Researching epidemic flow and social behaviour in virtual environment

Authors:
Nazar Simchuk

Lyceum of innovative technologies
Russia, Khabarovsk

May 2020

Abstract

When we are dealing with hard diseases outbreak like epidemics and pandemics it's may be important to perform different analysis to be able to predict behaviour of peoples and disease itself. We spent some time to make this kind of analysis in our virtual environment. Also we need it to be absolutely user friendly. Any person should be able to understand its basics. It's main difference with science analogs, we wanted it be accessible and open project for non-scientists to understand. For this reason Virtual Epidemic Flow Environment, or VEFE for short, was created. Here we going to present our results and program which performs this kind of simulations.

Keywords: research, program, virtual, disease, simulation

Contents

1 Math of virtual environment

1.1 SIR

First things first, we need to create basic model of our environment and define mathematical base of this model. In our simulation we using improved SIR model. Before we will give you description of our version of SIR model we going to describe what is SIR model in general. According to SIR model we dividing our agents(aka. virtual representation of peoples) into 3 categories:

- *Susceptible* - a person who could potentially become infected
- *Infected* - an agent with actual infection
- *Removed* - agent that can't be infected again and can't spread infection(*e.g. dead or with immunity*)

As we can understand, agents are changing their state according to this sequence: $S \to I \to R$. Based on this we can create simulation following this plan and record it's results in parallel:

1. Initialize all agents as Susceptible
2. Insert one Infected agent(aka. Zero Patient)
3. Begin epidemic flow
4. Calculate flow until there are no Susceptible or Infected left

Next we going to present changes which we made to improve this simulation and add additional functionality to it.

1.2 Improvements

Now we are going to review the improvements we have made.

Hidden state. First improvement is adding new agent's transition class between S and I types. We implemented Hidden agent[1]. This type of agent works as Infected one, but it's state is unknown for other agents. It adds new layer of freedom to create different social behaviours(*e.g. quarantine, distancing*) which happens only when society knows about someone's infection. It may reflect on agent's behaviour itself when it knows or didn't knows about its infection. Also we can make higher or lower infectious rate[2] for agent to simulate different states of infection in organism.

Scenarios. We implemented idea of scenarios. This makes user able to add different behaviours to agents and change the rules of disease. Some of the other improvements below comes as scenarios which user can activate or deactivate if he wants.

Areas. The way it works users are able to divide environment into different zones(analog of countries). According to different scenarios they can demonstrate interactions(*e.g. travailing, help, remedy-research*).

[1] Originally, this was idea on the 3Blue1Brown "Simulating an Epidemic" video by Grant Sanderson

[2] For now we considering that infectious rate is probability of being infected during the contact with infected agent.

Transport system. This idea isn't new, but we improved it a lot. User can create it's own transport ways. This works as separate scenario. Different areas are able to close their border in cases of self-isolation or other scenarios. User can also close borders during the simulation.

Immunity. Remedy. Mutations. This is really big improvement. Now we can control remedy creation. Remedy can help agents to recover into S state. More over, after recover they are able to save immunity which lower the infectious rate. Also disease can mutate, which means that some of it's parameters are about to change during the simulation and immunity can no longer withstand newer forms of viruses. All of this comes as separate scenarios.

Agent's states flow and infectious rate. We changed the way of calculating the infectious rate and the way of states flow. It may cause more natural behaviour and makes us able to create more variable simulations. Now our environment can simulate HIV, Hepatitis and etc. More about it will be it **Math** section.

Miscellaneous. Scenario system makes us able to simulate police and government behaviour, social places, crowds, epidemic wars and even zombie apocalypse. Keep reading. We hope you'll like it.

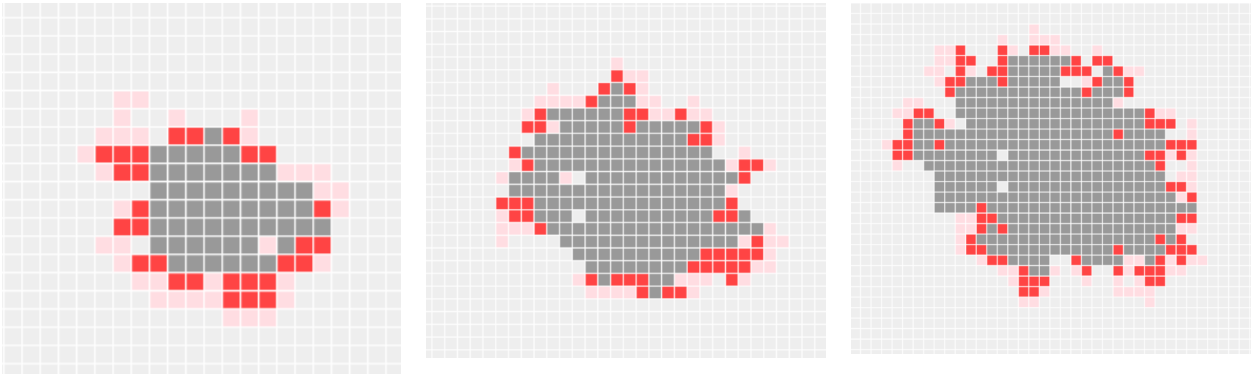
1.3 Math base

Now we are going to go through all mathematical basics. To spread infection between different agents we need to calculate their probability of being infected Let's suppose that we have two agents on a distance of $\sqrt{X}$. First agent is S and second one is I or H[1]. We can calculate S-agent infecting probability P according to this formula:

$$P = R \cdot e^{-\left(\frac{X}{\sigma}\right)^{\alpha}}$$

where X - square of distance and α, σ, R - different values that we can control to change P - infectious probability. But before diving deeper in understanding of this formula and variables roles we want to make side step to the usual and more common method of spreading infection.

We want to show you an example approach to this kind of problems in Kevin Silmer's article "Outbreak" published in March 16, 2020. His approach was to take agents as cells of square grid. His program spreads infection to the nearest infected cell neighbours in certain range. This way we can get great simulations:



Also here you can see gray cells, which represent R state of agents. We will talk about it later.

All in all, this is example of more standard approach and it can handle the most of simulations. Our goal was representativeness. Of course, this is one of the most optimized approach and nothing can be more representative. But we wanted to make control more representative in the surrounding world and nature. When we talking about nature we can't say that disease can spread in particular sharp finite range where it has some constant infectious rate which drops to the zero outside of it, name of this kind infection distribution is uniform distribution and it's looks like rectangular function graph. We talking about more even natural distribution and variables which are not so abstract as constant range.

1.4 Math of the infectious rate

"King" of the natural-kind distribution is Gaussian's distribution. This is function that represents probability distribution over all possible input values. And now take a look at it:

$$P = e^{-x^2}$$

[1]As we said before, this is hidden state of infection

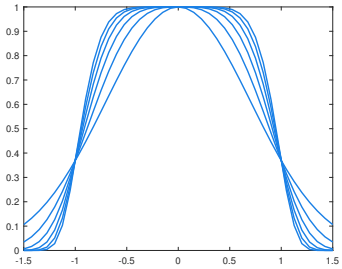
where $e = 2.718281\ldots$ This is close enough to what you seen in our formula[2]. Now you may understand why we had chosen exponential function. This function is great in representing nature behaviours. We using minus in the power value of this function to determine that rate decreases over the distance. Also in our formula we use absolute value of distance which means that our distance can't be negative value because for non even α-variable value our function has non even distributed graph. Now we going to tell, what our variables in probability function represent:

Look at this formula again. Then we will show you how distribution graph changes according to changes of 3 main variables α, σ and R:

$$P = R \cdot e^{-\left(\frac{X}{\sigma}\right)^{\alpha}}$$

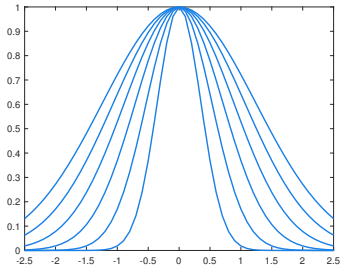
Changes of the distribution according to the values of variables:

$\alpha - value:$



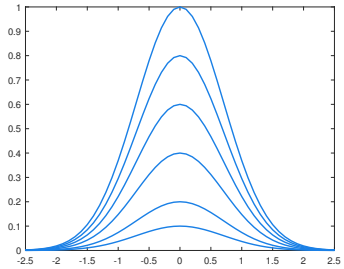
Alpha represents the ability of infection to linger in the environment. It means that infection can stay longer in some areas which makes it more concentrated in certain range.

$\sigma - value:$



Sigma is analogy to the infectious range in more standard simulations about which we talked earlier. When it's higher, effective range is higher too.

$R - value:$



R value is representation of the general infectious probability or as we talked earlier - *infectious rate* itself. It show how probably agent about to be infected with this infection over distance distribution.

Note. But one important note is that we can get same kind of range distribution about which we talked earlier. If $\alpha \to \infty$ our graph(fig. 1) represents distribution of constant infectious rate in fixed range, were range is σ. It shows that we haven't gone so far from original infection distribution. In fact, we saved this method in case we need to make fast compute. We may implement different kinds of distribution in the future. For now we consider using only normal and uniform distribution like in Kevin Silmer's article.

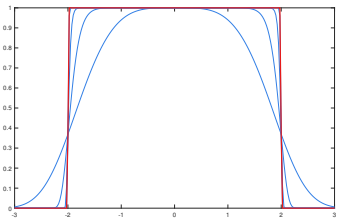


Figure 1: Range graph

[2]Note that this formula is not in the form of probability density function

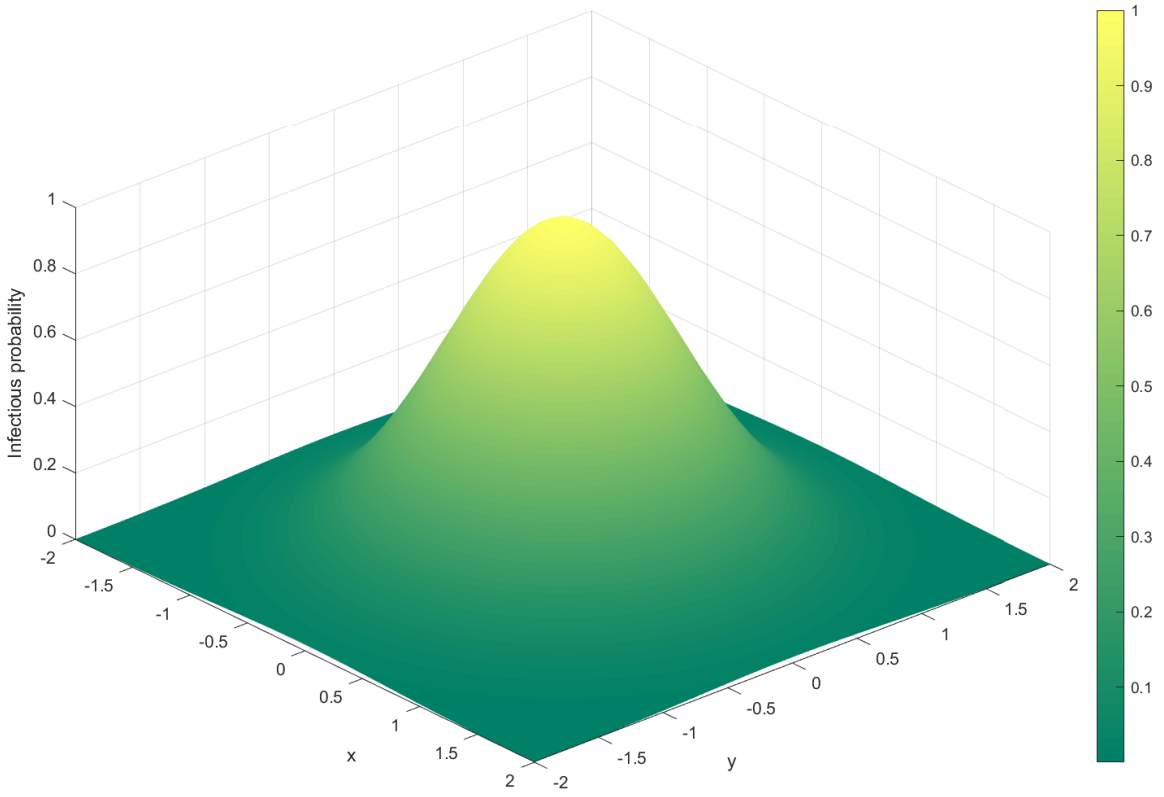


Figure 2: Range graph in expanded coordinates form

Few words about distances As you may remember, we used X as square of the distance but not distance itself. It's little optimisation trick which helps us to make better program performance. Now we can say that we can take pair of coordinates of S agent - (x_s, y_s) and pair of I agent coordinates - (x_i, y_i), then we can define that $(x, y) = (x_s - x_i, y_s - y_i)$ and substitute it in formula:

$$P = R \cdot e^{-\left(\frac{x^2+y^2}{\sigma}\right)^{\alpha}}$$

After that we can see this distribution in its full form. Also, we want to note that we calculating this infectious probability only if square of the distance between agents is lower than $(2\sigma)^{2\alpha}$ which means that we not calculating it if agents on range where infectious probability is so low that we can neglect it.

1.5 Cycles

When we talking about calculating infectious probability we talking about probability of being infected during the contact. Here word *'during'* has special meaning. If we calculate this probability each frame we will get some strange results because at frame rate 30 we have $1-(1-P_0)^{30}$ probability of being infected after only one second were P_0 is our calculated probability for one animation frame. Suppose $P_0 = 0.02$ then according to this formula after only one second we will have $P = 1-(1-0.02)^{30} = 0.45$ and this is enormous probability if we compare it with P_0. Also, there are big performance issue. Imagine we calculating this for each pair of agents. If we have n agents then we will have to make $n \cdot (n-1) = n^2 - n$ calculations each frame. For $n = 100$ we will have 9900 calculations each frame. So we need to create epidemic's calculation update rate. So, each t frames we need to do calculations. But we must remember that we can't have them do it simultaneously, so we need each agent to track it's on *cycles* and after each new cycle do calculations. Imagine each agent making it's own infection *pulse* after constant time t. This pulse is correspond to single act of spreading an infection by this agent. Also each cycle we can update our agent's state. Now let's look at it.

State change Now we need to change our states over the time period. As mentioned before or probability calculates like: $P = 1 - \Pi_{i=0}^{d}(1 - P(i))$. This also known as 'or' probability. We wanted to make natural epidemic progression and for this we can came up with formula $P(d)$ - for probability of flowing to the next state in cycle d:

$$P(d) = \frac{1 - e^{-x}}{1 + e^{-\frac{x-d_0}{\delta}}}$$

where d_0 is half probability circle number[1] and δ - is smoothness:

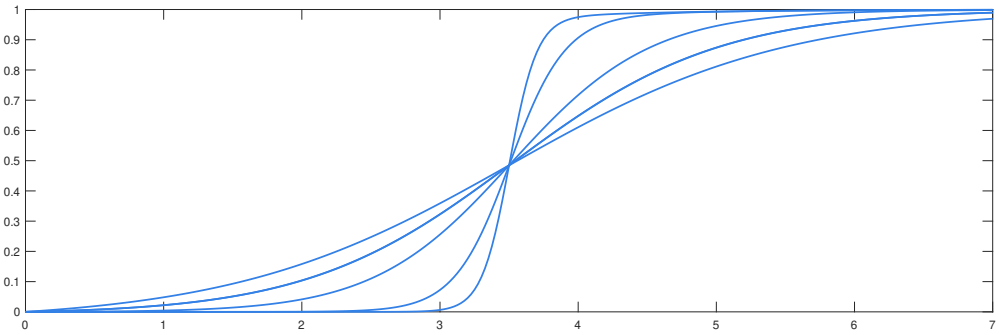


Figure 3: δ graph affect

This 2 variables makes us able to create simulations of diseases which are very unpredictable when changing states. We can simulate diseases which are hidden during long time period and can very unpredictably become known like HIV, Hepatitis, Tuberculosis and etc. This can be done by setting high values for δ and d_0.

[1]Circle on which probability of changing state will be 0.5

Now we can do this algorithm:

1. Initialise each agent with variable $d = 0$ and variable $timer$ equal to random number from 0 to s where s is time step length.

2. If agent not in S state: add 1 to $timer$, if $timer = s$: set $timer = 0$, make calculations[2] and add 1 to d, if state changes: set $d = 0$.

3. Repeat step 2 until simulation ends.

1.6 Motion

To make this system behave in natural way we need to add random motion to the agents. While we were working with motion of agents we thought about moving them according to the Perlin's noise but this idea was bad because of the noise artifacts and motion was very strange. Then we came up with idea of randomly changing direction of the agent's motion but this kind of the motion was very uncontrollable and more strange.

Finally, we got idea of using Bézier curves. We randomly creating path according to generated curve by setting random position for four points and then using linear interpolation between this points to get Bézier curve path. This method has very good advantages which we will show later.

[2]As we said earlier this corresponds to calculating probability of near agents being infected and making calculations of changing state with probability according to $P(d)$ formula.

2 Inside of virtual environment

Taking overall look, it's amazing to understand that such a natural behaviour created by such simple rules.

Stepping into our virtual environment we want to show you most of it's functionality which wasn't mentioned in math section. We previously said that we implemented different scenarios which adds different advantages to the system and gives freedom in manipulations with the environment. There are some interesting mechanics that can be useful while creating different kinds of simulations.

Math described in the previous section is more general way of creating epidemic simulations with our additions and implementations. But now we want to focus on different thing which makes this simulations more integrative with real situations. This mechanics result in creating very interesting behaviour of the agents and environment itself. This virtual environment becomes very powerful tool while working with real-world situations.

2.1 Quarantine

We implemented great idea of the quarantine. We got this idea from video about epidemic simulation form Grant Sanderson. Regions are able to enclose their agents to block their way out to lower infection spread rate.

As we said earlier we implemented idea of using the hidden state of agent. This is very useful because now we can make our quarantine to detect only known infected agents. This idea helps us to take knowledge about infected agents and isolating them skipping unknown agents which are able to spread infection. It brings us closer to real world situations and makes quarantine work hard to catch all of the infected agents.

We can't skip opportunity making quarantine limited to some amount per day and amount in total, which brings us more interesting results. This may result in different spread of the infection.

2.2 Transport

As mentioned before, we added special system to make agents travel between regions. More over, we can add system which makes regions able to close their borders which helps in cases of self-isolation. Agents traveling also based on population density. Agents tends to travel to the regions which have higher population density. This makes counties save their population count. Density level drops when region has big infectious rate and this makes agents behave more naturally - they start travelling to other regions to leave that dangerous area.

Then we can add transport system screen layout where user can edit transport ways. This is very interesting idea which we planing to work on in future.

2.3 Public Places, Distancing and Isolation

This idea isn't new, but it collaboration with ours agents movement system it gives good results. The reason for this is that we can generate our Bézier curve's points not only in random way but also modifying their positions making agents follow more specified path. Now we can make agents to choose specified points for the path more likely that the others. This points can be called public places.

Also we can implement idea of social distancing by making some portion of agents to avoid others. This can be done by making them selecting path points far from path points of the other agents but Bézier curves still can intersect each other. Also this is not the efficient solution for this. We can make agents repel each other which represents the distancing in natural way.

2.4 Remedy, Mutations, Immunity

Areas able to create their own remedy which also can be spread around the world but now disease is able to pass mutations which makes new types of viruses that resistant against previous remedy and formed immunity.

Remedy can be created with following types:

- Immunity remedy.

  This type of remedy created based on antibodies obtained from the blood plasma of those who are immune. This means, that power of remedy is proportional to the number of cured agents.

- Inactivated vaccine.

  This is remedy that created based on weak virus organisms. Reproduction of this remedy is proportional to the number of infected agents but can be reproduced in laboratories which means, that creation speed of this remedy never drops below constant boundary.

- Standard remedy.

  This remedy creates based on other methods. It's can be affected by both number of infected and number of cured agents and can be more deeply customised.

Remedy production is not equal for all regions. It appears in some regions, but other regions can get remedy production only if they connected to the region which has remedy production. All regions have special parameter - remedy reproduction level which goes from 0 to 1. Country, where remedy firstly was created, affects connected countries to rise their level of remedy reproduction. This makes remedy reproduction spread around the world. Agents may be configured to travel into the regions which has greater remedy reproduction level and this may cause either positive or negative impact on disease spreading.

Immunity system, which we have mentioned before, is created by returning agent to the S state, but not to the R state. When we removing agent we giving him some immunity rate. It helps him to withstand new infection by lowering an infectious probability by some amount, which depends on the infection it-self.

Also, we can mutate infection by changing it's values σ, α and R values during it's transition to the other's agents. More absolute difference with initial parameters more the virus power against immunity. But if immunity was formed on the stronger virus then immune agent will be stronger against new mutations. This is kind of the natural balance.

Then we can run the simulation to look what happening with this implementations and how they work in the simulation.

2.5 Updated social behaviour

We implemented different social behaviour scenarios. Now we can deal with the police. Now society can try to create groups to collectively protect from the I-agents. If we talking about simulating removed-based infection which spreads trough the removed agents then we can simulate zombies. We can add chasing behaviour to make this simulation more interesting.

But more interestingly is how we can change behaviour of the normal agents. We can make them go into a groups. People can manage groups where people isolating them-self in the city and protecting from infected ones. Group behaviour can be controlled in different ways.

After some iterations in simulation we can start creating some graphs of disease that will show how our infection spreads.

3 Research

This is some text for this part of the article

References

[1] Michel Goossens, Frank Mittelbach, and Alexander Samarin. *The LaTeX Companion*. Addison-Wesley, Reading, Massachusetts, 1993.

[2] Albert Einstein. *Zur Elektrodynamik bewegter Körper*. (German) [*On the electrodynamics of moving bodies*]. Annalen der Physik, 322(10):891–921, 1905.

[3] Knuth: Computers and Typesetting,
`http://www-cs-faculty.stanford.edu/~uno/abcde.html`