Simulation Paper

Benjamin Vandersmissen[1], Lars Van Roy[2],
Evelien Daems[3], and Frank Jan Fekkes[4]

[1] benjamin.vandersmissen@student.uantwerpen.be
[2] lars.vanroy@student.uantwerpen.be
[3] evelien.daems@student.uantwerpen.be
[4] franciscus.fekkes@student.uantwerpen.be

Abstract. In this paper we will examine the Stride tool and discover its functionalities. We will discuss some findings about the use of different parameters, populations and more. In the end there is a brief discussion of the performance of the program, a very important topic within computer related problems.

1 Simulation

1.1 Stochastic variation

We use the Stan (STochatsic ANalysis) controller to examine the influence of stochasticity on the results obtained from the simulation.

In Figure 1 the number of cummulative cases per time-step is plotted. Here we can observe an exponential grow of the number of cases througout time. This is not surprisingly because it can be deducted from common reasoning. If per time more people are affected, a larger contactpool is possibly infected. These people who are now new carriers of the disease will enter their personal contactpool and again more people will be reached.

Towards the end a flattening of the curve occurs. This is not something totally unexpected because the population is obviously not infinite. At one point anyone who can be infected will effectively become a carrier of the disease.

The same reasoning can explain the curve in Figure 2. Now the cases are not the cummulative ones but the number of new cases in each time-step. A similar course of the curve can be observed.

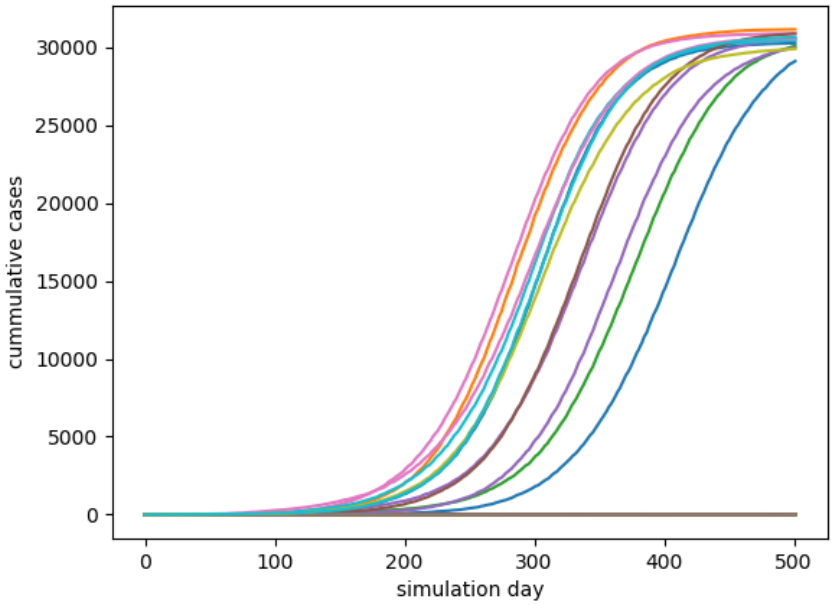

Fig. 1. Result of a number of stochastic runs. The figure displays the distribution of the number of cummulative cases per time-step.

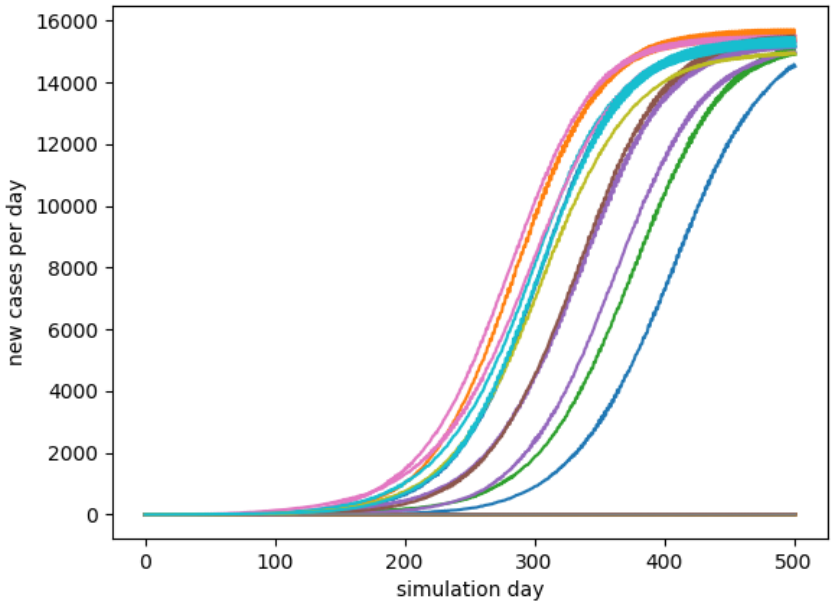

Fig. 2. Result of a number of stochastic runs. The figure displays the distribution of the number of new cases per time-step.

1.2 Determining an extinction threshold

In the previous section 1.1 (Stochastic variation) we found that extinction will influence the outcome. It is neccessary to be able to exclusively look at outbreaks. If we can find some threshold where there is a clear difference between large outbreaks and extinctions we can seperate the two scenarios.

After creating a number of simulations we can plot the total number of infected cases and their occurences. We used the file "stochastic_analysis.xml" for the simulations. After running the simulator the outcome is plotted in the histogram in Figure 3 where the frequency of the amount of infected cases is plotted.

There is a clear distinction between large outbreaks and smaller ones. The smaller ones are again plotted in the second histogram "extinction_small.jpg". There it can be noticed that small outbreaks are really small (25 maximum). Which can be called an extinction after 500 days. The threshold can be set between than 50 and 25 000. Either of those thresholds should eliminate all extinctions in this case.

A very low threshold might allow some extinctions to be passed while a high threshold might eliminate an outbreak. What can be noticed is the total lack of simulations between 100 and 25000 infected cases. But there can still be exceptions in the infected cases. A threshold of 1000 would be more than adequate. It will eliminate all extinctions while keeping the outbreaks.

It should be noted that this threshold will change for al lot of variables. Variables like time and population will affect the threshold. A new threshold should be determined for each simulation.

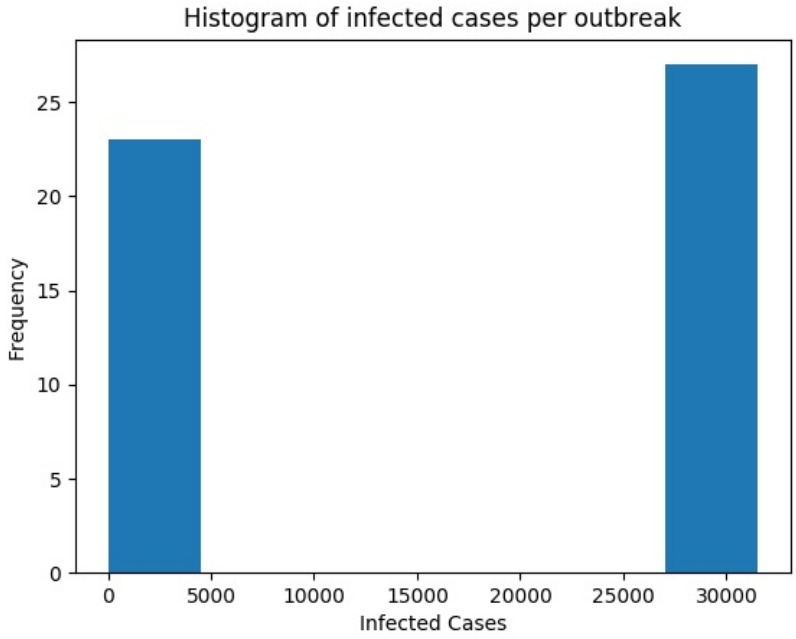
Histogram of infected cases per outbreak
Frequency
Infected Cases
0
5
10
15
20
25
0
5000
10000
15000
20000
25000
30000

Fig. 3.

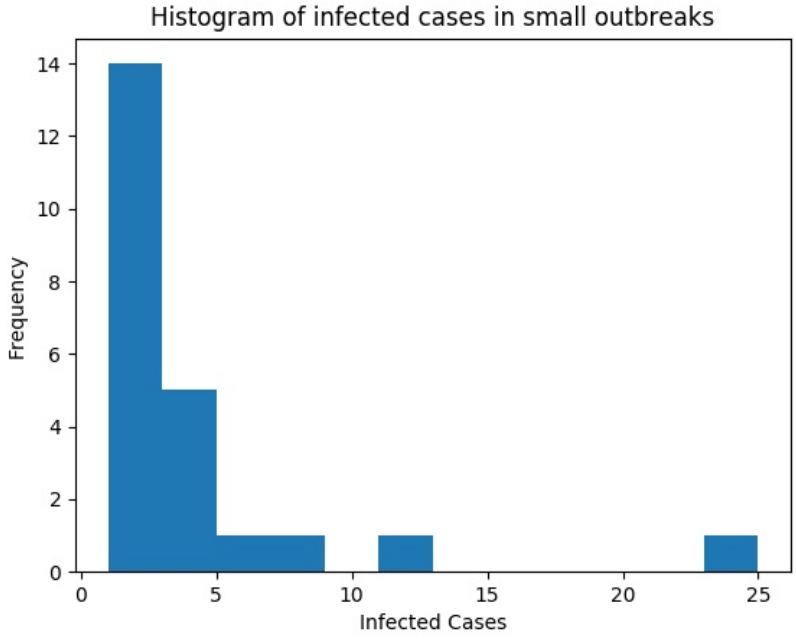
Histogram of infected cases in small outbreaks
Frequency
Infected Cases
0
2
4
6
8
10
12
14
0
5
10
15
20
25

Fig. 4.

2.3 Is commuting to work important for disease spread?

One could easily assume that working at a different location affects the rate at which a disease spreads, as it enhances it's reach. In a first simulation we generated simulations for 5 different commuting percentages. As you can clearly see, it does affect the rate at which the disease spreads, but it has no, or little, effect on whether the disease does or does not spread. In all cases, the entire population got the disease, be it that it took a few days longer to get to that point. Another thing we can remark is that the highest "peak" of newly diseased people is lower the lower the commuting factor gets. A possible cause of the lack of serious effect can come from the fact there are a lot of college commuters who will have the same effect as the workplace commuters.

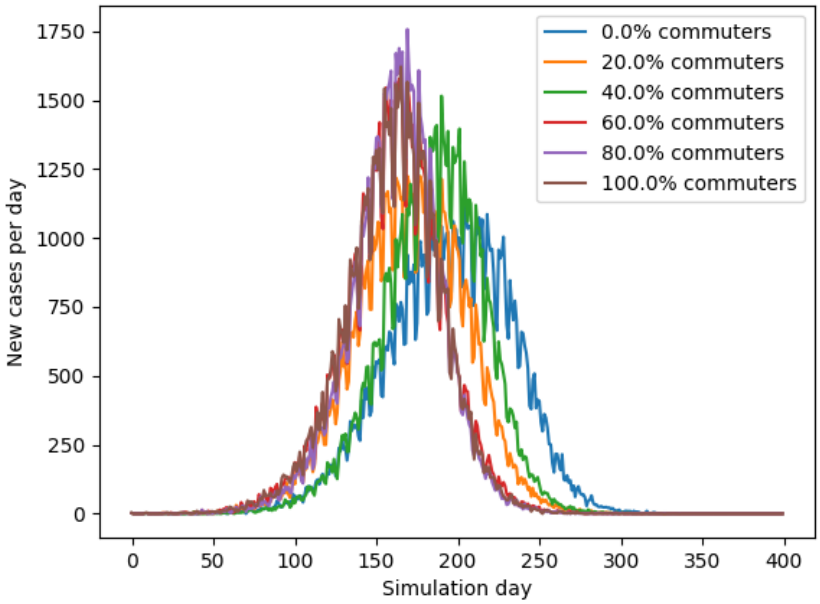

Fig. 5. Results of 6 different percentages of commuters in a range from 0 to 100.

If you watch the top of the graph, you can see that from a certain fraction and onwards, there is little to no difference in their behavior. To get a better view we graphed a closeup of percentages between 30 and 70. In the next graph you can see that the peeks are almost equally in height. You can even notice that some of the higher percentages have their peeks later then the ones with a lower percentage.

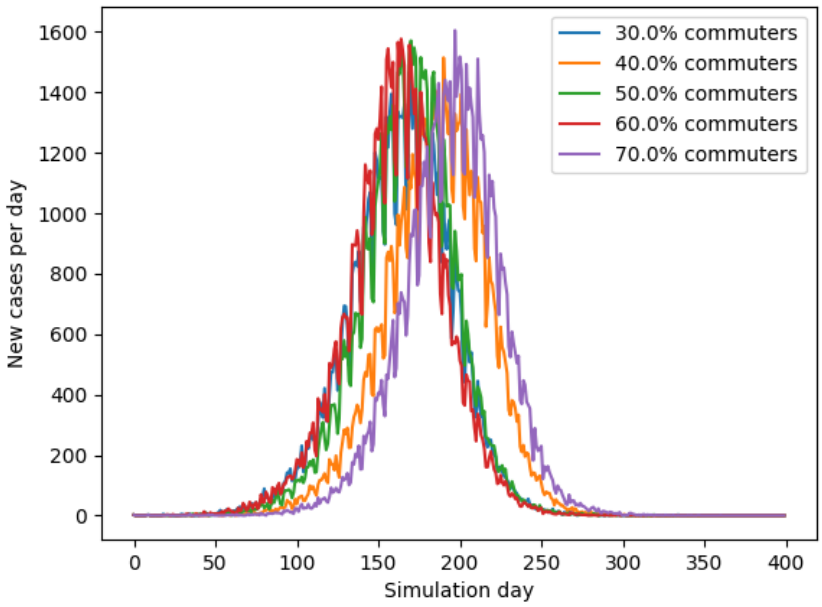

Fig. 6. Results of 5 different percentages of commuters in a range from 30 to 70.

Considering the recorded data we can see that solely changing the percentage of people who commute to work will not affect the spreading of disease in a significant manner. The disease will still spread all the same, but at a slower pace.

3 Performance profiling of sequential code

To study the performance of the code we will discuss a few parameters. We used the GPROF tool to profile the code. Based on these result we could see the influence of different parameters on the runtime.
First we will choose a random number of days to run a simulation and look at the time needed to complete the algorithm.

Number of days	Time needed
10	00:00:03:192:028
50	00:00:04:827:135
150	00:00:09:313:779
500	00:00:22:090:867
1000	00:00:39:660:327

As could be expected, there is an increase in execution time when we take a larger amount of days. But surprisingly enough is the simulation even with 1000 days rapidly. This is because after a period of time everyone will be infected and thus no further computation is necessary for the remaining amount of days. Only increasing the number of days will not make a difference in the performance of the simulation.

Next, we vary the parameter of population size. From the following table it is clear that the larger the population the longer the simulation needs to finish. From the GPROF analysis we notice that the most work is done in getting the count of the infected.

Population size	Time needed
500	00:00:00:332:810
1000	00:00:00:348:895
10000	00:00:00:315:920
50000	00:00:00:472:412
100000	00:00:00:677:304

Varying the immunity rate, there is not a significant difference in runtime for different configurations. In order for this variable to have an influence on the final result, it is necessary to give other parameters different values.

Immunity rate	Time needed
0.2	00:00:04:869:367
0.4	00:00:04:873:811
0.6	00:00:04:966:409
0.8	00:00:05:035:361
0.99	00:00:04:921:399

SEEDING

...

Seeding rate	Time needed
0.000001	00:00:04:684:663
0.00001	00:00:04:568:581
0.0001	00:00:04:525:856
0.001	00:00:04:810:693
0.01	00:00:05:369:309

The contact log mode has a significant impact on the running time of a simulation. When the standard algorithm is used (all or susceptibles) it requires a lot more time to complete the simulation. It forms a large contrast with the running time needed when using the optimized algorithm with all the members of the contact pool sorted. By reducing the number of loops in the algorithms the necessary time to complete the algorithmcan be reduced with it.

Contact log mode	Time needed
All	00:16:09:012:346
Susceptibles	00:16:48:490:700
None	00:00:06:609:545
Transmissions	00:00:06:747:454