Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Lecture with Computer Exercises:
Modelling and Simulating Social Systems
in MATLAB (or Python)

Project Report

Modeling and Simulating Disease Outbreaks for Arbitrary Topological Systems

Thomas Hahn
Severin Jäger
Yannick Schubert

Zurich
December 2018

Agreement for free download

We hereby agree to make our source code for this project freely available for download from the web pages of the COSS chair. Furthermore, we assure that all source code is written by ourselves and is not violating any copyright restrictions.

Thomas Hahn   Severin Jäger   Yannick Schubert

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Declaration of originality

The signed declaration of originality is a component of every semester paper, Bachelor's thesis, Master's thesis and any other degree paper undertaken during the course of studies, including the respective electronic versions.

Lecturers may also require a declaration of originality for other written papers compiled for their courses.

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor.

Title of work (in block letters):

MODELING AND SIMULATING DISEASE OUTBREAKS FOR ARBITRARY TOPOLOGICAL SYSTEMS

Authored by (in block letters):
For papers written by groups the names of all authors are required.

Name(s):	**First name(s):**
Hahn	Thomas
Jäger	Severin
Schubert	Yannick

With my signature I confirm that
- I have committed none of the forms of plagiarism described in the 'Citation etiquette' information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work.

I am aware that the work may be screened electronically for plagiarism.

Place, date	**Signature(s)**
Zürich, 8.12.2018	Thomas Hahn
	S.
	Y. Schubert

For papers written by groups the names of all authors are required. Their signatures collectively guarantee the entire content of the written paper.

Abstract

This paper discusses a very powerful model that classifies the spread of infectious diseases. It provides the tools with which we can predict whether or not a disease will turn into an epidemic. In our work, we are able to show the underlying concept, as well as validate the model experimentally for practically relevant cases. Additionally, we found circumstances, under which the model reaches its limits.

Contents

1 Individual contributions

Thomas Hahn mainly focuses on the theory behind the model. This includes understanding when the model can potentially fail and why it works otherwise. Yannick Schubert and Severin Jäger implement the theory and verify results by simulating the time evolution of the SIR model and comparing outcomes. In addition, they analyze what happens when certain important nodes are infected.

2 Introduction and Motivations

The social sciences have long tried to predict the time evolution propagation of viruses, ideas, etc. Obviously, a deep understanding of how and why diseases spread would allow us to better come up with prevention measures. Their first attempt was to parametrize the disease and analyze the spreading during short time-steps. The topology of the system, i.e. a structure that describes the connection of any all players (or lack thereof), was completely ignored. It was just assumed that everyone knew everyone. More recently, experts in this field have managed to not only create a theory that includes both aspects, but also achieve a decoupling. This, of course, is by no means trivial.

We will base our work on the paper 'Threshold Conditions for Arbitrary Cascade Models on Arbitrary Networks" from 2012 [2]. They claim to be the first ones to decouple the virus propagation model form the topology of the network. There, they define a virus strength as the product of the largest eigenvalue of the connectivity matrix and a constant, which only depends on the virus propagation model.

The G2-Threshold Theorem, which is formulated and proven in the paper, states that there exists a threshold for the effective strength below which a virus dies out quickly. For many "what-if"-scenarios this is already the necessary information. Since there exist very fast algorithms for computing the largest eigenvalue of a matrix, this information can be obtained quickly for an arbitrary virus propagation model and an arbitrary topology.

In this paper, we will investigate whether the G2-Theorem still applies for different initial conditions. Thereto, we will vary the number of initially infected nodes in a network and infect its most important nodes. Additionally, we will figure out, whether virus propagation in very small networks can still be analyzed with this theorem.

3 Description and Proof of the G2-Threshold Theorem

For any given epidemiological model, there are two key components that will describe our system; namely the virus propagation model and the topological constraints of the system. It will be shown that these two components can in fact be de-coupled and described by two constants. To fully describe arbitrary systems, some definitions must be introduced.

Definition:

The connectivity matrix is a symmetric $n \times n$ matrix, where n is the number of nodes in the system. If two nodes, i and j, are connected, the $(i, j)^{th}$ entry is equal to 1. Otherwise it is equal to 0. In addition, all diagonal entries are automatically set to 0.

Throughout this paper, the connectivity matrix will fully describe the interaction between two different people. In later chapters, we may allow values ranging from 0 to 1, to better describe the relationship of any two given nodes.

To mathematically characterize any given virus propagation model (VPM), the following parameters are needed.

Parameters:

(i) β is the attack or transmission probability via a link between two nodes

(ii) γ is the immunization loss probability after a nodes has recovered

(iii) δ is the healing probability once a node has been infected

(iv) ϵ is the virus maturation probability after a node has been exposed. This therefore means that $1 - \epsilon$ defines the virus incubation probability

(v) θ is the direct immunization probability whilst a node is susceptible to the virus

We now have the tools to prove the theorem, which not only allows us to decouple the topology from the VPM, but also determines whether or not viruses exceed the epidemic threshold.

3.1 G2-Threshold Theorem

For any given VPM that can be thought of as a $S^*I^2V^*$ model (there are two infectious states, but not conditions for the number of susceptible and vigilant states), which operates on an arbitrary topology, the virus will be eradicated quickly if

$$s < 1 \tag{1}$$

where s is the effective strength and is defined as

$$s = \lambda_1 \cdot C_{VPM} \tag{2}$$

Here, λ_1 is the largest eigenvalue of the connectivity matrix and C_{VPM} is a constant that is derived from the VPM.

3.2 Proof Overview

We shall start by first giving a broad overview of why the G2-Threshold Theorem makes sense and in the following subsection we shall go into more detail.

Terminology:

In the general $S^*I^2V^*$ model, we have the following three classes:

(i) Susceptible Class: Elements of this state can be infected by neighboring elements; i.e.

the i^{th} node can be infected by the j^{th} node if $A_{i,j} = 1$. We can have $n \geq 0$ arbitrary susceptible states. In addition, nodes can only enter this class via endogenous transitions from the to two other classes. This means that these transitions do not depend on neighboring elements. Transitions within this class are also endogenous.
(ii) Infected Class: Elements in one of the two possible infected classes can spread the infection to susceptible neighbors with a certain probability. For those wondering why we have two classes instead of one, we need only give the following example. If infectious state I has a transmissibility parameter $\beta_I > 0$ and infectious state E with $\beta_E = 0$, we can elegantly describe scenarios where the disease need not always be contagious. Elements can enter this class via exogenous transitions (neighbor-dependent) from susceptible states. Transitions within this class are endogenous.
(iii) Vigilant Class: Elements of these states can neither infect, nor be infected. This class can be used to describe dead, vaccinated, and/or recovered nodes (only in models, where recovered patients are no longer susceptible). Once again, there can be $m \geq 0$ possible states in this class. Elements can enter this class via endogenous transitions from all other states. Transitions within this class are also endogenous.

Assumptions:

We will be making two assumptions, which not only simplify calculations, but are rather intuitive.
(i) The infection can only be spread from node i to node j if $A_{i,j} = 1$. In essence, this means that the infection can only be spread via neighbors.
(ii) When an element from a susceptible state becomes infected, it must always transition into the "first" susceptible state I_1. This is trivially fulfilled for models with one infectious state. Models with two susceptible states could then describe infections that first need an incubation period. In this case I_1 would be the state "E" that was mentioned before.

Proof Sketch:

We start by defining a vector $\mathbf{P}_t$ that will describe the system at time t. In a model where m is the total number of states and n is the total number of nodes, this vector will have $m \times n$ entries, i.e. $\mathbf{P}_t \in \mathbb{R}^{m \cdot n}$. More specifically, we can write $\mathbf{P}_t$ in the following way:

$$\mathbf{P}_t = (P_{s_1,1,t}, P_{s_1,2,t}, ..., P_{s_1,n,t}, P_{s_2,1,t}, ..., P_{s_m,n,t})^T, \tag{3}$$

where $P_{s_i,j,t}$ is given by the probability that node j is an element of state s_i at time t. For any given VPM, $\mathbf{P}_{t+1}$ can be determined by $\mathbf{P}_t$, i.e.

$$\mathbf{P}_{t+1} = g(\mathbf{P}_t), \tag{4}$$

where g is a (potentially non-linear) function. For a fixed point, we have that

$$\mathbf{P}_{t+1} = \mathbf{P}_t. \tag{5}$$

At these points, one can calculate the stability, i.e. will we return to this fixed point under small perturbations or not, by analyzing the spectrum of the Jacobi matrix at these locations. We will in essence calculate the stability of systems, where no nodes are infected. As we will see, the constraints for the spectrum of the Jacobian will allow us to decouple the topology from the VPM and deliver the desired result.

3.3 Proof

To make the proof more readable, we shall introduce additional notation.

3.3.1 Notation

(i) m denotes the total number of states in the VPM.
(ii) q denotes the total number of states in both the Susceptible, as well as the Vigilant class. This means that $m = q + 2$.
(iii) w denotes the total number of states in the Susceptible Class.
(iv) $S_1, ..., S_w$ denote the states in the Susceptible class.
(v) E, I denote the two states in the Infected class.
(vi) α_{KU} denotes the probability of a transition from state K to state U. We assume that α_{KU} is not only constant, but also given to us beforehand.
(vii) β_1 denotes the transmission probability to for state E.
$(viii)$ β_2 denotes the transmission probability to for state I.
(ix) $\zeta_{i,t}(E, I)$ denotes the probability of node i not being infected by E or I at time t.
(x) $\mathbf{x}$ denotes the fixed point vector of our system that has no infected node.
(xi) $p^{\star}_{S_y}$ denotes the probability of being an element of the S_y state at vector $\mathbf{x}$. We assume that this value is independent of the node.
(xii) $\mathcal{J}$ denotes the Jacobian matrix of the non-linear differential equation at $\mathbf{x}$.

3.3.2 Deriving the System Equations

We will now describe the function g in terms of our parameters. The first step will be to derive the probability that an arbitrary node i will not be infected by its neighbors during one time-step. In other words, we want to calculate $\zeta_{i,t}(E, I)$. We know that the virus will not be transmitted if one of the following holds:
(i) The neighbor is not in any of the two infected states E or I.
(ii) A neighbor is in state E, but the virus is not transmitted with a probability of $1 - \beta_1$
(iii) A neighbor is in state I, but the virus is not transmitted with a probability of $1 - \beta_2$
To only consider first-order effects, we shall be assuming that our time steps are infinitesimal, i.e. $\Delta t \to 0$. In addition, we postulate that all neighbors are independent of one another. By doing so, we can write $\zeta_{i,t}(E, I)$ as follows:

$$\zeta_{i,t}(E, I) = \prod_{J \in N\mathcal{E}(i)} (P_{E,j,t}(1 - \beta_1) + P_{I,j,t}(1 - \beta_2) + (1 - (P_{E,j,t} - (P_{I,i,t})), \tag{6}$$

where $N\mathcal{E}(i)$ is just the set of all neighboring nodes. This just says that for each neighbor, we must consider him to be in (i) state E with probability $P_{E,j,t}$ and not spread the disease with probability $(1 - \beta_1)$, (ii) state I with probability $P_{I,j,t}$ and not spread the disease with probability $(1 - \beta_2)$, and (iii) none of the Infectious states with probability $(1 - (P_{E,j,t} - (P_{I,i,t}))$, where he cannot spread the disease at all. As all neighbors are independent, we must multiply the probabilities. We can rewrite equation (6) and attain

$$\zeta_{i,t}(E, I) = \prod_{J \in \{1,...,n\}} (1 - \mathbf{A}_{i,j}(\beta_1 P_{E,j,t} + \beta_2 P_{I,j,t})). \tag{7}$$

Here, we just count over all the indices and if i and j are neighbors $A_{i,j} = 1$ and we get the same terms as above. If, on the other hand i and j are not neighbors, $A_{i,j} = 0$ and we are in essence multiplying everything by 1, which changes absolutely nothing.

We also demand that for any node i at any given time t, it must be in exactly one of the states. This just means that

$$\forall i, t : \sum_{K} P_{K,i,t} = 1. \tag{8}$$

Having finally found a formula for $\zeta_{i,t}(E, I)$, we are now ready determine g. If a node is in a particular state S_y, which is located in the Susceptible class at time $t + 1$, it must be for one of the following reasons:
(i) The node was already an element of the state S_y and stayed there.
(ii) The node was in a different state U and transitioned into that state during our time step. This transition must of course be internal (See Terminology).
We can now use this information to attain the following result:

$$\forall y = 1, 2, ..., w : P_{S_y,i,t+1} = \sum_{K \neq S_y} \alpha_{KS_y} P_{K,i,t} + P_{S_y,i,t} \left(\zeta_{i,t}(E, I) - \sum_{K \neq E, S_y, I} \alpha_{S_y,K} \right) \tag{9}$$

The term $\alpha_{KE} P_{K,i,t}$ denotes all the transitions from other susceptible state to S_y, whereas the second term gives us the probability that the element in S_y stays in S_y. Through analogous reasoning we attain similar results for elements in state E at time $t + 1$

$$P_{E,i,t+1} = \sum_{K \neq S_1,...S_w} \alpha_{KE} P_{K,i,t} + \sum_{y=1}^{w} P_{S_y,i,t} \left(1 - \zeta_{i,t}(E, I)\right), \tag{10}$$

as well as for all other states $U \neq \{S_1, ..., S_w, E\}$ at time t+1

$$P_{U,i,t+1} = \sum_{K} \alpha_{KU} P_{K,i,t}. \tag{11}$$

Equations (9) to (11) now define g, which is what we wanted.
We will now need a theorem, which will not be proven (e.g. see [3]).
Asymptotic Stability Theorem:
A system that is determined by $\mathbf{P}_{t+1} = g(\mathbf{P}_t)$ is asymptotically stable at some fixed point $\mathbf{P} = \mathbf{x}$, if the absolute value of all eigenvalues of the the Jacobian $\mathcal{J} = \nabla g(\mathbf{x})$ are less than one. The entries of the Jacobian matrix are given by

$$\mathcal{J}_{i,j} = [\nabla g(\mathbf{x})]_{i,j} = \frac{\partial g_i}{\partial x_j}|_{\mathbf{x}} \tag{12}$$

3.3.3 A More Thorough Analysis of Fixed Points

We are only interested in fixed points, where none of the nodes are infected. If that is the case, the only possible transitions will be from endogenous transitions within and between the Susceptible and Vigilant class. This means that now the topology plays no role whatsoever in determining our preferred fixed point, which allows us to use the Markov Chain model. Using this method, we will attain a unique steady state probability for each node (same for each one). The steady state vector $\pi^\star$, whose entries will contain the probability that any node is found in an arbitrary state (In either the Susceptible or Vigilant class). It therefore is a vector with q entries, where q is the number of states in the Susceptible and Vigilant class. We then know from Markov chain analysis that

$$\pi^{\star T} \mathrm{Trans}_{MC_{SV}} = \pi^\star, \tag{13}$$

where we set the constraint that

$$\sum_{i=1}^{q} \pi^{\star} = 1. \tag{14}$$

Here, $\text{Trans}_{MC_{SV}}$ denotes the stochastic matrix of the Markov chain MC_{SV}. To create a probability vector for an arbitrary node i with has entries for all states, including the Infected states, we just add two 0 entries to $\pi^{\star}$. We shall call this new vector $p^{\star}$. To finally attain the fixed point that describes all nodes, we define $\mathbf{x}$ to be

$$\mathbf{x} = \left(p^{\star}, p^{\star}, ..., p^{\star}\right)^T, \tag{15}$$

where $p^{\star}$ is included n times. We define $p^{\star}_{S_y}$ to be the steady state probability of an arbitrary node i being an element in state S_y. In addition, let

$$p^{\star}_S = \sum_{y=1}^{w} p^{\star}_{S_y}. \tag{16}$$

This can be interpreted as the probability of any given node i being in the Susceptible class.

3.3.4 Deriving the Jacobian

Here, we rely on equations (9) to (11) to express g and then take all of the partial derivatives. As all of our n nodes can be elements of m possible states, we expect our Jacobian matrix to be of the form $(m \cdot n) \times (m \cdot n)$. By plugging in the values, we get the following results:

	S_y	K	$\cdots$	E	I
S_y	$(1 - \sum_{K \neq S_y, E} \alpha_{S_y,K})\mathbb{1}$	$\alpha_{KS_y}\mathbb{1}$	$\cdots$	$\alpha_{ES_y}\mathbb{1} - p^*_{S_y}\beta_1\mathbf{A}$	$\alpha_{IS_y} - p^*_S\beta_2\mathbf{A}$
$\vdots$		$\ddots$			
U	$\alpha_{SyU}\mathbb{1}$	$\alpha_{KU}\mathbb{1}$	$\cdots$	$\alpha_{EU}\mathbb{1}$	$\alpha_{IU}\mathbb{1}$
$\vdots$			$\ddots$		
E	$\alpha_{S_yE}\mathbb{1}$	$\alpha_{KE}\mathbb{1}$	$\cdots$	$\alpha_{EE}\mathbb{1} + p^*_S\beta_1\mathbb{A}$	$\alpha_{IE}\mathbb{1} + p^*_S\beta_2\mathbb{A}$
I	$\alpha_{S_yI}\mathbb{1}$	$\alpha_{KI}\mathbb{1}$	$\cdots$	$\alpha_{EI}\mathbb{1}$	$\alpha_{II}\mathbb{1}$

Table 1: Denotes the Components of The Jacobian. The first row and columns represent the states

Here, the state K denotes all states not equal to E or I, whereas $U \neq \{S_1, ..., S_w, E, I\}$. We shall now recall one of our previous assumptions, namely that there can be no endogenous transitions into the Infected class. Mathematically speaking this means

$$\forall K \neq E, I : \alpha_{KE} = \alpha_{KI} = 0 \tag{17}$$

This, of course simplifies our Jacobian, which is reduced to

	S_y	K	$\cdots$	E	I
S_y	$(1 - \sum_{K \neq S_y, E} \alpha_{S_y,K})\mathbb{1}$	$\alpha_{KS_y}\mathbb{1}$	$\cdots$	$\alpha_{ES_y}\mathbb{1} - p^*_{S_y}\beta_1\mathbf{A}$	$\alpha_{IS_y} - p^*_S\beta_2\mathbf{A}$
$\vdots$		$\ddots$			
U	$\alpha_{SyU}\mathbb{1}$	$\alpha_{KU}\mathbb{1}$	$\cdots$	$\alpha_{EU}\mathbb{1}$	$\alpha_{IU}\mathbb{1}$
$\vdots$			$\ddots$		
E	$\mathbf{0}_{N\times N}$	$\mathbf{0}_{N\times N}$	$\cdots$	$\alpha_{EE}\mathbb{1} + p^*_S\beta_1\mathbb{A}$	$\alpha_{IE}\mathbb{1} + p^*_S\beta_2\mathbb{A}$
I	$\mathbf{0}_{N\times N}$	$\mathbf{0}_{N\times N}$	$\cdots$	$\alpha_{EI}\mathbb{1}$	$\alpha_{II}\mathbb{1}$

Table 2: Here we set the constants of equation (17) to 0

Now that we have attained our matrix, we must calculate all of its eigenvalues.

3.3.5 Calculating the Eigenvalues of our Jacobian

As we can see in Table 2, the Jacobian is of the form

$$\mathcal{J} = \begin{pmatrix} \mathbf{B}_1 & \mathbf{B}_2 \\ \mathbf{0} & \mathbf{B}_1 \end{pmatrix} \tag{18}$$

where $\mathbf{B}_1$ is a $((m-2)\cdot n) \times ((m-2)\cdot n)$ matrix, $\mathbf{B}_2$ is a $((m-2)\cdot n) \times (2\cdot n)$ matrix, and $\mathbf{B}_3$ is a $(2\cdot n) \times (2\cdot n)$ matrix. Any eigenvector $\mathbf{v}$ with eigenvalue $\lambda_{\mathcal{J}}$ can now be thought of a composition of two smaller vectors, $\mathbf{v}_1$ and $\mathbf{v}_2$, such that

$$\mathbf{v} = (\mathbf{v}_1, \mathbf{v}_2)^T. \tag{19}$$

Since $\mathbf{v}$ is an eigenvector, we know that

$$\begin{pmatrix} \mathbf{B}_1 & \mathbf{B}_2 \\ \mathbf{0} & \mathbf{B}_1 \end{pmatrix} \begin{pmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \end{pmatrix} = \lambda_{\mathcal{J}} \begin{pmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \end{pmatrix}. \tag{20}$$

This, in turn, can be written as

$$\mathbf{B}_1\mathbf{v}_1 + \mathbf{B}_2\mathbf{v}_2 = \lambda_{\mathcal{J}}\mathbf{v}_1 \tag{21}$$

$$\mathbf{B}_3\mathbf{v}_2 = \lambda_{\mathcal{J}}\mathbf{v}_2, \tag{22}$$

which tells us that one of the following must be true:
(i) $\mathbf{v}_2 = 0$
(ii) $\mathbf{v}_2$ can be thought of as an eigenvector of $\mathbf{B}_3$ with eigenvalue $\lambda_{\mathcal{J}}$
We shall now examine both cases.

3.3.6 Case (i)

Scenario (i) simplifies equation (21) considerably, as the second term is dropped and we left with

$$\mathbf{B}_1\mathbf{v}_1 = \lambda_{\mathcal{J}}\mathbf{v}_1, \tag{23}$$

which then again again implies one of the following:
(i) $\mathbf{v}_1 = 0$
(ii) $\mathbf{v}_1$ is an eigenvector
Since $\mathbf{v}_2 = 0$ by assumption, $\mathbf{v}_1 = 0$ would yield a zero vector as the solution to equation (20), which is not a very interesting outcome (as $\mathbf{v} = 0$ would not be considered an

eigenvector). We can therefore consider $\mathbf{v}_1$ to be an eigenvector of $\mathbf{B}_1$.

Definition: For two arbitrary matrices A and B, where A and B are of the form $m \times n$ and $r \times s$, respectively, the Kronecker product $A \otimes B$ is defined as

$$A \otimes B = \begin{pmatrix} a_{11} \cdot B & ... & a_{1n} \cdot B \\ ... & ... & ... \\ a_{m1} \cdot B & ... & a_{mn} \cdot B \end{pmatrix}. \tag{24}$$

This is clearly a $(m \cdot r) \times (n \cdot s)$ matrix.

We shall now note that with the help of the Kronecker product, $\mathbf{B}_1$ can be written as

$$\mathbf{B}_1 = \mathbf{T} \otimes \mathbb{1}, \tag{25}$$

where $\mathbf{T}$ is given by

$$\mathbf{T} = \begin{pmatrix} \left(1 - \sum_{K \neq S_y, E}\right) & \alpha_{KS_y} & \cdots \\ \vdots & \vdots & \vdots \\ \alpha_{S_y U} & \alpha_{KU} & \cdots \\ \vdots & \ddots & \vdots \end{pmatrix}. \tag{26}$$

It can be shown that if $A = B \otimes C$, $A_{diag} = B_{diag} \otimes C_{diag}$. This means that if we diagonalize A, we have diagonalized B and C as well. Now, since in our case C is the identity, $\mathbf{T}$ will have the same eigenvalues as $\mathbf{B}_1$. We shall now analyze case (ii).

3.3.7 Case (ii)

Here, we know that $\mathbf{v}_2$ is an eigenvector of $\mathbf{B}_3$. From Table 2 it follows that

$$\mathbf{B}_3 = \begin{pmatrix} \alpha_{EE}\mathbb{1} + p_S^\star \beta_1 \mathbf{A} & \alpha_{IE}\mathbb{1} + p_S^\star \beta_2 \mathbf{A} \\ \alpha_{EI}\mathbb{1} & \alpha_{II}\mathbb{1} \end{pmatrix}. \tag{27}$$

For pragmatic reasons, we shall split $\mathbf{v}_2$ into to components, $\mathbf{u}_1$ and $\mathbf{u}_2$, such that both are vectors of length n. Since $\mathbf{v}_2$ is an eigenvector with eigenvalue $\lambda_{\mathcal{J}}$, we know that the two components satisfy:

$$\left(\alpha_{EE}\mathbb{1} + p_S^\star \beta_1 \mathbf{A}\right)\mathbf{u}_1 + \left(\alpha_{IE}\mathbb{1} + p_S^\star \beta_2 \mathbf{A}\right)\mathbf{u}_2 = \lambda_{\mathcal{J}} \mathbf{u}_1 \tag{28}$$

$$\alpha_{E1}\mathbf{u}_1 + \alpha_{II}\mathbf{u}_2 = \lambda_{\mathcal{J}} \mathbf{u}_2 \tag{29}$$

Equation (29) tells us that

$$\mathbf{u}_1 = \left(\frac{\lambda_{\mathcal{J}} - \alpha_{II}}{\alpha_{EI}}\right)\mathbf{u}_2, \tag{30}$$

which can be plugged into (28) to get:

$$\left(\left(\alpha_{EE}\mathbb{1} + p_S^\star \beta_1 \mathbf{A}\right)\left(\frac{\lambda_{\mathcal{J}} - \alpha_{II}}{\alpha_{EI}}\right) + \left(\alpha_{IE}\mathbb{1} + p_S^\star \beta_2 \mathbf{A}\right)\right)\mathbf{u}_2 = \lambda_{\mathcal{J}}\left(\frac{\lambda_{\mathcal{J}} - \alpha_{II}}{\alpha_{EI}}\right)\mathbf{u}_2 \tag{31}$$

We can now reorder the equation such that we end up with:

$$\mathbf{A}\mathbf{u}_2 = \left(\frac{\lambda_{\mathcal{J}}^2 - \left(\alpha_{II} + \alpha_{EE}\right)\lambda_{\mathcal{J}} + \alpha_{II}\alpha_{EE} - \alpha_{IE}\alpha_{E1}}{p_S^\star \beta_1 \left(\lambda_{\mathcal{J}} - \alpha_{II}\right) + p_S^\star \beta_2 \alpha_{EI}}\right)\mathbf{u}_2 \tag{32}$$

Since we know that $\mathbf{u}_2 \neq 0$ (otherwise $\mathbf{u}_1 = 0$ and $\mathbf{v} = 0$), $\mathbf{u}_2$ is an eigenvector of the adjacency matrix. As we can see in (32), the eigenvalue is dependent on many parameters.

To simplify matters, we shall call the eigenvalue $\lambda_{\mathbf{A}}$. We can now replace **A** with its eigenvalue in equation (31), so that it can be rewritten as:

$$\lambda_{\mathcal{J}}^2 - \lambda_{\mathcal{J}} \left(\alpha_{II} + \alpha_{EE} + p_S^{\star}\beta_1\lambda_{\mathbf{A}}\right) + \left(\alpha_{II}\alpha_{EE} - \alpha_{IE}\alpha_{E1} + p_S^{\star}\lambda_{\mathbf{A}}\left(\beta_1\alpha_{II} - \beta_2\alpha_{EI}\right)\right) = 0 \quad (33)$$

We've now found a quadratic equation that will give us two (possibly different) eigenvalues for $\mathcal{J}$ as a function of eigenvalues of **A**. The calculations done in this chapter thusly show that the following lemma holds:

Lemma:

The eigenvalues of $\mathcal{J}$ are either acquired via the matrix **T** or by solving the quadratic equation (33), whose roots are dependent in the eigenvalues of **A**.

3.3.8 Stability Requirements

The asymptotic stability theorem requires that all eigenvalues of $\mathcal{J}$ have an absolute value less than 1. We can now use the previous subsections to find out when this is true. The previous lemma states that one of the following two conditions must hold:

(i) The eigenvalues of **T** must all have an absolute value less then 1.

(ii) The solutions of the quadratic equation given by (33) must all have an absolute value less than 1(for all eigenvalues of **A**).

Let's first analyze scenario (1).

3.3.9 Case (i)

Now this is quite simple. Since **T** is a stochastic matrix (This means that summing over the values in each column will always return the value 1), it can only have eigenvalues whose absolute value are less than 1. Our requirements are therefore automatically fulfilled. It should be noted that **T** does not depend on **A**.

3.3.10 Case (ii)

In this case, we must find the conditions for which our eigenvalues have an absolute value less than 1. To do this, we will prove the following lemma:

Lemma:

The solutions of the quadratic equations, which are given by (33) for all eigenvalues of **A**, have an absolute value less than 1 if the following inequality holds:

$$\lambda_1 p_S^{\star}\left(\frac{\beta_1\left(1-\alpha_{II}\right) + \beta_2\alpha_{EI}}{\left(1-\alpha_{II}\right)\left(1-\alpha_{EE}\right) - \alpha_{IE}\alpha_{E1}}\right) < 1, \quad (34)$$

where λ_1 is the largest eigenvalue of **A**.

Proof:

We define r_1 and r_2 to be the solutions of equation (33) for a given $\lambda_{\mathbf{A}}$. We obviously want that

$$|r_1| < 1 \ \& \ |r_1| < 1. \quad (35)$$

While both r_1 and r_2 can in general be complex numbers, let us first analyze the solutions, where both are real numbers. In this case we know that $r_1, r_2 \in (-1, 1)$. For a general quadratic equation $f(x) = ax^2 + bx + c$, where $a > 0$, both solutions can lie in the desired interval if and only if the following conditions hold:

$$\begin{aligned} a - c &> 0 \quad (36)\\ a - b + c &> 0 \quad (37)\\ a - b + c &> 0 \quad (38)\end{aligned}$$

But for our quadratic equation, this means that:

$$\begin{aligned} \alpha_{II}\alpha_{EE} - \alpha_{IE}\alpha_{EI} + p_S^\star \lambda_{\mathbf{A}} \left(\beta_1 \alpha_{II} - \beta_2 \alpha_{EI}\right) &< 1 \quad (39) \\ 1 + \alpha_{EE} + \alpha_{II} + p_S^\star \beta_1 + \alpha_{II}\alpha_{EE} - \alpha_{IE}\alpha_{EI} + p_S^\star \lambda_{\mathbf{A}} \left(\beta_1 \alpha_{II} - \beta_2 \alpha_{EI}\right) &> 0 \quad (40) \\ 1 - \alpha_{EE} - \alpha_{II} - p_S^\star \beta_1 + \alpha_{II}\alpha_{EE} - \alpha_{IE}\alpha_{EI} + p_S^\star \lambda_{\mathbf{A}} \left(\beta_1 \alpha_{II} - \beta_2 \alpha_{EI}\right) &> 0 \quad (41) \end{aligned}$$

We can now rewrite the last two equations to get

$$\begin{aligned} \lambda_{\mathbf{A}} p_S^\star \left(\frac{-\beta_1 \left(1 + \alpha_{II}\right) + \beta_1 \alpha_{EI}}{\left(1 + \alpha_{II}\right)\left(1 + \alpha_{EE}\right) - \alpha_{IE}\alpha_{E1}} \right) &< 1 \quad (42) \\ \lambda_{\mathbf{A}} p_S^\star \left(\frac{\beta_1 \left(1 - \alpha_{II}\right) + \beta_1 \alpha_{EI}}{\left(1 - \alpha_{II}\right)\left(1 - \alpha_{EE}\right) - \alpha_{IE}\alpha_{E1}} \right) &< 1. \quad (43) \end{aligned}$$

This should hold for all eigenvalues of $\mathbf{A}$. Since $\mathbf{A}$ is a symmetric real-valued matrix, it must be diagonalizable with real eigenvalues. Additionally, we can use the Perron-Frobenius theorem, which states that the algebraically largest eigenvalue λ_1 must not only be a positive number, but must also have the largest magnitude (in comparison to all the other eigenvalues). This tells us that if equations (42) and (43) are valid for $\lambda_{\mathbf{A}} = \lambda_1$, they must be valid for all other eigenvalues as well. We also know that the following two inequalities have to hold:

$$\begin{aligned} \left(1 + \alpha_{II}\right)\left(1 + \alpha_{EE}\right) - \alpha_{IE}\alpha_{E1} &> \left(1 - \alpha_{II}\right)\left(1 - \alpha_{EE}\right) - \alpha_{IE}\alpha_{E1} \quad (44) \\ \beta_1 \left(1 - \alpha_{II}\right) + \beta_1 \alpha_{EI} &> -\beta_1 \left(1 + \alpha_{II}\right) + \beta_1 \alpha_{EI} \quad (45) \end{aligned}$$

These follow from the fact that all the parameters must be positive and less than or equal to 1. With these two inequalities, it should be clear that if (43) holds, (42) must hold as well. We can thus conclude, that if the roots of the quadratic equation are real

$$\lambda_1 p_S^\star \left(\frac{\beta_1 \left(1 - \alpha_{II}\right) + \beta_1 \alpha_{EI}}{\left(1 - \alpha_{II}\right)\left(1 - \alpha_{EE}\right) - \alpha_{IE}\alpha_{E1}} \right) < 1. \quad (46)$$

must hold to attain values between $(-1, 1)$. On the other hand, if we allow for complex roots, we must allow the discriminant to be negative. Since our quadratic equation has real coefficients, r_1 and r_2 must be the complex conjugate of one another. This just means that $|r_1| = |r_2| = \sqrt{r_1 r_2}$. It is also easy to show that the product of both roots is equal to $\frac{c}{a}$, where the quadratic equation is given by $ax^2 + bx + c = 0$. This means that since we require all roots to have an absolute value of less than 1, $c/a < 1$. This just means that the following inequality must hold:

$$\left(\alpha_{II}\alpha_{EE} - \alpha_{IE}\alpha_{E1} + p_S^\star \lambda_{\mathbf{A}} \left(\beta_1 \alpha_{II} - \beta_2 \alpha_{EI}\right)\right) < 1 \quad (47)$$

By rearranging the equation though, it turns out that this gives us equation (42). We already know though, that (42) is implied by (43). Thus, if the condition given by (43) is fulfilled, the roots must have an absolute value of less than 1.

3.3.11 Finalizing the Result

We have finally shown that our fixed point is stable if

$$\lambda_1 p_S^\star \left(\frac{\beta_1 \left(1 - \alpha_{II}\right) + \beta_1 \alpha_{EI}}{\left(1 - \alpha_{II}\right)\left(1 - \alpha_{EE}\right) - \alpha_{IE}\alpha_{E1}} \right) < 1. \quad (48)$$

By setting

$$C_{VPM} = p_S^\star \left(\frac{\beta_1 \left(1 - \alpha_{II}\right) + \beta_1 \alpha_{EI}}{\left(1 - \alpha_{II}\right)\left(1 - \alpha_{EE}\right) - \alpha_{IE}\alpha_{E1}} \right), \quad (49)$$

we have finally proven the G2 stability theorem.

4 Experimental Verification of the Model

Our aim is to reproduce the model as well as to push it towards its boundaries and find cases, where the epidemic threshold is not described by the effective strength. To do this, we conduct SIR simulations by implementing the Gillespie algorithm, which is described in the following section. Initially, we focus on a setting similar to the one used by [2]. In a further step, we vary the simulation parameters to find possible weaknesses of the model. In particular, we focus on the influence of the initially infected nodes.

4.1 Simulating the Time Evolution of the SIR model on Arbitrary Networks with the Gillespie Algorithm

During the course of this chapter, we will analyze the time evolution of the propagating virus. To do this, we consider the fraction of infected people and the threshold condition for an outbreak of an epidemic in the SIR model shown in Figure 5.
The SIR model states that individuals can either be susceptible for the disease (S), infected (I) or recovered (R). The only two possible events are the following:

(i) An infected node infects a susceptible neighbour

(ii) An infected node gets recovered

Hereinafter we will call the set of susceptible nodes that have at least one infected neighbor "at_risk_nodes". The model is entirely described by the transmission probability over a contact-link β and the healing probability once infected δ. The parameter β is defined such that the probability that (a) happens in the infinitesimal time interval $[t, t + dt)$ is given by βdt. Analogously the probability that an infected node at time t gets healed (event (b)) within the time interval $[t, t+dt)$ is δdt. Therefore the probability that exactly one event happens in the time interval $[t, t + dt)$ is given by

$$\begin{aligned}
\text{total\_rate} \times dt &= (\text{total\_infection\_rate} + \text{total\_recovery\_rate}(t)) \times dt \\
\text{total\_infection\_rate} &= \sum_{u \in \text{at\_risk\_nodes}} \beta \times \text{number\_of\_infected\_neighbours}[u] \\
\text{total\_recovery\_rate}(t) &= \delta \times \text{number\_of\_infected\_nodes}
\end{aligned}$$

To solve this problem numerically we have to discretize the time and therefore replace dt with a finite time step Δt. The smaller Δt is chosen, the more accurate the algorithm describes the model.
We first want to describe a naive algorithm that could simulate the disease outbreak with the SIR model. Assuming we know the state of all nodes at time t, we can simulate the states of the next time step $t + \Delta t$ with the following algorithm:

1. generate a random number r_1 uniformly distributed in the interval $(0, 1)$
2. (a) If $r_1 < \text{total\_rate} \times \Delta t$: generate another number r_2 uniformly distributed in the interval $(0, \text{total\_rate})$
    - If total_recovery_rate $< r_2$: choose a random infected node u, set its state to "recovered", and reduce the number_of_infected_neighbors[ν] for every neighbor ν of u by 1
    - Else: Choose u from at_risk_nodes with probability $\frac{\beta \times \text{number\_of\_infected\_neighbours}[u]}{\text{total\_infection\_rate}}$, remove it from the at_risk_nodes, set its state to infected and add its neighbors to at_risk_nodes
    - Update all rates and start over again with 1. for the time $t + \Delta t$

(b) Else: No event occurs in this time interval and we start again with 1. for the time $t + \Delta t$

Since r_1 is a random number uniformly distributed in $(0, 1)$, the probability that $r_2 <$ total_recovery_rate × Δt is equal to total_recovery_rate × Δt. So for infinitely small Δt, 2 (a) would exactly simulate the probability that an event will happen in the time step $t + \Delta t$. If an event happens in the considered time interval, the next question is which event that is. The probability that a node recovers during this time-step is given by the probability that total_recovery_rate $< r_2$. In the remaining cases this event will be a transmission of the disease. The probability that a at_risk_node gets infected is simply proportional to its number of infected neighbors.

The problem with this algorithm is that by decreasing Δt (and therefor increasing the accuracy), the computational intensity increases. During most of the time steps, the generated random number r_1 is created, even though no event took place.

The Gillespie follows a different approach: Instead of analyzing each time step, it calculates the time $t + \tau$ at which the next event occurs. It is a model that was developed in the middle of the 20^{th} century and was mainly used to efficiently simulate chemical or biochemical reactions [10]. Instead of explaining the general algorithm, we want to focus on how it can be used to simulate the SIR model. Our paper will closely follow the work done in [8], where it's shown how the Gillespie algorithm can be used to describe chemical reactions.

We shall define $f(\text{total\_rate}(t), s)ds$ as the probability that for a given total_rate at time t, the next event will happen during the time interval $[t+s, d+s+ds)$ and $g(\text{total\_rate}(t), s)ds$ as the probability, that no event happens in the time interval $[t, t + s)$.

The probability that for a given time t the next event will happen during the time interval $[t+s, d+s+ds)$ can now be written as the product of the probabilities, that no event happens in the time interval $[t, t+s)$ and the probability that exactly one event happens in the time interval $[t+s, t+s+ds)$: $f(\text{total\_rate}(t), s)ds = g(\text{total\_rate}(t), s) \times \text{total\_rate}(t+s)ds$. Since no event happens in $[t, t + s)$ we know that $\text{total\_rate}(t + s) = \text{total\_rate}(t)$. Therefore:

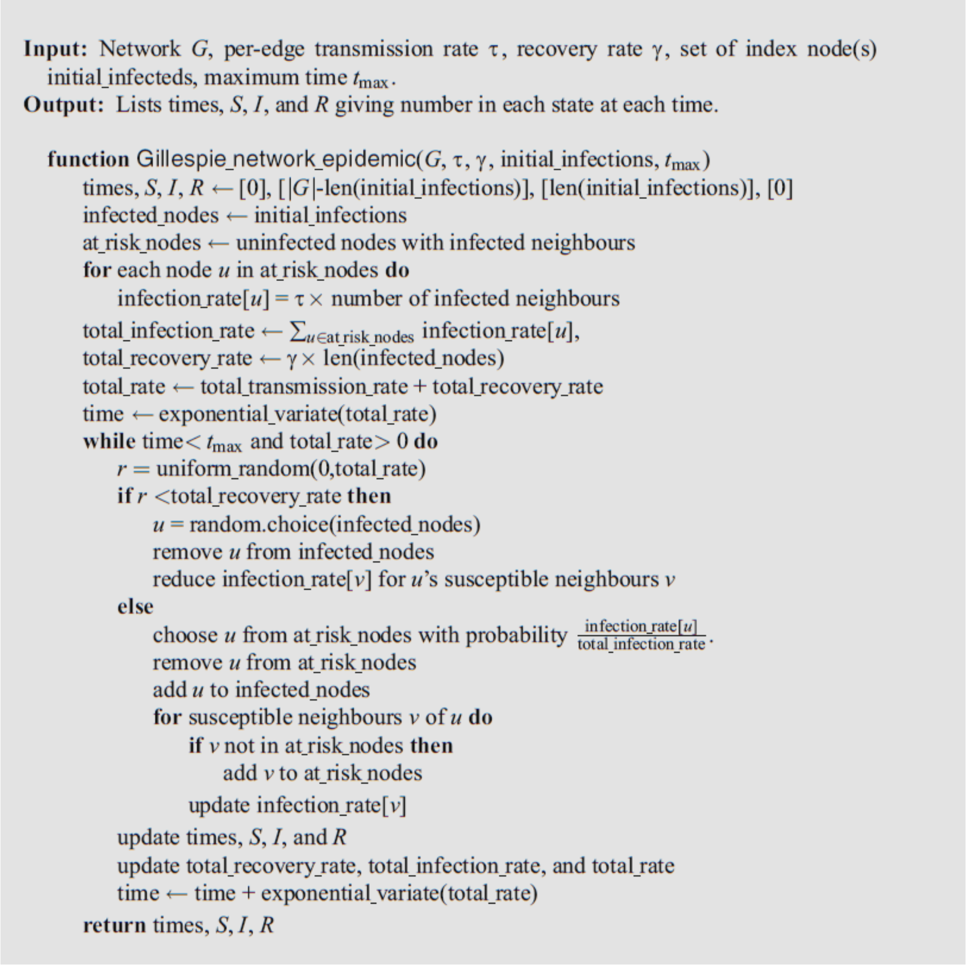
$$f(\text{total\_rate}(t), s)ds = g(\text{total\_rate}(t), s) \times \text{total\_rate}(t)ds \qquad (50)$$

Now we want to calculate $g(\text{total\_rate}(t), \sigma)$ for all $\sigma > 0$. For an infinitesimal time step $d\sigma$ the probability that no event occurs in the time interval $[t, t + \sigma + d\sigma)$ is given by: $g(\text{total\_rate}(t), \sigma + d\sigma) = g(\text{total\_rate}(t), \sigma)[1 - \text{total\_rate}(t + \sigma)]d\sigma$. Here we have used the fact that $d\sigma$ is infinitesimally small and therefore the probability that more than one event happens in the time interval $[t + \sigma, t + \sigma + d\sigma)$ is negligible. In this expression we can replace $\text{total\_rate}(t + \sigma)$ with $\text{total\_rate}(t)$, since no event took place in the the time interval $[t, t + \sigma)$. This leads to the following differential equation for $g(\text{total\_rate}(t), \sigma)$:

$$\begin{aligned}\frac{dg(\text{total\_rate}(t), \sigma)}{d\sigma} &= \lim_{d\sigma \to 0} \frac{g(\text{total\_rate}(t), \sigma + d\sigma) - g(\text{total\_rate}(t), \sigma)}{d\sigma} \\ &= -\text{total\_rate}(t) \times g(\text{total\_rate}(t), \sigma)\end{aligned}$$

With the initial condition that $g(\text{total\_rate}(t), 0) = 1$, we obtain the unique solution $g(\text{total\_rate}(t), \sigma) = \exp[-\text{total\_rate}(t) \times \sigma]$. From this we know that $f(\text{total\_rate}(t), s)ds = \exp[-\text{total\_rate}(t)s] \times \text{total\_rate}(t)ds$. Therefore the time τ, such that the next event happens at the time $t + \tau$, is exponentially distributed according to the probability distribution $f(\lambda, x) = \lambda \exp[-\lambda x]$ where $\lambda = \text{total\_rate}(t)$.

A Pseudocode for the Gillespie algorithm, which can be used to with the SIR model, is shown in figure 1.



```
Input: Network $G$, per-edge transmission rate $\tau$, recovery rate $\gamma$, set of index node(s)
   initial_infecteds, maximum time $t_{\max}$.
Output: Lists times, $S$, $I$, and $R$ giving number in each state at each time.

  function Gillespie_network_epidemic($G$, $\tau$, $\gamma$, initial_infections, $t_{\max}$)
      times, $S$, $I$, $R$ ← [0], [$|G|$-len(initial_infections)], [len(initial_infections)], [0]
      infected_nodes ← initial_infections
      at_risk_nodes ← uninfected nodes with infected neighbours
      for each node $u$ in at_risk_nodes do
          infection_rate[$u$] = $\tau \times$ number of infected neighbours
      total_infection_rate ← $\sum_{u \in \text{at\_risk\_nodes}}$ infection_rate[$u$],
      total_recovery_rate ← $\gamma \times$ len(infected_nodes)
      total_rate ← total_transmission_rate + total_recovery_rate
      time ← exponential_variate(total_rate)
      while time< $t_{\max}$ and total_rate> 0 do
          $r$ = uniform_random(0,total_rate)
          if $r$ <total_recovery_rate then
              $u$ = random.choice(infected_nodes)
              remove $u$ from infected_nodes
              reduce infection_rate[$v$] for $u$'s susceptible neighbours $v$
          else
              choose $u$ from at_risk_nodes with probability $\frac{\text{infection\_rate}[u]}{\text{total\_infection\_rate}}$.
              remove $u$ from at_risk_nodes
              add $u$ to infected_nodes
              for susceptible neighbours $v$ of $u$ do
                  if $v$ not in at_risk_nodes then
                      add $v$ to at_risk_nodes
                  update infection_rate[$v$]
          update times, $S$, $I$, and $R$
          update total_recovery_rate, total_infection_rate, and total_rate
          time ← time + exponential_variate(total_rate)
      return times, $S$, $I$, $R$
```

Figure 1: Pseudocode for the Gillespie algorithm simulating an SIR epidemic in a network [4]

Assuming all states of the system are known for a time t_0, the algorithm perform the following steps:

1. Compute all rates
2. Compute the time $t + \tau$ at which the next event will happen. Therefore the random number τ exponentially distributed according to $f(s) = \exp[-\text{total\_rate}(t)s] \times \text{total\_rate}(t)$ is generated
3. Compute which event will happen
   (i) If total_recovery_rate $< r_2$: choose a random infected node u, set its state to "recovered" and reduce the number_of_infected_neighbors[ν] for every neighbor ν of u by 1
   (ii) Else: Choose u from at_risk_nodes with probability $\frac{\beta \times \text{number\_of\_infected\_neighbours}[u]}{\text{total\_infection\_rate}}$, remove it from the at_risk_nodes, set its state to infected and add its neighbors to at_risk_nodes
4. Update all rates and start over again with 1. for the time $t + \tau$

4.2 Centrality Measures for Initial Infections

In order to initially infect the most important nodes we use the mathematical concept of centrality. There exist several different centrality measures, however, for our purposes the concept of eigenvector centrality turned out to be suitable. The vector of the eigenvector centralities x of all nodes in a network is defined via

$$\mathbf{A}x = \lambda x \tag{51}$$

where $\mathbf{A}$ denotes the networks' adjacency matrix and λ denotes the largest eigenvalue of $\mathbf{A}$, which is real and positive by the Perron-Frobenius theorem. Hence, x is the eigenvector belonging to the spectral radius of $\mathbf{A}$. [9]

4.3 Implementation

To simulate the infections, we used an implementation of the Gillespie algorithm from [7], as well as Python functions, which enable us to plot the time behaviour of the infection. In addition we computed the footprint of the infection as a function of the effective strength. Our Code uses the libraries scipy, matplotlib, networkX, and EoN.

Data sets

For the experiments described in the following section, we use the following data sets.

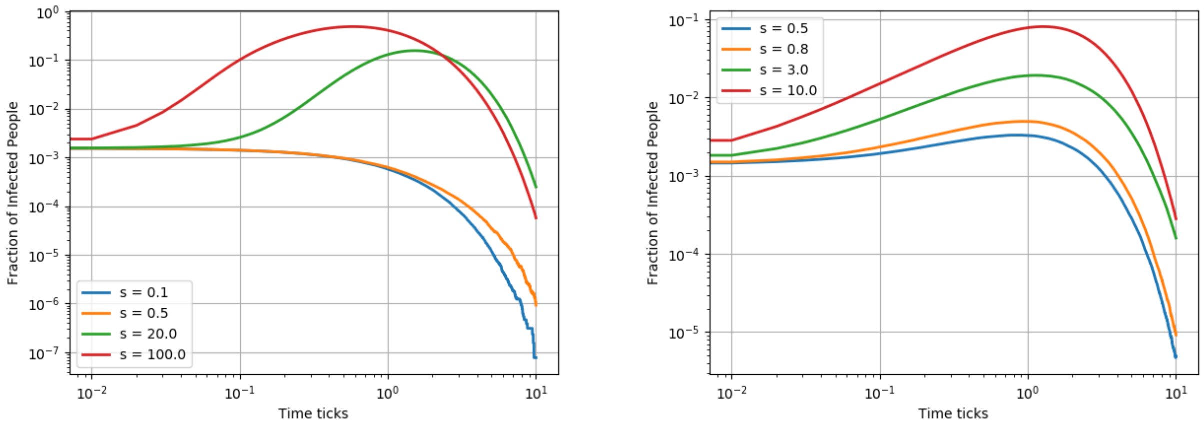
(i) AS-OREGON: Internet connection data from January 2, 2000 collected by University of Oregon Route Views Project [5]. 6474 nodes, 13895 edges, spectral radius 46.9.

(ii) FACEBOOK-EGO: Data from a survey on Facebook app usage [6]. 4039 nodes, 88234 edges, spectral radius 162.4.

(iii) TERRORIST: Social associations between the terrorists involved in the 9/11 attacks [1]. 62 nodes, 304 edges, spectral radius 8.7.

4.4 Simulation Results and Discussion

For our experiments, we conducted computer simulations using Python 3.7. To create meaningful plots, we ran at least 200 independent simulations per figure and plotted their arithmetic mean.



(a)

(b)

Figure 2: Time behaviour of the SIR simulations for random **(a)** and critical **(b)** initial infection on the AS-OREGON network.

Reproduction of previous work

To reproduce the work in [2], we used the AS-OREGON data set, which is similar to one of the data sets in their experiments. Initially we infected 10 random nodes and investigated the time behaviour of the SIR simulation, which is shown in figure 2a. The result matches the expectations, as s-Values greater 1 lead to a significant increase of infected nodes, while under-critical infections ($s < 1$) die out quickly. This behavior is confirmed by the simulation results shown in figure 3, which support the assumption that the tipping point is at $s = 1$.

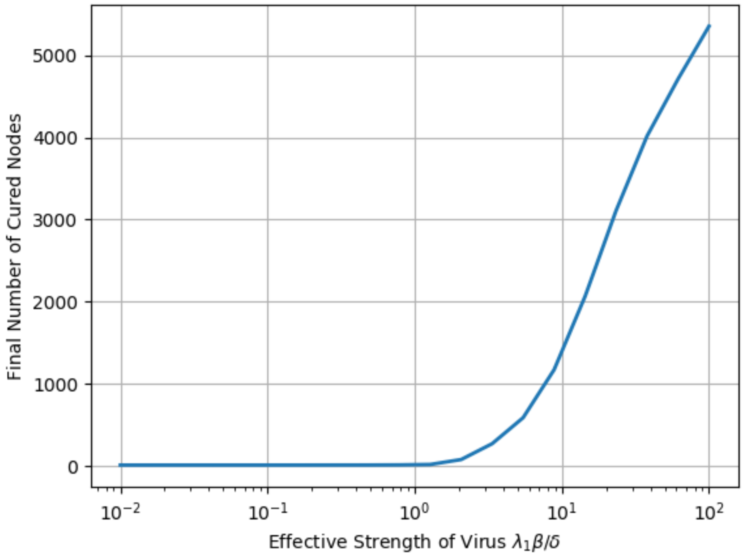


Figure 3: Infection footprint for different SIR virus strengths on the AS-OREGON network.

Initial infection size

As we were able to reproduce the results from [2], we investigated the influence of the size of the initially infected population. We therefore ran the same simulation from before, but randomly altered the initial number of infected nodes. The result is shown in figure 4a, which suggests, that the model is still valid, even for notable sizes of initially infected nodes. However, the tipping point slightly moved to the left with increasing numbers of initial infections.

Infection of important nodes

In the next step, we initially infected the nodes with the highest eigenvector centrality (see section 4.2). As expected, this is beneficial to the spreading of the virus. Figure 5 compares the footprint of random infection with critical node infection. We can clearly see that the initial infection of the nodes with the highest eigenvalue centrality moves the tipping point slightly to the left of 1. Hence, infection of critical nodes brings the model already close to its limit. This fits the time behavior of the system, which is shown in figure 2b, where the below-critical infections spread slightly before dying out.

Additionally, we conducted experiments with larger numbers of critical initial infections. The results are shown in figure 4b and again indicate that by initially infecting

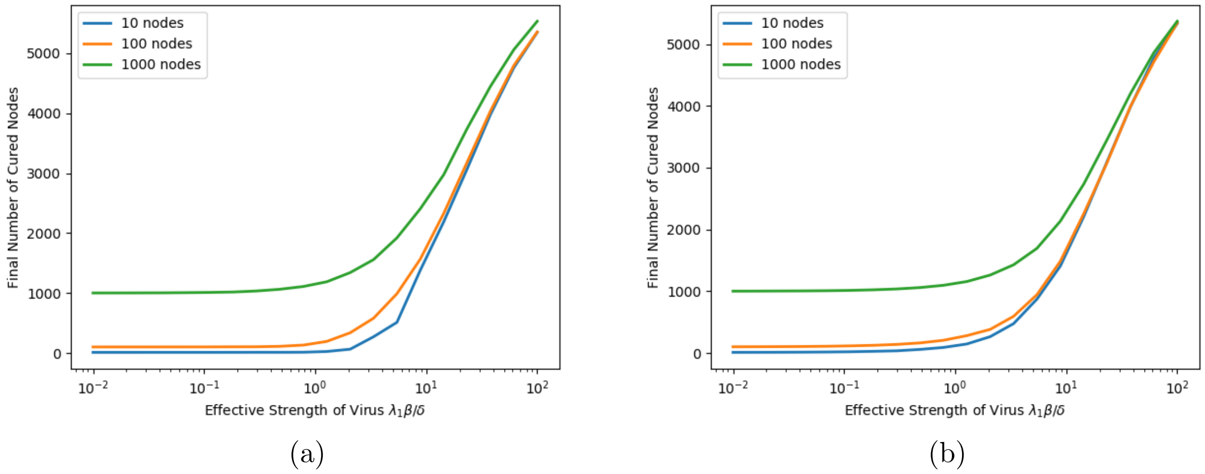


Figure 4: SIR infection footprint for different sizes of the random **(a)** and critical **(b)** initial infection on the AS-OREGON network.

nodes with the highest centrality, we create a tipping point very close to 1, particularly for larger numbers of initially infected nodes. However, the model can still be considered valid, as the footprint for effective strengths below 1 remains very close to the initial infection. This meets, due to the generality of the proof (see section 3.3), our expectations.

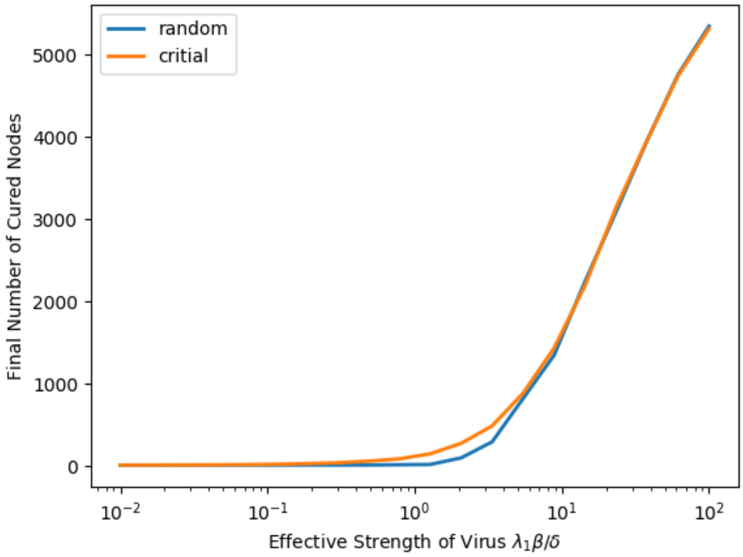


Figure 5: Infection footprint for different SIR virus strengths for random (blue) and critical (orange) initial infection on the AS-OREGON network.

Verification with highly connected network

To verify the results of the previous section, we ran the same simulation on a different network. We used the FACEBOOK-EGO network, which has a slightly smaller number of nodes, but nevertheless a larger spectral radius and therefore a higher density of connections. For this data set we were able to reproduce all the outcomes from the AS-OREGON network. As expected, the model is also valid for all our test cases on this network.

Behavior of small networks

As the behavior for very large graphs was already investigated in [2], we were interested if the model is also applicable for small networks. We therefore ran simulations on the TERRORIST network with 62 nodes and a high density of connections. We again infected the nodes with the highest eigenvector centrality. Figure 6 shows the results for different sizes of the initial infection. Now the tipping point is, especially for larger numbers of initially infected nodes, below one. This is interesting, as for the larger networks even higher ratios of initially infected nodes led to no disease spreading. Hence, we draw the conclusion that the model is not generally valid for small networks.

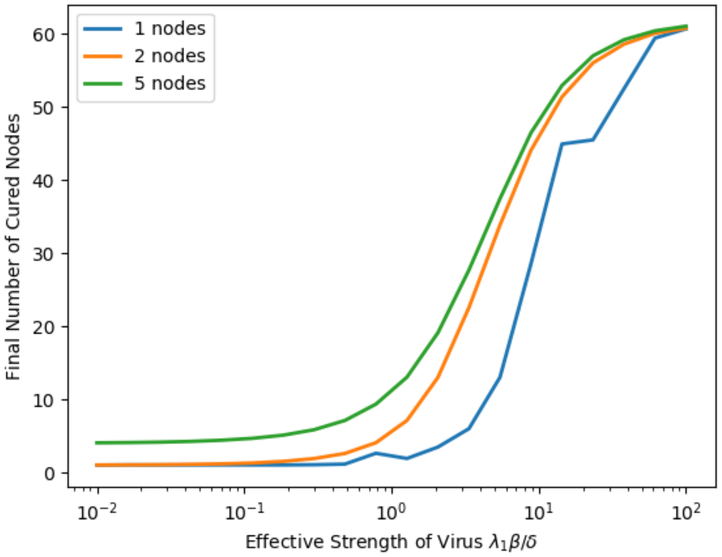


Figure 6: Infection footprint for different numbers of initially infected nodes on the TERRORIST network.

5 Summary and Outlook

In section 3, we saw that the G2 theorem is a powerful tool, as it creates a simple condition for analyzing VPMs on arbitrary networks and is almost always applicable (This is because the assumptions were reasonable and therefore do not really count as constraints). We clearly saw the vast scope of its utility in our experimental results. Only for small networks did we observed scenarios where the tipping point was off.

However, our experiments were limited to the SIR virus propagation model. It could very well be that more sophisticated VPMs may lead to different results, in which case the G2 theorem would cease to hold.

Another thing that has to be kept in mind is the fact the results are averages over many runs. We saw that particular simulations were strongly deviated from the mean.

By taking the average over many peaks that do not align, the resulting curve will have a lower, wider peak which does not resemble a single realization [8].

Nevertheless, the G2-threshold theorem has shown to be very robust and reliable for the SIR model and there is no reason to suspect that this would be any different for other prevalent virus propagation models.

References

[1] Aaron Clauset. Terrorist associations for 9/11 attacks. `http://tuvalu.santafe.edu/~aaronc/datacode.htm`. [Online; accessed 8-December-2018].

[2] B. Aditya Prakash, Deepayan Chakrabarti, Nicholas Valler, Michalis Faloutsos, Christos Faloutsos. Threshold conditions for arbitrary cascade models on arbitrary networks. Knowledge and Information Systems manuscript No. KAIS-12-3483R1, 2012.

[3] Hirsch MW. *Differential Equations, Dynamical Systems and Linear Algebra.* Morris W. Hirsch Stephen Smale Robert L. Devaney Stephen Smale, 1974.

[4] István Z. Kiss, Joel C. Miller, Péter L. Simon. *Mathematics of Epidemics on Networks*, volume 46, chapter Appendix A, page 382. Springer, Cham, 2017.

[5] Jure Leskovec. Autonomous systems AS-733. `https://snap.stanford.edu/data/as-733.html`. [Online; accessed 8-December-2018].

[6] Jure Leskovec. Social cirlces: Facebook. `https://snap.stanford.edu/data/ego-Facebook.html`. [Online; accessed 8-December-2018].

[7] Joel C. Miller. Git repository mathematics-of-epidemics-on-networks. `https://github.com/springer-math/Mathematics-of-Epidemics-on-Networks/blob/4005900b7307a455f5912fb4ec3281a47be0530d/EoN/simulation.py`. [Online; accessed December-2018].

[8] Radek Erban, S. Jonathan Chapman, Philip K. MAINI. A practical guide to stochastic simulations of reaction-diffision processes. https://arxiv.org/pdf/0704.1908.pdf. [Online; accessed 7-December-2018].

[9] Wikipedia contributors. Centrality — Wikipedia, the free encyclopedia. `https://en.wikipedia.org/w/index.php?title=Centrality&oldid=861906659`, 2018. [Online; accessed 7-December-2018].

[10] Wikipedia contributors. Gillespie algorithm — Wikipedia, the free encyclopedia. `https://en.wikipedia.org/w/index.php?title=Gillespie_algorithm&oldid=867948442`, 2018. [Online; accessed 7-December-2018].