Stride*

Andrei Bondarenko, Mathias Ooms, Stan Schepers, and Laurens Van Damme

University of Antwerp, Prinsstraat 13, 2000 Antwerpen, Belgium

Abstract. A summary of explanations why some variables have influence on the results of simulations run with Stride and if they have an impact on the performance.

Keywords: Stride · Epidemiology · Simulations.

1 Simulations

This section contains the results of the simulation exercises using Stride and our interpretation of these results. Simulations were run using the STAN and pyStride controllers.

1.1 Stochastic variation

After running multiple simulations, using 10 and 100 seeds, it seems that the chance has a big influence over the results. The graph in figure 1 shows the two possible outcomes:

- Outbreak: The amount of infected people starts small but quickly starts to grow. Around 30000 people will be infected at the end.
- Extinction: A few people get infected (a maximum around 35) and the amount stays constant throughout the remaining time. Which is noticeable on figure 2.

Meaning it really depends on the first infected people. No other situations where, for example, only 10000 people were infected are present. With this configuration there's a ± 50% chance for an outbreak or extinction with this configuration.

* Supported by organization COMP.

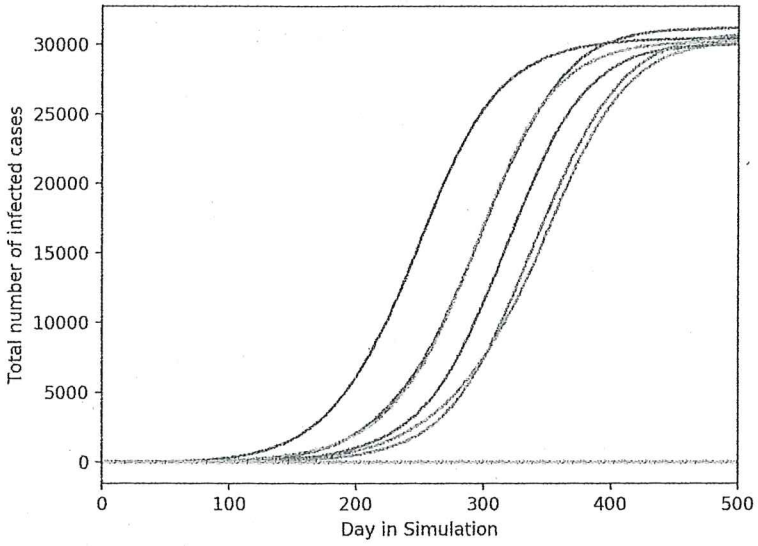


Fig. 1. Cumulative plot of 10 simulations using 10 random seeds

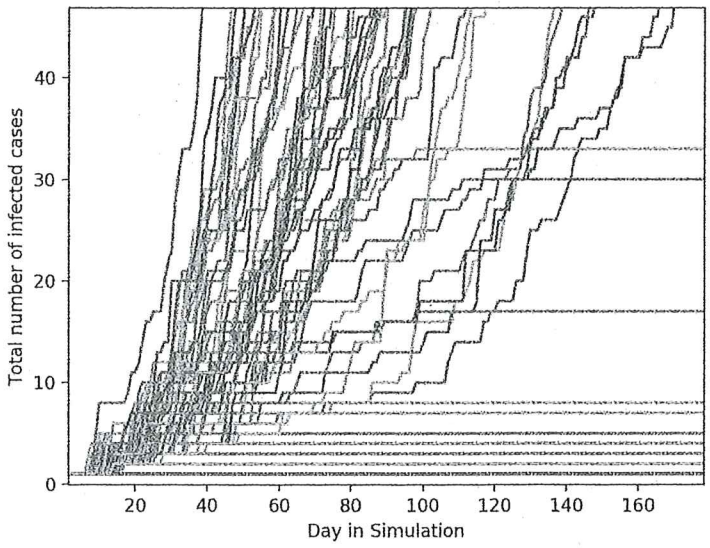


Fig. 2. Zoomed cumulative plot of 100 simulations using 100 random seeds

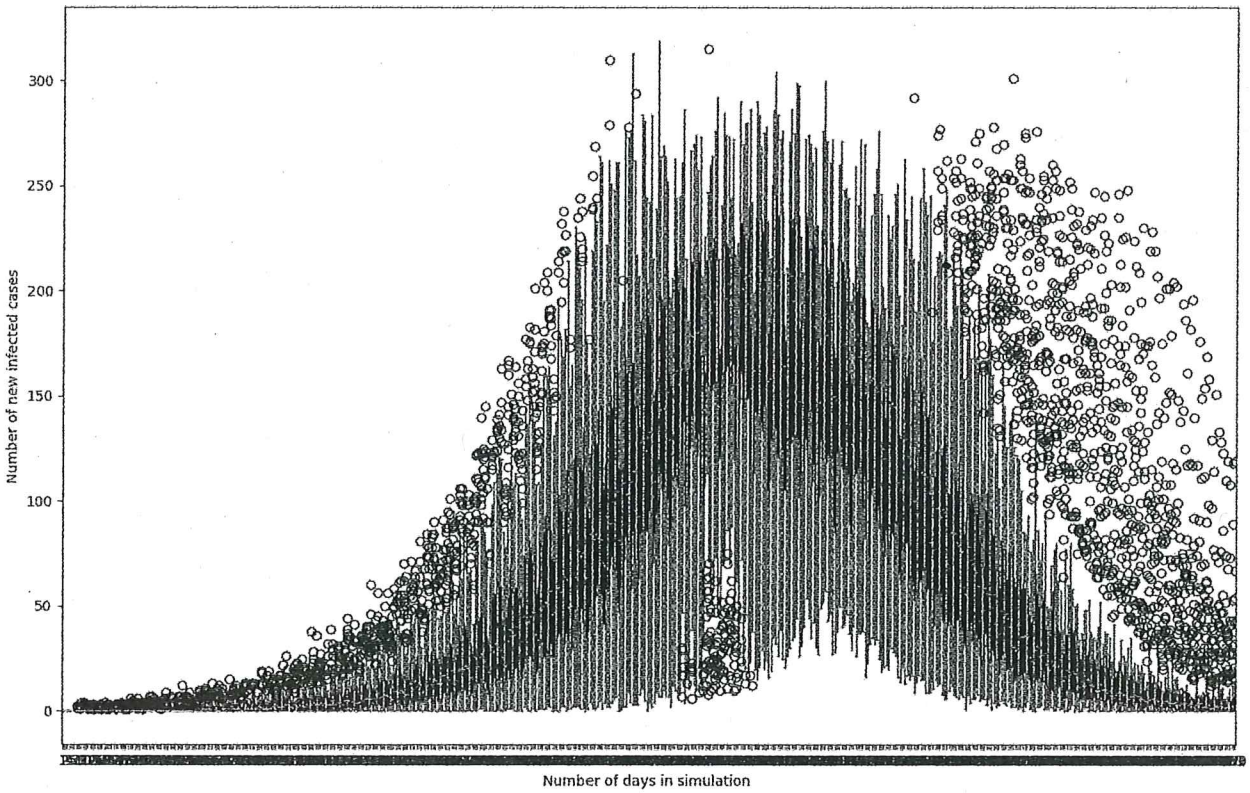


Fig. 3. Box plot of amount of new cases per day in 100 simulations using 100 random seeds

Plotting the number of new cases per day in a box plot graph of only the simulations where an outbreak is present, as seen in figure 3, gives to much information on a small graph which causes it to be chaotic. For this reason a sample is taken, as seen in figure 4, in which the measurements are only shown every tenth day and also the number of new cases is now an average over the passed ten days.

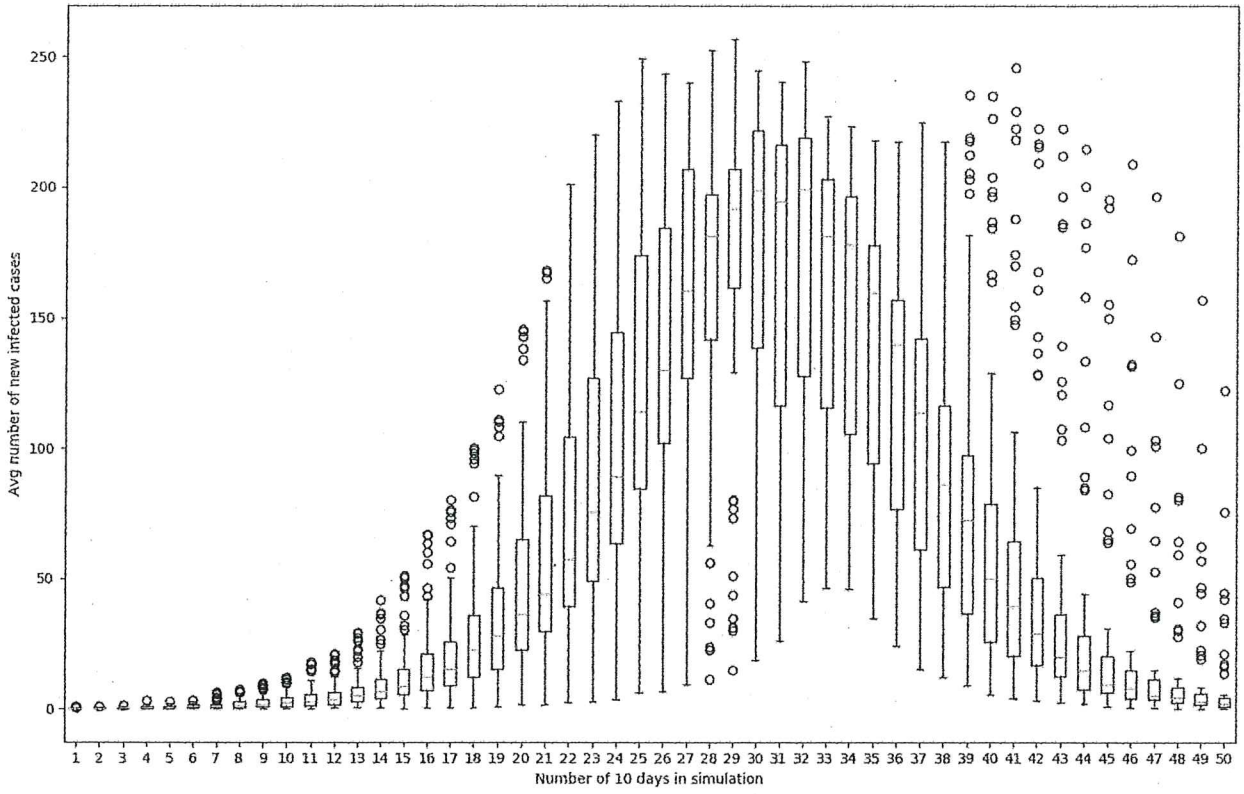


Fig. 4. Box plot sample of figure 3, average new cases over 10 days for every tenth day.

The graph shows that the number of new cases per day follow a bell curve. It seems that the reason for this is the fact that at the beginning of an outbreak there are few infected people that can infect others. As time passes more people get infected and the number of new cases per day will thus increase as well, until a point is reached where there are more people left with a lower chance of getting infected, thus resulting in a lower number of new cases per day. The outliers are data from simulations where the outbreak starts a bit earlier or later, this also causes the whiskers to be so big.

Introduce what you will do.

1.2 Determining an extinction threshold

Plotting the final number of infected cases from the previous section in a histogram, results in figure 5. The two possible outcomes that could be seen in the previous section, can be seen here as well. Either the final frequency remains very low or it becomes very high. A very rough estimate of 2500 can be made for the extinction threshold based on this plot.

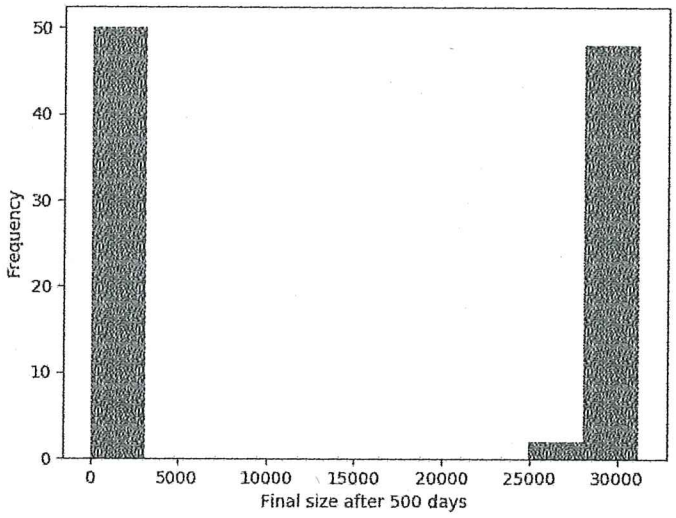


Fig. 5. Histogram

Unnecessary white space?

1.3 Estimating the immunity level

To find the best estimate for the immunity level of the population the vaccination profile was set to *None* and the immunity profile to *Random*. Then for each immunity level, ranging from 60% to 80%, 10 simulations were run using different seeds, and the average was then plotted to compare with the given plot. Immunity levels of 70% and 71%, figures 6 and 7 respectively, bore the most resemblance to the given plot.

Since these plots still differed quite a bit from the given plot, more simulations were run, this time for every 10th of a percentage between 70% and 71%, which can be seen plotted in figure 8. Several of these plots, like for example figure 9, looked like a good approximation. So taking into account the stochasticity of the simulations it is quite safe to say that the the immunity level of the given population is somewhere between 70% and 71%.

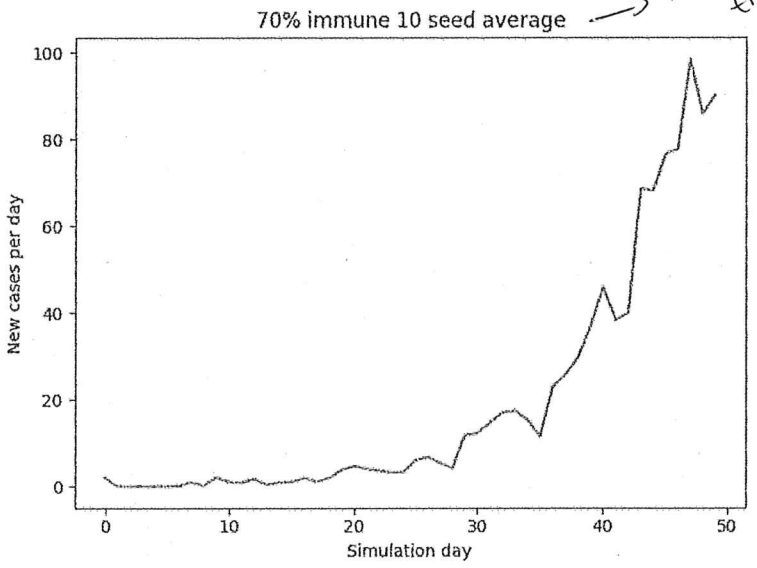


Fig. 6. New cases per day for 70% immunity level

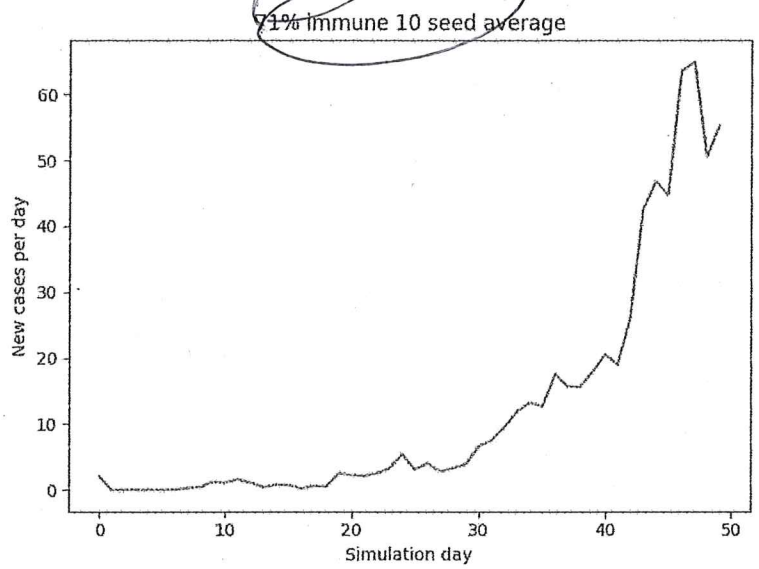


Fig. 7. New cases per day for 71% immunity level

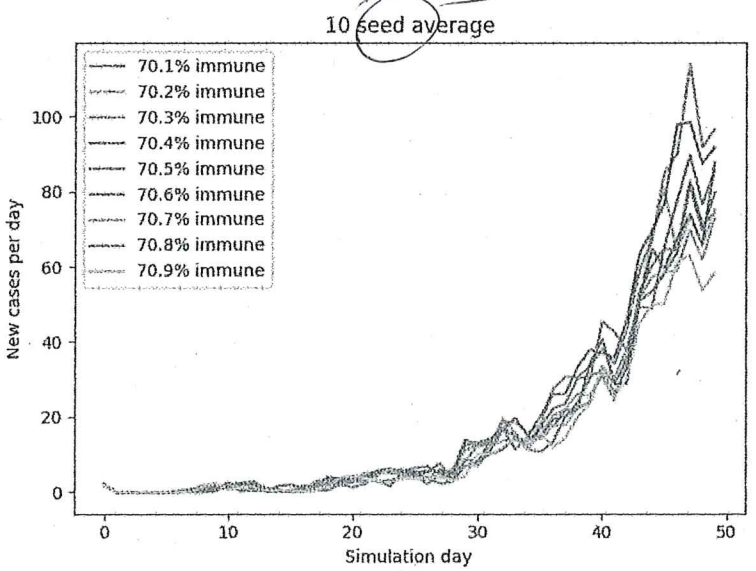


Fig. 8. New cases per day for immunity levels of 70.1% to 70.9%

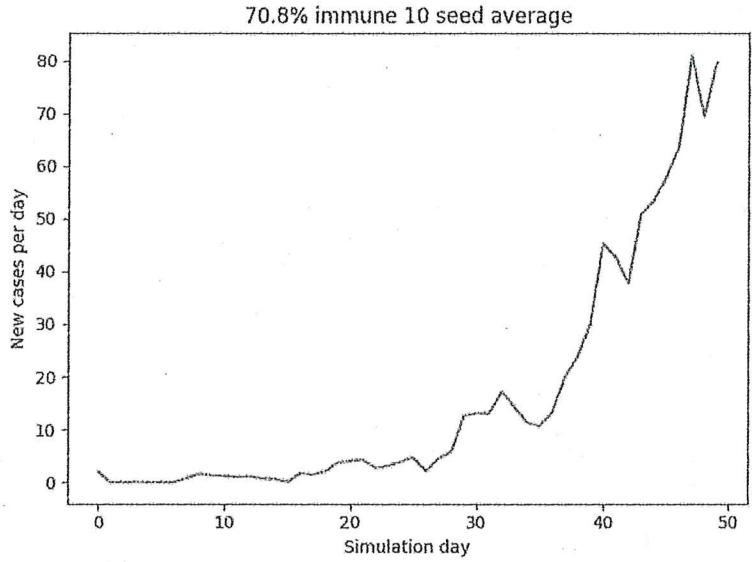


Fig. 9. New cases per day for 70.8% immunity level

1.4 Estimating R_0

During the simulations in the previous section a fixed value of 15 was used for the basic reproduction number, R_0. To see whether or not the conclusion of that section is dependant upon this value, the immunity level was fixed at a good approximation, i.e. 70,8% (figure 9), and simulations were run for R_0 ranging from 12 to 18, a range used for the basic reproduction number of measles.

Plotting the results of those simulations in one graph resulted in figure 10, which clearly shows that the results of the simulations and thus the conclusion of the previous section are indeed dependant upon the value for R_0. It is, however, noticeable that all the plots have similar oscillations, with the main difference being that plots of simulations with a higher R_0 value grow quicker.

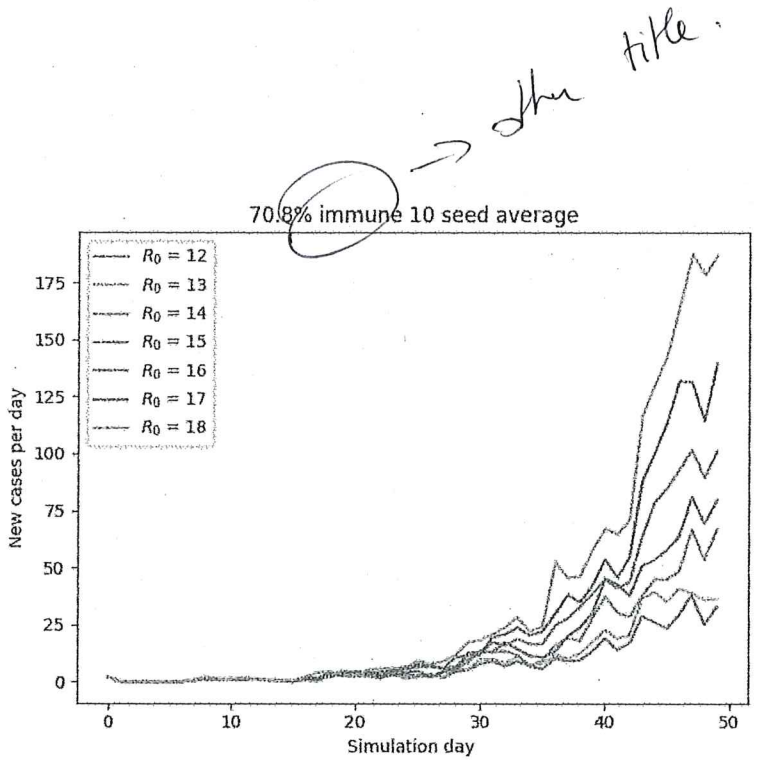


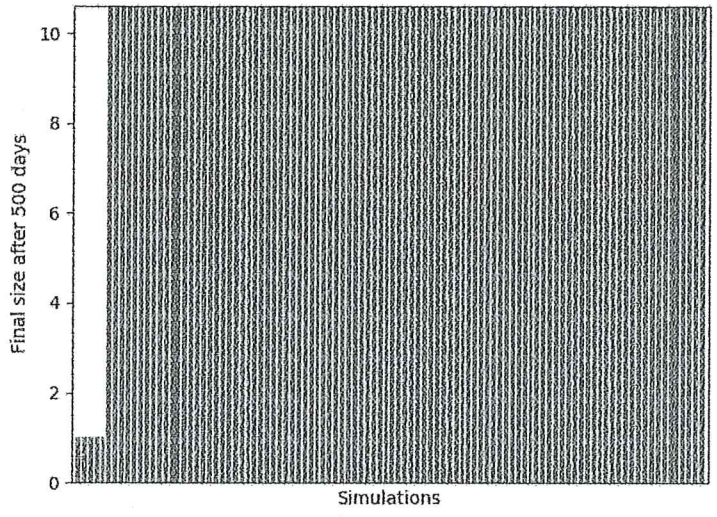
Fig. 10. New cases per day for 70.8% immunity level for several R_0

2 Population generation

Simulation results can also depend on the population in which a simulation is run. The following sections describe how they do that.

2.1 The influence of demography on epidemics

Two populations were created, named Region A and Region B. Both are quite similar, but what is important to notice is the difference in age. The people in Region A are much younger than those in Region B. The results from running simulations on both populations show that in 95% of the cases, an outbreak is present for Region A while Region B only shows outbreaks in 89% of the cases. So the chance for an outbreak is bigger in Region A with more younger people. Younger people are going out much more and to more different places which causes them to meet much more people than older people. This way the sickness can be spread more easily and that explains why the chance is higher in Region A.



→ "Zoomed" plot of what?

Fig. 11. Zoomed plot of final infected cases of Region A of 100 simulations using 100 random seeds

↳ Wouldn't it be possible to combine Fig 11 & Fig 12 in one graph. Take the average of both region A & B and compare them in a bar plot. Also plot the standard deviations.

unnecessary whitespace ?

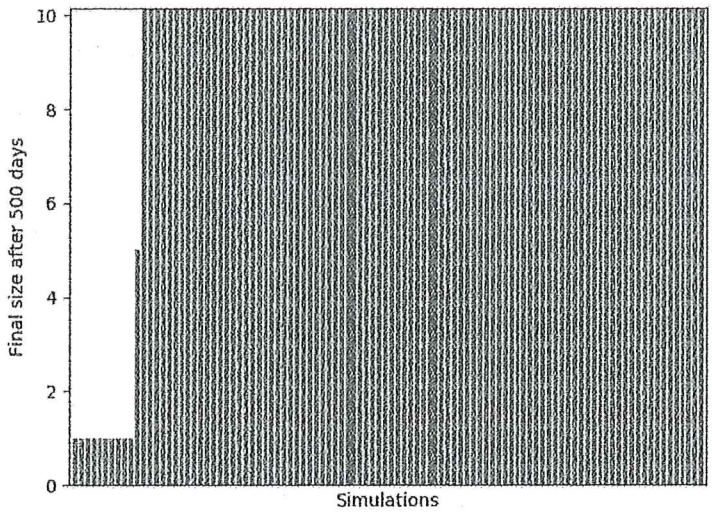


Fig. 12. Zoomed plot of final infected cases of Region B of 100 simulations using 100 random seeds

whitespace ?

2.2 Vaccinating on campus

When students get vaccinated, the amount of new cases per day is more often 0 then when they aren't vaccinated. The line which represents the simulation with vaccination doesn't only have less spikes, the spikes it has, are also lower. That vaccination will cause the spread to go slower.

This becomes even more clear when we create a cumulative ten seed average of the plot. By plotting the cumulative version, it gives us a much better overview of the total number of infected. Now we can confirm firmly that vaccinating students after 7 days has a positive effect (less infected per day) when an outbreak is suspected.

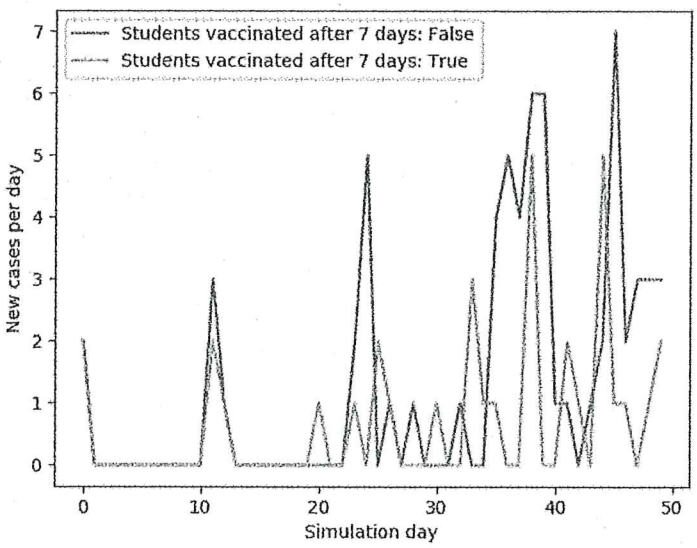


Fig. 13. New cases per day plot of 2 simulations with the same seed and one where students are vaccinated after 7 days

white space?

should be in color.

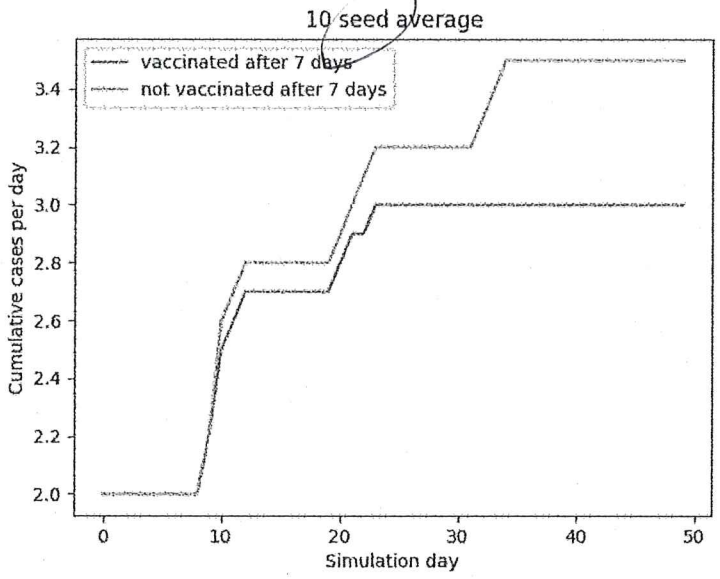


Fig. 14. The average cumulative cases per day plot of 10 simulations each, where students are vaccinated after 7 days

white space?

2.3 Commuting to work

The more people that have to commute to work, the more people they meet and thus the faster the disease will spread. This is not very noticeable in figure 15, where we used 10 different random seeds and calculated the average new cases per day.
A better result can be acquired by taking the average cumulative number of new cases per day (figure 16), this way we avoid the oscillations from the different random seeds. Here we can see there's a difference between the commuting levels, with the most significant one being between the 75% and the 100%. To put this in perspective we also calculated the same simulations over 500 days (figures 17 and 18). Here we can see that a population that commutes for 100% has a major but quick outbreak, where a commuting level of 0% has a more slowly increasing outbreak spread over a larger period.
The intuitive interpretation of these results can be derived as follows: If nobody has to go to their work space, so assuming they stay in the same city, the new cases will rise slow. If you don't meet a lot of new people, then passing the disease to someone new is much more difficult. The higher the amount of people that have to commute to work, the faster pikes of new cases for a day will appear. However the peak-values stay the same, which is visible in figure 15. So eventually the same amount of people will get infected.

Outbreak percentages for the five commuting levels: 0.00 = 98% 0.25 = 99% 0.50 = 100% 0.75 = 99% 1.00 = 99%

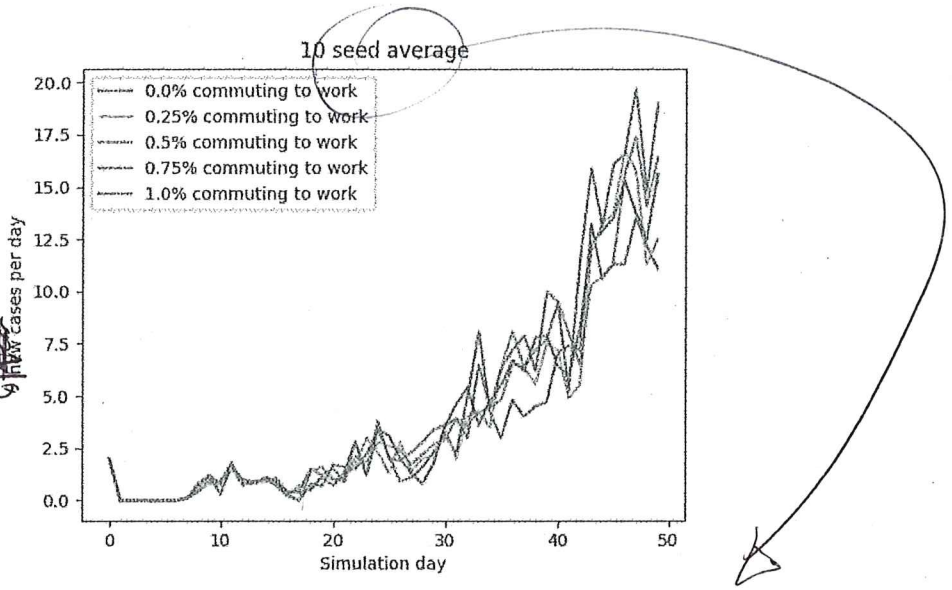


Fig. 15. Plots of the (10-seed) average new cases per day of 5 simulations with different commuting percentages over 50 days

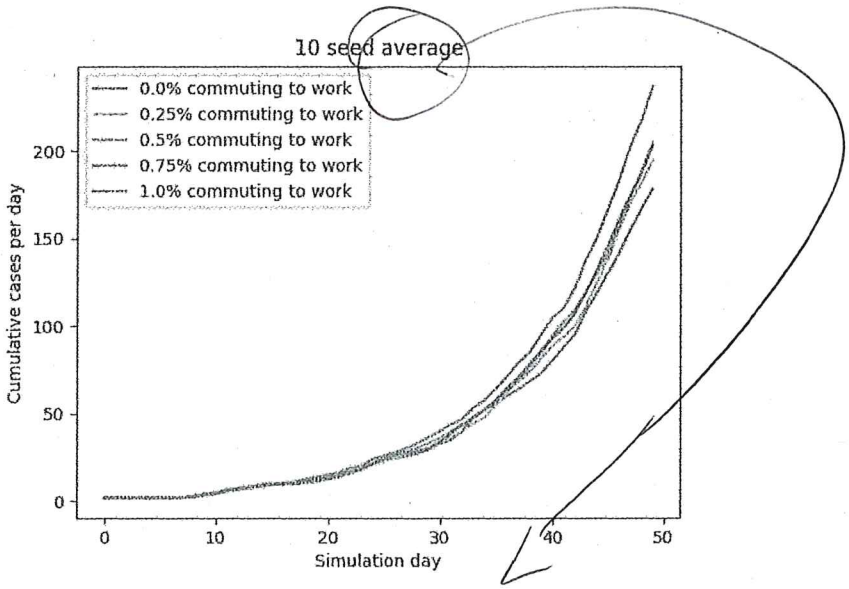


Fig. 16. Plots of the (10-seed) cumulative cases per day of 5 simulations with different commuting percentages over 50 days

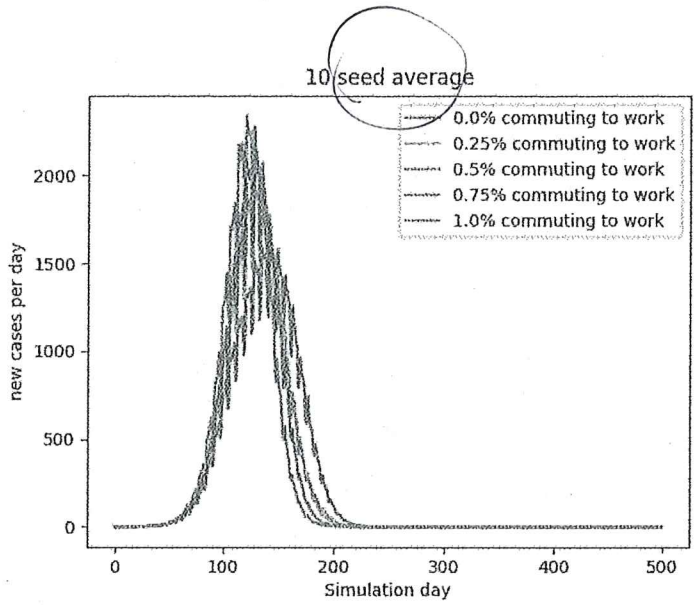


Fig. 17. Plots of the (10-seed) average new cases per day of 5 simulations with different commuting percentages over 500 days

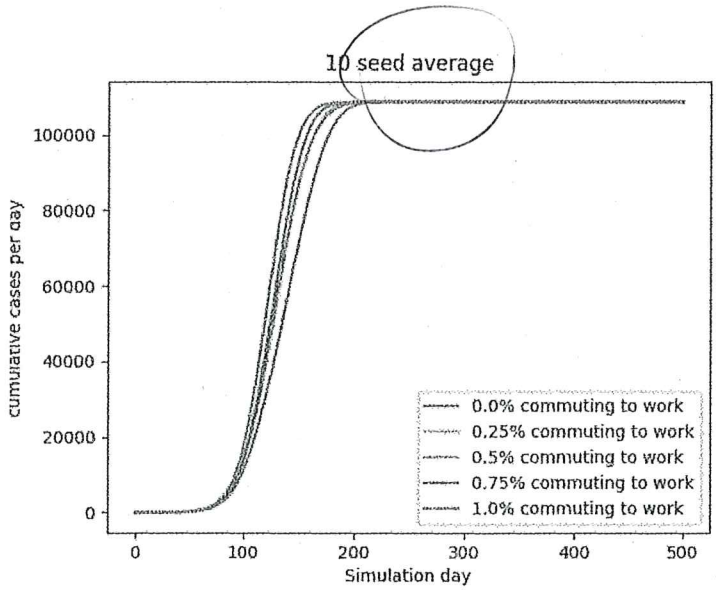


Fig. 18. Plots of the (10-seed) cumulative cases per day of 5 simulations with different commuting percentages over 500 days

3 Performance profiling of sequential code

In this section we will discuss the results from the performance profiling with different parameters. For the profiling we disabled OpenMP to get the performance of the sequential code. We run the default configuration and change only the parameter and measure the wall clock time. We don't evaluate absolute measurements. The results shown are averages from 3 tests so that they are not interfered with other processes on the computer.

3.1 Simulated Days

In figure 19 we see that the amount of days is linearly proportional to execution time.

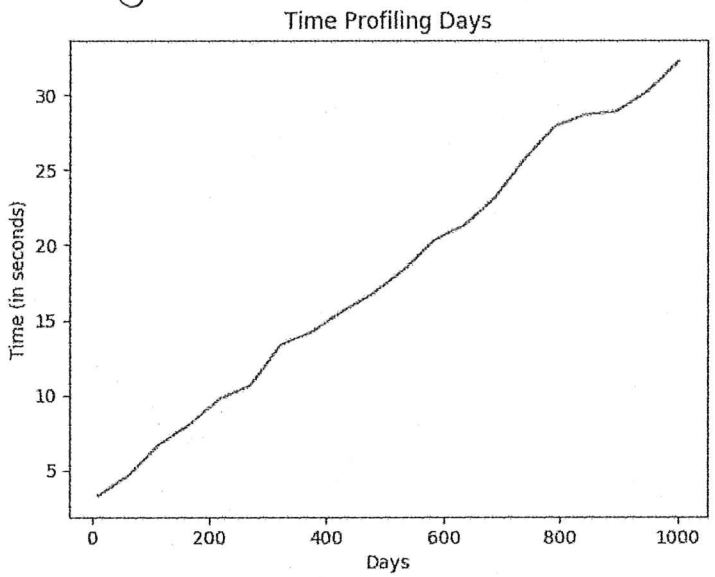


Fig. 19. Profiling plot days versus time

3.2 Population size

In figure 20 it also appears to be a linearly proportionality between the population size and time.

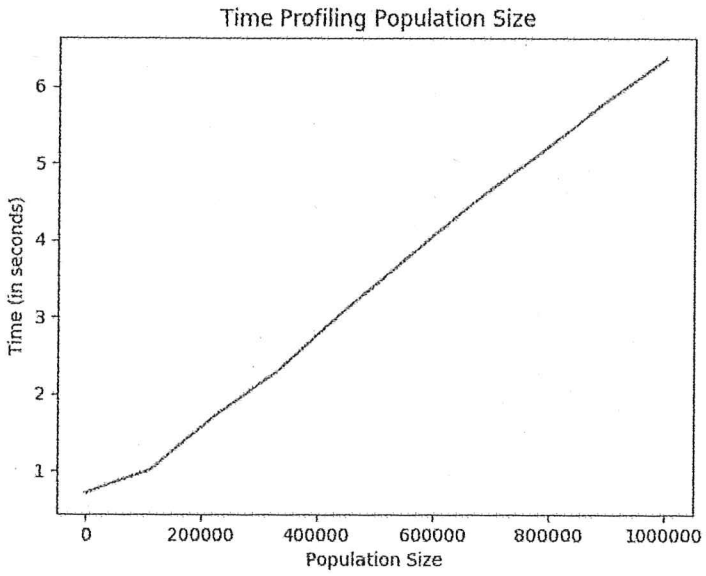


Fig. 20. Profiling plot population size versus time

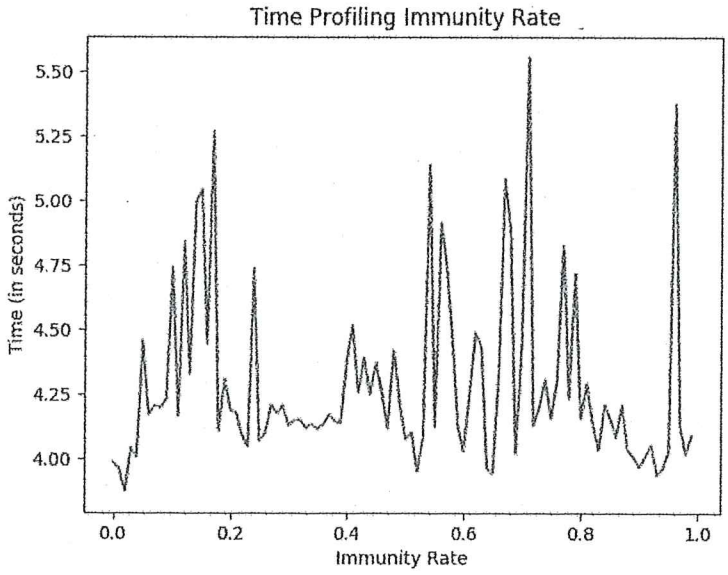


Fig. 21. Profiling plot immunity rate versus time

3.3 Immunity rate

In figure 21 we see that the differences with all immunity rates are minimal and not significant. Thus, the immunity rate doesn't affect the wall clock time of the simulation.

-> why?

3.4 Seeding rate

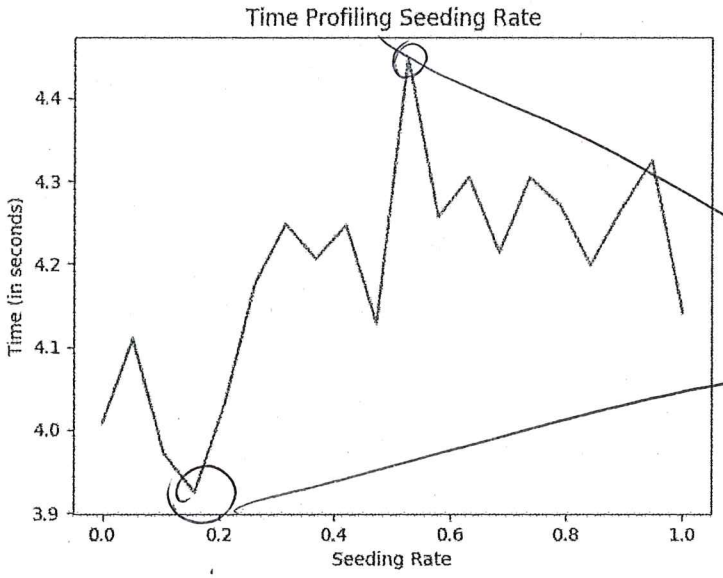


Fig. 22. Profiling plot seeding rate versus time

In figure 22 we see little differences between the execution times so we can conclude that the seeding rate solely has no direct influence to the execution time.

-> why?

difference of 0.5

$\Rightarrow \frac{0,5}{4,5} \approx 10\%$

is this a "little" difference?

3.5 Contact log mode

Here we going to monitor the time that is needed for different levels of contact log. In figure 23 we see that 'All' and 'Susceptibles' require more time for simulations. This is because the contact log modes 'All' and 'Susceptibles' use a algorithm that use that simulates all possible combinations instead of the more optimal algorithm that only simulates the interesting cases used by 'None' and 'Transmissions'. An explantion can be found in the fact that 'All' and 'Susceptibles' will loop over all possible contacts and 'None' and 'Transmissions' will only the interessting ones.

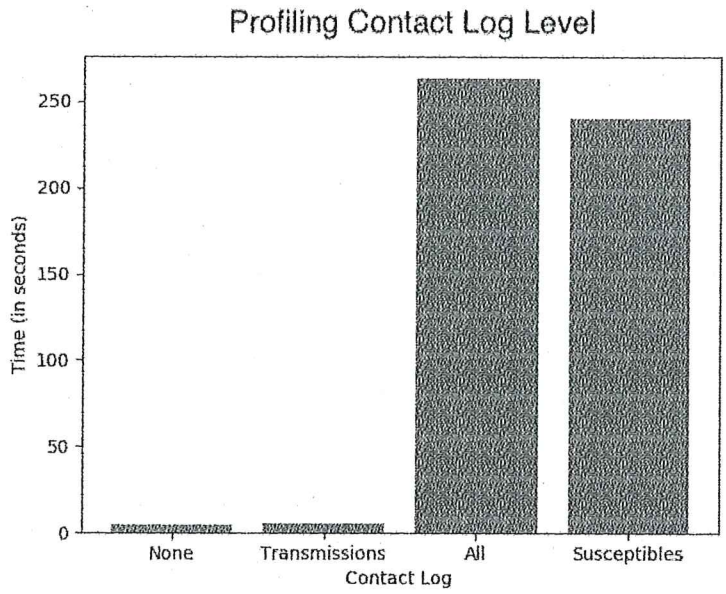


Fig. 23. Profiling plot time per type of contact log level

3.6 Call graph

In this section we try to see a overview of function timings so we can compare them later function timings. We run Stride with the default settings found in *run_default.xml* configuration. For this tests parallelization is turned off. We give a top 20 table of functions which relatively take the longest. These 20 functions take about 7.43 seconds of the 14.88 seconds the simulation ran. They occupy almost 50% of the simulation time so they are a good indication of the whole simulation. The results were acquired with *gprof*, a profiling tool from GNU.

Name			
time (%)	time (s)	calls	ns/call
stride::Health::IsImmune() const			
13,68	2,04	23085278	88,3680066577496
stride::ContactPool::SortMembers()			
4,50	0,67	10360620	64,6679445824671
std::bitset<6ul>::reference::reference(std::bitset<6ul>&,unsigned long)			
3,97	0,59	150000000	3,93333333333333
std::_Base_bitset<1ul>::_S_whichbit(unsigned long)			
3,46	0,52	307366575	1,69179098280286
stride::ContactType::IdSubscriptArray<bool>::operator[] (stride::ContactType::Id)			
2,82	0,42	150000000	2,8
std::bitset<6ul>::reference::operator=(bool)			
2,62	0,39	150000000	2,6
std::_Base_bitset<1ul>::_S_maskbit(unsigned long)			
2,49	0,37	157366575	2,35119814992479
stride::Person::Update(bool,bool,bool)			
2,45	0,37	30000000	12,3333333333333
std::bitset<6ul>::operator[](unsigned long)			
1,95	0,29	150000000	1,93333333333333
stride::util::SVIterator<stride::Person,512ul,true,stride::Person const*,stride::Person const&,true>::operator++()			
1,48	0,22	61200000	3,59477124183007
stride::util::SVIterator<stride::Person,512ul,true,stride::Person const*,stride::Person const&,true>::operator*()			
1,48	0,22	61200000	3,59477124183007
stride::Population::GetInfectedCount() const			
1,48	0,22	102	2156862,74509804
stride::Health::IsInfected() const			
1,41	0,21	91200010	2,30263132646586
std::_Base_bitset<1ul>::_M_getword(unsigned long)			
1,31	0,20	150000000	1,33333333333333
boost::algorithm::detail::is_any_ofF<char>::is_any_ofF (boost::algorithm::detail::is_any_ofF<char>const&)			
1,01	0,15	30000040	4,99999333334222
stride::ContactType::ToSizeT(stride::ContactType::Id)			
0,97	0,15	198968934	0,753886533864628
unsigned int& std::forward<unsigned int&>(std::remove_reference<unsigned int&>::type&)			
0,67	0,10	56005212	1,78554810220163
__gnu_cxx::__atomic_add(int volatile*,int)			
0,67	0,10	10361106	9,65147929188255
stride::Infector<(stride::ContactLogMode::Id)1,false,true>::Exec (stride::ContactPool&,stride::AgeContactProfile const&,stride::TransmissionProfile const&,stride::ContactHandler&,unsigned short,std::shared_ptr<spdlog::logger>)			
0,67	0,1	10360620	9,65193202723389
stride::Health::IsSusceptible() const			
0,64	0,10	21404785	4,67185257875751
(total) 49,73	(total) 7,43		

A scientific paper should always have a conclusion, mentioning the most important contributions of the paper and the most important results.