A TypeScript based agent-based disease model

Heyan Zhu (Anson)

Mathmatics & Computer Science Class

heyan.zhu@ulink.cn

July 4, 2024

Abstract

This papers brings an agent-based model implemented by TypeScript, simulating the spread of diseases.

1 Introduction

An agent-based model is a computer simulation of the interaction of agents (people) that is used to predict and demonstrate the spread of diseases. Center [2]

1.1 Math theory of spread of disease

In the field of mathematics, there also exists a model known as the SIR model, which is a set of differential equations that describes the number of people who are susceptible (S, not infected), infected (I), and recovered (R, have recovered from the illness and are no longer susceptible to the same disease). The equations of SIR model are as follows.

$$\frac{\mathrm{d}S}{\mathrm{d}t} = -\beta S I \tag{1}$$

$$\frac{\mathrm{d}I}{\mathrm{d}t} = \beta S I - \gamma I \tag{2}$$

$$\frac{\mathrm{d}R}{\mathrm{d}t} = \gamma I \tag{3}$$

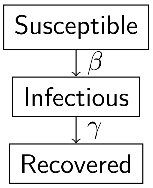

Figure 1: The flow chartNguyen [3] of SIR model.

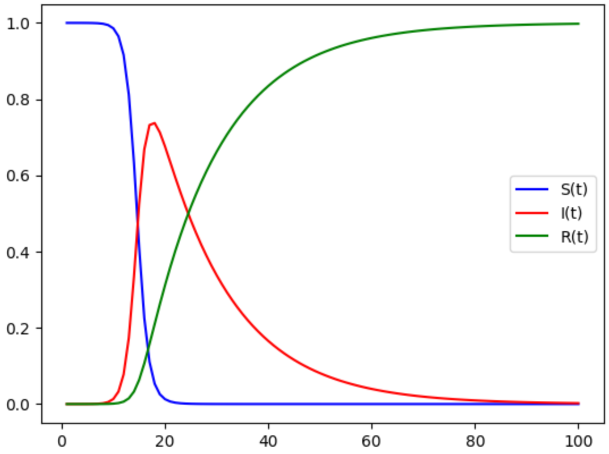

Figure 2: The diagramCastro [1] of SIR model. ($\beta = 1$ and $\gamma = 0.1$)

2 Methodology

2.1 Agent-based Model

In order to construct an agent-based model, it is necessary to define the agents, their behaviours, the interactions between them, and the environment in which the agents operate. The agent-based model was implemented in TypeScript due to the author's familiarity with this programming language. In general, agents are collections of properties, such as name, age, gender, health, and so on. They are used to represent the individuals, playing an important role in the simulation. The behaviours of the agents in this model represent their daily routines and the locations of the agents at a specific time.

2.2 Environment

The environment can be defined as the space in which the agents reside. The dynamics of the agents can be understood as the movement of the agents within this space, which is influenced by their routines.

2.3 Dynamics

2.3.1 Infection Possibility

In order to simulate the spread of the disease, we need to consider how the healthy agents interact with the infected agents and get infected. If the infected agents move close to the healthy agents, the risks of infection will be increased. A simple formulae which can roughly represent this is as follows.

$$\mathbb{P}(\text{Infection}) = \begin{cases} \dfrac{1}{d^2} & \text{(for } d > 1) \\ 1 & \text{(for } d \leq 1) \end{cases} \quad (d \text{ is the between the agents})$$

The graph of the above formula is shown in the figure below.

Where the x is the distance between the agents, calculated by the formula.

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

Every agents will randomly get infected with a certain probability.

2.3.2 Restoration

As soon as the agents are infected, a timer will start to count down. When the timer reaches 0, the agents will be restored. The restored agents will not get infected by the same disease.

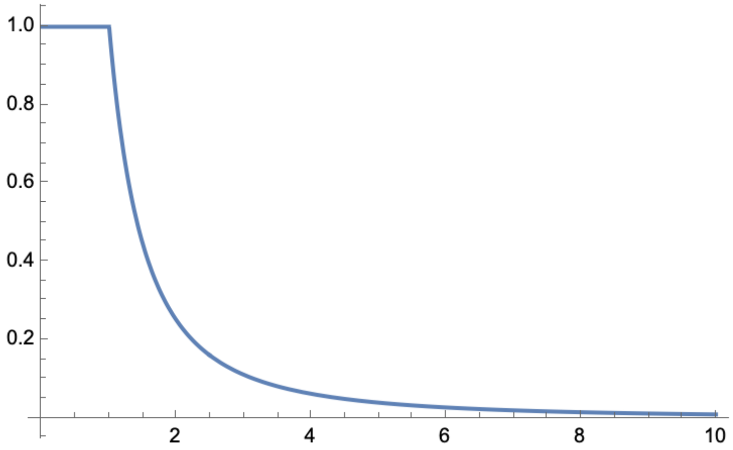

Figure 3: The graph of possibility of infection.

2.4 Implemention

In general, the model is implemented in TypeScript for the type safety which can structure the model well and make the code easier to understand, however, the graphing of the model is implemented in R, becuase of the powerful plotting package `ggplot2`.

The graph of the model .2 which shows the design and the way that the modules cooperates with each other is shown in the appendix.

The flow chart of the model .3 is also placed in the appendix.

3 Results

The log of the model can be found in the appendix. The result of the model is shown in the figure below. The parameters set in this model:

- Simulation Time: 60
- Agent Number: 40
- Room Size: 40

The disease which in this simulation model:

- Disease Name: Cold

- Disease Severity: 100
- Disease Time to Restore: 10

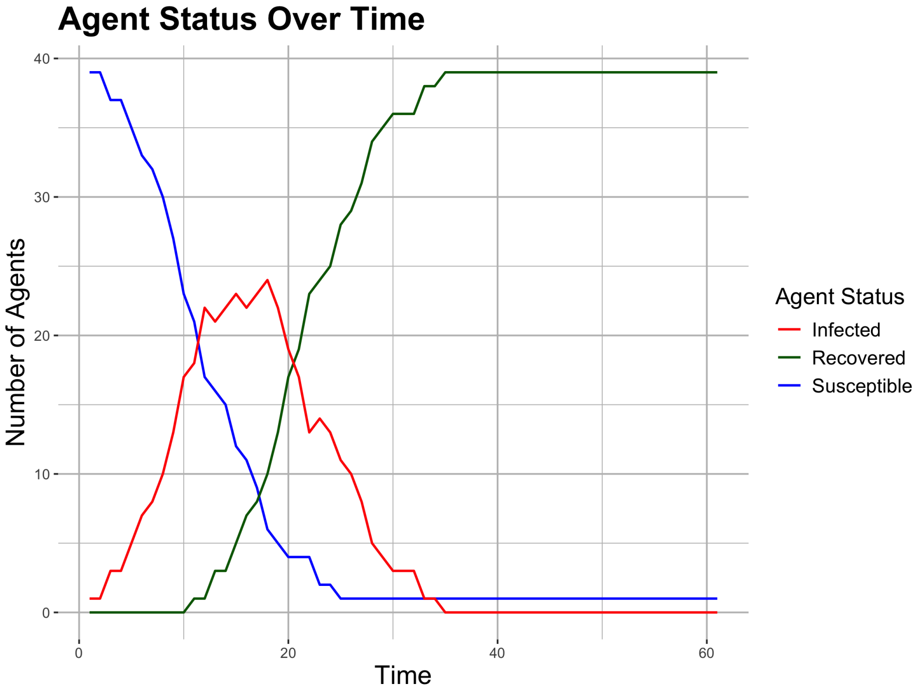

Figure 4: The diagram of the created agent-based model

It is obvious that this model can partially simulate the spread of the disease with given parameters with the agents and diseases, because of the similiarty between Figure 2 and Figure 4 with the outbreaks and extinctions.

4 Discussion

This model is a good model to simulate the simple spread of the disease, written in TypeScript, for the sake of simplicity, but lack of ability to simulate the multiple diseases with complex dynamics. Some feature that can migrate and control the parameters of β and γ are missing, should be added in the future. Becuase of the familiarity of the TypeScript, author choose to write in this language, which is not common for the data analysis. It can be better if the model can be implemented with Python or R.

5 Conclusion

The above model can successfully simulate the spread of the disease, with the outbreak and extinctions.

6 References

[1] Joaquin de Castro. *Graphing The SIR Model With Python*. `https://pub.towardsai.net/graphing-the-sir-model-with-python-e3cd6edb20de`. Accessed: 2024-07-01. 2020.

[2] Columbia University Irving Medical Center. *Agent-Based Modeling*. `https://www.publichealth.columbia.edu/research/population-health-methods/agent-based-modeling`. Accessed: 2024-07-01. 2024.

[3] Le Nguyen. *Simulation 102: Agent-Based Disease Modeling*. `https://medium.com/@ln8378/simulation-102-agent-based-disease-modeling-17307f2de2f3`. Accessed: 2024-07-01. 2023.

7 Appendix

.1 The log of the agent-based model

The log for the run from the model can be found in this URL. `https://pastecode.io/s/jb6npovz`. The routine of the agent has been set to randomly move.

.2 The structrue of the agent-based model

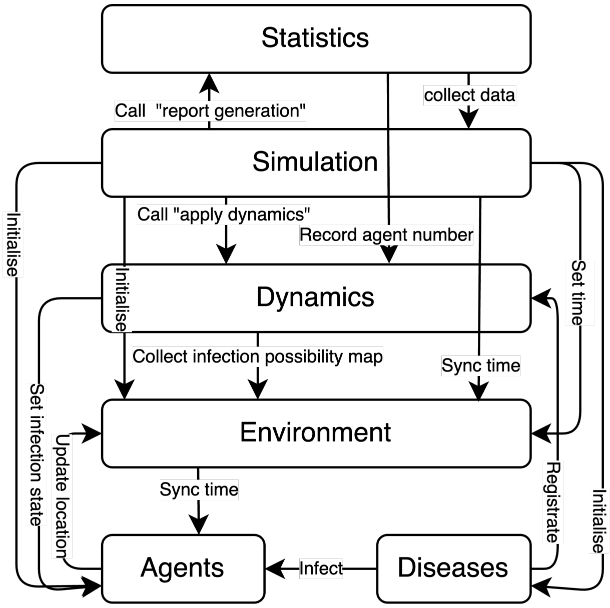

Figure 5: The structure of the created agent-based model

.3 The flow chart of the agent-based model

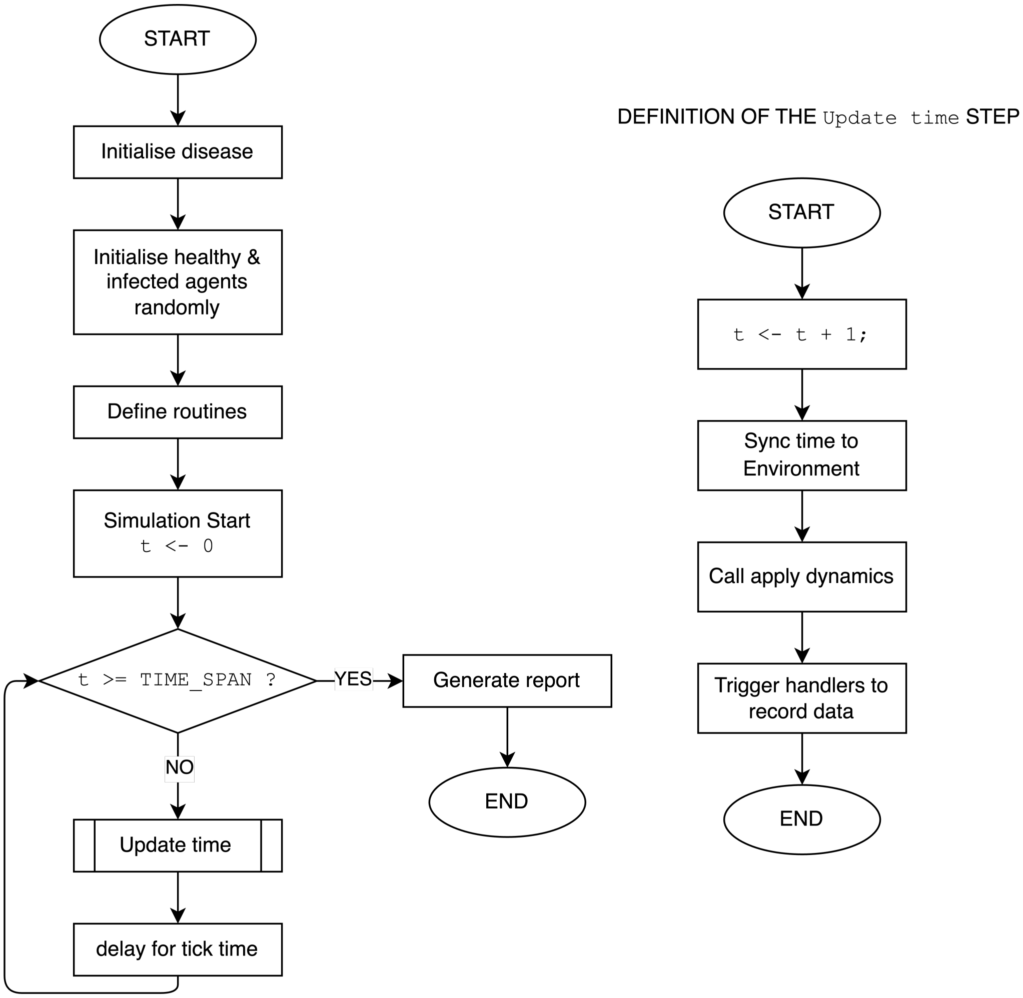

Figure 6: The flow chart of the created agent-based model