Program Structures & Algorithms

Spring 2021

Covid-19 Simulation

Members

- Kan Zhang 001529338
- Yumeng Zhang 001543248
- Xinyi Wu 001529267

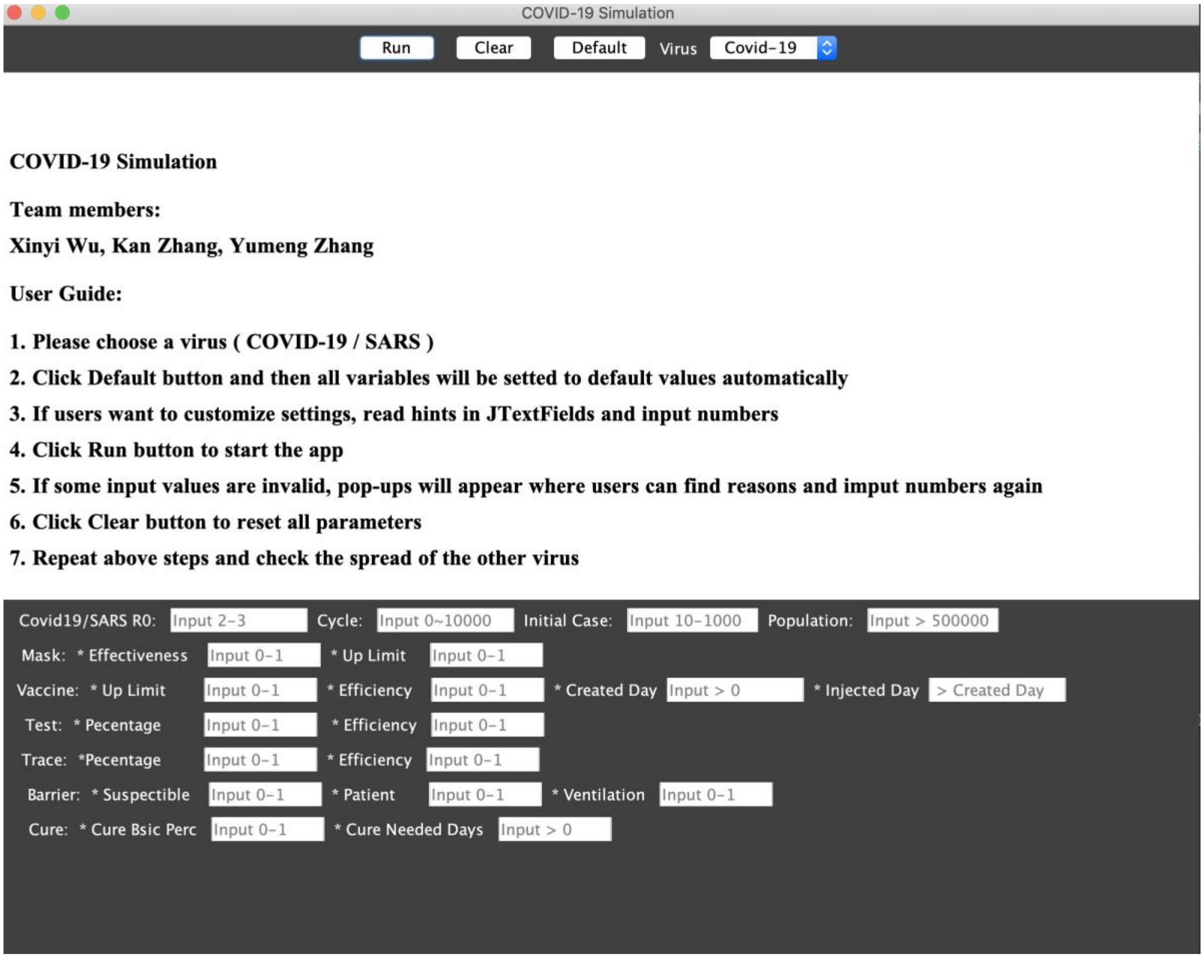
Introduction

This project aims to simulate the spread of a virus such as Covid-19 or SARS. The simulation takes into account of basic reproduction number, mask usage, vaccine availability, barrier actions, prevalence of testing and contact tracing, population and recovery. We use Java Swing as User Interface, Python to simulate the spread model and Bokeh to draw the result.

Guidance

We designed a user interface to allow soft coding from users and execute the program. Users can execute the program by clicking "Run", set the default value by clicking "Default" or clean all the inputs by clicking "Clear". Then it will pop up a page with figures of simulation result.

The main entrance shows as below:



Algorithms & Assumptions

The basic idea is to modify R0 corresponding to the external and internal variants. Core formula is:

$$new\ case = current\ case * R$$

where R is the combination of R0 and all variants.

We count new case at the end of every day and renew the input variants for next status.

Variants

1. Mask

We have two soft coded variants as “Up Limit” and “Effectiveness”. Both values are between 0 and 1. The formula is:

$$R = R_{init} * (1 - usage * Effectivness)$$

where usage is determined by:

$$e = (\frac{day}{100})^{0.1} \text{ if } day < 100 \text{ else } 1$$

$$usage = Up\ Limit\ * e\ * (\frac{current\ case}{TOTAL\ PPL})^{0.1}$$

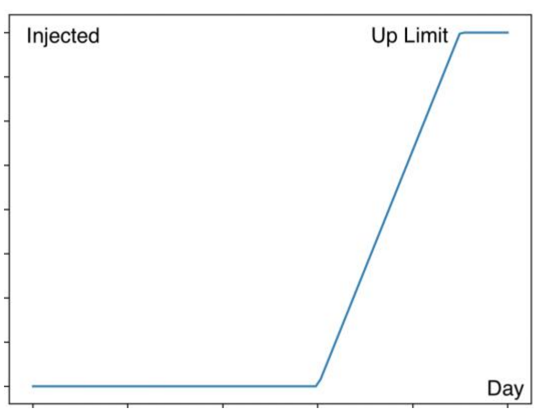
“day” is the spanning day from simulation begins. “TOTAL PPL” is total population as an invariant in this simulation.

This is to consider people raise the attentiveness when day goes by and case number goes up.

2. Vaccine

Four soft coded variants: “Up Limit”, “Efficacy”, “Created Day”, and “All Injected Day”

Before “Created Day”, there are no people injected of vaccine. After “All Injected Day”, “Up Limit” of people get injected considering some would not like to take the vaccine. The popularizing progress is linear. It goes like this:



And the effect to R is like:

$$R = R_{init} * (1 - Injected * Efficacy)$$

3. Barrier

In this case, we consider two measures – “Quarantine” and “Ventilation”. “Quarantine” is parted into “Susceptible” and “Patient” due to different medications to the crowd. We will count “Patient” in Recovery later. The basic formula is:

$$R = R_{init} * (1 - Efficiency)\ for\ all\ barrier\_measures$$

4. Test & Trace

We have four variants, “Test Efficiency”, “Test Percentage”, “Trace Efficiency”, and “Trace Percentage”. Like Barrier, it goes like:

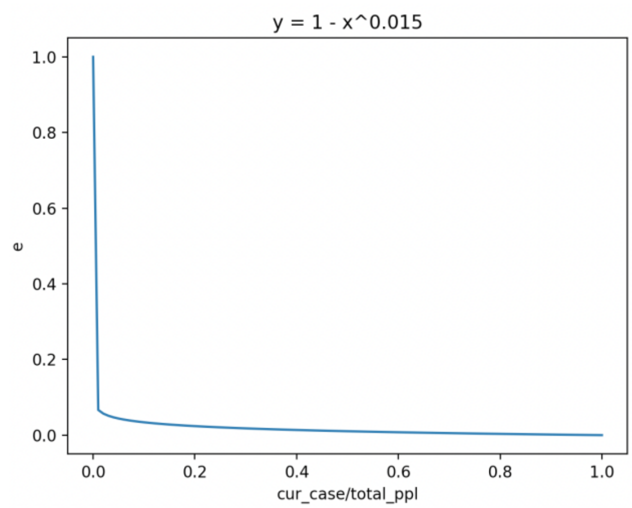
$$R = R_{init} * (1 - Efficiency * Percentage)\ for\ Test\ and\ Trace$$

5. Population

When infected case goes up, the spread of virus would get faster due to the large patient crowd. However, when it comes to certain level, people raise the attentiveness and be more conscious of protection. Thus, the proportion would be smaller.

$$R = R_{init} * \left(1 - \frac{current\ case^{0.015}}{TOTAL\ PPL}\right)$$

Y = 1 – X^0.015 goes like this:



When virus just exploded which means current case is small, we assume most of people would be careful about that so the reproduction number drops significantly. However, there are still remaining part that go to party regardless of the pandemic.

The case number does little to their lifestyle. So population limitation drops slowly afterwards.

6. Recovery

Patients would recover from Covid-19 due to the quarantine or other medications. We suppose an average recovery day in this case as “Cure Need Days”. Other variants go like “Cure Percentage” and “Quarantine Effectiveness of Patient”. When we do good medications and quarantine, the “Cure Percentage” raise up.

$$Cure\ Cases = Case(Cure\ Need\ Days\ ago) * Cure\ Percentage$$

And we add arbitrary factor to cover those unexpected situations:

$$Random\ Cure\ Cases = Case * Random\ Cure\ Percentage$$

The Random Percentage is set up to 0.05.

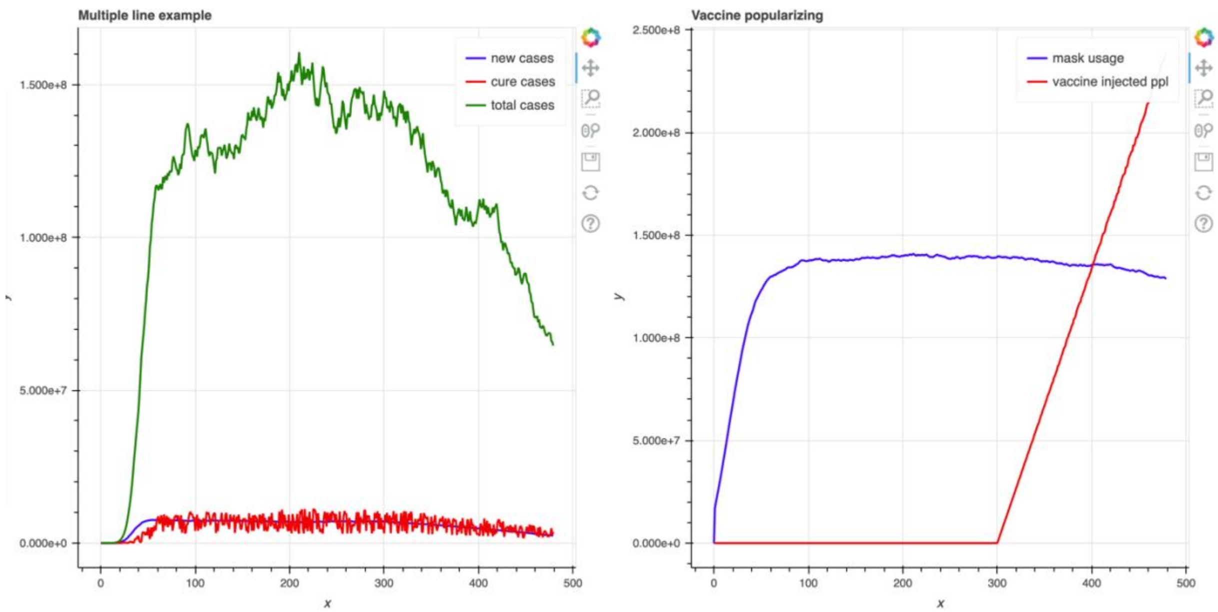
Output

Here we provide with an example of an anonymous virus which R0 is set up to 15. It shows an obvious change: https://youtu.be/Wj48V1YD-8c

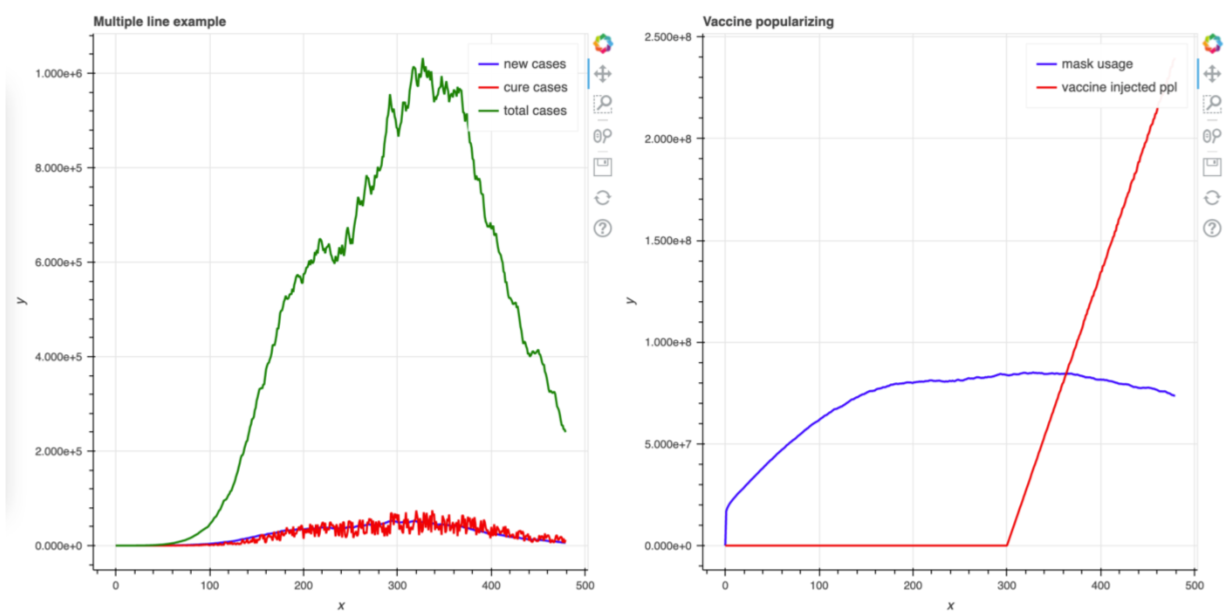
Here are animated pie charts of Covid-19 and SARS

<animated pie chart of Covid-19 R0 = 2>

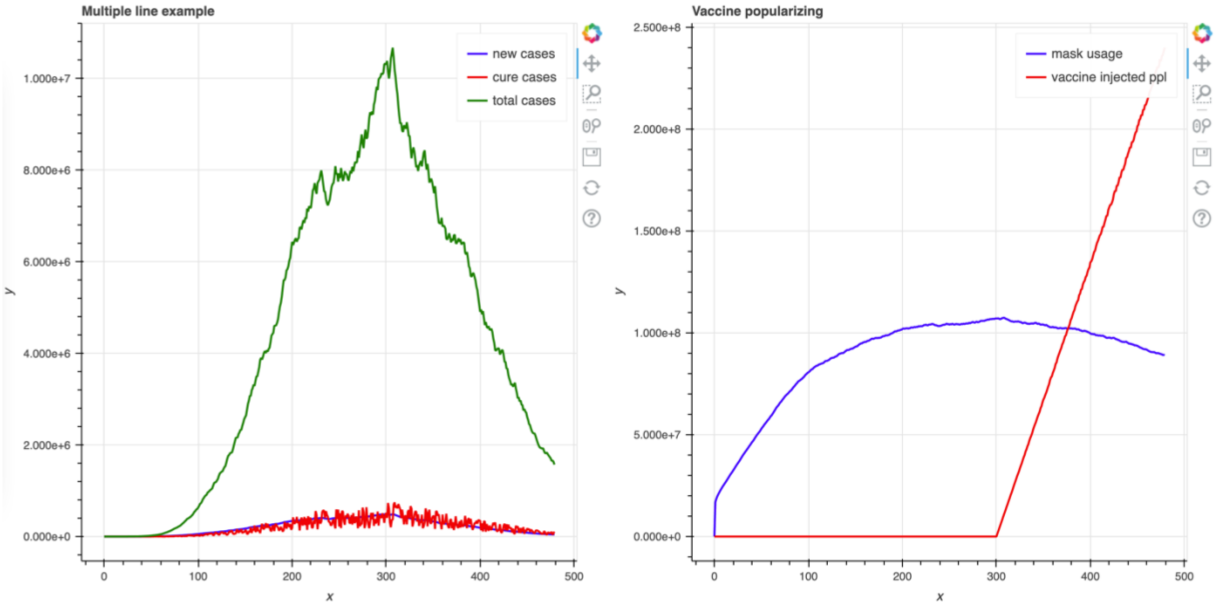
<animated pie chart of Sars R0 = 3>



<example line graphs of anonymous virus R0=15>



<line graphs of Covid-19 R0=2>



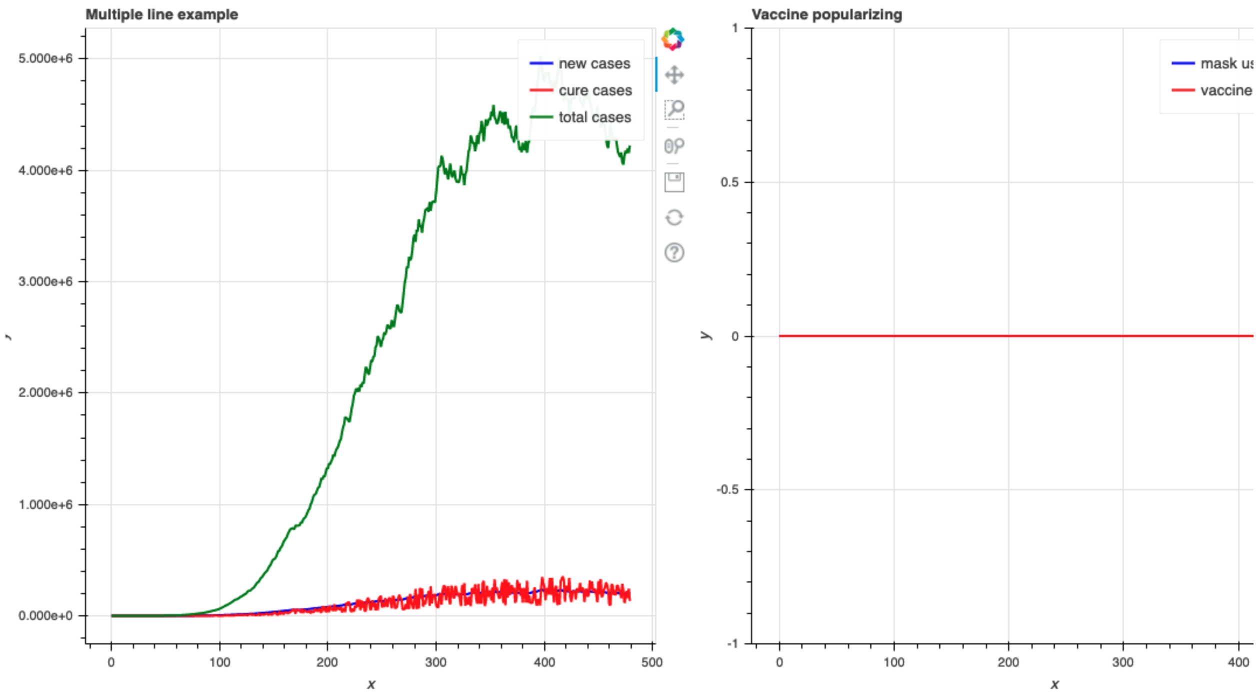
<line graphs of SARS R0=3>

Mathematical Analysis

- Analysis of two virus
  The R0 for COVID-19 is 2 – 3. If other parameters are constant, total case of COVID-19 is above 5.000e+6 after 300 days and then decreases due to popularizing mask usage and vaccine injection.

  The R0 for SARS is 1.8 – 3.6. If other parameters are constant, total case of SARS is also above 5.000e+6 after 300 days. However, the default R0 of COVID-19 is less than that of SARS, so that we assume COVID-19 is more transmissible.

- Worst case
  If we set the effectiveness of mask and vaccine to be zero but maximize R0 which means the disease is highly transmissible and will not decline, then the simulation shows that the number of total cases is always increasing significantly.



Conclusion

The rapid spread of coronavirus disease 2019 (COVID-19) creates significant challenges for economies and healthcare systems of many countries around the world. The situation evolves extremely quickly and, to date, there is a high degree of uncertainty about the future outcomes of the pandemic.

In this report, we examine how the likelihood of reinfection affects epidemiological dynamics at the population level of 0.3 billion. A key finding is that mitigation measures not only delayed the peak of infection, but also delayed the moment when the difference between reinfection and non-reinfection options became apparent.

Unit tests result

```
❯ python -m test.test -v
test_add_case (__main__.TestPandemicSimulation) ... ok
test_cure (__main__.TestPandemicSimulation) ... ok
test_cure_perc_raise (__main__.TestPandemicSimulation) ... ok
test_initial_R0 (__main__.TestPandemicSimulation) ... ok
test_load_json_file (__main__.TestPandemicSimulation) ... ok
test_mask (__main__.TestPandemicSimulation) ... ok
test_mask_usage_rise (__main__.TestPandemicSimulation) ... ok
test_ppl_limit (__main__.TestPandemicSimulation) ... ok
test_vaccine (__main__.TestPandemicSimulation) ... ok
test_vaccine_avail_raise (__main__.TestPandemicSimulation) ... ok

----------------------------------------------------------------------
Ran 10 tests in 0.001s

OK
```