

# SIR-Modell einer Epidemie

im Rahmen der 2. Modularbeit zur Veranstaltung Modellbildung und Simulation im Sommersemester 2020 an der Hochschule München

**Benedikt Beil<sup>1</sup>, Benjamin Eder<sup>2</sup>, Konstantin Schlosser<sup>3</sup>  
und Andreas Stiglmeier<sup>4</sup>**

<sup>1</sup>Hochschule München, Master Informatik, SWE [bbeil@hm.edu](mailto:bbeil@hm.edu)

<sup>2</sup>Hochschule München, Master Informatik, SWE [beder@hm.edu](mailto:beder@hm.edu)

<sup>3</sup>Hochschule München, Master Informatik, SWE [k.schlosser@hm.edu](mailto:k.schlosser@hm.edu)

<sup>4</sup>Hochschule München, Master Informatik, SWE [stiglmeier.andreas@hm.edu](mailto:stiglmeier.andreas@hm.edu)

---

## Zusammenfassung

Im Zuge der aktuellen Situation rund um das neuartige Coronavirus COVID-19 implementieren wir unter Zuhilfenahme des SIR-Modells ein grafisches Programm zur Simulation der Ausbreitung einer Infektion. Auf Basis der entstandenen Anwendung untersuchen wir das Basismodell sowie verschiedene Erweiterungen und deren Auswirkungen. Außerdem führen wir eine Grenzwertbetrachtung des SIR-Modells durch. Die resultierenden Graphen und Schaubilder geben Einblick in mögliche Ausbreitungsszenarien von Krankheiten. In einer Stellungnahme zur Dauer und Wirksamkeit von Maßnahmen gegen die Ausbreitung des neuartigen Coronavirus bestätigen wir die Wirksamkeit und geben eine Schätzung zur Dauer der Epidemie ab. Abschließend geben wir eine Zusammenfassung und Bewertung ab und ordnen das Projekt in den Modellierungszyklus ein.

---

---

## Abstract

In the course of the current situation regarding the novel Coronavirus COVID-19, we are implementing a visual application to simulate the spread of an infection using the SIR model. Using the resulting application, we investigate the base model as well as various extensions and their effects. Furthermore, we perform a boundary value analysis of the SIR model. The resulting graphs and diagrams provide insight into possible scenarios of the spread of diseases. In a statement on the duration and effectiveness of measures against the spread of the novel Coronavirus, we confirm the effectiveness and provide an estimate of the duration of the epidemic. Finally, we provide a summary and assessment followed by a positioning of the project in the modelling cycle.

---

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>3</b>
1.1	SIR-Modell (Eder)	3
1.2	Projektbeschreibung (Eder)	3
1.3	Aufbau (Eder)	3
<b>2</b>	<b>Methoden</b>	<b>4</b>
2.1	Verwendete Werkzeuge (Schlosser)	4
2.2	Simulationslogik (Schlosser)	4
2.3	Geplante Erweiterungen (Eder)	7
	Todesfälle und Immune	7
	Beschränkter Bewegungsradius	7
	Ausgangssperre	7
	Inkubationszeit und Krankheitsdauer	7
	Quarantäne	7
	Impfstoff	7
2.4	Grafische Benutzeroberfläche (Eder)	7
	Basiseinstellungen und Steuerung	8
	Visualisierungen	9
	Steuerung der Modellerweiterungen	9
	Berechnung der Nettoerproduktionsrate	12
<b>3</b>	<b>Ergebnisse</b>	<b>16</b>
3.1	Implementierung des Basismodells (Stiglmeier)	16
3.2	Betrachtung des Basismodells (Beil)	17
3.3	Grenzwertbetrachtungen (Beil)	22
3.4	Validierung des Basismodells (Beil, Eder)	28
	Validierung anhand Grenzwertbetrachtung	28
	Validierung anhand Boccarda und Cheong (1992)	29
3.5	Erweiterungen	31
	Aufteilung in Todesfälle und Immune (Schlosser)	31
	Beschränkter Bewegungsradius (Eder)	33
	Ausgangssperre (Stiglmeier)	38
	Inkubationszeit und Krankheitsdauer (Beil)	40
	Quarantäne (Stiglmeier)	42
	Impfstoff (Schlosser)	44
3.6	Stellungnahme: Dauer und Wirksamkeit von Maßnahmen gegen COVID-19 (Eder)	46
<b>4</b>	<b>Zusammenfassung und Bewertung (Eder)</b>	<b>49</b>
4.1	Anmerkungen zum Programm	49
4.2	Einordnung in den Modellierungszyklus	49
4.3	Abschließende Worte	50

# 1 Einleitung

Vor dem Hintergrund der aktuellen Pandemie aufgrund des neuen Virus aus der Familie der Coronaviren „COVID-19“ (*Coronavirus disease 2019* [2020](#)) entsteht diese Arbeit als Modularbeit für die Veranstaltung „Modellbildung und Simulation“ im Sommersemester 2020 an der Hochschule München für angewandte Wissenschaften.

## 1.1 SIR-Modell (Eder)

Die Ausbreitung von Epidemien wird in zahlreichen Modellen behandelt. Diese beschreiben den Verlauf meist mit Differenzialgleichungen oder durch stochastische Prozesse. In dieser Arbeit jedoch wird auf Grundlage eines zweidimensionalen Zellularautomaten ein SIR-Modell abgebildet. *SIR* ist hierbei ein Akronym für **S**usceptible, **I**nfective und **R**emoved (Boccara und Cheong [1992](#)):

- *Susceptible*: Personen, welche infiziert werden können.
- *Infective*: Personen, welche eine Krankheit übertragen und damit verbreiten können.
- *Removed*: Personen, welche entweder an der Krankheit gestorben oder von dieser genesen sind und daher immun.

Zur Simulation werden Zellen des Automaten mit verschiedenen Zuständen initialisiert. Entweder sind die Zellen leer oder durch eine gesunde, bzw. infizierte Person besetzt. Im Zeitverlauf können die Zellen ihren Zustand nach den folgenden Regeln ändern:

- Zufällig ausgewählte Individuen können sich auf dem Gesamtgebiet in freie Zellen bewegen. Allerdings bewegen sich nicht alle Personen.
- Nachdem sich die Personen bewegt haben, können infizierte Individuen andere gesunde Zellen in einer definierten Nachbarschaft anstecken.
- Anschließend genesen oder sterben die infizierten Zellen mit einer gewissen Wahrscheinlichkeit.
- Diese Regeln werden in einem festen Zeitintervall iterativ angewandt.

Neben diesen Regeln wird davon ausgegangen, dass die ursprüngliche Population geschlossen ist, also weder neue Personen hinzukommen noch durch andere Gründe als die Krankheit sterben.

## 1.2 Projektbeschreibung (Eder)

Im Zuge dieser Arbeit implementieren wir eine Simulation des soeben beschriebenen Modells und testen dieses im Basiszustand, sowie mit bestimmten Erweiterungen und Anpassungen. Ebenso führen wir eine Grenzwertbetrachtung des SIR-Modells durch. Auf Basis der Erkenntnisse unternehmen wir einen Versuch, die Ausbreitung des aktuellen Coronavirus zu simulieren. Auch nehmen wir zu aktuellen Maßnahmen gegen das besagte Virus Stellung.

## 1.3 Aufbau (Eder)

Der nachfolgende Teil befasst sich einleitend mit den verwendeten Methoden und Werkzeugen, sowie einer Beschreibung unserer Implementierung und Verständnis des Modells.

Danach wollen wir unsere Ergebnisse für das Basismodell, wie auch für bestimmte Erweiterungen präsentieren und analysieren. Ein weiterer Teil berichtet von unserer Schätzung für Parameter, welche zu COVID-19 passen könnten, gefolgt von einer kurzen Stellungnahme zur Dauer und Wirksamkeit der aktuellen Maßnahmen gegen die weitere Ausbreitung der Krankheit.

Am Ende geben wir eine Zusammenfassung und Bewertung des Projekts und unserer Erkenntnisse.

## 2 Methoden

### 2.1 Verwendete Werkzeuge (Schlosser)

Um die in 1.1 vorgestellte Simulation durchführen und konfigurieren zu können, erstellen wir ein Programm, das diese Funktionalität beinhaltet.

Als Programmiersprache wurde Python aufgrund der großen Verfügbarkeit von Bibliotheken und des hohen Bekanntheitsgrades gewählt. Es wurde sich teamintern auf die Python-Version 3.8 geeinigt, da diese die momentan aktuellste ist und somit am längsten unterstützt werden wird.

Die Oberfläche wurde mithilfe der Bibliothek *Qt* erstellt, welche es erlaubt, interaktive Oberflächen in vielen Programmiersprachen zu erstellen. Die Anbindung an Python übernahm die Bibliothek *PyQt5*, welche ein Framework für Graphen und Visualisierungen, *PyQtGraph* anbietet. Insbesondere diese Werkzeuge werden im Abschnitt 2.4 genauer vorgestellt.

Um diese Graphen zu erstellen, wird in *PyQtGraph* *numpy* verwendet. Da diese Bibliothek auch weitere, nützliche Features für die Entwicklung bot, wie zum Beispiel *choices* zur Vereinfachung der Auswahl der Zufallswerte (*numpy.random.choice* 2020) wurde sie für diese Zwecke genutzt.

Um das Programm besonders flexibel zu gestalten, wurde sich entschieden, die Logik mithilfe von Events umzusetzen. Diese erlauben eine entkoppelte Architektur, was in einer besonders einfachen Erweiterbarkeit resultiert. Diese Eigenschaft ist für ein Programm, welches innerhalb von drei Wochen entwickelt wird, besonders interessant.

Um diese Logik möglichst einfach implementieren zu können, fand eine weitere Bibliothek Verwendung, *pyObservable* (*timofurrer/observable minimalist event system for Python* 2018). Für diese wurde sich entschieden, da sie eine besonders einfache Handhabung bietet, aber dennoch flexibel genug ist, um auch mögliche Erweiterungen des Programms in der Zukunft zu unterstützen.

### 2.2 Simulationslogik (Schlosser)

Um das Basismodell, welches schon in der Einleitung in Kapitel 1.1 beschrieben wurde, umzusetzen, wurde sich für ein sog. *event-driven* System entschieden.

Das heißt, dass die Schritte in der Simulation nicht statisch vorgegeben sind, sondern durch Events gesteuert werden. Diese erlauben einem Programmteil, wie einer Klasse, bestimmte Methoden für bestimmte Events zu registrieren. Diese Methoden werden dann zusammen mit Argumenten aufgerufen, welche beim Auslösen des Events bestimmt werden.

Um die Logik der Simulation zu verstehen, müssen drei Klassen genauer erläutert werden: *Scheduler*, *Agent* und *Grid*.

#### **Agent**

Die Klasse *Agent* repräsentiert ein Individuum auf dem Simulationsfeld. Er hat Kenntnis seiner Eigenschaften wie Status, Infektions- sowie Inkubationszeit. Diese Klasse hat in der Logik keine aktive Aufgabe inne, stellt aber einen wichtigen Baustein dar.

#### **Grid**

Den zweidimensionalen Zellularautomaten, das "Feld" der Simulation, wird von der Klasse *Grid* verwaltet. Diese hat Kenntnis über alle aktiven Agenten, deren Positionen und ist für die Ereignisbehandlung der Agenten verantwortlich. Daher ist sie auch für die Bewegungen und Statusupdates zuständig.

#### **Scheduler**

Um einen Simulationsschritt vorzunehmen, sind mehrere Events nacheinander nötig. Das Auslösen dieser Events geschieht im sog. *Scheduler*. Dieser kontrolliert die registrierten Eventhandler, Events und den Ablauf des Programms. Einen beispielhaften Ablauf eines Simulationsschritts ist

in Algorithmus 1 zu finden. Beim Konzipieren des Schedulers wurde darauf geachtet, dass in Zukunft für rechenintensivere Anwendungen auch mehrere Threads verwendet werden können. So kann beispielsweise problemlos das Zeichnen der Benutzeroberfläche von der Abarbeitung der Programmlogik getrennt werden.

```
1 def __next_step(self, state: SimState):
2     self.__logger.info("Next step triggered, starting agent movement")
3     self.trigger_event(Events.AGENT_MOVEMENT, {"state": state})
4     self.__logger.info("Agent movement finished, starting status update")
5     self.trigger_event(Events.STATUS_UPDATE, {"state": state})
6     self.__logger.info("Status update finished, starting additional events")
7     self.__logger.info("Additional events finished. Starting repaint")
8     self.trigger_gui_event(Events.REPAINT)
```

**Listing 1.** Logik des Schedulers für einen Simulationsschritt

Um auf das Beispiel aus Algorithmus 1 zurückzukommen, soll ein Simulationsschritt des Basismodells beschrieben werden. Als erstes wird von der Nutzoberfläche das Event „NEXT\_STEP“ ausgelöst. Dieses wird im Scheduler mithilfe der aufgeführten Methode behandelt und erlaubt so die sequentielle Abarbeitung darauf folgender Events.

Als nächstes wird das Event „AGENT\_MOVEMENT“ ausgelöst. Dieses wird über das Grid weitergereicht und in der momentan aktiven Bewegungsstrategie behandelt. So wird für jeden Agenten ein Simulationsschritt der Bewegungen ausgeführt.

Im Basismodell gibt es eine sogenannte Durchmischungszahl,  $m$ , welche bestimmt, welcher Anteil der Agenten sich in einem Schritt bewegt. Die Agenten, welche sich tatsächlich bewegen, werden in jedem Schritt erneut zufällig bestimmt. Grundsätzlich kann ein Agent, der im Bewegungsanteil ist, auf ein beliebiges freies Feld springen.

Daraufhin werden mittels des Events „STATUS\_UPDATE“ die Zustände der Agenten verändert. Hier liegt die besondere Schwierigkeit darin, dass mehrere Modifikatoren gleichzeitig aktiv sein können. Beispielsweise muss für die Impf-Erweiterung ebenfalls eine Unterteilung in Tote und Immune vorgenommen werden. Durch diese Vorhergehensweise ist es möglich, eine Kombination der verschiedenen Erweiterungen zu untersuchen.

Die Modifikation der Agenten-Zustände erfolgt gesammelt nach dem Status der Agenten. Die verschiedenen Strategien haben jeweils einen zugeordneten Zustand, den sie behandeln. Diese Modifikationen werden nacheinander, sortiert nach Wichtigkeit, aufgerufen, sofern sie aktiv sind. Der Zustand (aktiv oder inaktiv) wird dynamisch bestimmt und nur aktive Strategien tatsächlich berechnet, die für den jeweiligen Agenten ausschlaggebend sind. Diese Logik ist in Algorithmus 2 aufgeführt.

Im Basismodell werden zunächst erneute Ansteckungen der Agenten bestimmt, welche sich im Zustand *Infective* befinden. Diese können, über die Infektionschance geregelt, ihre vier direkten Nachbarn anstecken.

Ist diese Berechnung erfolgt, werden als nächstes die infizierten Agenten über die Entfernungsschance gegebenenfalls in den Status *Removed* gesetzt.

```
1 self.__status_strategies = {
2     AgentState.SUSCEPTIBLE: [
3         (VaccineStatusStrategy(self.get_scheduler()), lambda state: state.
4           vaccine_toggle()),
5         (IncubationStatusStrategy(), lambda state: state.incubation_period_enabled
6           ()),
7     ],
8     AgentState.INFECTIVE: [
9         (QuarantineStatusStrategy(self.get_scheduler()), lambda state: state.
10          quarantine_enabled()),
11         (DefaultInfectionStrategy(), lambda state: True),
12         (LethalityStatusStrategy(), lambda state: state.lethality_toggle()),
```

```

10         (DefaultRemoveStrategy(), lambda state: not state.lethality_toggle()),
11     ],
12 }

```

**Listing 2.** Auflisten der Statusstrategien, die Bestimmung der aktiven Strategien wird mittels des Lambda-Ausdrucks durchgeführt.

Die Liste der Agenten liegt, wie erwähnt, in der Klasse *Grid* vor, welches das Simulationsfeld darstellt. Es wurde sich aus Performancegründen entschieden, die Behandlung der Events, welche die Agenten betreffen, nicht in den einzelnen Agenten selbst, sondern in dem Grid, welches alle Agenten enthält, vorzunehmen. So können zum Beispiel Funktionsaufrufe, welche aufgrund des aktuellen Status nicht nötig sind, vermieden werden (bspw. Bewegung der Agenten im Status *Removed*).

Die Zustände der Agenten werden ebenfalls über Events an die GUI übermittelt. Diese geht zu Beginn von einem leeren Feld aus, auf welchem die Agenten hinzugefügt werden. Über die Methode in Algorithmus 3 wird der Oberfläche nun mitgeteilt, die interne Repräsentation des Feldes an Position (row, col) auf den Wert `agent_state.value` zu setzen.

```

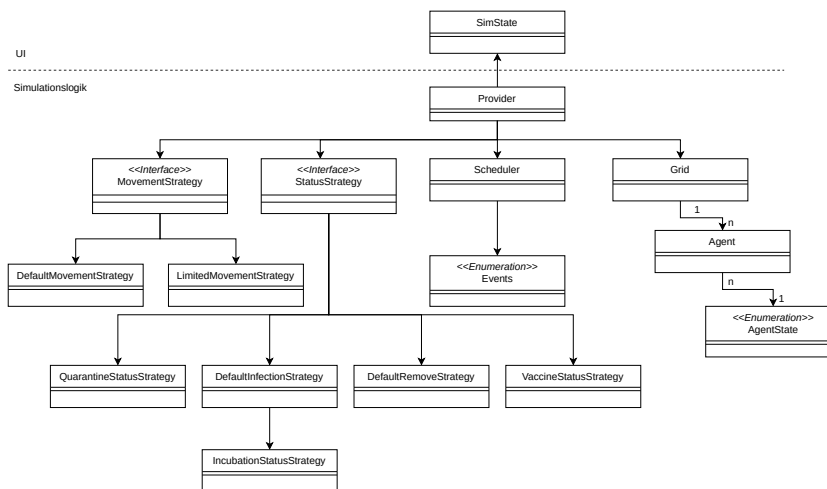
1 def update_gui_state(self, grid_pos: GridPos, agent_state: AgentState):
2     self.trigger_gui_event(Events.AGENT_CHANGE_GUI, {"row": grid_pos.row(), "col":
3         grid_pos.col(), "state": agent_state.value})
4 self.__logger.debug(f"Agent update, at position {grid_pos.row()}, {grid_pos.col()},
5     new state: {agent_state}")

```

**Listing 3.** Übermittlung einer Zustandsänderung an die GUI

Mit dem Event "REPAINT" wird der Oberfläche signalisiert, dass alle Updates abgeschlossen sind und die Graphen neu gezeichnet werden können.

Somit ist der Simulationsschritt abgeschlossen.



**Abbildung 1.** Übersicht über das Architekturschema der Simulationslogik

## 2.3 Geplante Erweiterungen (Eder)

Zusätzlich zum Basismodell haben wir vor diverse Erweiterungen einzubauen. Von diesen erhoffen wir uns einerseits eine realistischere Simulation, andererseits ein besseres Verständnis über die Ausbreitung von Epidemien, gerade im Kontext der aktuellen Situation rund um *COVID-19*.

### **Todesfälle und Immune**

Im Basismodell wird auf eine Unterscheidung zwischen toten und immunen Individuen verzichtet. Aus unserer Sicht ergibt eine Aufteilung aber durchaus Sinn, da sich Immune bewegen, Tote jedoch naturgemäß nicht. Wir werden daher ein paar Änderungen vornehmen und dessen Auswirkungen untersuchen.

### **Beschränkter Bewegungsradius**

Die Bewegungen der Individuen des Basismodells können mit „Teleportationen“ auf freie Zellen verglichen werden. Uns interessieren die Auswirkungen, wenn Bewegungen nur noch in einer limitierten Umgebung möglich sind.

### **Ausgangssperre**

Eine Maßnahme zur Eindämmung der Ausbreitung von Epidemien stellen Ausgangssperren dar. Wir wollen untersuchen, inwiefern diese den gewünschten Effekt haben, auch in Bezug auf die aktuellen Ausgangsbeschränkungen gegen COVID-19 (Bayerisches Staatsministerium für Familie, Arbeit und Soziales [2020](#)).

### **Inkubationszeit und Krankheitsdauer**

Die Inkubationszeit beschreibt die Zeit zwischen der Infektion mit einer Krankheit und dem Auftreten der ersten Symptome (*Inkubationszeit* – [Wikipedia 2020](#)). In dieser Zeit können die Personen schon ansteckend sein, jedoch sicherlich reduziert, da zum Beispiel Symptome wie Husten maßgeblich zur Ausbreitung beitragen, in der Inkubationszeit jedoch noch nicht vorhanden sind. Vereinfachend wollen wir in dieser Arbeit davon ausgehen, dass in dieser Zeit die Personen genauso ansteckend sind wie andere Infizierte mit Symptomen und daraus folgende Ergebnisse analysieren.

### **Quarantäne**

Eine weitere beliebte Maßnahme zur Eindämmung von Epidemien ist die Quarantäne, also das Isolieren von kranken Personen. Das erhoffte Ergebnis ist eine Verringerung der Neuinfektionen, welches wir mit unserem Versuch ebenso nachstellen möchten.

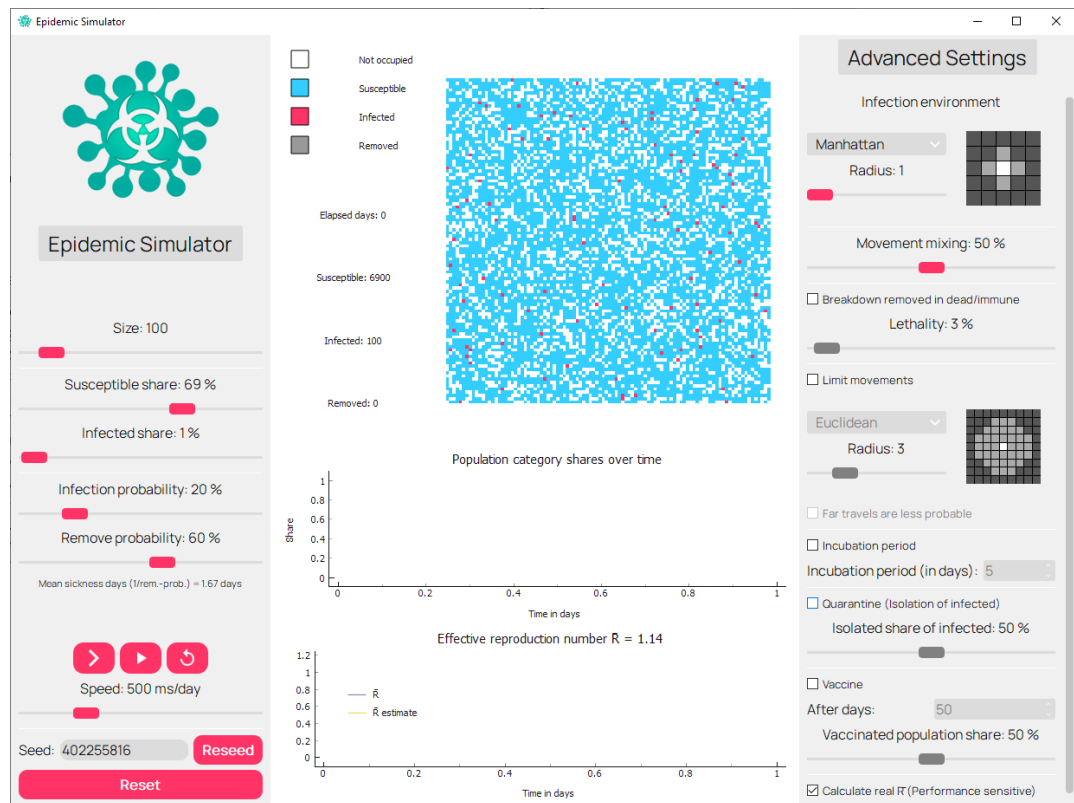
### **Impfstoff**

Zuletzt wollen wir die mögliche Entwicklung eines Impfstoffs untersuchen. Falls ein solches Medikament fertiggestellt wird - vereinfacht angenommen – ein bestimmter Anteil der Bevölkerung mit sofortiger Wirkung immun gegen die Krankheit. Dabei wollen wir vor allem die Unterschiede in den Todesfällen untersuchen und welche Rolle der Impfzeitpunkt und -anteil spielt.

## 2.4 Grafische Benutzeroberfläche (Eder)

Die Simulation bietet schon im Basismodell zahlreiche Einstellungsmöglichkeiten, wie die Größe des zweidimensionalen Zellularautomaten oder das Verhältnis von gesunden zu infizierten Individuen zu Beginn. Zusammen mit geplanten Erweiterungen bedarf es einer Möglichkeit die Simulation einfach zu steuern und geeignet zu visualisieren. Wir haben uns eine grafische Lösung ausgedacht. Genauer eine grafische Benutzeroberfläche mit dem Framework Qt (The Qt Company [2020](#)).

Zusätzlich erlaubt uns die Bibliothek *pyqtgraph* (*PyQtGraph - Scientific Graphics and GUI Library for Python* 2016) das Erstellen von interaktiven Visualisierungen.



**Abbildung 2.** Screenshot der Benutzeroberfläche zur Steuerung der Simulation

Die entstandene Benutzeroberfläche - wie in **Abbildung 2** dargestellt - ist horizontal in drei Teile aufgeteilt: **Basiseinstellungen und Steuerung**, **Visualisierungen** und die **Steuerung der Modellerweiterungen** (in **Abbildung 2** die mit unterschiedlichen Hintergrundfarben separierten Abschnitte, von links nach rechts).

### Basiseinstellungen und Steuerung

Hier werden Einstellungen getroffen, welche das Basismodell betreffen. Im Folgenden beschreiben wir die einzelnen Möglichkeiten wie im Screenshot oben dargestellt von oben nach unten:

- „**Size**“: die Größe des zweidimensionalen Zellularautomaten
- „**Susceptible share**“  $S(0)$ : der Anteil an gesunden Personen im Automaten zu Beginn
- „**Infected share**“  $I(0)$ : der Anteil an infizierten Personen im Automaten zu Beginn
- „**Infection probability**“  $p_i$ : die Wahrscheinlichkeit, dass eine infizierte Person eine gesunde Person in direkter Nachbarschaft mit der Krankheit ansteckt
- „**Remove probability**“  $p_r$ : die Wahrscheinlichkeit, dass eine infizierte Person entweder stirbt oder geneset und immun wird. Unter dem „Slider“ zur Steuerung der Wahrscheinlichkeit befindet sich ein Hinweis, dass diese Einstellung die durchschnittliche Krankheitsdauer direkt bestimmt und wie folgt berechnet wird:  $1/p_r$
- **Zeitsteuerung**: Drei Knöpfe sowie ein weiterer „Slider“ erlauben die zeitliche Steuerung der Simulation. Mit dem linken Knopf wird lediglich ein weiterer Tag simuliert. Der mittlere Knopf startet die fortlaufende iterative Simulation mit der mittels „Slider“ eingestellten Geschwindigkeit (in **Abbildung 2** werden zum Beispiel alle 500 Millisekunden ein Tag simuliert). Mit demselben Knopf kann die gestartete Simulation auch wieder pausiert werden. Der letzte Knopf setzt die



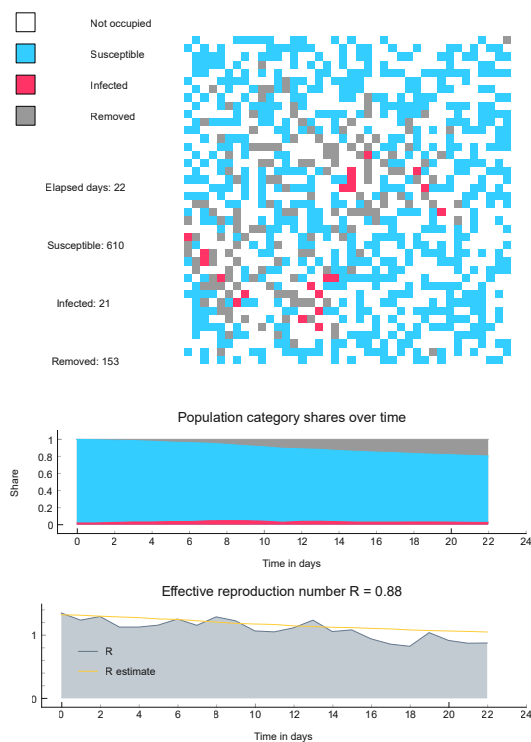
Simulation mit denselben Einstellungen zurück.

- **„Seed“:** Die Simulation basiert auf einem Zufallsgenerator und führt daher zu immer anderen Ergebnissen. Um trotzdem reproduzierbare Ergebnisse zu erhalten, kann hier eine 9-stellige Zahl - dem sogenannten „Seed“ - für den Zufallsgenerator eingestellt werden. Für denselben Seed wird beispielsweise immer derselbe Startzustand des zweidimensionalen Zellularautomaten generiert. Der Knopf neben dem Textfeld erzeugt einen neuen zufälligen 9-stelligen Seed.
- **„Reset“-Knopf:** dieser setzt die Simulation und die Einstellungen auf den Startzustand zurück

### Visualisierungen

Eine Simulation ohne sichtbares Ergebnis hat keinen großen Nutzen. Daher stellen wir verschiedene Visualisierungen in der Benutzeroberfläche zur Verfügung, welche unter anderem den aktuellen Zustand des Zellularautomaten darstellen. Diese sind noch einmal in Abbildung 3 zu sehen.

Der Momentanzustand der Simulation wird primär durch ein Bild visualisiert, in dem jede Zelle durch eine Farbe dargestellt wird. Eine nebenstehende Legende hilft bei der Zuordnung der Farben auf deren Bedeutung.



**Abbildung 3.** Darstellung der Visualisierungen in der Benutzeroberfläche

### Steuerung der Modellerweiterungen

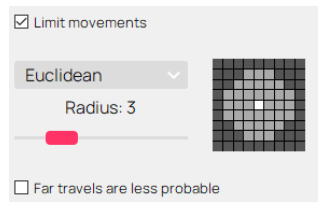
Zu den einzelnen Modellerweiterungen haben wir Steuerungsmöglichkeiten entworfen und implementiert.



**Abbildung 4.** Erweiterungssteuerung zum Aufteilen in Tote oder Immune

Abbildung 4 zeigt die Steuerung der Aufteilung in Tote und Immune. Diese Erweiterung kann aktiviert oder deaktiviert werden. Falls sie aktiviert ist, besteht zusätzlich die Möglichkeit die Töd-

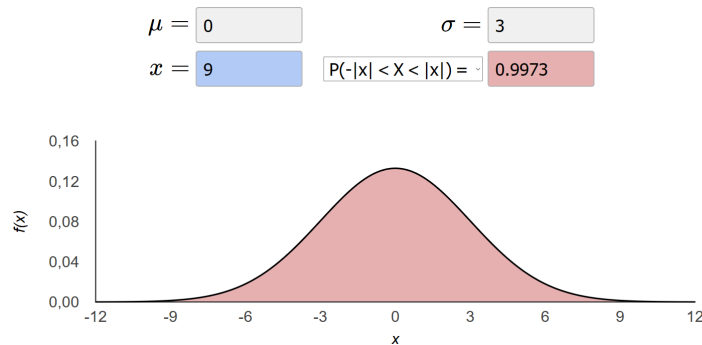
lichkeit der Krankheit über einen „Slider“ einzustellen. In der Legende der Visualisierungen wird *Removed* durch entsprechende *Tote* und *Immune* Elemente ersetzt.



**Abbildung 5.** Einstellungen für die Beschränkung der Personenbewegungen

Die Beschränkung der Bewegungen kann mit den in Abbildung 5 dargestellten Reglern eingestellt werden. Wenn die Checkbox „Limit movements“ gesetzt ist, ist die Bewegungslimitierung aktiv. Die folgenden Regler ermöglichen die Änderung der erlaubten Bewegungsumgebung um eine weiß dargestellte Zelle im Automaten. Neben der euklidischen Distanzmetrik steht auch die Manhattan-Metrik zur Verfügung.

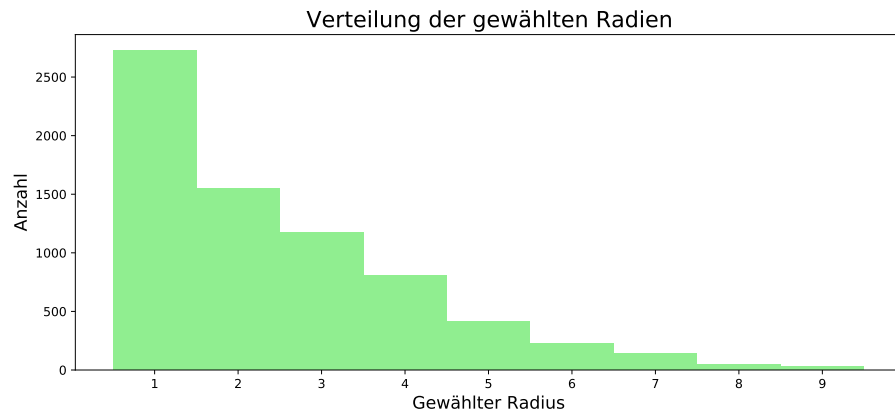
Des Weiteren kann über die Checkbox „Far travels are less probable“ entschieden werden, ob kürzerer Entfernungen wahrscheinlicher zurückgelegt werden als weite. Wenn diese aktiviert ist, wird aus einer Normalverteilung mit dem Mittelwert  $\mu = 0$  und Standardabweichung  $\sigma = r/3$  gezogen, wobei  $r$  der eingestellte Radius ist. Dadurch wird mit einer Wahrscheinlichkeit von  $P(-|x| < \mathbf{X} < |x|) = 0,9973$  ein Radius aus dem Intervall  $[0; r]$  gewählt. Falls wider Erwarten ein größerer Radius gewählt wird, wird stattdessen der maximale Radius  $r_{max} = r$  genommen. Ebenso muss der minimale Radius  $r_{min} = 1$  sein, da sich die Person sonst gar nicht bewegt. Daher wird der minimale Radius  $r_{min}$  gewählt, falls der gezogene Radius  $r < 1$  ist. Die Verteilung ist in Abbildung 6 dargestellt.



**Abbildung 6.** Normalverteilung, aus der gezogen wird bei einer beschränkten Bewegung mit der Option „Far travels are less probable“. Abbildung entnommen von *Normal Distribution Applet/Calculator* (2019).

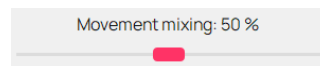
Es ist ersichtlich, dass aus dieser Verteilung häufiger kleine Radien gezogen werden, als große. Demnach überwinden Personen mit Vorliebe kurze Entfernungen und vermeiden große.

Zur Validierung dieser Aussage wurde die oben beschriebene Radius-Auswahl entsprechend oft ausgeführt und ein Histogramm über die erhaltenen Ergebnisse gebildet, welches in Abbildung 7 zu sehen ist.



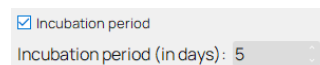
**Abbildung 7.** Verteilung der gezogenen Radien

Dieses Histogramm bestätigt, dass Personen wahrscheinlich kürzere Distanzen zurücklegen. „Im Jahr 2017 legte jede Person mit 39 km im Mittel einen Kilometer mehr zurück als im Jahr 2008“ (BMVI - *Mobilität in Deutschland* 2020). Der tatsächliche Bewegungsradius ist jedoch nur ein Bruchteil der 39 Kilometer Gesamtlänge. Für eine realistischere Bewegungssimulation kann es daher Sinn ergeben, kürzere Radien zu bevorzugen.



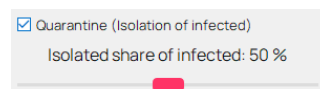
**Abbildung 8.** Steuerung für die Erweiterung Ausgangssperre

Der in Abbildung 8 zu sehende „Slider“ dient zur Steuerung der Durchmischungszahl  $m$ . Diese gibt den Grad der Durchmischung, also konkret die Anzahl der sich in einem Simulationsschritt bewegendenden Personen, an. Bei niedrigeren Werten bewegen sich weniger Personen, wodurch eine Ausgangssperre oder -beschränkung effektiv simuliert wird.



**Abbildung 9.** Steuerung der Inkubationszeit

Eine weitere Erweiterung erfordert die Einstellung der Inkubationszeit in Tagen. Abbildung 9 zeigt die entstandene Benutzeroberfläche.



**Abbildung 10.** Quarantäneinstellungen in der Benutzeroberfläche

Des Weiteren zeigt Abbildung 10 eine Möglichkeit zur Aktivierung einer Quarantänesimulation. Der dazugehörige „Slider“ regelt dabei den Anteil an Infizierten, welche von der restlichen Bevölkerung isoliert werden sollen.

**Abbildung 11.** Einstellungen der Impfstoffweiterung

Zu guter Letzt präsentieren wir unsere Steuerungsmöglichkeit für die Impfstoffweiterung der Simulation in [Abbildung 11](#). Anzugeben ist dabei die Anzahl an Tagen, nach denen der Impfstoff fertiggestellt wird und der zu impfende Teil der Bevölkerung.

### Berechnung der Nettoreproduktionsrate

Wie bereits in [Abschnitt 2.4](#) erwähnt, visualisieren wir die Nettoreproduktionsrate  $R(t)$  über die Zeit. Zur Wiederholung: Die Nettoreproduktionsrate stellt die durchschnittlichen Neuinfektionen dar, welche durch alle aktuell infizierten Individuen verursacht werden (*Basisreproduktionszahl* – [Wikipedia 2020](#)). Anhand dieser Beschreibung der Metrik haben wir einen Berechnungsalgorithmus entwickelt, welchen wir im Folgenden aus Gründen der Nachvollziehbarkeit vorstellen.

Für die Berechnung benötigen wir die folgenden Eingabedaten:

- Den Zustand des zweidimensionalen Zustandsautomaten zum Zeitpunkt  $t$ .
- Die Wahrscheinlichkeit, nach der ein Infizierter wieder gesundet oder stirbt: Remove probability  $p_r$ .
- Die Wahrscheinlichkeit, nach der ein Infizierter andere gesunde Personen ansteckt: Infection probability  $p_i$ .
- Die Umgebung  $\mathbf{U}$ , in der infizierte Personen gesunde anstecken können: Radius  $U_r$  und Metrik  $U_m$  (Euklidische oder Manhattan Distanzmetrik).

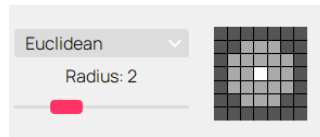
Wir gehen nach der folgenden Vorgehensbeschreibung, je infizierter Person  $x$ , vor:

- Für die infizierte Person  $x$  finden wir die Anzahl der infizierbaren (gesunden) Personen  $a$  in der definierten Umgebung  $\mathbf{U}$  um  $x$ .
- Weiter überprüfen wir die bereits von dieser Person  $x$  verschuldeten Neuinfektionen  $I_{neu}(x)$  sowie dessen aktuelle Krankheitsdauer  $T_{krank}(x)$ .
- Anhand der durchschnittlichen Krankheitsdauer (Berechnet durch  $1/p_r$ ) schätzen wir nun wie lange die Person  $x$  noch krank sein wird, wobei das noch mindestens ein Tag sein muss (Der Infektionsschritt erfolgt vor dem Gesunden oder Sterben des Infizierten in der Simulationslogik). Die Berechnung der restlichen Krankheitsdauer  $T_{rest}(x)$  ist also wie folgt:  $T_{rest}(x) = \max(\text{round}(1/p_r - T_{krank}(x)), 1)$ .
- Nachfolgend schätzen wir mit  $T_{rest}(x)$  wie viele Personen die infizierte Person  $x$  noch anstecken wird mit:  $I_{est}(x) = T_{rest}(x) * a * p_i$ .
- Die geschätzten  $I_{est}(x)$  und bisherigen Infektionen  $I_{neu}(x)$  durch die infizierte Person  $x$  werden aufsummiert als  $I_{total}(x)$ .

Zusammenfassen berechnen wir je aktuell infizierter Person die insgesamt voraussichtlich verursachten Neuinfektionen. Die Nettoreproduktionsrate ist der Mittelwert aller dieser berechneten Werte. Im folgenden wird die Anzahl dieser Werte mit  $n$  bezeichnet.

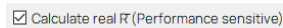
$$R(t) = \bar{R} = \frac{1}{n} * \sum_{i=1}^n I_{total}(i)$$

Zusätzlich zu den Steuerungsmöglichkeiten für die Erweiterungen gibt es noch zwei Möglichkeiten, welche direkt in die Berechnung der Nettoproduktionsrate einfließen.



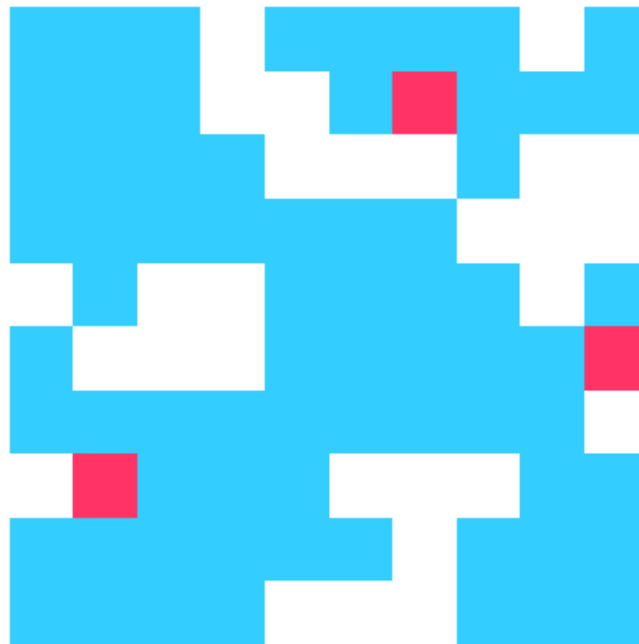
**Abbildung 12.** Steuerung der Infektionsumgebung

In Abbildung 12 ist die Benutzerschnittstelle zur Steuerung der Infektionsumgebung  $U$  zu sehen. Einstellbar ist wieder die Distanzmetrik (Euklidisch oder Manhattan) sowie der dazugehörige Radius. Diese wird nicht nur bei der Berechnung von  $R(t)$  zu Rate gezogen, sondern auch beim Infektionsschritt in der Simulationslogik.



**Abbildung 13.** Deaktivierung der Berechnung der tatsächlichen Nettoproduktionsrate

Des Weiteren zeigt Abbildung 13 eine Möglichkeit, die Berechnung der Nettoproduktionszahl zu deaktivieren. Der Grund dafür ist, dass die Berechnung teilweise sehr performanceintensiv ist, je nach Größe des zweidimensionalen Zustandsautomaten und der Infektionsumgebung. Ersatzweise, bzw. zusätzlich zur tatsächlichen Nettoproduktionsrate können wir die Metrik schätzen, um ein ungefähres Ergebnis zu erhalten. Laut *Basisreproduktionszahl – Wikipedia (2020)* kann eine Schätzung mit  $R(0) * \frac{S}{N}$  berechnet werden, wobei  $S$  der Anzahl an gesunden Personen und  $N$  die Anzahl der Gesamtpopulation entspricht. Das bedeutet, dass für die Schätzung der Nettoproduktionszahl lediglich der erste tatsächliche Wert  $R(0)$  berechnet werden muss und alle weiteren sich aus einer einfachen Funktion ergeben.



**Abbildung 14.** Beispiel für die Berechnung der Nettoproduktionsrate

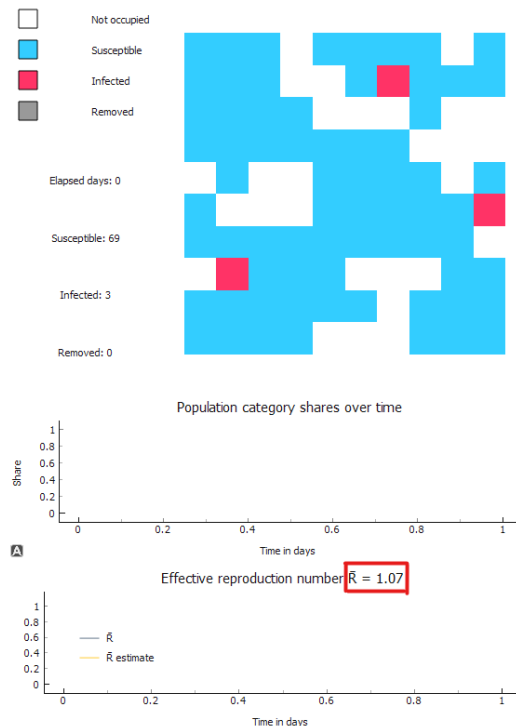
Zur Validierung der Berechnung der Nettoproduktionsrate möchten wir ein kleines Beispiel

präsentieren, welches in Abbildung 14 dargestellt wird. Zu sehen ist ein 10x10 zweidimensionaler Zellularautomat. Drei Zellen sind mit infizierten Personen belegt. Wir wollen nun die Netto-reproduktionsrate zum Zustand aus Abbildung 14 berechnen. Bevor wir mit der Berechnung beginnen können, brauchen wir weitere Informationen:

- Infektionsumgebung **U**: Die gesunden Zellen in direkter Nähe (oberhalb, unterhalb, links und rechts) von einer infizierten Zelle werden betrachtet (= Manhattan Metrik mit Radius 1).
- Infektionswahrscheinlichkeit  $p_i = 20\%$
- Remove-Wahrscheinlichkeit  $p_r = 50\%$  (= durchschnittlich 2 Tage Krankheitsdauer)

Nun die Berechnung:

- Die Zelle links hat drei gesunde Zellen in der Umgebung **U**. Daraus folgt eine Schätzung von Krankheitsdauer(2) \* Gesunde Zellen in der Nähe(3) \* Infektionswahrscheinlichkeit(20%) = 1, 2. Voraussichtlich wird diese Zelle also 1, 2 Neuinfektionen verursachen.
- Die Zelle oben in der Mitte erhält eine Schätzung von  $2 * 3 * 20\% = 1, 2$ .
- Die letzte Zelle auf der rechten Seite  $2 * 2 * 20\% = 0, 8$ .



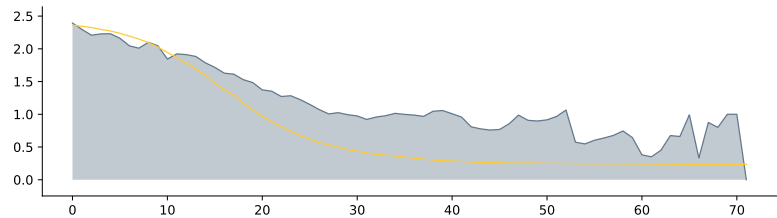
**Abbildung 15.** Ergebnis der Beispielberechnung der Netto-reproduktionsrate in der Benutzeroberfläche (rot gekennzeichnet)

Der Mittelwert und damit das Ergebnis  $R(0)$  beträgt gerundet 1, 07. Durchschnittlich verursacht eine infizierte Zelle zum aktuellen Zeitpunkt also knapp über eine Neuinfektion. Abbildung 15 zeigt das berechnete Ergebnis in der grafischen Benutzeroberfläche (Rot umrandet).

Des Weiteren wollen wir die Schätzung der Netto-reproduktionszahl überprüfen. Ein Beispiel mit den folgenden Parametern resultierte im Graph aus Abbildung 16:

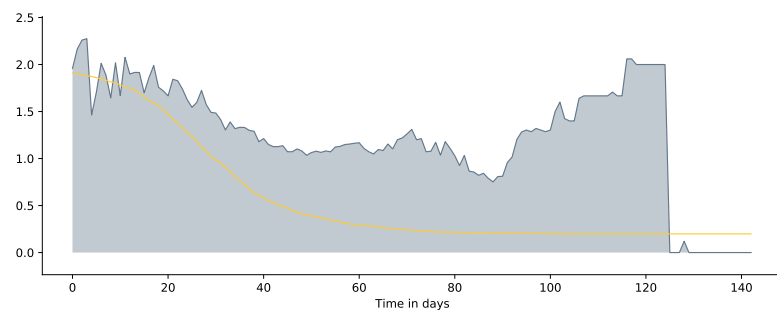
- Größe des Zellularautomaten: 50x50
- Susceptible share: 69%

- Infected share: 1%
- Infection probability: 15%
- Remove probability: 15%
- Seed (zum Reproduzieren): 968722973
- Alle anderen Parameter wurden auf dem Standardwert belassen.



**Abbildung 16.** Vergleich der Schätzung der Nettoproduktionsrate mit den tatsächlichen Werten im Zeitverlauf

Die graublau Linie kennzeichnet die tatsächlich berechnete Nettoproduktionsrate. Zusätzlich lässt sich in dem Diagramm eine gelbe Linie finden, welche den Schätzwert repräsentiert. Wie man sehen kann, approximiert die glatte Schätzungskurve die tatsächlichen Werte recht gut.



**Abbildung 17.** Verhalten der Nettoproduktionszahl gegen Ende der Simulation

Zu guter Letzt möchten wir noch erwähnen, dass unsere Simulation auf Zufall basiert. Gerade bei der Nettoproduktionsrate können unvorhergesehene Ergebnisse auftreten, wie auch in [Abbildung 17](#). Das Diagramm zeigt eine Erhöhung von  $R(t)$  gegen Ende der Simulationszeit. Grund dafür sind Langzeitkranke, die, wie der Zufall es will, nicht nur in der Realität, sondern auch in der Simulation in Erscheinung treten können. Der Graph von  $R(t)$  wird am Ende von nur wenigen Personen beeinflusst, da nur die *aktuell* noch infizierten mit in die Berechnung einfließen. Langzeitkranke mit einer bestehenden Zahl bereits verursachter Neuinfektionen führen zu einer hohen Nettoproduktionsrate. Diese Beobachtung führt uns zu der Annahme, dass die Metrik gegen Ende, bei wenigen Infizierten, mit Vorsicht zu genießen ist.

### 3 Ergebnisse

#### 3.1 Implementierung des Basismodells (Stiglmeier)

Basierend auf der in Kapitel 2.2 dargestellten Architektur wird die Logik, welche den Zustand des Modells verändert, in Methoden implementiert, welche aus der Sicht der Agenten heraus aufgerufen werden. In der Implementierung des Modells werden diese *Strategien* genannt. So existiert zum Beispiel für das Bewegen eines Agenten eine Bewegungs-Strategie und für die Erweiterung des Modells um eine Inkubationszeit eine Inkubations-Strategie. Das heißt, dass der Scheduler jeweils für alle Agenten auf dem Feld und je nach im User Interface aktivierter Erweiterung lediglich die passenden Strategien aufruft. Innerhalb der Strategie wird dann die relevante Veränderung ausgeführt. Im User Interface können neben der Aktivierung von Erweiterungen auch Parameter für die Strategien eingestellt werden. Die relevanten Parameter für das Basismodell sind auf der linken Seite des User Interfaces einstellbar (Vgl. Kapitel 2.4) und lauten:

- Size
- Susceptible share
- Infected share
- Infection probability
- Remove probability

Die Parameter *Size*, *Susceptible share* sowie *Infected share* sind dabei benötigt um zu Beginn das Feld mit Agenten zu füllen (Vgl. Kapitel 2.2). Zu Beginn wird ebenfalls der Parameter *Seed* für den Zufallszahlengenerator neu gesetzt. Das heißt, wenn das Feld neu erstellt wird, wird auch der Zustand des Generators gesetzt. Die beiden anderen Parameter *Infection probability* und *Remove probability* sind jedoch notwendig um das Verhalten der innerhalb des Basismodells verwendeten Strategien richtig zu konfigurieren. Die für das Basismodell aktiven Strategien sind dabei die Folgenden:

##### **Standardbewegungs-Strategie**

Diese Strategie wird beim Event „AGENT\_MOVEMENT“ aufgerufen. Dabei wird als erstes überprüft, ob eine freie Position innerhalb des Feldes existiert. Diese Überprüfung ist für den Extremfall relevant, dass die eingestellten Parameter *Susceptible share* und *Infected share* zusammen 100% ergeben und das Feld vollkommen ausgefüllt ist. Anschließend wird überprüft, ob der Agent entfernt ist. Entfernte Agenten dürfen sich nicht mehr bewegen. Anschließend erfolgt eine Überprüfung, ob eine gezogene Zufallszahl über oder unter der Grenze des Parameters *Movement mixing* ist. Dabei wird das Modell so gesteuert, dass sich nur ein durch den Parameter festgelegter Anteil überhaupt bewegen darf. Der Parameter kann zwar jederzeit im User Interface konfiguriert werden, ist aber im Basismodell standardmäßig bei 100% festgelegt. Änderungen sind nur für Erweiterungen relevant. Sind alle Überprüfungen erfolgreich bestanden, dann wird solange zufällig eine Koordinate im Feld ausgewählt und überprüft, ob diese frei ist, bis eine freie Koordinate gefunden wurde. Dann bewegt sich der Agent auf die neue Position.

##### **Standardinfektions-Strategie**

Diese Strategie wird beim Event „STATUS\_UPDATE“ aufgerufen. Dabei wird als erstes überprüft, ob der Agent andere Agenten anstecken darf. Also ob er infektiös ist, nicht unter Quarantäne steht oder sich noch in der Inkubationszeit befindet. Diese Überprüfungen sind zwar nicht für das Basismodell relevant, aber da die Architektur so festgelegt ist, dass jede Einstellung am Modell in sich kompatibel ist, müssen auch hier Überprüfungen für die Erweiterungen erledigt werden. Anschließend werden je nach den im User Interface konfigurierten Einstellungen zur Größe und Art des



Infektionsradius alle umliegenden Agenten ausgewählt. Für jeden ausgewählten Agenten in diesem Schritt wird dann überprüft, ob eine Zufallszahl über oder unter der Grenze des Parameters *Infection probability* liegt und der Agent dementsprechend angesteckt wird oder nicht. Da diese Logik für jeden infektiösen Agenten ausgeführt wird, steigt auch die Wahrscheinlichkeit für einen gesunden Agenten angesteckt zu werden, wenn mehrere infektiöse Agenten in der Nähe sind, da er sich mehrmals der Ansteckungswahrscheinlichkeit aussetzt.

### **StandardEntfernungs-Strategie**

Diese Strategie wird beim Event „STATUS\_UPDATE“ aufgerufen. Dabei wird überprüft, ob der Agent überhaupt entfernt werden darf. Also ob er im infektiösen Zustand ist. Anschließend wird überprüft, ob eine Zufallszahl über oder unter der Grenze des Parameters *Remove probability* liegt und der Agent dementsprechend den entfernten Zustand annimmt.

Um die allgemeine Performance zu erhöhen, werden wichtige Größen, wie die Anzahl der aktuellen Infizierten, im Simstate (Vgl. Abbildung 1) gespeichert um möglichst wenig über das gesamte Feld iterieren zu müssen. Dadurch wird zum Beispiel die Berechnung der Nettoerproduktionsrate  $R$  beschleunigt. Die Aktualisierung der wichtigen Größen erfolgt dabei direkt in den Strategien.

## **3.2 Betrachtung des Basismodells (Beil)**

Das Basismodell besitzt folgende Einstellungen:

- Size  $L = 100$
- Susceptible share  $S(0) = 69 \%$
- Infected share  $I(0) = 1 \%$
- Infection probability  $p_i = 20 \%$
- Remove probability  $p_r = 60 \%$
- Movement mixing  $m = 100 \%$
- Infection environment = Manhattan mit einem Radius von 1
- Keine Bewegungseinschränkung
- Keine Impfung
- Keine Inkubationszeit
- Keine Quarantäne
- Keine Unterscheidung von Immunen und Tote
- Seed = 550475099

d	0	5	10	15	20	25	30	35	38
S(t)	6900	6685	6472	6362	6290	6243	6220	6201	6200
I(t)	100	60	36	27	13	11	9	1	0
R(t)	0	348	492	611	697	746	771	796	800

**Tabelle 1.** Werte aus der Basissimulation



**Abbildung 18.** Ergebnisse des Basismodells für einen Seed von 550475099.

Es gibt nur die drei Kategorien Susceptible, Infective und Removed. Es gibt insgesamt  $L \times L = 100 \times 100 = 10.000$  Felder. Davon sind 6.900 von Empfängern und 100 von Infizierten besetzt. 3000 Felder sind dadurch frei. Jeder Agent bewegt sich nach jeder Zeiteinheit auf ein zufälliges freies Feld. Nachdem sich alle Agenten bewegt haben, werden die Empfänger mit einer Wahrscheinlichkeit von  $p_i = 20\%$  infiziert, falls sich dieser in einem Ansteckungsgebiet eines Infizierten befindet. Anschließend wird der Status der Infizierten mit der Wahrscheinlichkeit  $p_r = 60\%$  auf Removed gesetzt.

### Susceptible

Die Anzahl der Empfänger nimmt im Laufe der Simulation um 700 ab. Da Empfänger infiziert werden können, es aber keine neuen Empfänger entstehen können, war dieser Verlauf zu erwarten. Wenn an einem Tag niemand neues infiziert wurde, bleibt die Anzahl der Empfänger gleich. Dies ist nur wahrscheinlich, wenn sehr wenige Infizierte vorhanden sind. Dies ist am Tag 36 und Tag 38 aufgetreten. Der Schwund an Empfängern ist zum Beginn am höchsten, da dort die Anzahl an Infizierten am höchsten ist. Vorausgesetzt die Anzahl an Empfängern ist viel größer wie die Anzahl an Infizierten und Entfernten, dann gilt, umso mehr Infizierte, umso größer ist der Schwund an Empfängern, da sich mehr Empfänger anstecken können.

### Infected

Die Anzahl der Infizierten beträgt zum Ende der Simulation 0. Damit sind keine weiteren Infektio-

nen möglich und die Simulation ist beendet. Die Anzahl an Infizierten sinkt dabei Anfangs schneller wie zum Ende. Dies liegt an der höheren Anzahl an Infizierten.  $p_r = 60\%$   $\Rightarrow$  wenn 100 Personen infiziert sind, werden im Durchschnitt 60 Personen entfernt. Sind aber nur noch 10 Infizierte vorhanden, dann werden auch nur 6 Personen entfernt. Daher sinkt die Anzahl an Infizierten Anfangs schneller wie zum Ende. Weil  $p_r > p_i$  werden mehr Infizierte Agenten im Durchschnitt entfernt als neue angesteckt. Daher sinkt die Anzahl der Infizierten im Durchschnitt konstant. Das Maximum an Infizierten ist daher zum Beginn der Simulation vorhanden und beträgt 100. Mehr dazu im Abschnitt 3.3. Während der Simulation steigt die Anzahl der Infizierten 8 mal. Dies geschieht, wenn mehr Empfänger infiziert werden, als Infizierte entfernt werden. Dazu müssen viele Agenten günstig für eine Infizierung gesetzt werden (4 Empfänger umgeben einen Infizierten).

### **Removed**

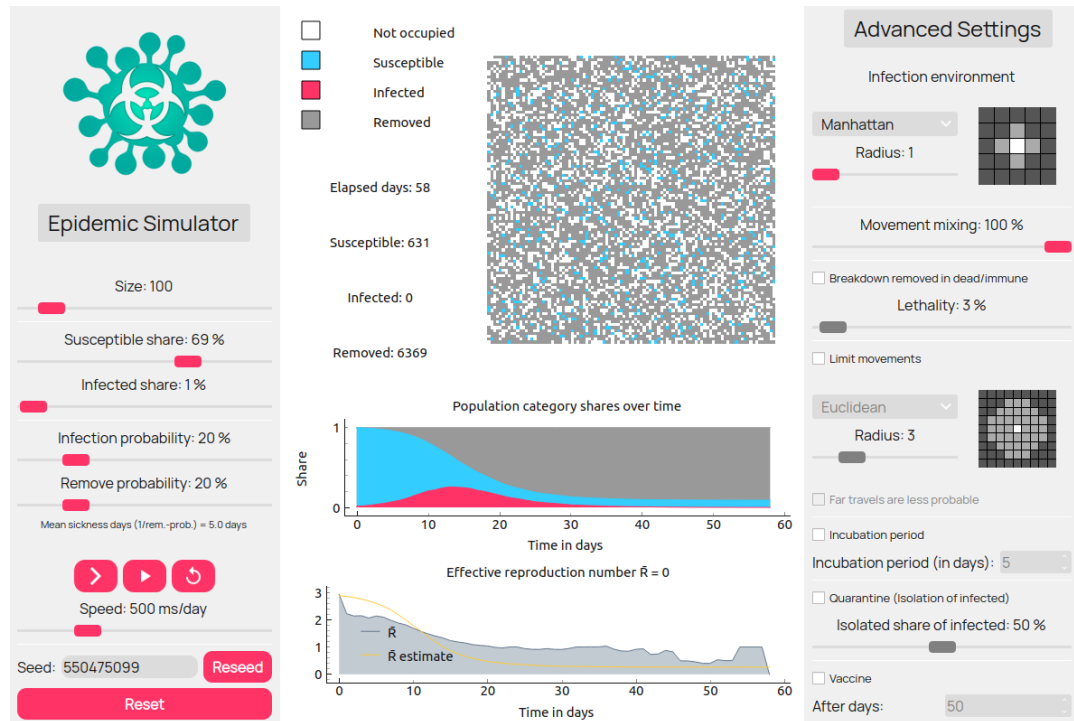
Die Anzahl der Entfernten steigt von Tag zu Tag. Zum Beginn der Simulation beträgt die Anzahl an Entfernten 0, da es zum Beginn der Simulation noch keine Entfernten geben soll. Der Anstieg ist zum Beginn der Simulation am höchsten. Dieser ist im Durchschnitt  $p_r \times$  Anzahl an Infizierten. Da die Anzahl an Infizierten zum Beginn am höchsten ist, ist der Anstieg an Entfernten zum Beginn auch am höchsten.

### **Mittlere Reproduktion**

Die mittlere Reproduktion aller Agenten beträgt im Durchschnitt über die Simulation 1,12. Die Schwankungen von  $\pm 0,5$  entstehen durch den Zufall. Sind zufälligerweise viele Empfänger im Infektionsbereich vieler Empfänger, dann steigt die Reproduktionszahl. Sind zufälligerweise wenig Empfänger im Infektionsbereich von den Empfängern, dann sinkt die Reproduktionszahl. Am Tag 36 der Simulation steigt die Reproduktionszahl auf 3,4. Hier kommt es zu dem Phänomen, welches auch bei dem Bild 17 auftritt. Zum Ende der Simulation sinkt die Anzahl an Infizierten. Dadurch ist die Reproduktionszahl nur noch von wenigen Infizierten abhängig. Weswegen Ausreiser, die überdurchschnittlich lange Leben, den Wert hoch treiben. Am letzten Tag (Tag 38) beträgt die Reproduktionszahl 0, da es keine Infizierte mehr gibt.

### Zusätzliche Untersuchung

Das Basismodell wird nun mit  $p_r = 20\%$  ausgeführt und analysiert. Alle anderen Einstellungen bleiben bestehen.



**Abbildung 19.** Ergebnisse des Basismodells für einen Seed von 550475099 mit  $p_r = 20\%$ .

d	0	10	11	12	13	14	15	20	30	40	50	58
S(t)	6900	4260	3812	3289	2817	2424	2072	1138	713	639	631	631
I(t)	100	1438	1542	1704	1794	1766	1733	1084	224	58	5	0
R(t)	0	1302	1646	2007	2389	2810	3195	4778	6063	6303	6364	6369

**Tabelle 2.** Werte aus der Basissimulation mit  $p_r = 20\%$ .

### Susceptible

Im Vergleich zur Basissimulation, befindet sich der maximale Schwund nicht zum Beginn der Simulation. Er hat sein Maximum vom Tag 13 auf den Tag 14 mit 393 Verlusten, da das Maximum an Infizierten am Tag 13 ist, anstatt zum Beginn.

### Infected

Wie bei allen SIR-Modellen endet diese Simulation mit 0 Infizierte. Im Vergleich zum Basismodell, ist hier  $p_r = p_i$ . Durch das ausgewogenere Verhältnis zwischen den neuen Infektionen und Entfernungen, steigt die Anzahl an Infizierten bis diese am Tag 13 ihr Maximum mit 1794 Infizierten erreicht. Der Hochpunkt bildet sich, da die Anzahl der Empfänger verhältnismäßig gering wird. Mehr dazu im Abschnitt 3.3.

### **Removed**

Agenten der Kategorie Removed machen am Ende der Simulation den größten Anteil mit 6369 Personen aus. Das Maximum des Anstieges befindet sich vom Tag 13 auf den Tag 14 mit 421 zusätzlichen entfernten Agenten. Dieses Maximum befindet sich hier, da sich das Maximum an Infizierten am Tag 13 befindet.

### **Mittlere Reproduktion**

Die mittlere Reproduktion ist zum Start der Simulation höher wie im Basismodel, da  $p_r$  geringer ist. Dadurch infiziert ein Infizierter im Mittel mehr Personen, da er länger diesen Status besitzt. Für  $p_r = 0,6$  besitzt ein Agent den Status Infiziert im Durchschnitt für 1,66 Tage. Für  $p_r = 0,2$  sind es im Durchschnitt 5 Tage. Da im Laufe der Simulation die Anzahl der Empfänger stark abnimmt und die Anzahl an Entfernten stark steigt, sinkt die Infektionsmöglichkeit mit dem Vorranschreiten der Simulation. Dadurch sinkt die Reproduktionszahl mit dem Vorranschreiten der Simulation. Der Anstieg am Ende der Simulation ist wieder auf das Phänomen zurückzuführen, welches durch die geringe Anzahl an Infizierten und dem Zufall zustande kommt.

### 3.3 Grenzwertbetrachtungen (Beil)

Es gilt im Mittel von einem Schritt auf den nächsten:

$$S_{t+1} = p - I_{t+1} - R_{t+1} \quad (1)$$

$$I_{t+1} = I_t + S_t(1 - (1 - p_i I_t)^4) - p_r I_t \quad (2)$$

$$R_{t+1} = R_t + p_r I_t \quad (3)$$

$S_t$ : Dichte an Empfängern

$I_t$ : Dichte an Infizierten

$R_t$ : Dichte an Entfernten

Die Dichte ist ein Fließkommerzahl im reellen Bereich zwischen 0 und 1.

$p$ : Summe aus  $S_t$ ,  $I_t$ ,  $R_t$  und hatt dadurch konstant den Wert 1

$p_i$ : Wahrscheinlichkeit einer Ansteckung

$p_r$ : Wahrscheinlichkeit, dass ein Infizierter entfernt wird

Die Wahrscheinlichkeiten sind eine Fließkommerzahl im reellen Bereich zwischen 0 und 1.

#### Allgemeine Betrachtung

Durch diese Regel entsteht ein Verlauf. Dieser Verlauf wird zuerst im Allgemeinen anhand dieser Regel untersucht.

#### Susceptible

$$S_{t+1} = p - I_{t+1} - R_{t+1} \quad (4)$$

Die Dichte der Empfänger wird bestimmt, indem von der Gesamtdichte  $p = 1$  die Dichte der Infizierten und die Dichte der Entfernten abgezogen werden. Da die Summe aus  $I_{t+1}$  und  $R_{t+1}$  mit dem voranschreiten der Tage steigt, sinkt die Dichte der Empfänger mit dem voranschreiten der Tage. Der Anstieg des Empfängerschwundes ist dann Maximal, wenn der Zuwachs der Dichte von Infizierten und Entfernten maximal ist.

#### Infected

$$I_{t+1} = I_t + S_t(1 - (1 - p_i I_t)^4) - p_r I_t \quad (5)$$

Die Dichte der Infizierten kann im Vergleich zu den anderen Dichten steigen und fallen.  $I_t$  ist dabei der aktuelle Bestand der Dichte.  $S_t(1 - (1 - p_i I_t)^4)$  ist die Dichte an neuen Infizierten. Die Potenz zu 4 kommt durch die 4 möglichen Felder, die angesteckt werden können.  $-p_r I_t$  sind die Infizierten, die nach diesen Tag den Status "removed" bekommen. Werden mehr Agenten infiziert, wie entfernt, so steigt die Dichte der Infizierten. Werden mehr Agenten entfernt, wie infiziert, dann sinkt die Dichte der Infizierten. Ob die Dichte steigt oder fällt ist von  $S_t$ ,  $p_i$  und  $p_r$  abhängig. Hohe  $S_t$  und  $p_i$  mit einen geringen  $p_r$  begünstigen eine Erhöhung der Dichte. Ein hohes  $p_r$  und ein geringes  $S_t$  und  $p_i$  begünstigen eine Senkung. Da  $S_t$  mit voranschreitenden Tage immer kleiner wird, sinkt die Dichte an Infizierten immer ab einen bestimmten Tag. Dieser Tag ist das Maximum. Von dort an sinkt die Dichte, da  $S_t$  immer weiter abnimmt. Wenn  $p_r$  deutlich größer wie  $p_i$  ist, sinkt die Dichte an Infizierten von Beginn an.  $\Rightarrow I_t$  für  $t \rightarrow \infty$  ist immer 0. Vorausgesetzt man nimmt an, dass  $R_0 = 0$  und  $I_0 \ll S_0$ , dann ist auch  $I_1$  klein und näherungsweise gilt:

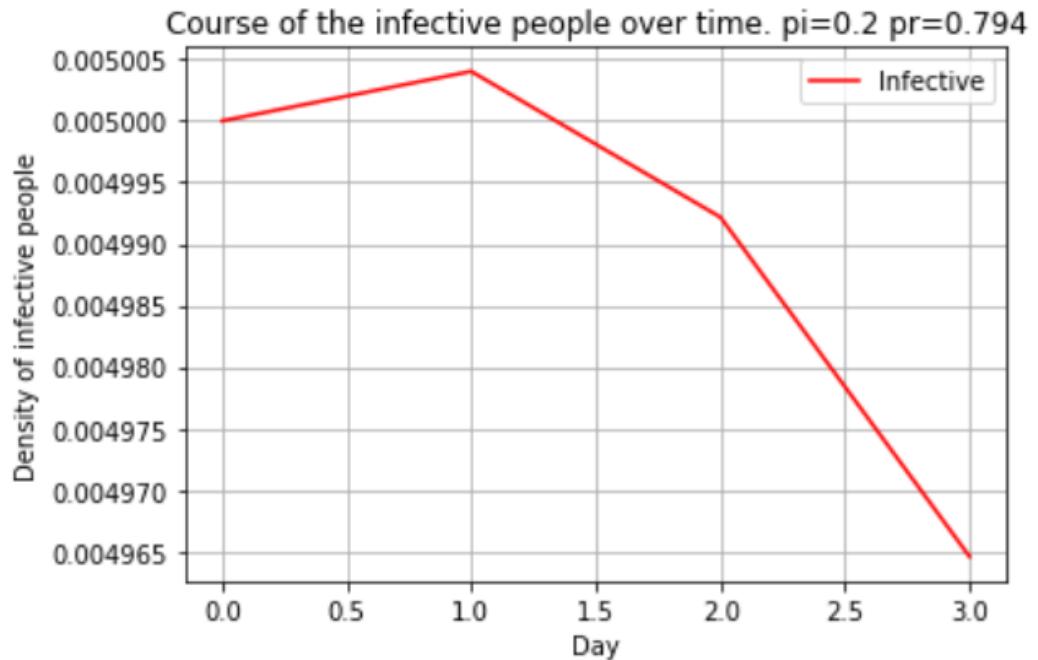
$$I_1 - I_0 = (4p_i S_0 - p_r) I_0 + O(I_0^2) \quad (6)$$

Für  $S_0 = \frac{p_r}{4p_i}$  gilt  $I_1 - I_0 = O(I_0^2)$ . Für kleine  $I_0$  ist  $O(I_0^2)$  vernachlässigbar und damit 0.  $\Rightarrow$  Ist  $S_0 > \frac{p_r}{4p_i}$ , dann ist  $I_1 > I_0$  und somit steigt anfangs die Dichte der Infizierten, bis das Maximum erreicht wird. Anschließend verläuft  $S_t$  gegen 0. Ist  $S_0 < \frac{p_r}{4p_i}$ , dann ist  $I_1 < I_0$  und somit sinkt die Dichte an Infizierten von Beginn an. In den folgenden Bildern ist dieses Verhalten zu beobachten.

$$S_0 = 0.995$$

$$p_r / (4 * p_i) = 0.9925$$

$$I_0^2 = 2.5e-05$$



**Abbildung 20.** Beispiel für  $S_0 > \frac{p_r}{4p_i}$

$$S_0 = 0.995$$

$$p_r / (4 * p_i) = 0.9975$$

$$I_0^2 = 2.5e-05$$



**Abbildung 21.** Beispiel für  $S_0 < \frac{p_r}{4p_i}$

### Removed

$$R_{t+1} = R_t + p_r I_t \quad (7)$$

$R_t$  ist der aktuell Bestand der Dichte an Entfernten.  $p_r I_t$  sind die Infizierten, die diesen Tag den Status Removed erhalten.  $R_t$  startet immer mit den Wert 0 und steigt bis zum Ende der Simulation immer weiter an. Der Zuwachs der Dichte der Entfernten ist umso größer, umso höher die Dichte der Infizierten am vorherigen Tag war. Daher ist der Zuwachs maximal am Tag nach dem Hochpunkt der Dichte der Infizierten.



### Betrachtung eines Beispiels

Zur Veranschaulichung dieser Regeln, wurde ein Jupyter Notebook erstellt, welches diese Regeln implementiert. Der Verlauf wird graphisch dargestellt. Es benötigt folgende Eingangsparameter:

$p_i$ : Wahrscheinlichkeit einer Ansteckung

$p_r$ : Wahrscheinlichkeit, dass ein Infizierter entfernt wird

$I_0$ : Dichte der Infizierten zum Beginn der Simulation

Für diese Veranschaulichung werden folgende Werte verwendet:

$p_i = 0,2$

$p_r = 0,4$

$I_0 = 0,01$

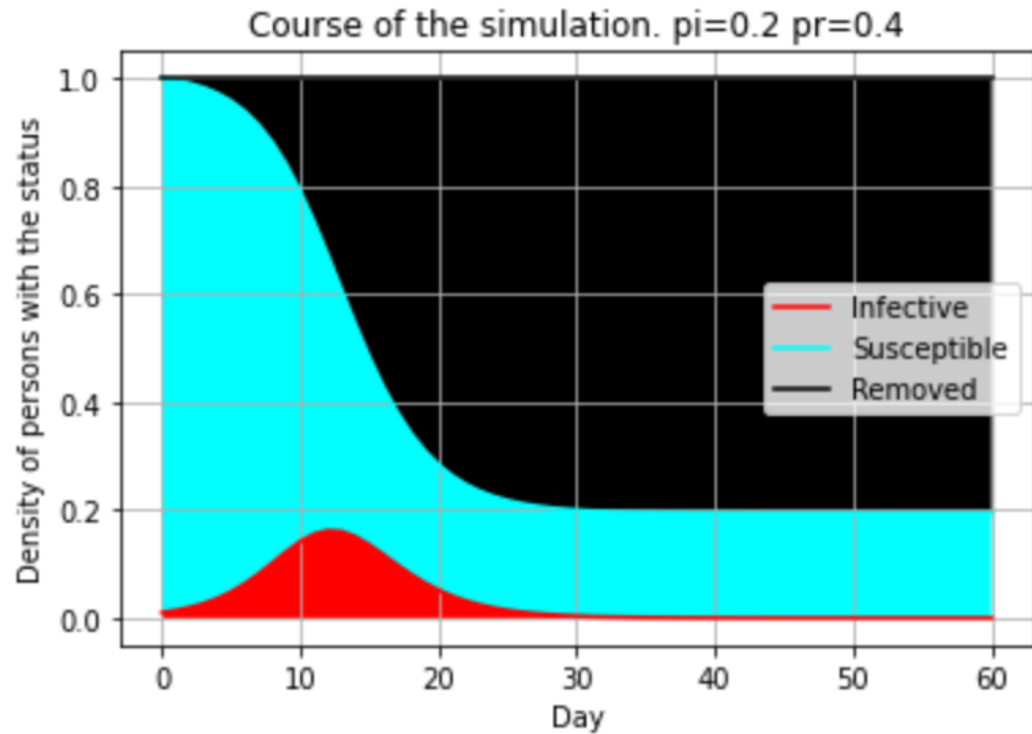
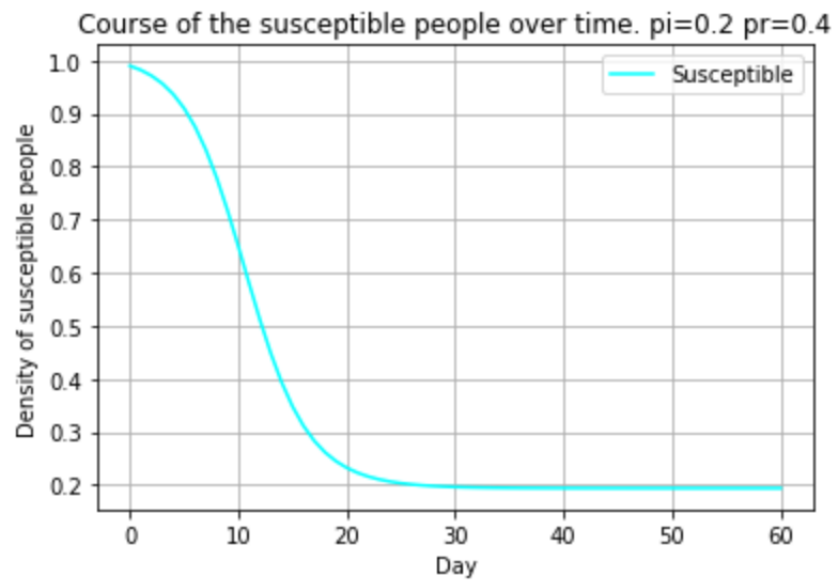
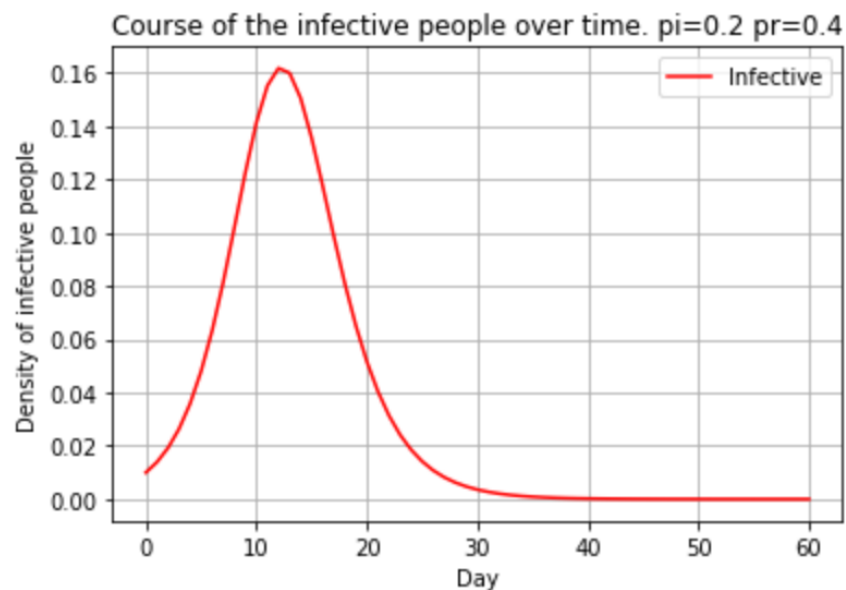


Abbildung 22. Verlauf der Dichten in einem Plot



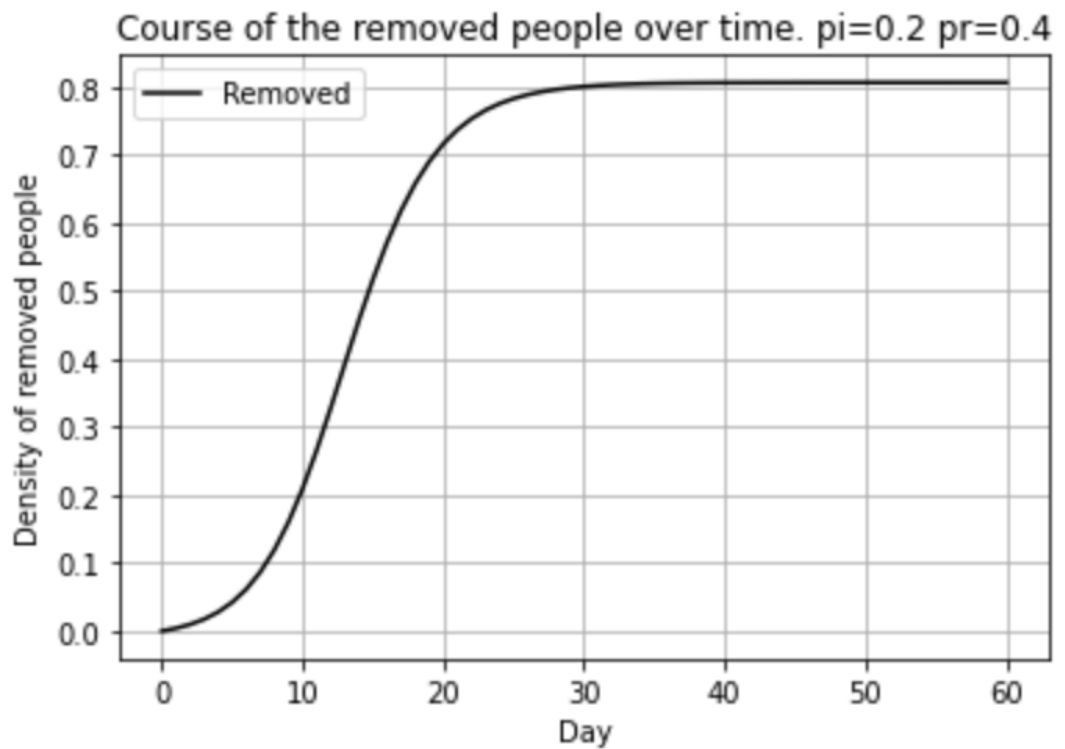
**Abbildung 23.** Verlauf der Dichte der Empfänger

Wie beschrieben sinkt die Dichte der Empfänger beim Vorranschreiten der Simulation. Das maximum des Schwundes befindet sich beim Maximum der Dicht der Infizierten.



**Abbildung 24.** Verlauf der Dichte der Infizierten

Da  $S_0 > \frac{\pi_r}{4\pi_i}$  ist, steigt die Dichte der Infizierten zum Beginn. Durch den Einbruch der Dichte der Empfänger dominiert der entfernende Anteil und die Dichte beginnt zu sinken. Da die Empfänger weiterhin sinken, geht die Dichte der Empfänger gegen 0.



**Abbildung 25.** Verlauf der Dichte der Entfernten

Wie beschrieben steigt die Dichte der Entfernten beim Vorranschreiten der Simulation. Das maximum des Zuwachses befindet sich beim Maximum der Dichte der Infizierten.

### 3.4 Validierung des Basismodells (Beil, Eder)

Um zu überprüfen, ob unser Basismodell korrekt ist, können wir einen qualitativen Vergleich mit anderen Modellen durchführen, welche als korrekt befunden wurden. Dies wird im Abschnitt 3.4 durchgeführt. Des weiteren können wir Überprüfen, ob unser Basismodell qualitativ mit unserer Grenzwertbetrachtung übereinstimmt. Dies wird im folgenden Abschnitt überprüft.

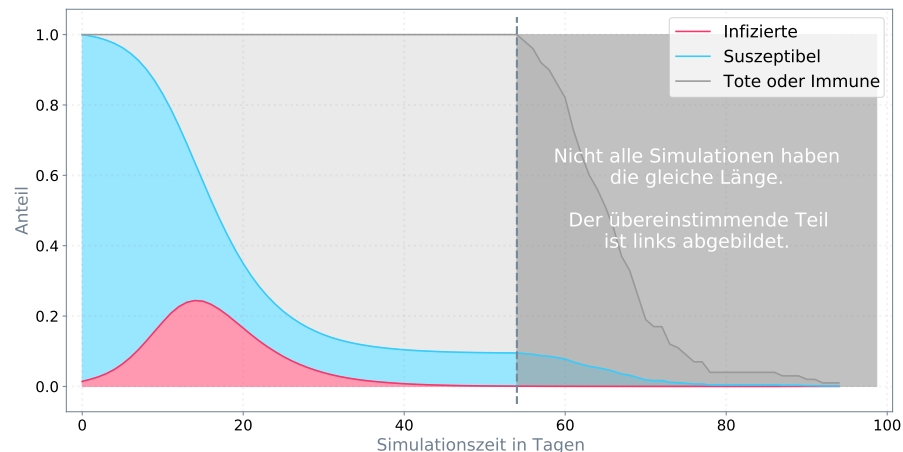
#### Validierung anhand Grenzwertbetrachtung

Die Grenzwertbetrachtung gibt uns folgende Merkmale vor:

- $S_t$  sinkt im Laufe der Simulation
- Das Maximum des Schwundes von  $S_t$  ist dann, wenn  $I_t$  sein Maximum erreicht
- $I_t$  für  $t \rightarrow \infty$  ist 0
- Für  $S_0 > \frac{p_r}{4p_i}$  steigt  $I_t$  zum Beginn der Simulation bis das Maximum erreicht ist. Anschließend verläuft  $I_t$  gegen 0
- Für  $S_0 < \frac{p_r}{4p_i}$  verläuft  $I_t$  von Beginn an gegen 0
- $R_t$  steigt im Laufe der Simulation
- $R_t$  wächst maximal, am Tag nach dem Maximum von  $I_t$

Treffen diese Merkmale auf unser Basismodell zu, ist dies ein Indikator auf die Korrektheit unseres Modells. Als Beispielwerte, werden die Konfigurationen verwendet, welche im Abschnitt 3.2 beschrieben sind. Das Modell wird über 100 Simulationen gemittelt mit  $p_r = 20\%$  und  $p_r = 60\%$ . Die Ergebnisse sind in den Bildern 26 und 27 zu begutachten.

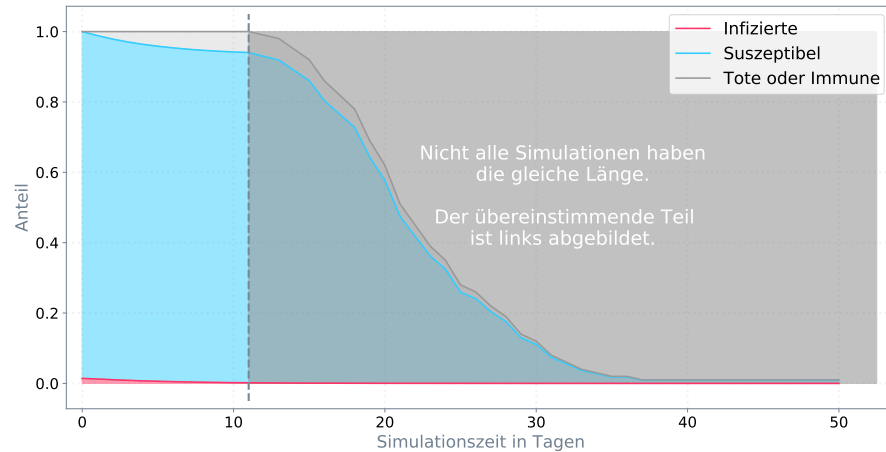
Gemittelte Anteile über die Simulationszeit mit  $p_r = 0,2$  im Basismodell



**Abbildung 26.** Die Anteile der verschiedenen Personenkategorien im Basismodell mit  $p_r = 20\%$ . Gemittelt über 100 Simulationen auf einem Automaten der Größe  $100 \times 100$ .

Für  $p_r = 20\%$  ist  $S_0 > \frac{p_r}{4p_i}$ . Daher muss  $I_t$  Anfangs steigen, bis dieser sein Maximum erreicht. Anschließend muss  $I_t$  gegen 0 laufen. Dieses Verhalten ist in der Abbildung 26 zu sehen. Die restlichen Merkmale sind vorhanden.

Gemittelte Anteile über die Simulationszeit mit  $p_r = 0,6$  im Basismodell



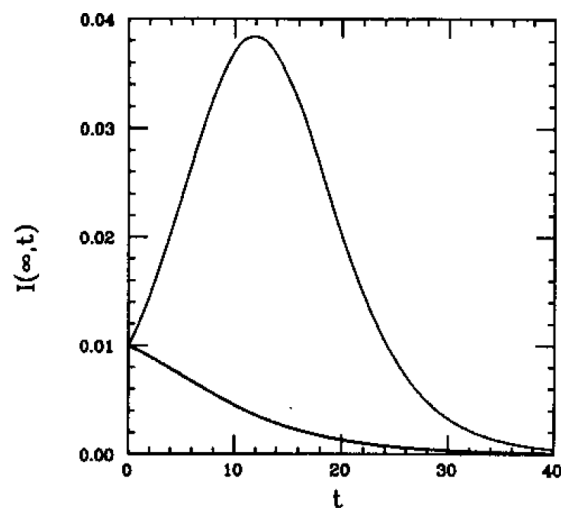
**Abbildung 27.** Die Anteile der verschiedenen Personenkategorien im Basismodell mit  $p_r = 60\%$ . Gemittelt über 100 Simulationen auf einem Automaten der Größe  $100 \times 100$ .

Für  $p_r = 60\%$  ist  $S_0 < \frac{p_r}{4p_i}$ . Daher muss  $I_t$  von Anfang an gegen 0 gehen. Dies ist in der Abbildung 27 zu sehen. Die restlichen Merkmale sind vorhanden.

**Da qualitative alle Merkmale vorhanden sind, ist unser Modell korrekt.**

#### Validierung anhand Boccara und Cheong (1992)

Im Werk von Boccara und Cheong (1992) findet sich auf Seite 2450 die folgende Abbildung 28.



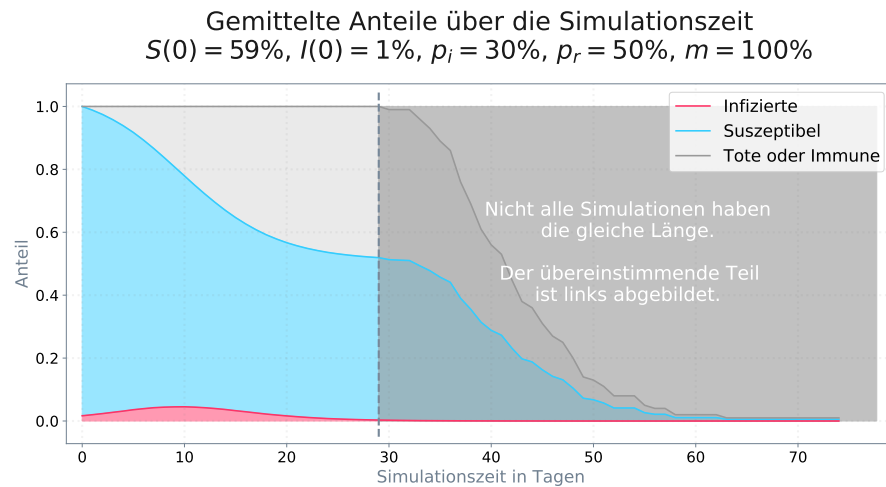
**Abbildung 28.** Auszug der Infizierten Dichteverteilung über die Zeit aus Boccara und Cheong (1992, S. 2450)

Sie zeigt die Dichte der Infizierten über die Zeit. Ist unsere Simulation korrekt, müsste sich im Mittel ein ähnliches Bild für die Parameter darstellen. Die Parameter in Boccara und Cheong (1992) auf Seite 2450 sind wie folgt:

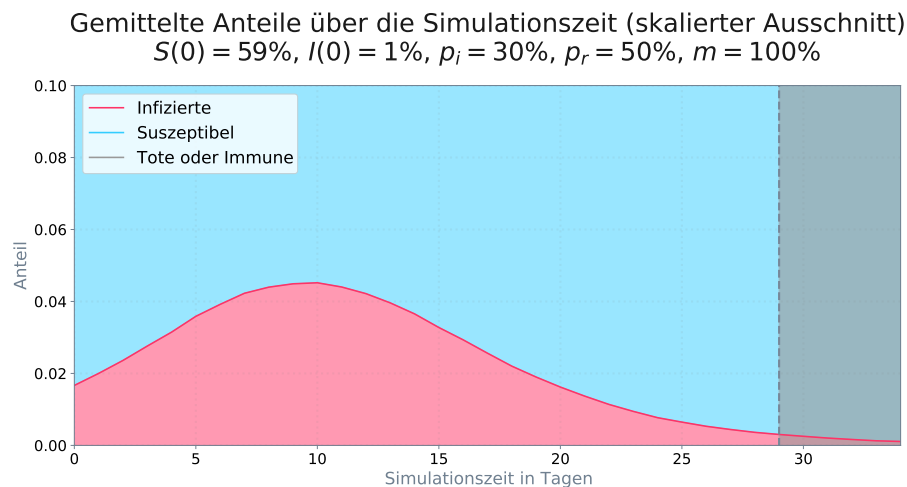
- Suszeptibler Anteil  $S(0) = 59\%$
- Infizierter Anteil  $I(0) = 1\%$
- Infektionsumgebung: Die Manhattan Metrik zusammen mit Radius  $r = 1$

- Infektionswahrscheinlichkeit  $p_i = 30\%$
- Remove-Wahrscheinlichkeit  $p_r = 50\%$  (in Abbildung 28 die obere Kurve) und  $p_r = 75\%$  (in Abbildung 28 die untere Kurve)
- Durchmischungszahl  $m = 100\%$

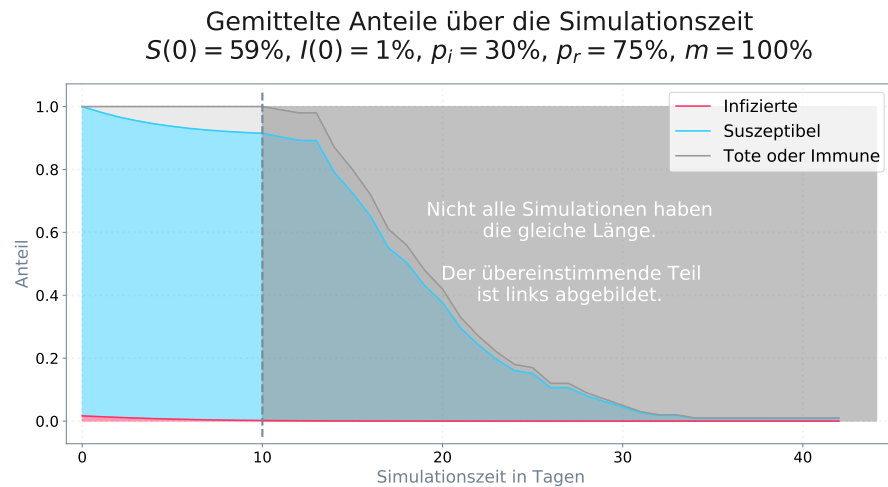
Zur Bestätigung simulieren wir je Testfall ( $p_r$  unterschiedlich) 100 Durchgänge und mitteln die erhaltenen Dichten pro Zeiteinheit. Das Ergebnis besteht aus Abbildung 29 und 31. Diese zeigen die gemittelten Personenanteile über alle 100 Simulationen. Der Vergleich mit Boccarda und Cheong (1992) wird durch skalierte Varianten der Abbildungen in Diagramm 30 und 32 ermöglicht.



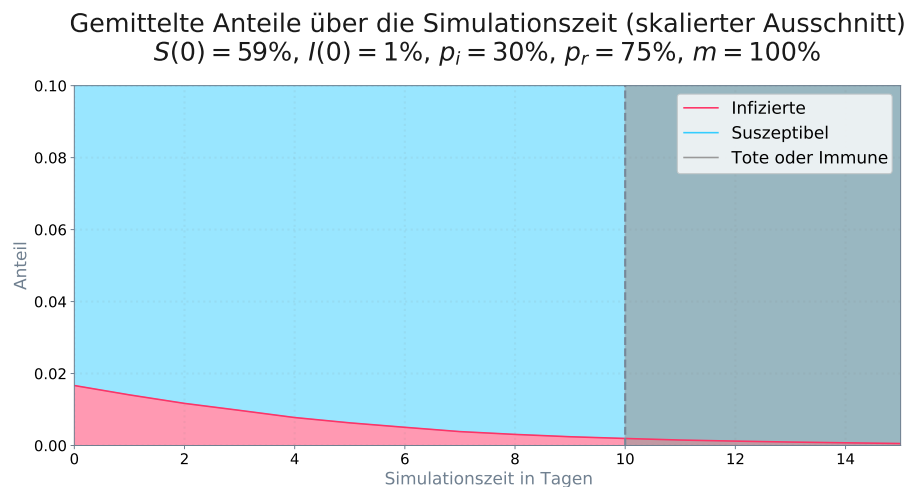
**Abbildung 29.** Die Anteile der verschiedenen Personenkategorien. Nach den Parametern für den ersten Fall aus Boccarda und Cheong (1992, S. 2450). Gemittelt über 100 Simulationen auf einem Automaten der Größe  $100 \times 100$ .



**Abbildung 30.** Diagramm aus Abbildung 29 skaliert



**Abbildung 31.** Die Anteile der verschiedenen Personenkategorien. Nach den Parametern für den zweiten Fall aus Boccarda und Cheong (1992, S. 2450). Gemittelt über 100 Simulationen auf einem Automaten der Größe  $100 \times 100$ .



**Abbildung 32.** Diagramm aus Abbildung 31 skaliert

Die skalierten Abbildungen zeigen eine minimale Abweichung von der Abbildung aus Boccarda und Cheong (1992). Wir gehen daher davon aus, dass unser Basismodell annähernd korrekt ist.

### 3.5 Erweiterungen

Als weitere Aufgaben waren einige Erweiterungen des Basismodells zu implementieren. Die Ergebnisse dieser Erweiterungen werden im Folgenden dargelegt.

#### **Aufteilung in Todesfälle und Immune (Schlosser)**

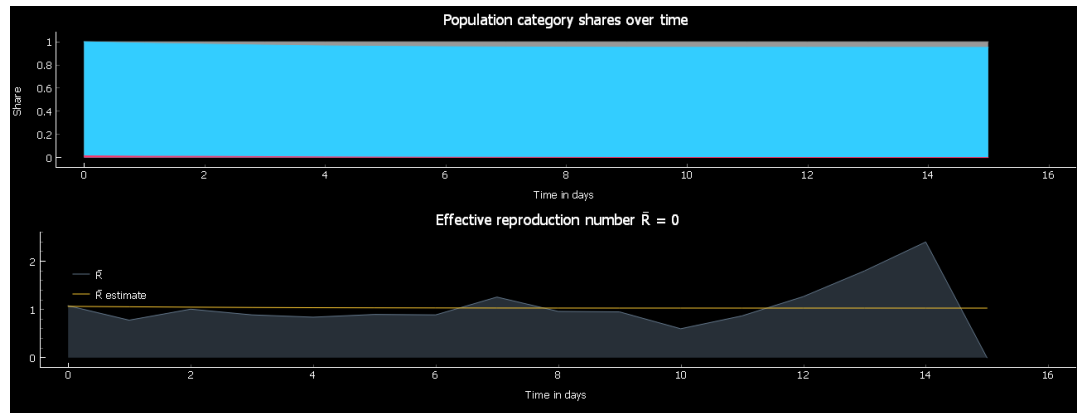
Um die Darstellung der Simulation realitätsnäher zu machen, wurden die Personen, welche den Status *Removed* haben, noch weiter in *Immune* (d.h. genesene Personen) und *Tote* eingeteilt. Die Wahrscheinlichkeit, dass eine Person stirbt ist konfigurierbar, wurde aber standardmäßig ungefähr auf die Todesrate in Deutschland, 3% festgelegt (COVID-19: Fallzahlen in Deutschland und weltweit

2020) (genau: 3,66%). Ein weiterer Unterschied zur Standardsimulation ist, dass im Gegensatz zu entfernten Personen, Genesene sich weiterhin bewegen, was eine Steigerung der zu bewegenden Agenten bewirkt.

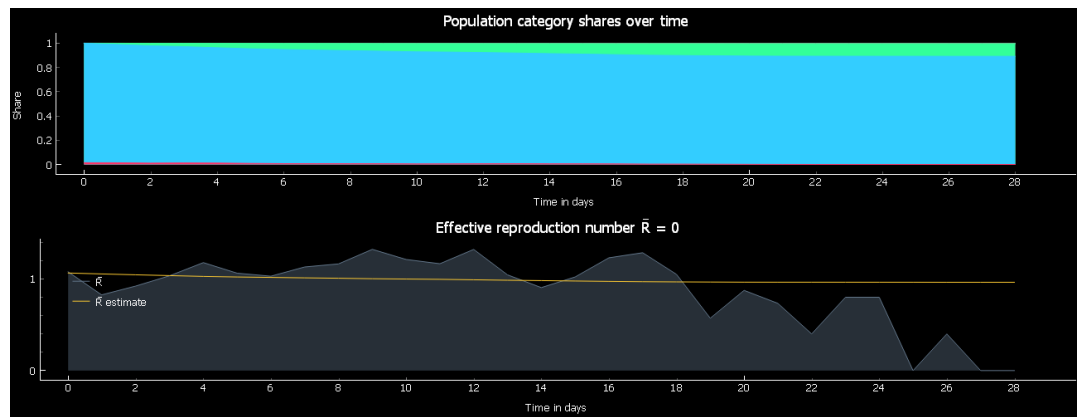
Zum Überprüfen der Erweiterung wurden zwei Testfälle durchgeführt:

Eine Simulation mit den vorgegebenen Standardparametern jeweils mit und ohne Betrachtung der Aufteilung und eine Simulation mit einer durchschnittlichen Krankheitsdauer von 14 Tagen, welche bei Personen mit leichten Symptomen am häufigsten ist (*Covid-19: Krankheitsanzeichen* 2020).

In Abbildung 33 ist der Krankheitsverlauf mit den Standardparametern zu sehen. Zum Vergleich ist in Abbildung 34 der Infektionsverlauf mit der Aufteilung der Personen in Genesene und Gestorbene aufgeführt.



**Abbildung 33.** Anteile der Bevölkerung und R der Standardsimulation

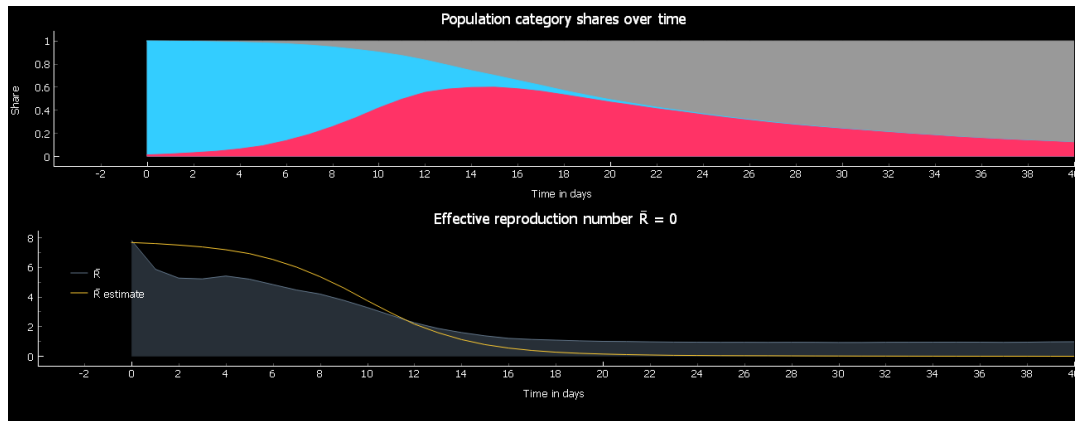


**Abbildung 34.** Anteile der Bevölkerung und R der Simulation mit Aufteilung der Todesfälle und Immune

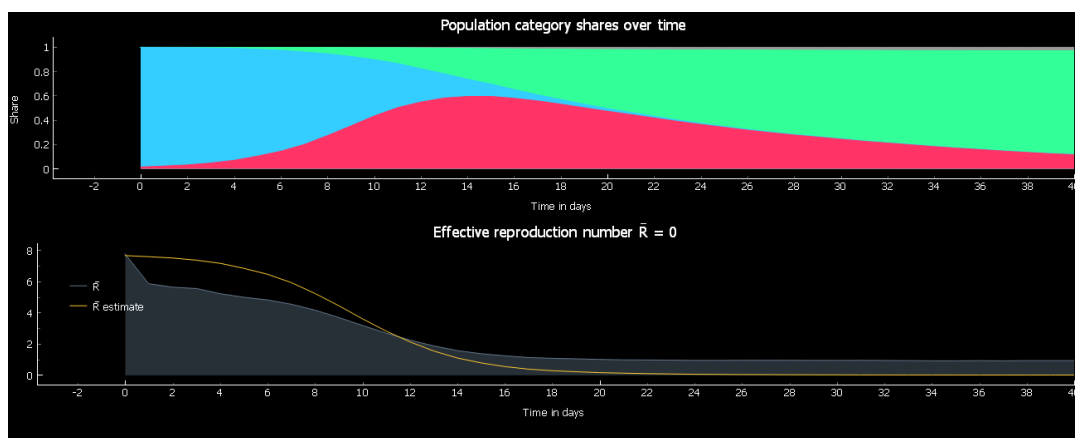
Es wurden in der Standardsimulation mit einer Feldgröße von 100 mal 100 767 "entfernte" Personen mit einer Infektionsdauer von 15 Tagen beobachtet, in der Simulation der Todesfälle mit einer Sterblichkeitsrate von 3% wurden 726 Immune und 29 Tote gemessen. Damit betrug die tatsächliche Todesrate der Simulation 3,9% und liegt damit sehr nahe an der eingestellten Wahrscheinlichkeit. Der Unterschied der absoluten Anzahl infizierter Personen liegt an der zufälligen Natur der Simulation, weswegen diese nicht eins zu eins die gleichen Zahlen ergeben.

Die Aussagekraft der Untersuchungen kann durch direkten Vergleich gesteigert werden. Daher wurde auch eine Simulation mit veränderter Krankheitsdauer durchgeführt. Diese Ergebnisse sind in Abbildung 36 und 36 zu sehen.





**Abbildung 35.** Anteile der Bevölkerung und  $\bar{R}$  der Standardsimulation



**Abbildung 36.** Anteile der Bevölkerung und  $\bar{R}$  der Simulation mit Aufteilung der Todesfälle und Immune

Hier wurden in der Standardsimulation 6994 entfernte Personen gezählt, die Infektion hatte eine Gesamtdauer von 138 Tagen.

In der Simulation mit der Aufteilung der entfernten Personen wurden 6785 Immune und 221 Tote gezählt, bei einer Gesamtdauer von 129 Tagen. Damit betrug die gemessene Sterberate 3,2% und liegt ebenfalls innerhalb der Toleranzgrenzen.

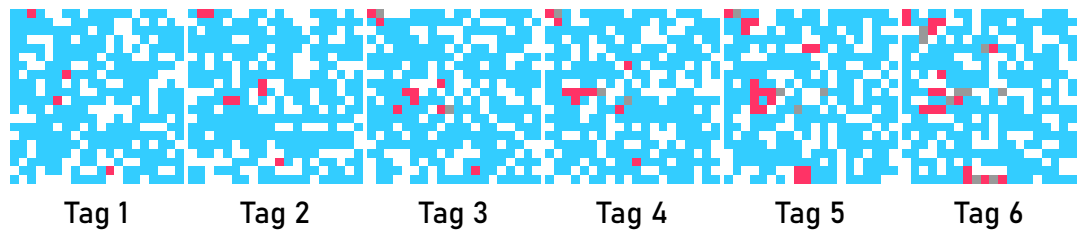
Abschließend ist zu sagen, dass eine Einteilung in Immune und Tote die Krankheitsdauer und Gesamtanzahl der Infizierten nicht beeinflusst, sondern lediglich eine Steigerung des Informationsgehalts der Simulation bietet. Diese Zahlen sind allerdings sehr interessant, weswegen es eine lohnenswerter Erweiterung ist. Es kann davon ausgegangen werden, dass die meisten Personen, welche an der Infektion sterben, zumindest zeitweise einen Platz der Intensivpflege benötigen. Somit kann eine begrenzte Aussage über die Belastung des Gesundheitssystems getroffen werden.

### **Beschränkter Bewegungsradius (Eder)**

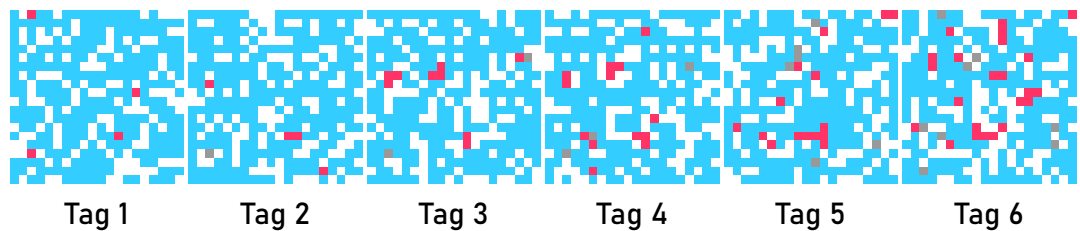
Wie bereits in Teil 2.3 erwähnt, entsprechen die Bewegungen eher „Teleportationen“ im Rahmen des zweidimensionalen Zellularautomaten. Das bedeutet, dass diese lediglich durch die Ergebnisse des verwendeten Zufallsgenerators bestimmt sind. Um stattdessen ein sinnvolles Bewegungsmuster in die Simulation zu pflegen, entschieden wir uns für die in Abschnitt 2.4 geschilderte Benutzerschnittstelle.

*Zur Wiederholung:* Die Umgebung, in der sich eine Person bewegen kann, ist einstellbar (d. h. Radius und Distanzmetrik). Des Weiteren kann mit einer Checkbox entschieden werden, ob sich Personen bevorzugt in kleinem Radius bewegen.

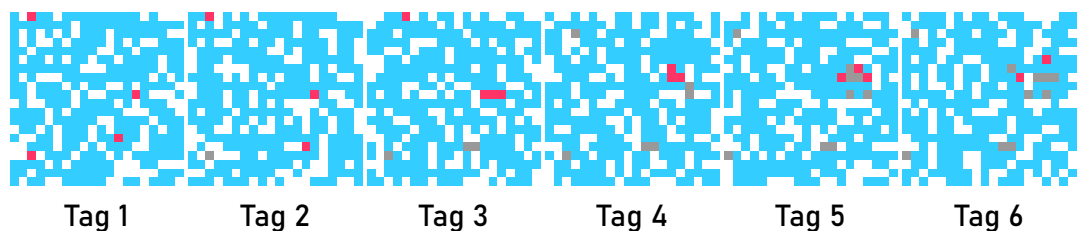
Nun möchten wir die Erweiterung in die Simulationslogik einfließen lassen und daraus gewonnene Ergebnisse darstellen und interpretieren. Als ersten Schritt wollen wir einen Überblick über das geänderte Verhalten gewinnen auf einem Automaten mit der Größe  $20 \times 20$ , einem Anteil susceptibler Personen von 69%, einem Anteil infizierter Personen von 1%, einer Durchmischungszahl  $m = 50\%$ , der Infektionswahrscheinlichkeit  $p_i = 20\%$  sowie der Remove-Wahrscheinlichkeit  $p_r = 20\%$  und 60%. Die Umgebung wird mit dem Radius 2 und der euklidischen Distanzmetrik bestimmt. Auf die Aktivierung der Checkbox für die Bevorzugung von kurzen Distanzen wird vorerst noch verzichtet.



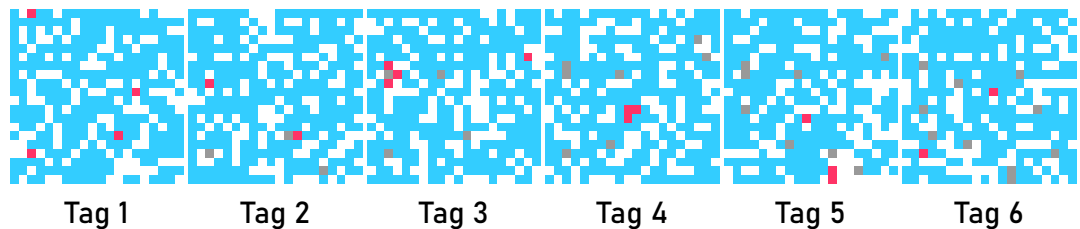
**Abbildung 37.** Beschränkter Bewegungsradius - Überblick mit  $p_r = 0,2$



**Abbildung 38.** Unbeschränkte Bewegung - Überblick mit  $p_r = 0,2$



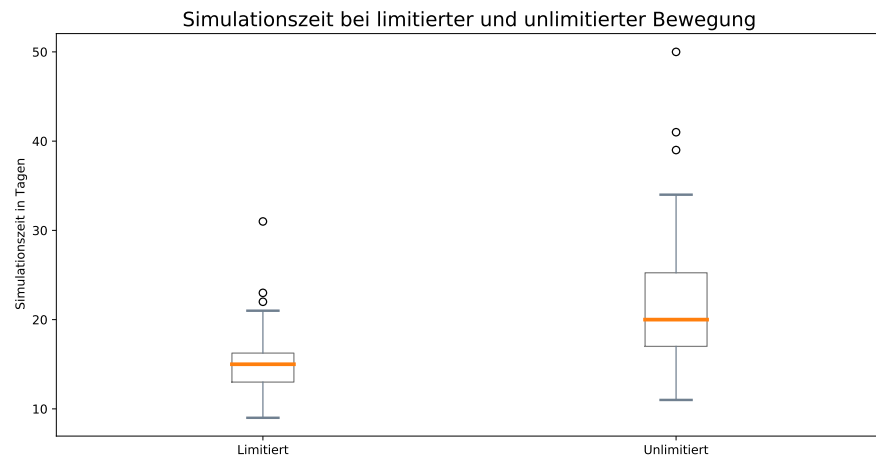
**Abbildung 39.** Beschränkter Bewegungsradius - Überblick mit  $p_r = 0,6$



**Abbildung 40.** Unbeschränkte Bewegung - Überblick mit  $p_r = 0,6$

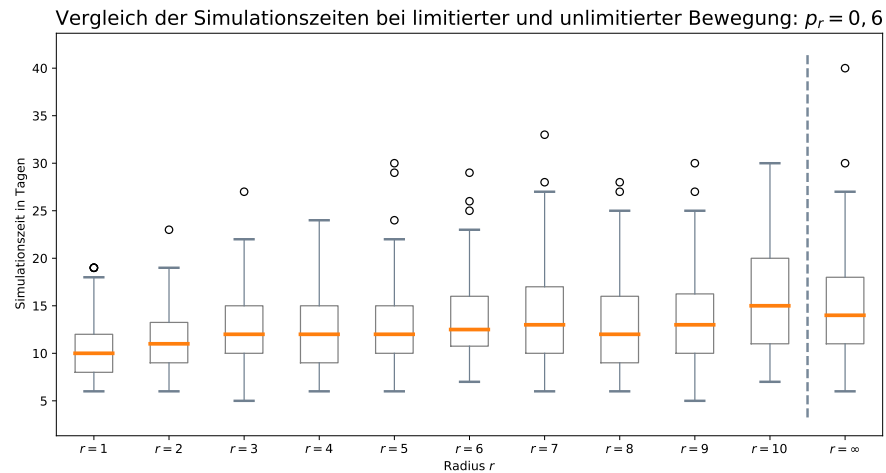
Abbildung 37 und 38 zeigen den Verlauf der Simulation über 6 Tage, einmal mit der Bewegungs-limitierung, einmal ohne. Die Ausbreitung der Infektion ist lokaler, relativ gesehen zu den Startpo-sitionen der ersten Infizierten, als bei der unbeschränkten Bewegung. In der Anzahl der Infizierten ist allerdings in diesem Beispiel kein großer Unterschied zu sehen.

Im Vergleich dazu merkt man bei  $p_r = 0,6$ , dass die Simulation in beiden Szenarien (mit oder ohne Bewegungsbeschränkung) sehr schnell vorbei ist. Bei der limitierten Bewegung merkt man allerdings, dass die Infizierten sich nicht schnell genug von den Immunen oder Toten wegbewe-gen können und daher für die Infizierten weniger Neuinfektionen in direkter Umgebung möglich sind. Durch die hohe Remove-Wahrscheinlichkeit sind in der Regel die Simulationen auch schnel-ler vorbei. Um das zu bestätigen, führen wir diesmal mit einem größeren 100x100 Automaten 100 Simulationen mit den gegebenen Parametern durch und visualisieren die jeweiligen Simulations-zeiten (in Tagen).



**Abbildung 41.** Die Simulationszeiten von beschränkter zu unbeschränkter Bewegung mit sonst gleichen Pa-rametern im Vergleich

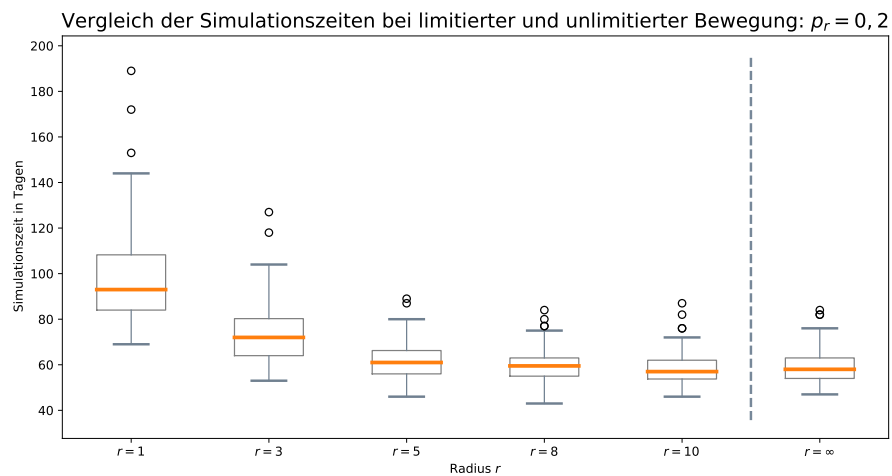
Wie man im Darstellung 41 zweifelsohne erkennen kann, sind die Simulationszeiten bei limi-tierter Bewegung und  $p_r = 60\%$  deutlich geringer. Genauere Auswirkungen für verschiedene Ra-dien werden im Folgenden untersucht. Zur Reduzierung der Laufzeit der Simulationen verringern wir die Größe des Automaten auf 50x50 und messen wieder 100-mal pro Radius von 1 bis 10. Das Ergebnis ist in Abbildung 42 zu sehen.



**Abbildung 42.** Vergleich der Simulationszeiten für verschiedene Radien für die Bewegungseinschränkung mit  $p_r = 60\%$

Wie in der ersten Abbildung angedeutet, ist eine leichte Steigerung der Simulationsdauer zu erkennen. Interessanterweise ist die Variante mit Radius 10 sogar im Mittel länger gelaufen als die unlimitierte Version.

Weiter fragen wir uns welchen Einfluss die Remove-Wahrscheinlichkeit auf dieses Verhalten hat. Daher verringern wir diese auf  $p_r = 20\%$  und lassen das Experiment erneut durchlaufen. Die Simulationen resultieren diesmal in Darstellung 43.

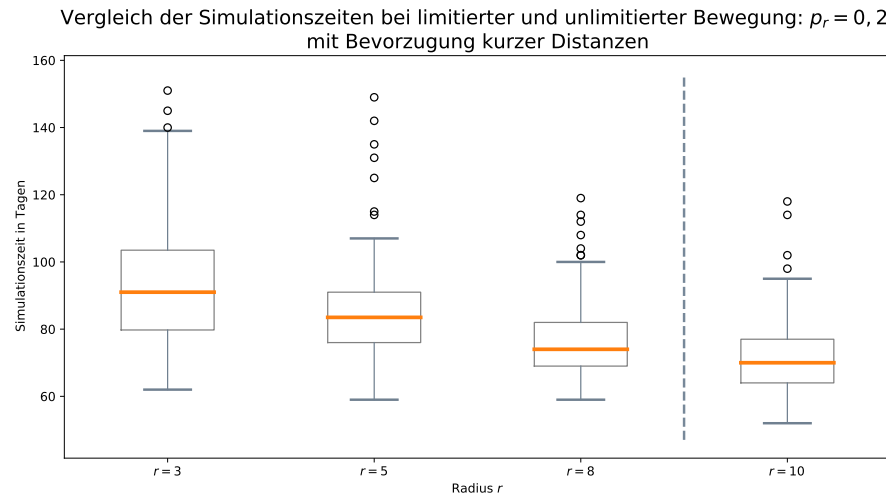


**Abbildung 43.** Vergleich der Simulationszeiten für verschiedene Radien für die Bewegungseinschränkung mit  $p_r = 20\%$

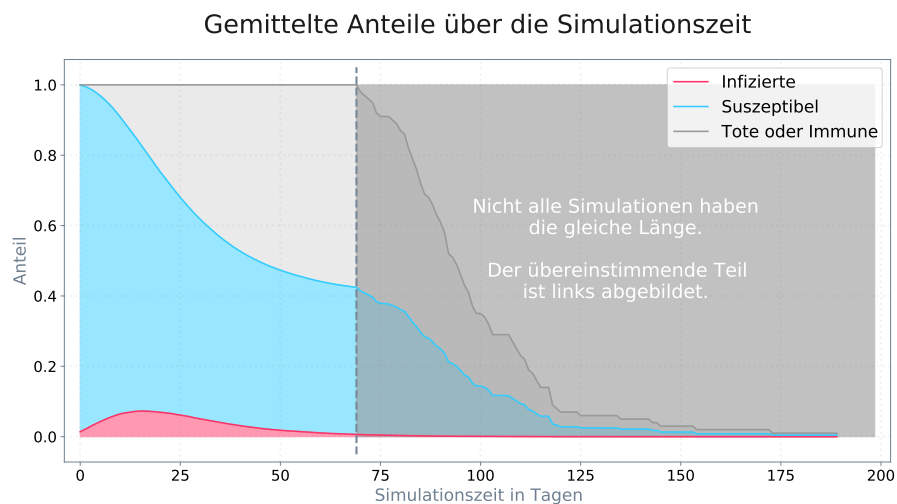
Diesmal ist die Situation umgekehrt und die Simulationszeit sinkt mit steigendem Radius. Unsere Vermutung ist, dass die höhere Krankheitsdauer – durch die niedrigere Remove-Wahrscheinlichkeit – zusammen mit kleinen Radien eine langsamere Ausbreitung gewährleistet. Bei großen Radien oder unlimitierter Bewegung kommen mehr anfällige Personen mit den Infizierten in Kontakt. Das führt zu einer schnelleren Ausbreitung, was unter anderem im Anteilsverlauf der Infizierten in Ab-

bildung 46 verglichen mit 45 sichtbar wird.

Als Abschluss für diese Erweiterung wollen wir die Wirkung der Checkbox zur Bevorzugung kleinerer Distanzen präsentieren. In Diagramm 44 wird mit 20-%iger Remove-Wahrscheinlichkeit, verschiedenen Radien und aktivierter Checkbox die Simulation 100-fach auf einem 50x50 Automaten wiederholt. Das Ergebnis ist ähnlich zur vorherigen Abbildung. Jedoch werden kleinere Radien bevorzugt, weshalb die Simulationszeiten bei höheren Radien im Vergleich zu Abbildung 43 etwas höher ausfallen.

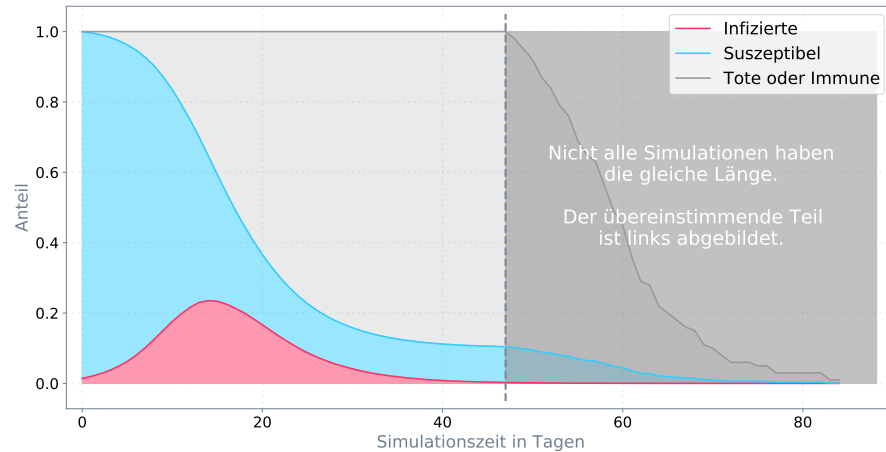


**Abbildung 44.** Vergleich der Simulationszeiten für verschiedene Radien für die Bewegungseinschränkung mit  $p_r = 20\%$  und Bevorzugung von kurzen Distanzen



**Abbildung 45.** Anteile der verschiedenen Personengruppen über die Zeit. Gemittelt über 100 Iterationen der Simulation auf einem 50x50 Feld mit  $p_r = 20\%$  und *beschränkter* Bewegung mit Radius  $r = 1$ .

### Gemittelte Anteile über die Simulationszeit



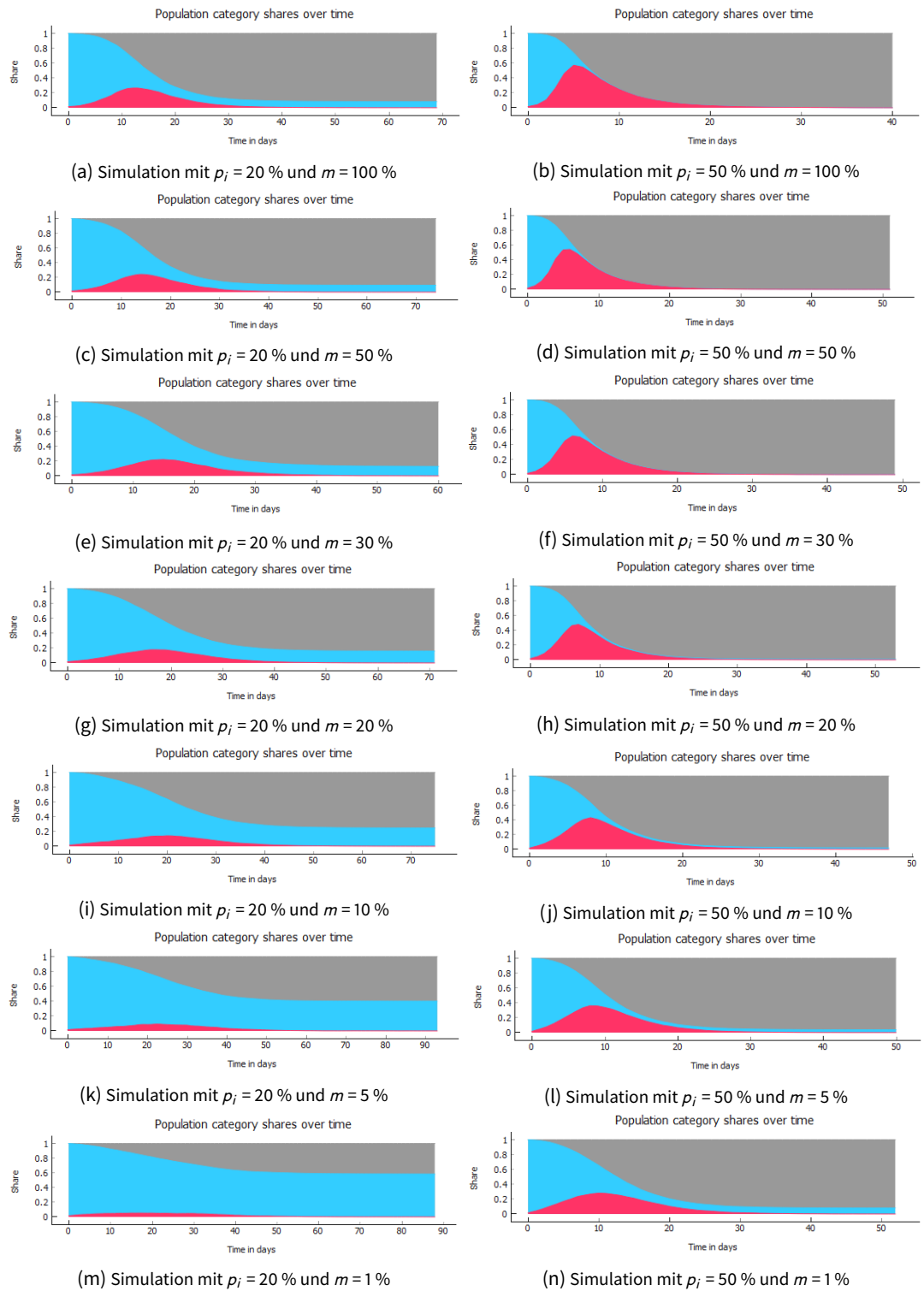
**Abbildung 46.** Anteile der verschiedenen Personengruppen über die Zeit. Gemittelt über 100 Iterationen der Simulation auf einem 50x50 Feld mit  $p_r = 20\%$  und *unbeschränkter* Bewegung.

#### Ausgangssperre (Stiglmeier)

Zur Untersuchung des Einflusses einer Ausgangssperre wird als Basiskonfiguration das Beispiel der zusätzlichen Untersuchung aus Kapitel 3.2 verwendet. Also  $S(0) = 69\%$ ,  $I(0) = 1\%$ ,  $p_i = 20\%$  und  $p_r = 20\%$ . Um eine Ausgangssperre zu modellieren, kann in diesem Modell der Durchmischungsgrad der Agenten verwendet werden, welcher durch den Parameter *Movement mixing* (kurz  $m$ ) realisiert ist. Hierbei wird nur der angegebene Teil der Agenten während des Bewegungsevents bewegt. Im Folgenden wurden verschiedene Simulationen mit jeweils unterschiedlichem Durchmischungsgrad durchgeführt. Desweiteren wurden Simulationen mit einer Infektionswahrscheinlichkeit  $p_i = 50\%$  durchgeführt um herauszufinden, ob eine Ausgangssperre bei höherem Infektionsgrad eine größere Rolle spielen kann.

Diese Simulationen sind in Abbildung 47 dargestellt. Geordnet sind die Simulationen auf der linken Seite mit  $p_i = 20\%$  und auf der Rechten Seite mit  $p_i = 50\%$  und von oben nach unten mit geringer werdendem Durchmischungsgrad. Qualitativ ist zu erkennen, dass die ersten Simulationen mit einer Durchmischung  $m$  von 100 %, 50 % und 30 % noch sehr ähnlich aussuchen. Erst ab etwa 20 % macht sich ein Unterschied zur ersten Simulation mit 100 % bemerkbar. Die Streckung der Infektionskurve bei verringerter Durchmischung ist über alle Simulationsschritte zu erkennen. Die Kurve ist jedes mal einige wenige Tage länger. Der Vergleich der Infektionsraten von 20 % auf der linken und 50 % auf der rechten Seite zeigt, dass die Ausgangssperre bei einer hohen Infektionsrate relativ gesehen weniger Unterschied macht. Zu sehen ist dies im Vergleich der Durchmischung von 100 % und 1 %. Die linke Seite ist um einen großen Faktor kleiner geworden während die rechte Seite mit der hohen Infektionsrate nur etwa halbiert wurde.

Werden nun die Infektionsanteile am Maximum der Infektionskurven miteinander verglichen (Vgl. Tabelle 3) ist zu erkennen, dass bei beiden Simulationen mit  $p_i = 20\%$  und 50% erst ab  $m = 10\%$  der maximale Anteil an Infizierten um mehr als 10 % gesunken ist. Weiterhin ist zu erkennen, dass bei einer größtmöglichen Ausgangssperre von  $m = 1\%$  beide Simulationen beim maximalen Anteil an Infizierten um etwa 20 % gesunken sind. Das heißt die Infektionswahrscheinlichkeit der Krankheit hat bei Umsetzung der Ausgangssperre einen zweitrangigen Einfluss. Eine Untersuchung von



**Abbildung 47.** Simulationen mit unterschiedlichem Durchmischungsgrad und Infektionswahrscheinlichkeit

einer geringeren Infektionswahrscheinlichkeit wie beispielsweise  $p_i = 5\%$  führt zu keinem brauchbaren Ergebnis, da dies in der Konfiguration ein Aussterben der Krankheit während der Simulation verursacht. Um das zu verhindern müsste die Entfernungswahrscheinlichkeit  $p_r = 20\%$  auch verringert werden, was dann wiederum zu einem gleichen Ergebnis wie bei  $p_i = 20\%$  führt mit dem einzigen Unterschied, dass sich die Simulationsdauer verlängert hat, qualitativ aber die gleiche Kurve entsteht.

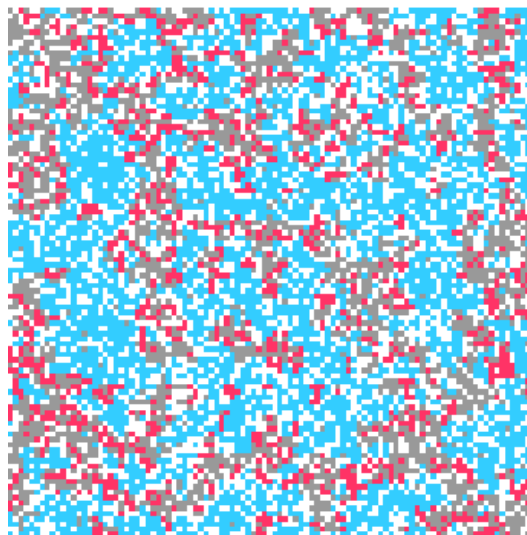
$p_i$	$m = 100\%$	$m = 50\%$	$m = 30\%$	$m = 20\%$	$m = 10\%$	$m = 5\%$	$m = 1\%$
20 %	26,5 %	24,2 %	22,2 %	17,7 %	14,3 %	9,1 %	5,2 %
50 %	57,2 %	54,2 %	51,7 %	48,0 %	43,2 %	36,1 %	28,1 %

**Tabelle 3.** Maximaler Anteil an Infizierten bei unterschiedlicher Ausgangssperre

Eine weitere Möglichkeit die Ausgangssperre in diesem Modell zu simulieren ist mit dem limitierten Bewegungsradius aus Kapitel 3.5. Also dass beispielsweise eine Person, die die Ausgangssperre missachtet nicht mehr mit dem Nahverkehr quer durch eine ganze Stadt in die Arbeit fährt, sondern nur noch limitiert bei einem Markt in der Nähe Lebensmittel einkauft.

Werden nun die Ausgangssperre mit dem Durchmischungsgrad  $m = 1\%$  und einem limitierten Bewegungsradius und einem euklidischen Radius von einer Einheit bei Infektionswahrscheinlichkeit von  $p_i = 50\%$  simuliert, dann lässt sich das Maximum der Infektionskurve auf 21,6 % reduzieren, was nochmals ein deutlicher Unterschied zu dem zuvor erreichten kleinsten maximalen Anteil von 28,1 % aus Tabelle 3 ist.

Zur Validierung der Erweiterung wurde ein sehr geringer Durchmischungsgrad von 1 % ausgewählt und simuliert (Vgl. Abbildung 48). Zu sehen ist, dass sich das Infektionsmuster nicht zufällig bildet, sondern eine regionale Ausbreitung aufweist. Damit gilt diese Erweiterung als validiert.



**Abbildung 48.** Snapshot einer Simulation mit einer Durchmischung von 1 % zur Darstellung des regionalen Ausbreitungsmusters

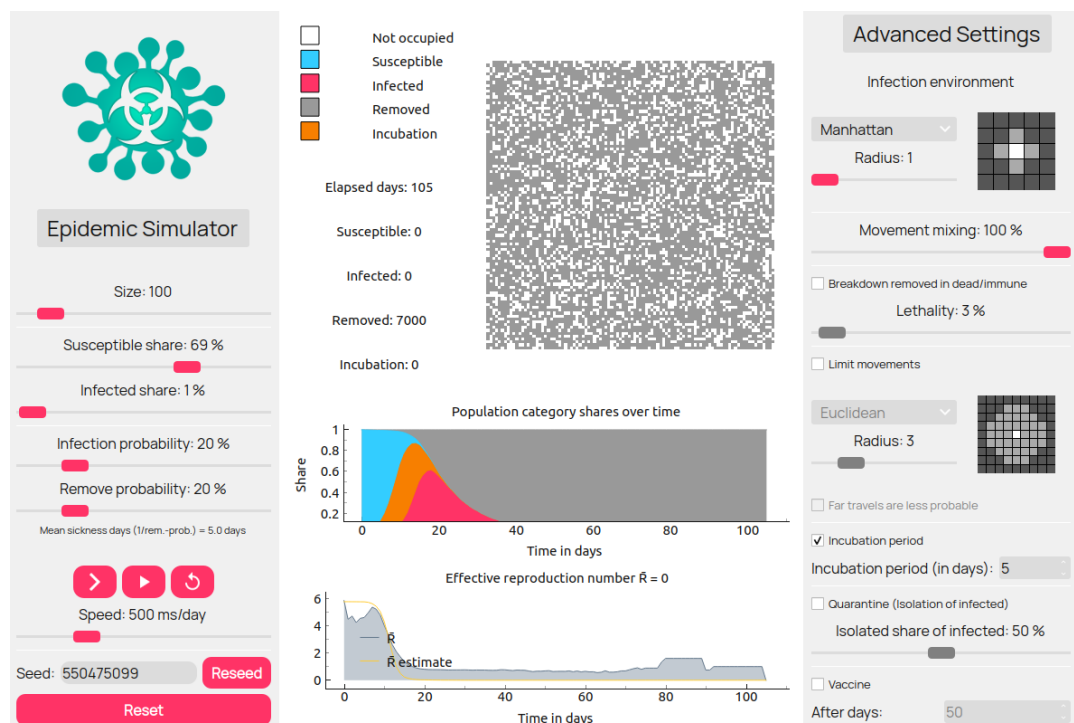
#### Inkubationszeit und Krankheitsdauer (Beil)

Das im Abschnitt 3.2 beschriebene Basismodell berücksichtigt nur die Krankheitsdauer. Die Krankheitsdauer wird durch  $p_r$  der Entfernungswahrscheinlichkeit festgelegt. Die mittlere Krankheitsdauer berechnet sich aus  $\frac{1}{p_r}$ . Für ein  $p_r$  von 0,5 ergibt sich eine mittlere Krankheitsdauer von 2 Ta-



gen. In der wirklichen Welt, gibt es eine Inkubationszeit. Das ist die Zeit zwischen der Ansteckung und den ersten Symptomen. In der Inkubationszeit sind die Personen schon ansteckend, können aber nicht immun werden oder sterben. Dieses Verhalten wird in dieser Modelerweiterung untersucht. Dazu wird der Status „Incubation“ eingeführt. Nun erhält jeder Agent bei einer neuen Infektion den Status „Incubation“ anstatt den Status „Infected“. Hat ein Agent den Status „Incubation“, dann kann er Agenten im Infektionsbereich anstecken. Diese erhalten Anfangs den Status „Incubation“. Ist die Inkubationszeit eines Agentens abgelaufen, dann ändert sich sein Status selbständig auf „Infected“. Agenten mit dem Status „Infected“ werden wie die Agenten aus dem Basismodell mit dem Status „Infected“ behandelt. Die Inkubationszeit kann über einen Schieberegler eingestellt werden. Dieses Verhalten wird anhand eines Beispiels erläutert. Das Beispiel besitzt folgende Konfigurationen:

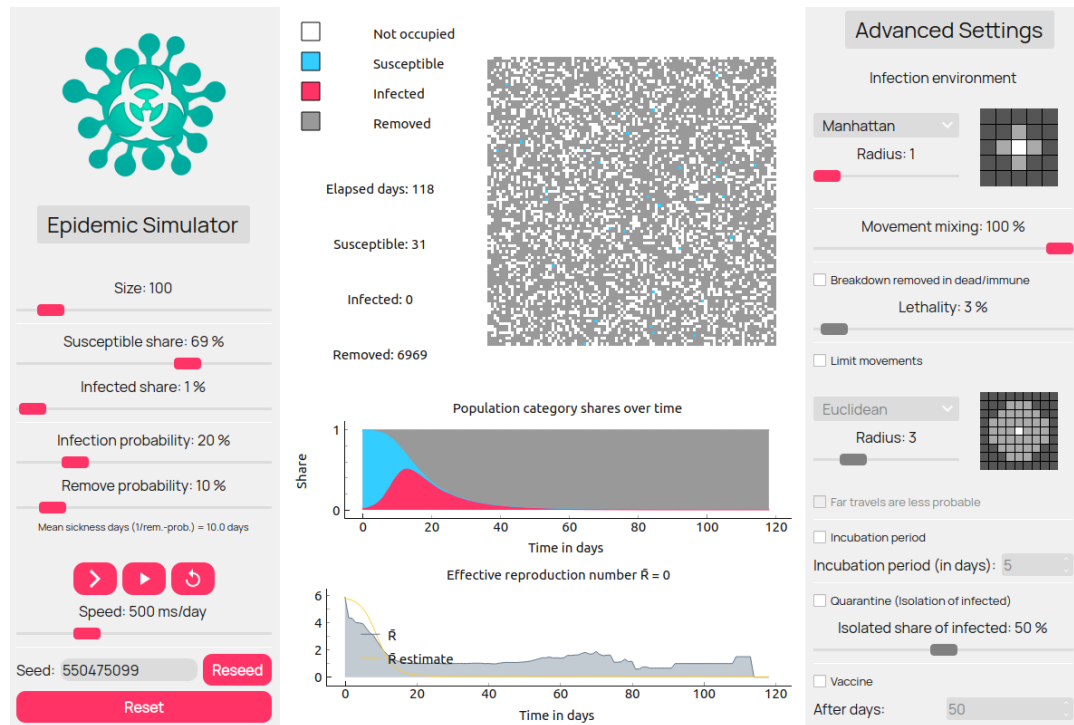
- Size  $L = 100$
- Susceptible share  $S(0) = 69\%$
- Infected share  $I(0) = 1\%$
- Infection probability  $p_i = 20\%$
- Remove probability  $p_r = 20\% \Rightarrow$  Krankheitsdauer von 5 Tagen
- Movement mixing  $m = 100\%$
- Infection environment = Manhattan mit einem Radius von 1
- Keine Bewegungseinschränkung
- Keine Impfung
- Inkubationszeit von 5 Tagen
- Keine Quarantäne
- Keine Unterscheidung von Immunen und Tote
- Seed = 550475099



**Abbildung 49.** Verlauf der Simulation mit den oben beschriebenen Einstellungen

Im Vergleich zum Basismodell mit  $p_r = 20\%$ , ist das Maximum an Infizierten (Agenten mit dem Status „Infected“ und „Incubation“) deutlich höher. Dieses liegt bei 6076 Personen am Tag 14 im Ver-

gleich zu 1794 beim Basismodell. Dieses Verhalten entsteht durch die Inkubation. Das Wachstum der Infizierten bremsen sich selbst aus, da umso mehr Infizierte entfernt werden, umso mehr Infizierte es gibt. Diese Bremswirkung ist bei den Agenten mit dem Status „Incubation“ nicht vorhanden, weswegen Sie stärker anwachsen können. Eine Inkubationszeit ist damit nicht vergleichbar mit einer längeren Krankheitsdauer, da dort der Bremseffekt vorhanden ist. Zur Veranschaulichung wurde der Verlauf des Basismodells mit  $p_r = 10\%$  in der Abbildung 50 dargestellt. Des weiteren fällt auf, dass durch die Inkubationszeit die Simulationsdauer von 58 auf 105 Tage steigt. Durch die Inkubationszeit überleben Infizierte viel länger. Da die Simulation erst beendet wird, wenn es keine Infizierten mehr gibt, dauert die Simulation dadurch länger.

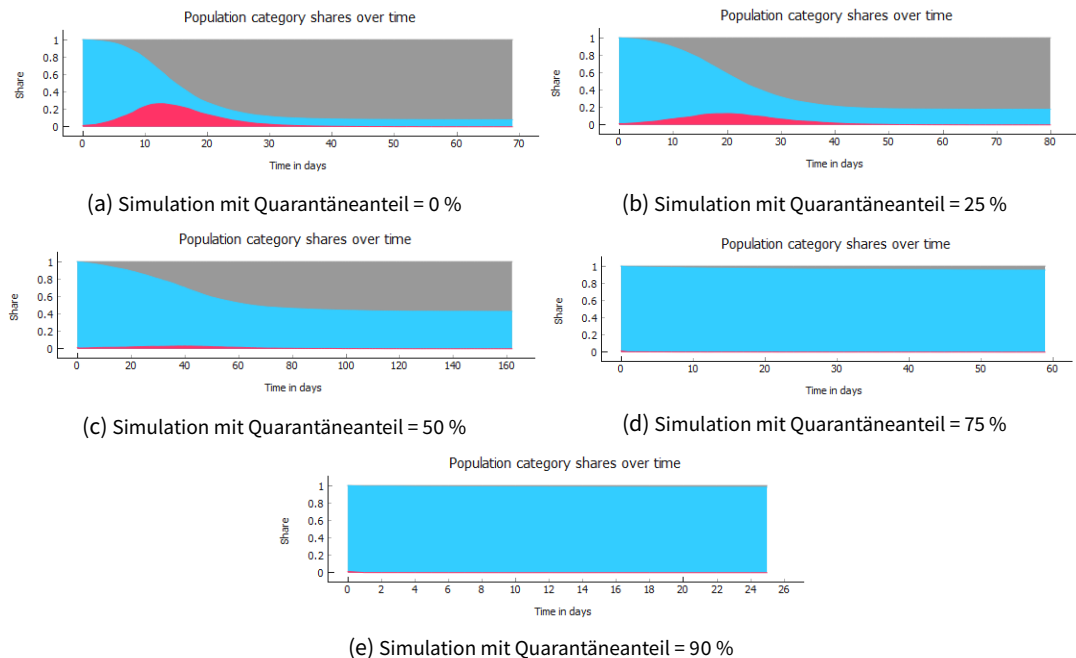


**Abbildung 50.** Verlauf mit  $p_r = 10\%$  aber ohne Inkubationszeit zum Vergleich

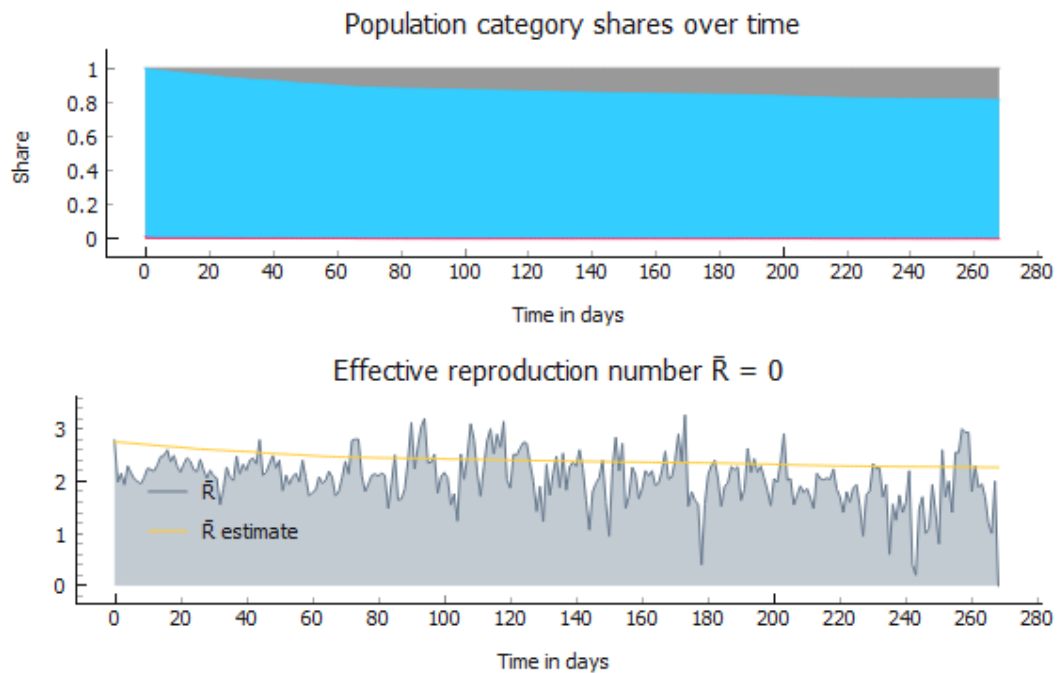
### Quarantäne (Stiglmeier)

Für die Umsetzung der Quarantäne wurde ein Algorithmus mit der folgenden Logik implementiert. Ist ein Agent infiziert, dann wird dieser aus dem Feld entfernt und in eine Quarantäne gesetzt. Wenn der Agent in der Quarantäne nicht mehr infiziert ist, als entweder tot, gesund oder entfernt ist, dann wird dieser wieder in das Feld eingefügt. Aufgrund der Aufrufreihenfolge der einzelnen Strategien (Siehe Kapitel 3.1 kann ein infizierter Agent keine umliegenden Agenten anstecken, da die Quarantäne-Strategie vor der Infektions-Strategie aufgerufen wird. Bevor ein Agent allerdings in Quarantäne gesetzt wird, wird erst überprüft, ob der Anteil der in Quarantäne sitzenden Agenten im Verhältnis mit den infizierten und freien Agenten dem im User Interface einstellbaren Parameter *Isolated share of infected* entspricht. Dadurch kann ein infizierter Agent auf dem Feld bleiben und weitere Agenten anstecken. Im Fall, dass im Feld durch die Entfernungswahrscheinlichkeit mehrere Agenten entfernt wurden und dann zu viele Agenten in Quarantäne sitzen, bleiben diese in Quarantäne. Es wird also nicht versucht, das Verhältnis von Infizierten zu in Quarantäne sitzenden Agenten in diesem Fall zu wahren. Weiterhin ist die Dauer der Simulation verlängert bis wieder alle Agenten zurück im Feld sind.

Zur Simulation wird als Basiskonfiguration das Beispiel der zusätzlichen Untersuchung aus Kapitel 3.2 verwendet. Also  $S(0) = 69\%$ ,  $I(0) = 1\%$ ,  $p_i = 20\%$  und  $p_r = 20\%$ . Es werden verschiedene Simulationen mit einem jeweils unterschiedlichem Quarantäneanteil durchgeführt. Diese sind in Abbildung 51 dargestellt. Es ist zu sehen, dass bereits ein geringer Quarantäneanteil von 25 % zu einer deutlichen Abflachung der Infektionskurve führt. Bei einem Quarantäneanteil von 50 % ist die Kurve bereits fast nicht mehr vorhanden und bei einem Anteil von 75 % hat sich der Anteil der entfernten Agenten ebenfalls stark reduziert. Ein Quarantäneanteil von 90 % lässt die entfernten Agenten für das Auge nicht mehr erkennen. Ebenfalls auffallend ist die deutlich längere Simulationsdauer bei einem Quarantäneanteil von 50 %. Hier dauerte die Simulation über 160 Tage an während der Rest im Schnitt bei etwa 60-80 Tagen liegt. Genauer betrachtet scheint es in dem Simulationsmodell einen optimale Einstellung zu geben, bei der sich die Neuinfektionen und Quarantänesetzungen die Waage halten und dafür sorgen, dass das Modell möglichst lange läuft. Diese Einstellung liegt etwa bei einem Quarantäneanteil von 65 %. Hier dauert die Simulation 268 Tage (Vgl. Abbildung 52). Während die Reproduktionszahl  $R$  pro Tag in der Regel normal verläuft, lässt sich bei Annäherung des Simulationsmodells and diesen optimalen Punkt bei dem Diagramm der Reproduktionszahl eine neue Beobachtung aufstellen. Wie in Abbildung 52 zu sehen, alterniert die Reproduktionszahl dann sehr stark zwischen wenig und viel Neuinfektionen. Normalerweise ist dies deutlich gleichmäßiger (Vgl. Abbildung 19).



**Abbildung 51.** Simulationen mit unterschiedlichem Quarantäneanteil



**Abbildung 52.** Quarantänesimulation mit Anteil 65 % und Dauer 268 Tage

Eine Betrachtung der Quarantäne von allen Infizierten führt zu keiner brauchbaren, neuen Erkenntnis. Alle zu Beginn infizierten Agenten, also 1 %, werden zu Beginn in Quarantäne gesetzt. Anschließend ist die Simulationsdauer lediglich davon abhängig, wann bei jedem Agenten in Quarantäne das Ereignis zur Entfernung eingetreten ist.

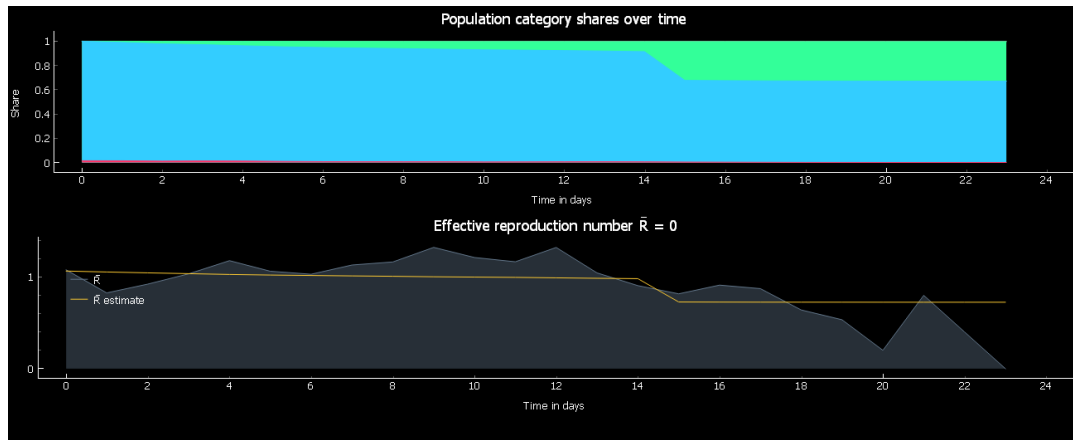
Zur Validierung des Modells wurden mehrere Simulationen mit unterschiedlichem Quarantäneanteil inklusive Maximalwerten durchgeführt. Bei allen Simulationen geht kein Agent verloren und die Simulationen weisen einen gleichen Verlauf wie das Basismodell bei Quarantäneanteil = 0 % auf. Bei steigendem Quarantäneanteil sinken die Infektions- und Entfernungszahlen stetig (Vgl. Abbildung 51). Damit gilt diese Erweiterung als validiert.

### **Impfstoff (Schlosser)**

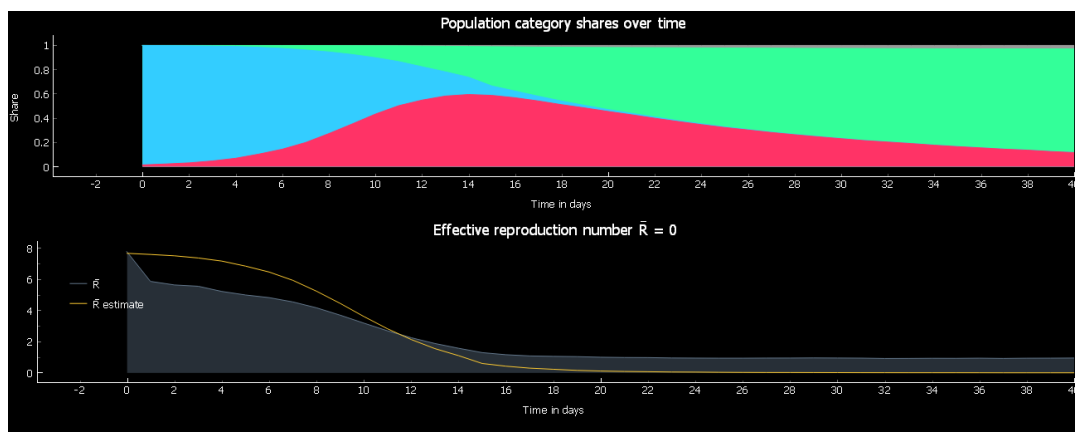
Um einen Impfstoff zu simulieren, wurde ein Algorithmus implementiert, der zu einem gewissen Zeitpunkt,  $t_{impf}$ , einen Prozentsatz der gesunden Bevölkerung,  $p_{impf}$ , immun gegen die Infektion macht.

Für diese Betrachtung wurden analog zu Kapitel 3.5 zwei Szenarien betrachtet: Ein Szenario mit den Standardparametern und eines mit einer durchschnittlichen Krankheitsdauer von 14 Tagen.

Die Ergebnisse dieser Simulationen mit den Parametern  $t_{impf} = 15$  und  $p_{impf} = 25\%$  sind in den folgenden Abbildungen zu finden. (Die Standardsimulation ist in Abbildungen 33 und 35 zu finden)



**Abbildung 53.** Anteile der Bevölkerung und  $\bar{R}$  der Impfsimulation mit Standardparametern



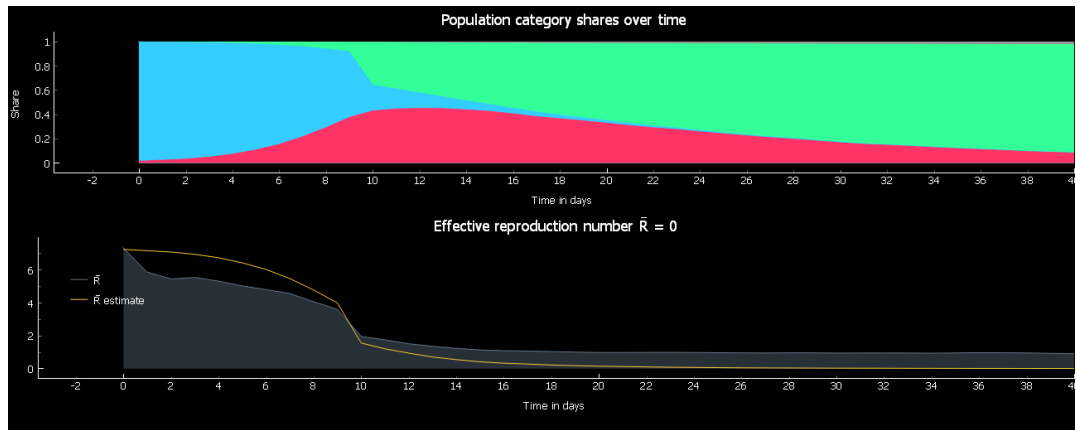
**Abbildung 54.** Anteile der Bevölkerung und  $\bar{R}$  der Impfsimulation mit durchschnittlicher Krankheitsdauer von 14 Tagen

Bei der Standardsimulation wurden insgesamt 2279 Immune und 25 Tote beobachtet. Die Infektionsdauer betrug 23 Tage. Diese Werte sind nicht wesentlich anders als die Simulation mit Einteilung in Tote und Genesene ohne Impfung, was allerdings zu erwarten ist, da die Simulation erst aufhört, nachdem keine kranken Agenten mehr auf dem Feld sind.

Bei der Simulation mit der verlängerten Krankheitsdauer wurden innerhalb der Infektionsdauer von 132 Tagen 6778 Imune und 221 Tote gemessen. Auch hier ist der Unterschied zur Vergleichssimulation nichts aussagend.

Allerdings lässt der geringe Unterschied in der gesamten Krankheitsdauer die Vermutung zu, dass ein nicht ausreichend großer Anteil der Bevölkerung geimpft worden ist. Daher wurden das Experiment mit einer Krankheitsdauer von 14 Tagen noch einmal mit einem Impfanteil von 50% wiederholt. Gleichzeitig wurde die Bevölkerung schon nach Tag 10 geimpft, da aus den Ergebnissen klar wurde, dass bei Tag 15 der Großteil der Agenten bereits infiziert war.

Die Ergebnisse dieser Simulation sind in [Abbildung 55](#) zu sehen.



**Abbildung 55.** Anteile der Bevölkerung und  $\bar{R}$  der Impfsimulation mit durchschnittlicher Krankheitsdauer von 14 Tagen, Impfung von 50% der Bevölkerung nach Tag 10

Bei diesem Durchlauf wurden 6829 Immune und 164 Tote gemessen, während die Infektionsdauer bei 125 Tagen lag. Diese Ergebnisse sind eine deutliche Verbesserung gegenüber dem Vergleichsmodell, sodass davon ausgegangen werden muss, dass eine Impfung von 25% der Bevölkerung noch keinen überwiegenden Schutz gegenüber einer Erkrankung bietet.

Dieses Ergebnis wird auch von den Informationen der Bundesregierung bestätigt, aus denen hervorgeht, dass eine Herdenimmunität erst bei einem langfristigen Anteil von 95% Immunen vorliegt (*Herdenimmunität: Schutz für den Einzelnen und die Gemeinschaft* 2018).

### 3.6 Stellungnahme: Dauer und Wirksamkeit von Maßnahmen gegen COVID-19 <sup>(Eder)</sup>

Zur Verminderung der Ausbreitungsgeschwindigkeit des neuen Virus, hat man in Deutschland verschiedene Maßnahmen in die Wege geleitet:

- Die Ein- und Ausreise ist stark eingeschränkt. Das wird in unserem Modell allerdings nicht beachtet.
- Durch die hohe Aufmerksamkeit rund um COVID-19 sind die Hygienemaßnahmen bei deutschen Bürgern deutlich erhöht worden (Mundschutz, öfter Händewaschen, ...).
- Vereinzelt wurden und werden Personen, welche positiv auf das Virus getestet werden, isoliert für die Dauer der Krankheit.
- Weitreichende Ausgangsbeschränkungen sind in Kraft getreten.

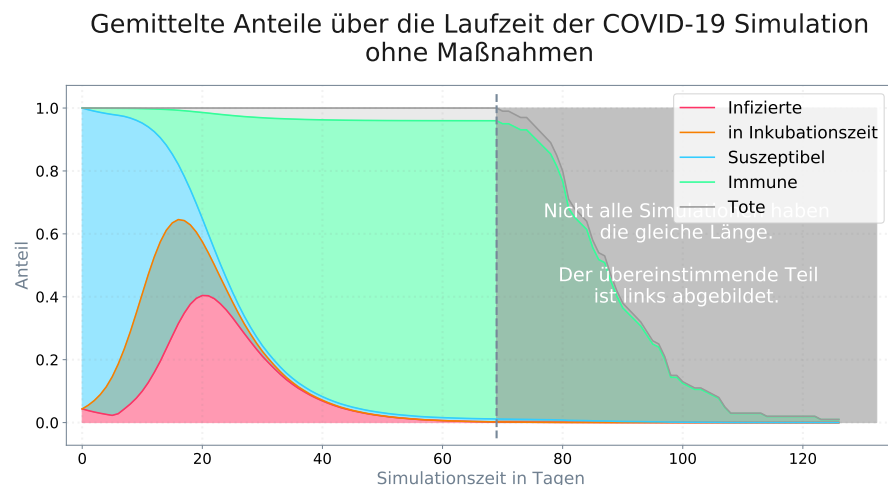
Wir wollen die Auswirkungen dieser Maßnahmen mit unserem Werkzeug überprüfen und auf Basis der Ergebnisse, Aussagen über die Dauer und Wirksamkeit treffen. Dazu benötigen wir im ersten Schritt geeignete Parameter für den Normalzustand (ohne Maßnahmen) und mit in Kraft getretenen Maßnahmen. Dabei hilft uns das Robert Koch Institut, welches einen Steckbrief für das neue Coronavirus verfasst hat (RKI 2020b). Dieser gibt allerlei Auskunft über den aktuellen Erkenntnisstand. Unter anderem wird die Basisreproduktionszahl  $R_0$ , die Krankheitsdauer sowie Inkubationszeit genannt.

- *Größe des Automaten:* 100x100 – die Größe ist von uns willkürlich gewählt, jedoch scheint diese keine großen Auswirkungen auf das Ergebnis zu haben.
- *Suszeptible und infizierte Startanteile  $S(0)$  und  $I(0)$ :* Die Gesamtgröße der Population wird von uns auf 23% geschätzt. Laut *Deutschland – Wikipedia* (2020) liegt die Einwohnerdichte in Deutschland bei 233 Personen pro  $km^2$ . Gehen wir bei der virtuellen Größe unserer simulierten Welt von einem Quadratkilometer aus, wären 23,3% der Zellen unseres Automaten belegt, sofern jede

Person  $1m^2$  repräsentiert. In unserer Anwendung kann man nur gerade Prozentzahlen einstellen, daher wählen wir  $S(0) = 22\%$  sowie  $I(0) = 1\%$ .

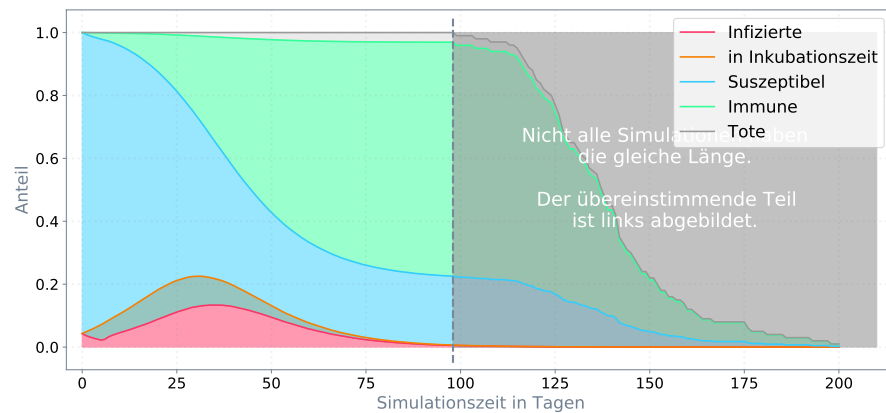
- *Aufteilung in Tote und Immune:* Diese Erweiterung wird aktiviert und wir wählen eine Sterblichkeitsrate von 4%, da laut RKI (2020a, Stand 30.04.2020) 159119 Gesamtfälle sowie 6288 Todesfälle gezählt wurden. Insgesamt sind also  $6288/159119 = 3,95\%$  Todesfälle zu beklagen.
- *Inkubationszeit:* Diese Erweiterung wird ebenfalls aktiviert. Die durchschnittliche Inkubationszeit ist laut RKI (2020b) 5 oder 6 Tage. Für unser Experiment wählen wir 5 Tage.
- *Quarantäne:* Des Weiteren gehen wir davon aus, dass falls die Maßnahmen aktiv sind, ein geringer Anteil der Infektionen erkannt und isoliert wird. Für das Experiment wählen wir einen Anteil von 5%.
- *Durchmischungszahl:* Ebenso wird die Durchmischungszahl im Falle von in Kraft getretenen Maßnahmen deutlich gesenkt auf 10%, sodass eine Ausgangsbeschränkung simuliert wird. Im Normalzustand gehen wir Ermangelns weiterer Informationen zum Bewegungsverhalten der Deutschen von 50% aus.
- Aus demselben Grund wird auch auf eine Beschränkung der Bewegung verzichtet.
- *Remove-Wahrscheinlichkeit:* Dieser Wert kann von der Dauer der Infektiosität berechnet werden, welche vom RKI (2020b) mit 8 Tagen angegeben wird:  $p_r = 12\% \approx 8,33\text{Tage}$ .
- *Infektionswahrscheinlichkeit:* Das RKI gibt auch die Basisreproduktionszahl  $R_0$  zwischen 2,4 und 3,3 an. Der Mittelwert davon beträgt 2,85. Unser Simulator kann für die eingestellten Parameter die Netto-Reproduktionszahl zum Startzeitpunkt berechnen. Durch Ausprobieren kommen wir auf einen Wert von  $\approx 2,85$  bei einer Infektionswahrscheinlichkeit von 40% (Je nach Seed). Diese Zahl wird von uns bei in Kraft getretenen Maßnahmen durch Vorsichts- und Hygienemaßnahmen um 10% verringert.

Nun da wir die benötigten Eingabeparameter beisammenhaben, wollen wir 100 Simulationen mit diesen laufen lassen. Die geballten Ergebnisse dieser Simulationen werden gemittelt und als Diagramme aufbereitet. So sind in Abbildung 56 die Anteile der Personen in den verschiedenen Kategorien über die Zeit für die Simulation ohne Maßnahmen und in 57 mit Maßnahmen dargestellt.



**Abbildung 56.** Diagramm der gemittelten Anteile über die Zeit der Simulation mit den geschätzten COVID-19 Parametern **ohne** Maßnahmen

### Gemittelte Anteile über die Laufzeit der COVID-19 Simulation mit Maßnahmen



**Abbildung 57.** Diagramm der gemittelten Anteile über die Zeit der Simulation mit den geschätzten COVID-19 Parametern **mit** Maßnahmen

Man kann in der Abbildung ohne Maßnahmen einen sehr raschen Anstieg der Infizierten beobachten. Des Weiteren ist die Simulationszeit im Vergleich zum Diagramm mit Maßnahmen deutlich geringer. Daraus können wir ableiten, dass die Maßnahmen Wirkung zeigen und tatsächlich die Ausbreitung des Virus verlangsamen. Das setzt allerdings voraus, dass die Maßnahmen über den kompletten Zeitraum bestehen bleiben.

Der Startbeginn unserer Simulation war mit 1% Infizierten Personen, also 1% der Bevölkerung in Deutschland:  $1\% \cdot 83.149.300 = 831.493$  Personen (Personenanzahl in Deutschland aus *Deutschland – Wikipedia* (2020)). Diese Anzahl an Infizierten haben wir allerdings laut RKI (2020a) noch nicht erreicht. Wir würden der Bundesregierung aufgrund des obigen Graphen raten, mindestens bis zu Tag  $\approx 60$  nach Erreichen der 831.493 Infektionen zu warten, bevor mit Lockerungen der Maßnahmen begonnen wird. Nach Erreichen dieses Tages sind ca. 70% der Population immun, sodass man von einer Herdenimmunität sprechen kann.



## 4 Zusammenfassung und Bewertung <sup>(Eder)</sup>

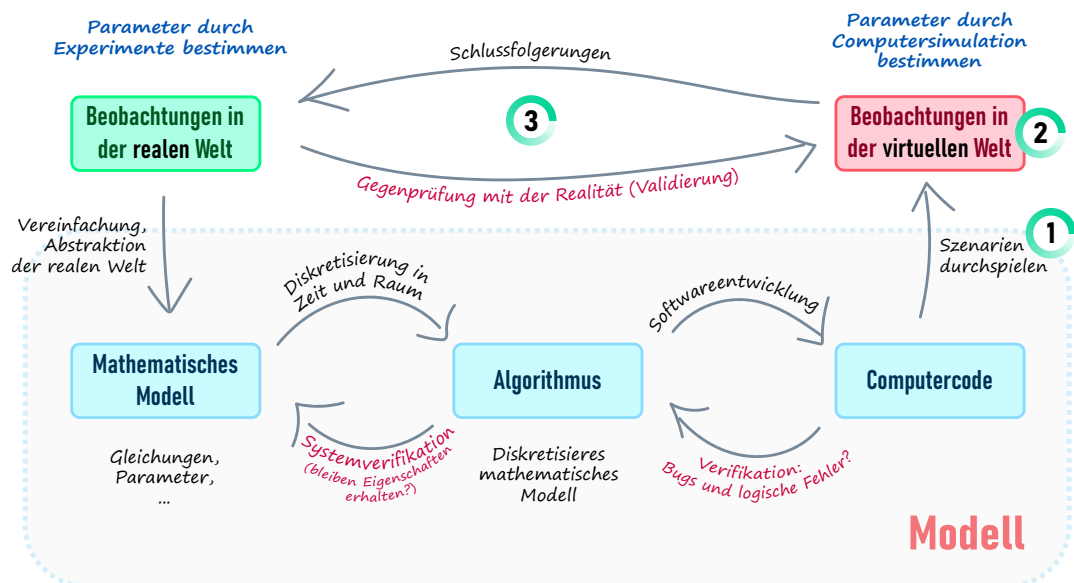
Die aufgeführten Ergebnisse geben einen – wie wir hoffen – interessanten und möglichst akkuraten Überblick über das Verhalten einer Epidemie auf Basis des SIR-Modells. Nichtsdestotrotz sollte man nicht vergessen, dass es sich um ein stark vereinfachendes Modell handelt und daher alle Ergebnisse mit Vorsicht zu genießen sind.

### 4.1 Anmerkungen zum Programm

Das im Zuge dieser Arbeit entstandene Programm bietet vielfältige Möglichkeiten zur Erweiterung des Basismodells und lädt zum Ausprobieren ein. Während der Durchführung der Simulationen haben wir jedoch festgestellt, dass die Laufzeit zu wünschen übrig lässt. Ein möglicher Faktor ist unser Fokus auf Einfachheit und Erweiterbarkeit bei der Architektur während der Implementierung. Ein weiterer die Verwendung von Python als Entwicklungssprache, da diese üblicherweise zur Laufzeit interpretiert wird (*Python (Programmiersprache)* – Wikipedia 2020). Für das nächste Projekt wollen wir nach Möglichkeit die Performance und damit die Laufzeit optimieren, sei es durch die Optimierung der Architektur oder Verwendung eines Ahead-of-time Compilers.

### 4.2 Einordnung in den Modellierungszyklus

Zum Abschluss haben wir vor das Projekt in den Modellierungszyklus einzuordnen. Speziell gehen wir dabei auf das Experiment, die Auswertung und das Analyseergebnis ein. In Abbildung 58 ist der Modellierungszyklus aus den Folien der Vorlesung „Modellbildung und Simulation“ von Prof. Dr. Gerta Köster an der Hochschule München dargestellt. Außerdem enthält das Schaubild drei Nummern zur Verortung der Elemente.



**Abbildung 58.** Modellierungszyklus - frei nach den Folien zur Veranstaltung „Modellbildung und Simulation“ von Prof. Dr. Gerta Köster an der Hochschule München

#### 1. Experiment

Dieser Teil entspricht dem Ausführen verschiedener Szenarien mit unserem Programm. Dabei stellen wir die gewünschten Parameter ein und beobachten live wie sich das Virus ausbreitet. Am Ende stehen verschiedene Graphen und Metriken als Ergebnis des Experiments.

#### 2. Auswertung

Für uns besteht die Auswertung aus dem Bestimmen von Parametern aus der *virtuellen* Welt. Diese stellen gleichfalls unsere Beobachtungen aus der Simulation dar. Beispielsweise haben

wir für die Stellungnahme zu COVID-19 versucht die Dauer der Epidemie zu bestimmen.

### 3. **Analyseergebnis**

Unter Analyseergebnis verstehen wir direkt im Anschluss an die Auswertung die Schlussfolgerung und Validierung mit der *realen* Welt. Für unser COVID-19 Beispiel sollte hier die ausgewertete Dauer der Epidemie mit der tatsächlichen - noch in der Zukunft liegenden - verglichen werden.

### 4.3 Abschließende Worte

Als ein spannendes Projekt mit interessanten Ergebnissen möchten wir das Projekt auch in Zukunft pflegen und erweitern. In diesem Zuge soll der Quellcode möglicherweise auf einer geeigneten Plattform für die Öffentlichkeit bereitgestellt werden.

## Literaturverzeichnis

- Basisreproduktionszahl – Wikipedia. 2020. [Online; accessed 25. Apr. 2020], April. <https://de.wikipedia.org/wiki/Basisreproduktionszahl>.
- Bayerisches Staatsministerium für Familie, Arbeit und Soziales. 2020. *Coronavirus: Ausgangsbeschränkungen in Bayern*. [Online; accessed 25. Apr. 2020], April. <https://www.stmas.bayern.de/aktuelle-meldungen/ausgangssperre.php>.
- BMVI - Mobilität in Deutschland. 2020. [Online; accessed 26. Apr. 2020], Februar. <https://www.bmvi.de/SharedDocs/DE/Artikel/G/mobilitaet-in-deutschland.html>.
- Boccara, N., und K. Cheong. 1992. "Automata network SIR models for the spread of infectious diseases in populations of moving individuals". *J. Phys. A: Math. Gen.* 25, Nr. 9 (Mai): 2447–2461. ISSN: 0305-4470. doi:10.1088/0305-4470/25/9/018.
- Coronavirus disease 2019. 2020. [Online; accessed 23. Apr. 2020], April. <https://www.who.int/emergencies/diseases/novel-coronavirus-2019>.
- COVID-19: Fallzahlen in Deutschland und weltweit. 2020. [Online; accessed 26. Apr. 2020], April. [https://www.rki.de/DE/Content/InfAZ/N/Neuartiges\\_Coronavirus/Fallzahlen.html](https://www.rki.de/DE/Content/InfAZ/N/Neuartiges_Coronavirus/Fallzahlen.html).
- Covid-19: Krankheitsanzeichen. 2020. [Online; accessed 26. Apr. 2020], März. <https://www.lungenaerzte-im-netz.de/krankheiten/covid-19/symptome-krankheitsverlauf/>.
- Deutschland – Wikipedia. 2020. [Online; accessed 30. Apr. 2020], April. <https://de.wikipedia.org/wiki/Deutschland>.
- Herdenimmunität: Schutz für den Einzelnen und die Gemeinschaft. 2018. [Online; accessed 26. Apr. 2020], Januar. <https://www.impfen-info.de/wissenswertes/herdenimmunitaet.html>.
- Inkubationszeit – Wikipedia. 2020. [Online; accessed 25. Apr. 2020], April. <https://de.wikipedia.org/wiki/Inkubationszeit>.
- Normal Distribution Applet/Calculator. 2019. [Online; accessed 25. Apr. 2020], September. <https://homepage.divms.uiowa.edu/~mbogner/applets/normal.html>.
- numpy.random.choice. 2020. [Online; accessed 25. Apr. 2020], Februar. <https://numpy.org/doc/stable/reference/random/generated/numpy.random.choice.html?highlight=numpy%20random%20choice#numpy.random.choice>.
- PyQtGraph - Scientific Graphics and GUI Library for Python. 2016. [Online; accessed 25. Apr. 2020], Dezember. <http://www.pyqtgraph.org>.
- Python (Programmiersprache) – Wikipedia. 2020. [Online; accessed 1. May 2020], Mai. [https://de.wikipedia.org/w/index.php?title=Python\\_\(Programmiersprache\)&oldid=198884601](https://de.wikipedia.org/w/index.php?title=Python_(Programmiersprache)&oldid=198884601).
- RKI. 2020a. *RKI - Coronavirus SARS-CoV-2 - COVID-19: Fallzahlen in Deutschland und weltweit*. [Online; accessed 30. Apr. 2020], April. [https://www.rki.de/DE/Content/InfAZ/N/Neuartiges\\_Coronavirus/Fallzahlen.html](https://www.rki.de/DE/Content/InfAZ/N/Neuartiges_Coronavirus/Fallzahlen.html).
- . 2020b. *RKI - SARS-CoV-2 Steckbrief zur Coronavirus-Krankheit-2019 (COVID-19)*. [Online; accessed 30. Apr. 2020], April. [https://www.rki.de/DE/Content/InfAZ/N/Neuartiges\\_Coronavirus/Steckbrief.html](https://www.rki.de/DE/Content/InfAZ/N/Neuartiges_Coronavirus/Steckbrief.html).
- The Qt Company. 2020. *Qt | Cross-platform software development for embedded & desktop*. [Online; accessed 25. Apr. 2020], April. <https://www.qt.io>.
- timofurrer/observable minimalist event system for Python. 2018. [Online; accessed 25. Apr. 2020], November. <https://github.com/timofurrer/observable%7D>.