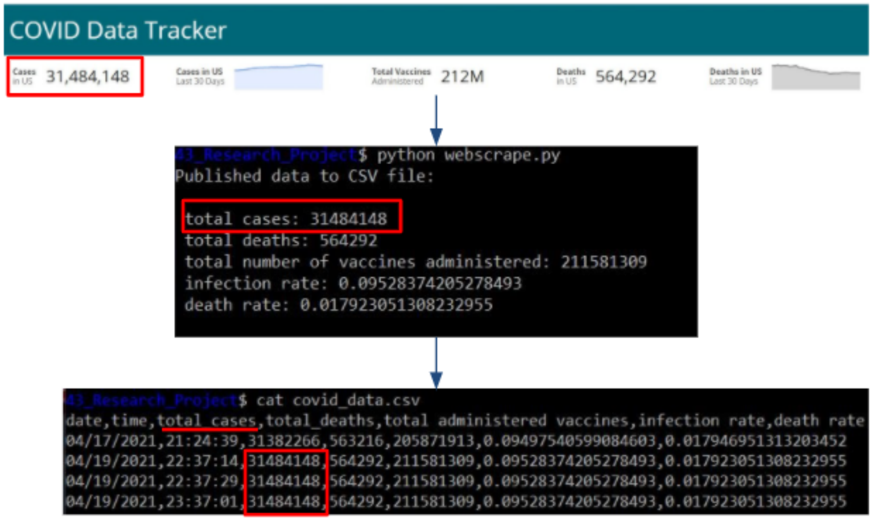
Web Scraping

For the most precise simulation results, the simulation parameters must be set using real-life data. To achieve this, we developed a web scraping Python script that transforms data from the CDC websites' Data Tracker status bar into usable data for the simulation model. The usable data is then either written to or appended to a CSV file; depending on if a CSV file already exists or not.

When assessing the CDC website's source code, the web page's data was determined to be loaded dynamically; therefore, traditional web scraping, using the Beautiful Soup Python library (which relies on static web pages), wasn't a viable option. Instead, we use HTML requests and a URL provided by the CDC's API to access the web page's JSON file, from which the web page's variables are dynamically loaded. Therefore, the web scraping script does not run into the same issue that traditional web scraping does, and the script is able to acquire the data needed. Afterwards, the script uses the equation in the Appendix to calculate both infection and death rates; the script uses the total number of cases in the US data as it's number of infected variable, the script also subtracts the number of infected variable from the total US population to set the population at risk variable, and the script uses a constant of 1. Also as an added feature, we calculate a daily infection rate by using the change in total cases between the previous day and the day before that. We do not use the current day's data, because depending on run-time, the website data may not be updated to the latest values and there will be variations in daily infection rate; causing the simulation run to produce less reliable results.

Once the CDC data is processed, the web scraping script performs a check to determine if a CSV file is present in the local directory. If the file is present, the script appends the collected data to a new line in the file.

Otherwise, the script creates a new CSV file, writes a header stating what each data value represents, and then writes a new line containing the collected data. The CSV file is then ready to be accessed and used for analysis.



```
43_Research_Project$ python webscrape.py
Published data to CSV file:

 total cases: 31484148
 total deaths: 564292
 total number of vaccines administered: 211581309
 infection rate: 0.09528374205278493
 death rate: 0.017923051308232955
```

```
43_Research_Project$ cat covid_data.csv
date,time,total_cases,total_deaths,total administered vaccines,infection rate,death rate
04/17/2021,21:24:39,31382266,563216,205871913,0.09497540599084603,0.017946951313203452
04/19/2021,22:37:14,31484148,564292,211581309,0.09528374205278493,0.017923051308232955
04/19/2021,22:37:29,31484148,564292,211581309,0.09528374205278493,0.017923051308232955
04/19/2021,23:37:01,31484148,564292,211581309,0.09528374205278493,0.017923051308232955
```

Figure 7: Web scraping pipeline

Simulation

The MATLAB code takes great inspiration from The Washington Post's article[2] in regards to viably modeling the spread of COVID-19 over a set period of time. Specifically, it focuses on modeling people as nodes and focusing on transmissibility due to collisions between the nodes. A series of vectors are used to continually update the current status of each node. Specifically:

- ➢ Current Status
- ➢ Time until status change (if susceptible / infected)
- ➢ Current Position and Movement

In effect, as time progresses, we can keep track of how nodes act and whether a discrete event, such as a collision occurs. In the case of a collision, we consider how each node changes after the interaction based on each node's previous status and movement. Moreover, after each computational iteration is done i.e. all the checks as shown in the activity flow diagram is performed, the simulation plotting starts and it continues till the end of the computation. Video is also being created by saving and processing each snapshot of the simulation process.

Our simulation follows **stochastic discrete model** and the working of it is shown below:

Pseudo-Code

Variable Initialization

```
For (time = 0; time = SIMUL_END; time++) {
        For(node_i = 0; node_i = Population; i++) {
                Check the states of infected nodes.
                For(node_j = 0; node_j = Population; j++) {
```

```
            If (collision == 1) {
                Recalculate movements.
                If(at least 1 colliding node is infected)
                    Update the states of colliding nodes.
            }
        }
    }
}
```

Model Plotting

Work-Flow Model

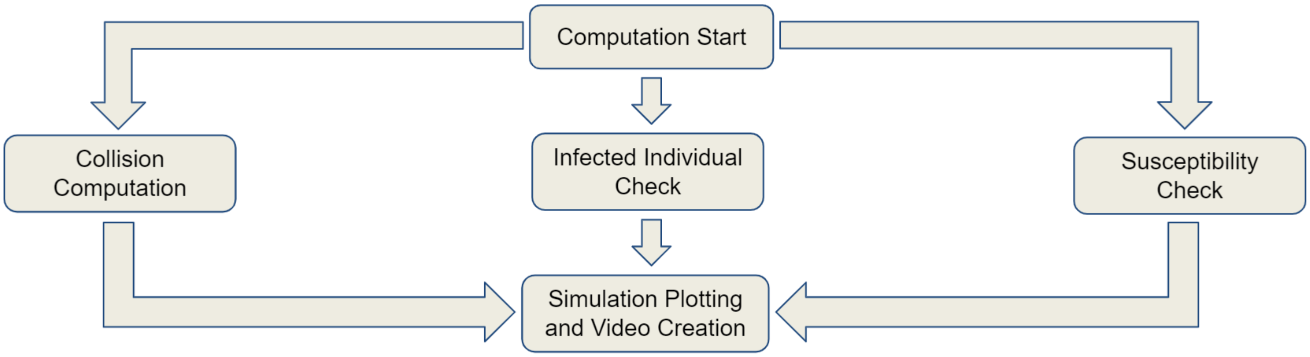

Computation Start
Collision Computation
Infected Individual Check
Susceptibility Check
Simulation Plotting and Video Creation

Figure 8 : Basic Activity Flow Diagram of the Simulation

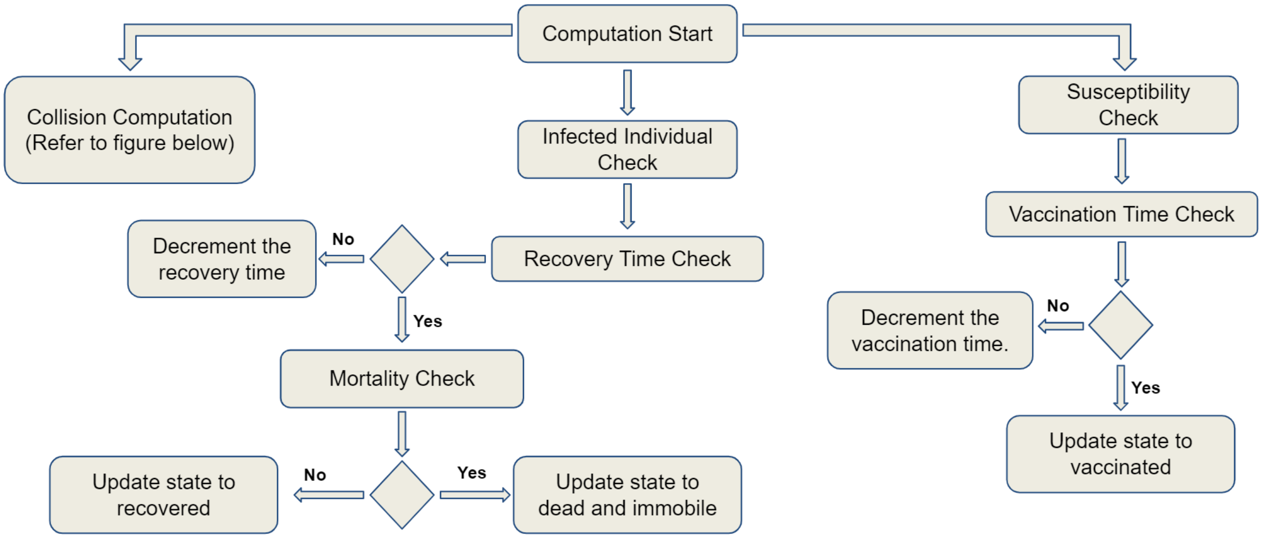

Computation Start
Collision Computation (Refer to figure below)
Infected Individual Check
Susceptibility Check
Vaccination Time Check
Recovery Time Check
Decrement the recovery time
No
Yes
Mortality Check
Decrement the vaccination time.
No
Yes
Update state to vaccinated
Update state to recovered
No
Yes
Update state to dead and immobile

Figure 9 : Activity Flow diagram explaining steps specifically related to Infection and Susceptibility.

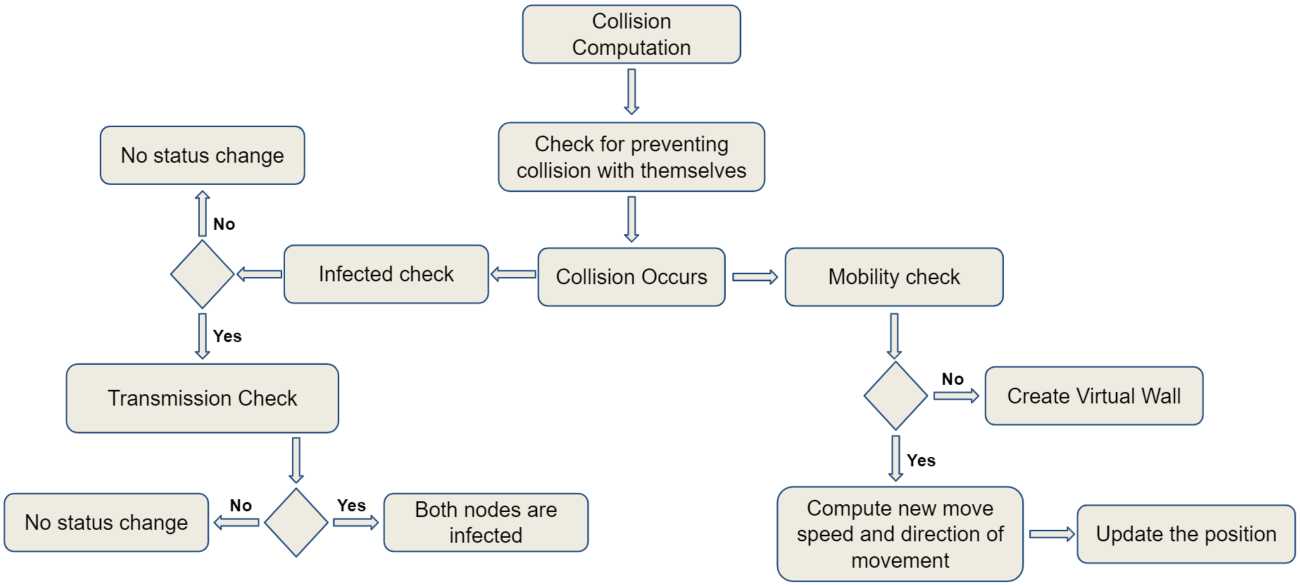


Figure 10 : Activity Flow Diagram explaining the steps involved in the Collision Computation

Variable Initialization

➢ **Time Specific** — The code takes into consideration the total duration of the simulation as well as how often the information within our stochastic discrete event simulation will be updated.

➢ **Map Specific** — The code takes into consideration the population density of a given area and attempts to downscale the model in order to reduce overhead. It also takes into account the percentage of the population who are active in social distancing.

➢ **COVID-19 Specific** — The code takes into consideration the initial infection status of the population at a given time as well as the transmissibility, mortality, and average recovery time rates of COVID-19. In specific, COVID-19 takes roughly 10 - 20 days to recover from, so we assumed an average recovery time of 15 ± 5 days[1].

➢ **Vaccine Specific** — The code takes into consideration the initial vaccination status of the population at a given time as well as the amount of time to become fully vaccinated under 2-dose vaccines, such as the Pfizer-BioNTech vaccine or the Moderna vaccine.

➢ **Population Specific** — The code initializes the vectors to keep track of the status of individuals as the simulation progresses. The five possible states are as follows: susceptible, infected, vaccinated, recovered, and deceased.

➢ **Movement** — The code takes into consideration population mobility, with the assumption that individuals who are socially distant remain immobile. Their current position, direction, and speed are considered. A collision vector is also initialized to determine which individuals come into contact with each other.

Infected Individual Checks

We consider the status of an infected individual as the simulation progresses. There are three cases that can occur:

- ➢ Case 1 — The infected individual dies. Update their state to deceased and prevent their mobility.
- ➢ Case 2 — The infected individual recovers from COVID-19. Update their state to recovered. Unlike case 1, their mobility is unaffected.
- ➢ Case 3 — The infected individual remains infected. No state update. We decrement how much longer it will take for the individual to recover.

Vaccination Checks

We consider the status of a susceptible individual waiting for a vaccine as the simulation progresses. There are two cases that can occur:

- ➢ Case 1 — The time to receive a vaccine has elapsed for a susceptible individual. Update their state to vaccinated.
- ➢ Case 2 — The time to receive a vaccine has not elapsed for a susceptible individual. No state update. We decrement how much longer it will take for the individual to be vaccinated.

Collision Checks

As the simulation progresses, we identify which individuals have collided and considered the following:

- ➢ Mobility — With the exception of immobile (either deceased or socially distant) individuals, as individuals collide, they follow the laws of momentum. Their direction, position, and movement speed are recalculated upon collision to determine their future movements.
- ➢ Boundaries — Individuals may not leave the bounds of the simulation map. If an individual encounters a map boundary, treat the interaction as if it was a collision and 'bounce' the object in an appropriate direction.

Transmission Checks

If two individuals collide with each other, we consider whether either person is infected. There are four possible cases:

- ➢ Case 1: There is one infected individual and one recovered, vaccinated, or deceased individual. No COVID-19 transmission occurs.
- ➢ Case 2: There is one infected individual and one susceptible individual. We perform a probability check based on the infectivity of COVID-19. This will determine whether the individuals will be infected. Their states are updated accordingly.
- ➢ Case 3: Both individuals are infected. There is no need to update the status of either.

- Case 4: Neither individual is infected. There is no need to consider COVID-19 transmission.

Plotting

The MATLAB code initializes the vectors to update the colors of each individual depending on their state. Furthermore, we compute each state's population percentage to give a better visualization of our simulation data.

Results

As it stands, the MATLAB code can effectively produce a visualization of the COVID-19 stochastic discrete event simulation. In comparison to data received from the New York Time's Massachusetts COVID-19 case information[3] we have found that our prediction model serves as a good model based on historical numbers received. We are able to isolate variables such as social distancing proportion, transmission rate, and average vaccination time and test such cases against a base case to determine the significance of the impact of certain variables. Our results utilize information taken from the CDC website as of April 19, 2021. We use the following base case:

- Simulation Duration: 90 days
- Population Size: 200
- Map Size: 600 feet
- Transmission Rate: 40%
- Initial Infected Population: 3%
- Infection Radius: 6 feet
- Average Vaccination Time: 30 ± 10 days
- Mortality Rate: 1.7%
- Average Recovery Time: 15 ± 5 days

Table 1: Effect of Social Distancing on Herd Immunity and COVID-19 Life-cycle

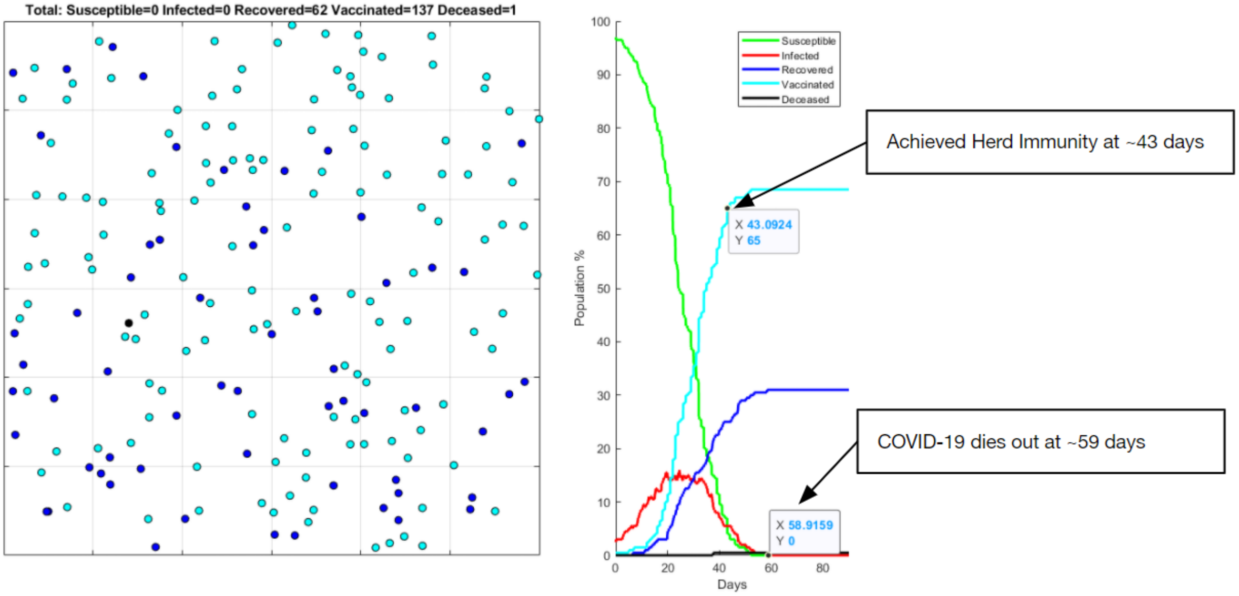
Social Distancing Proportion	Vaccination Herd Immunity Achieved (in days)	COVID-19 Duration (in days)
0%	41	64
20%	42	59
40%	**43**	**59**
60%	38	51
80%	37	38

Table 2: Effect of Transmission Rate on Herd Immunity and COVID-19 Life-cycle

Transmission Rate	Vaccination Herd Immunity Achieved (in days)	COVID-19 Duration (in days)
20%	37	41
40%	**43**	**59**
60%	57	64
80%	Not Reached	54
100%	Not Reached	50

Table 3: Effect of Vaccine Distribution on Herd Immunity and COVID-19 Life-cycle

Average Vaccination Time (in days)	Vaccination Herd Immunity Achieved (in days)	COVID-19 Duration (in days)
20 ± 10	29	40
25 ± 10	31	52
30 ± 10	**43**	**59**
35 ± 10	42	63
40 ± 10	49	64



Achieved Herd Immunity at ~43 days

COVID-19 dies out at ~59 days

Figure 11: Final Result of the Simulation for the base case parameters

Analysis

Overview

In the scope of the 90-day simulation period, the results demonstrate interesting but unsurprising trends. At best, we see that herd immunity can best be reached via distributing vaccinations as quickly as possible. With an average vaccination time of 20 ± 10 days, herd immunity will be reached in 29 days with COVID-19 dying out a mere 11 days after herd immunity is reached. In the worst-case scenario, social distancing is not actively practiced. We see that although the herd immunity date does not improve (we attribute the 2-day difference to random variance within our simulation), COVID-19 duration significantly worsens, dying out at 64 days. We discuss our findings in detail below.

Social Distancing

As the proportion of nodes that are socially distant increases, we see that vaccination herd immunity date gradually decreases until a threshold value of 37-38 days is reached. In regards to COVID-19 duration, we see that as more people are socially distant, COVID-19 dies out in a significantly shorter period of time — this is especially notable as the social distancing proportion increases from 60% to 80%.

Transmission Rate

As the transmission rate increases, we see that the vaccination herd immunity date rapidly increases at an exponential rate until at higher transmission rates (80% and 100%) we see that the vaccination herd immunity is not reached due to COVID-19 spreading rapidly to susceptible individuals. However, one notable observation is that at 80% and 100% transmission rate, COVID-19 dies out in a shorter period of time compared to 40% and 60% transmission rate. We hypothesize that this is due to COVID-19 spreading too quickly to susceptible nodes. With such a high transmission rate, many susceptible nodes will have become infected early into the simulation (i.e. before vaccination time occurs) leading to COVID-19 dying out due to a lack of susceptible nodes to infect.

Average Vaccination Time

As average vaccination time increases, we see that unsurprisingly, the vaccination herd immunity date increases. It is notable, however, that with an average vaccination time of 40 ± 10 days, a threshold value appears to be reached with the COVID-19 duration at 64 days.

Limitations

Nonetheless, the MATLAB model experiences one significant issue: scalability due to large computational time overhead ($O(N^3)$ time complexity), constraining us to downscale the all out the total population (i.e. lower node count) of Boston within the model. As a result, lower node count may result in lower precision. This may be resolved by utilizing data structures (e.g. hash maps) to improve the computational overhead and accordingly, consider for higher population valuations. Additionally, consistency of vaccine distribution rates may change contingent upon outer factors, such as arising COVID-19 strains, not considered within the scope of our simulation. Furthermore, it is difficult to maintain an exact and accurate simulation as the epidemiology landscape changes rapidly.

Future Scope

As it stands, the web scraper acquires its information from the CDC and computes them as a moving average. Due to COVID-19's inherent nature as a virus, COVID-19 rapidly changes over the course of time leading to less precise moving average values. We seek to implement usage of recent data to compute more precise values and parameters for input into the MATLAB simulation. Moreover, we aim to streamline parameter updating by improving the connectivity between the Python web scraper and the MATLAB simulation. We aim to remove the necessity of generating a .csv file for the simulation to parse its parameter data from. Optimally, parameter data from the web scraper will be directly fed into the simulation without the usage of intermediary files.
Additionally, we seek to improve the scalability of our model via optimization of our algorithm — this may be resolved via utilization of data structures. Furthermore, distributed simulations are well-known to give performance improvements for complex simulation models. Scalability improvements may be possible via usage of parallelization and perhaps even linking additional simulation models to this model.

We also aim to make the code more robust via various functionality implementations. Because COVID-19 is a virus, it inherently has the predisposition to rapidly mutate and evolve into several variants over a short period of time. We will take into consideration how these strains differ in the future. Improvements on both the vaccination prediction model and the mobility design can also be made. Currently, our model assumes a static approach for both, i.e. vaccinations distribution rate is constant and nodes mobility is linear until collisions occur. We seek to implement more dynamic considerations such as vaccination shortages and non-linear node mobility in the future. Population-specific improvements such as susceptibility conditions (e.g. immunocompromised or elderly) and node clustering for better representation of social distancing circles will also be considered.

Conclusion

COVID-19 has cemented itself globally as the worst pandemic to occur in recent history. As such, it is absolutely crucial for data to be quickly compiled and analyzed for better transparency and information in regards to the virus. We have developed a MATLAB simulation tool to help predict future trends, specifically herd immunity and COVID-19 remaining duration, based on various parameters such as social distancing, transmissibility, and vaccination rates. These simulation parameters are either taken directly or inferred from data on the CDC website via a Python web scraper. Our results show that, as of April 19, 2021, herd immunity will reasonably be reached between 29 - 57 days and COVID-19 will die out between 40 - 64 days. Furthermore, we find that vaccination distribution rates have the greatest impact on herd immunity whereas social distancing has the greatest impact on COVID-19 duration. Overall, these are very promising initial results and expectations — we will continue to develop our model to improve precision and robustness for future utilization.

References:

[1] CDC. (2020, March 28). COVID Data Tracker. Centers for Disease Control and Prevention. https://covid.cdc.gov/covid-data-tracker/#datatracker-home.

[2] Stevens, H. (2020, March 14). Why outbreaks like coronavirus spread exponentially, and how to “flatten the curve”. Retrieved April 26, 2021, from https://www.washingtonpost.com/graphics/2020/world/corona-simulator/

[3] The New York Times. “Massachusetts Coronavirus Map and Case Count.” The New York Times, The New York Times, 1 Apr. 2020, www.nytimes.com/interactive/2021/us/massachusetts-covid-cases.html?action=click.

[4] Lin, Q. , Zhao, S. , Gao, D. , Lou, Y. , Yang, S. , Musa, S. S. , Wang, M. H. , Cai, Y. , Wang, W. , Yang, L. , & He, D. (2020). A conceptual model for the coronavirus disease 2019 (COVID-19) outbreak in Wuhan, China with individual reaction and governmental action. International Journal of Infectious Diseases , 93, 211–216. https://doi.org/10.1016/j.ijid.2020.02.058

[5] Hellewell, J. , Abbott, S. , Gimma, A. , Bosse, N. I. , Jarvis, C. I. , Russell, T. W. , Munday, J. D. , Kucharski, A. J. , Edmunds, W. J. , Funk, S. , Eggo, R. M. , Sun, F. , Flasche, S. , Quilty, B. J. , Davies, N. , Liu, Y. , Clifford, S. , Klepac, P. , Jit, M. , Diamond, C. , & van Zandvoort, K. (2020). Feasibility of controlling COVID-19 outbreaks by isolation of cases and contacts. Lancet Global Health , 8(4), 488–496. https://doi.org/10.1016/S2214-109X(20)30074-7

[6] Tako, A. A. , & Robinson, S. (2009). Comparing discrete-event simulation and system dynamics: Users' perceptions. Journal of the Operational Research Society , 60(3), 296–312. 0160-5682. https://doi.org/10.1057/palgrave.jors.2602566

[7] United States. (n.d.). Retrieved April 22, 2021, from https://www.worldometers.info/coronavirus/country/us/

Appendix

[1] Our simulation code can be found at https://github.com/sn1026/COVID-19-Simul.

[2] The web scraping code can be found here: https://github.com/scheung97/EECE5643_Research_Project.

[3] Simulation Video Youtube Link: https://www.youtube.com/watch?v=nzHNcArtM8E

[4] Calculation of Infection Rates = $\frac{Total\ No.of\ Infected\ People}{Population\ at\ Risk} \times constant\ (k)$