Hochschule für
Technik und Wirtschaft
Dresden
University of Applied Sciences

Belegarbeit
Projektseminar
WS 2020/21

Parameteroptimierung eines SIR-Modells mittels Kleinste-Quadrate-Methode

Referent: Prof. Dr. rer. nat Fabian Schwarzenberger
Dipl.-Tech. Math. Tommy Etling

Vorgelegt von Felix Müller
s79138
allgemeine Informatik
Abgabetermin: 28.02.2021

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1 Einleitung

Anfang 2020 wurde das Virus SARS-CoV-2 als Auslöser der neuartigen Lungenkrankheit COVID-19 identifiziert. Es wird durch hauptsächlich durch Tröpfcheninfektion übertragen.[1]
Ziel des Projektseminars ist es zu ermitteln, wie gut dich die Parameter eines SIR-Modells nummerisch optimieren lassen und mit Daten des Robert-Koch-Instituts zur Coronaviruspandemie in Deutschland die Parameter eines SIR-Modells nummerisch zu optimieren. Diese Arbeit dokumentiert genutzte Quellen, die Vorgehensweise, die gewonnenen Ergebnisse; diskutiert Probleme, offene Fragestellungen und gibt Ausblick auf weitere Aspekte.
Im folgenden wird das Differentialgleichungsmodell SIR und die Methode der kleinsten Quadrate beschreiben.

2 SIR-Modell

Das SIR-Modell ist ein Differentialgleichungsmodell zur Modellierung von Infektionskrankheiten mit 3 Kompartimenten. Die 3 Kompartimente sind:

- $S(t)\ldots$ Anzahl nicht infizierter und nicht entfernter Individuen zum Zeitpunkt t (S ... susceptible)
- $I(t)\ldots$ Anzahl der infektiösen Individuen zum Zeitpunkt t (I ... infected)
- $R(t)\ldots$ Anzahl der aus dem Modell durch Tot oder Immunität entfernten Individuen zum Zeitpunkt t (R ... removed)

Ein einzelnes Individuum kann durch Ansteckung von **S** zu **I** übergehen und durch Genesung oder Tod von **I** zu **R**. (siehe Annahmen)

$$S \longrightarrow I \longrightarrow R$$

Es werden folgende Annahmen getroffen:

- Individuen, die von der Krankheit genesen sind können sich nicht erneut infizieren, sind also immun.
- Demographische Effekte werden nicht modelliert, dass bedeutet die Populationsgröße $N(t) = S(t) + I(t) + R(t)$ ist für alle t konstant.

- Der Erwartungswert der durch ein infektiöses Individuum neu infizierten Individuen beträgt pro Einheitszeitintervall $\beta \cdot S$ Im Einheitszeitintervall t finden demnach $\beta \cdot S(t) \cdot I(t)$ Infektionen statt.

- Infizierte verlassen die Gruppe der Infizierten mit der Rate α.
  Es gilt dadurch für einzelne Individuen: die Dauer der Infektiosität X in Einheitszeitintervallen ist exponentialverteilt ($X \sim \text{Exp}(\alpha)$) mit dem Erwartungswert $\mathbb{E} = \frac{1}{\alpha}$. Hier ist zwischen der Dauer der Infektiosität und der Dauer des Vorhandenseins von Krankheitssymptomen einzelner Individuen zu unterscheiden. Patienten können noch Krankheitssymptome aufweisen, auch wenn sie nicht mehr infektiös sind und Infizierte die (noch) keine Krankheitssymptome aufweisen können andere Individuen infizieren.

Die Ableitungen für S, I und R lauten wie folgt:

$$\frac{\mathrm{d}S}{\mathrm{d}t} = -\bar{\beta} \cdot S \cdot I \qquad \frac{\mathrm{d}I}{\mathrm{d}t} = \bar{\beta} \cdot S \cdot I - \alpha I \qquad \frac{\mathrm{d}R}{\mathrm{d}t} = \alpha \cdot I \tag{1}$$

3 Parameteroptimierung - Kleinste-Quadrate-Methode

Die Methode der kleinsten Quadrate verwendet die quadratische Abweichung zwischen einem Messwertvektor $\vec{y}$ mit n Komponenten (Messungen) und einer bestimmten Modellfunktion $\gamma : \mathbb{R}^k \to \mathbb{R}^n$ mit einem Parametervektor $\vec{\theta} \in \mathbb{R}^k$ als Argument. Die Minimierungsaufgabe kann dann numerisch mit Hilfe eines Rechners gelöst werden.
Wird die Summe der quadratischen Abweichungen minimiert, so findet man den Kleinste-Quadrate-Schätzer $\hat{\theta}$:

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \frac{1}{2} \sum_{i=1}^{n} (\gamma(\theta)_i - y_i)^2 \tag{2}$$

3.1 mehrdimensionale Optimierung am Beispiel

Zunächst soll mit einem Beispiel das Bestimmen der Lösung eines SIR-Modells, sowie die Simulation einer normalverteilten Störung dieser Lösung und das Anpassen der Modellparameter an die gestörte Lösung mit Hilfe des Kleinste-Quadrate-Verfahrens gezeigt werden.
Es soll in diesem Beispiel der Verlauf einer Pandemie vom ersten aufgetretenen Fall an betrachtet werden, deshalb wurden folgende Werte als Startwerte für das SIR-Modell festgelegt:

- $S_0 = 8 \cdot 10^7$

- $I_0 = 1$
- $R_0 = 0$

3.1.1 Bestimmen der Lösung für geeignete Parameter

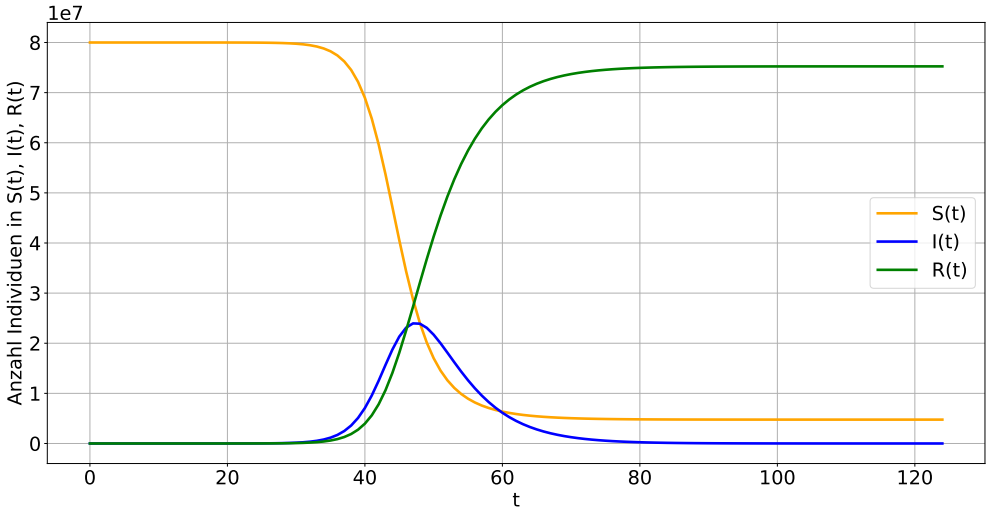

Abbildung 1: ungestörte Lösung: $S(t)$, $I(t)$, $R(t)$

Als Ausgangsparameter wurden: $\alpha = 0.2$ und $\beta = \bar{\beta} \cdot S_0 = 0.6$ gewählt. Der Parameter $\bar{\beta}$ wird mit S_0 skaliert, da er so, wie α ungefähr in der Größenordnung zwischen 10^{-1} und 10^{1} liegt. Es ist numerisch sinnvoller einen Parameter in dieser Größenordnung zu optimieren. Im Folgenden wird nur noch $\beta = \bar{\beta} \cdot S_0$ verwendet. Der skalierte Parametern kann verwendet werden, wenn die Modellgleichungen, die ihn verwenden, die Skalierung aufheben, sodass $\frac{\mathrm{d}S}{\mathrm{d}t} = -\frac{\beta}{N} \cdot S \cdot I$ und $\frac{\mathrm{d}I}{\mathrm{d}t} = \frac{\beta}{N} \cdot S \cdot I - \alpha I$ gilt.
Das Differentialgleichungsmodell wurde durch die Funktion SCIPY.INTEGRATE.ODEINT gelöst. Aus dieser Lösung lassen sich die Messwertevektoren M_I und M_R gewinnen, die die Funktionswerte des entsprechenden Kompartiments für die ganzzahligen Funktionsargumente zwischen $t_0 = 0$ und $t_n = 125$ enthalten.

3.1.2 Erzeugung normalverteilt gestörter Messdaten

Durch zwei Funktionen kann ein zufälliger normalverteilter Störungsvektor mit fester Varianz *var* in der Länge des ungestörten Datenvektors erzeugt und zu einem beliebigen Datenvektor addiert werden.

```
 #erzeugt normalverteilten Fehlervektor: Laenge = steps E=0 Varianz=var
def genNoise(var, steps):
  return np.random.normal(0,var,steps)

 # addiert normalverteilten Fehler auf Vektor data
def addNoise(var, steps, data):
  noisyData = data + genNoise(var, steps)
  return noisyData
```

Abbildung 2: Listing: Funktionen zum normalverteilten Stören eines Datenvektors

Die n Komponenten des Störungsvektors S sind n unabhängig und identisch verteilte Zufallsvariablen mit der Verteilung

$$X_i \sim \mathcal{N}(\mu, \sigma^2) \qquad \text{mit } i = 1, ..., n \quad \text{und } \mu = 0 \tag{3}$$

Der Störungsvektor A wird zum Messwertvektor M addiert, dadurch lassen sich für die Kompartimente I und R normalverteilt gestörte Messwertevektoren $N = M + A$ erzeugen. Das Hinzufügen der Störung führt bei einigen Komponenten der Messwertevektoren zu negativen Werten. Negative Zahlen für die Anzahl infizierter oder aus dem Modell entfernter Individuen sind zwar nicht realistisch, aber nummerisch leichter zu handhaben, als eine Verteilung mit $\mu \neq 0$, da sonst die Methode der kleinsten Quadrate nicht zur Optimierung geeignet ist.

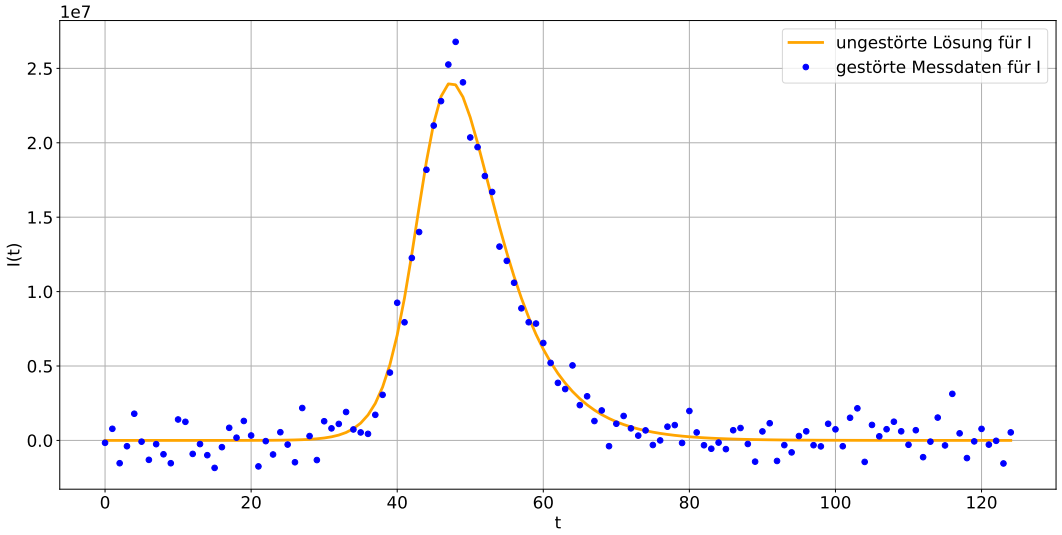

Abbildung 3: Kompartiment I der Lösung des SIR-Modells (orange) und gestörte Messdaten für I (blau) für vorgegebene Parameter $\alpha = 0.2$, $\beta = 0.6$ und $\sigma^2 = 1 \cdot 10^6$ bei 125 Zeitschritten

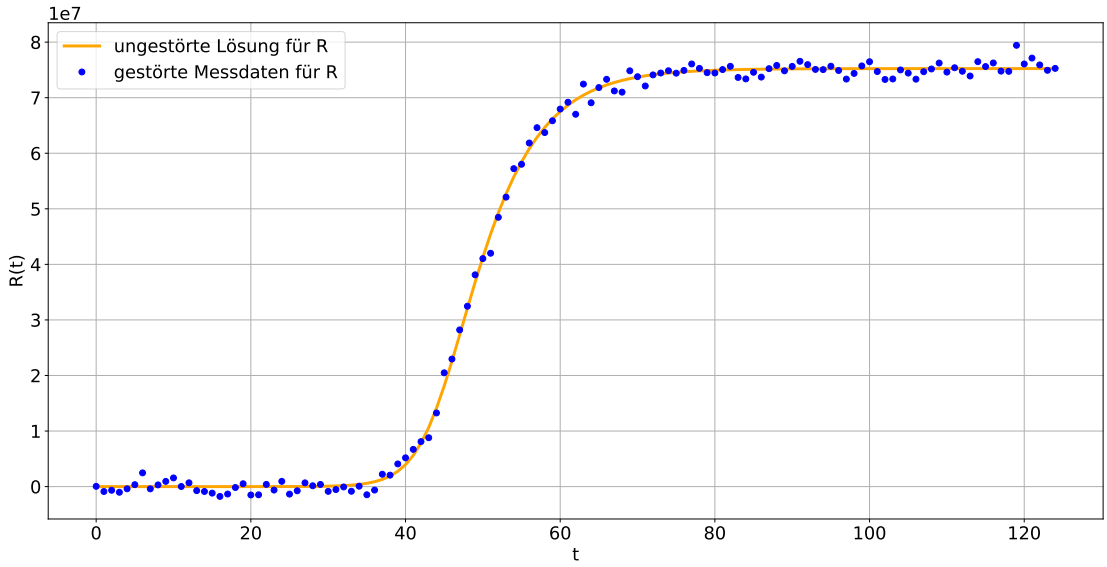

Abbildung 4: Kompartiment R der Lösung des SIR-Modells (orange) und gestörte Messdaten für R (blau) für vorgegebene Parameter $\alpha = 0.2$, $\beta = 0.6$ *und* $\sigma^2 = 1 \cdot 10^6$ *bei 125 Zeitschritten*

3.1.3 Optimierung der gestörten Messdaten

Im nächsten Schritt werden die gestörten Messdaten

$$N = \begin{pmatrix} N_I \\ N_R \end{pmatrix}$$

genutzt, um mit Hilfe der Methode der Kleinsten Quadrate die ursprünglichen Parameter des SIR-Modells zu bestimmen.
Die Modellfunktion $\gamma : \mathbb{R}^2 \to \mathbb{R}^{125}$ bildet den Parametervektor θ, bestehend aus den Komponenten α und β auf die Kompartimente I und R (mit je $n = 125$ Modellwerten) des zu Grunde liegenden SIR-Modells ab. Konvergiert das genutzte Optimierungsverfahren (*Trust Region Reflective algorithm*), dass die Zielfunktion $f(\theta) = \sum_{i=1}^{n} (\gamma(\theta)_i - N_i)^2$ minimieren soll, so werden die angepassten Parameter einer Liste hinzugefügt.

Beispielhaft wurden 200 angepasste Parameterpaare $(\alpha_{opt}, \beta_{opt})$ für die Varianzen $\sigma_1^2 = 1 \cdot 10^6$, und $\sigma_2^2 = 5 \cdot 10^6$ als Scatterplot dargestellt.
Die Anpassung der Parameter des Differentialgleichungsmodells auf Grundlage der gestörten

```
 # kompletter Prozess: erzeugt ungestoerte Loesung, fuegt Stoerung hinzu
 # und schaetzt aus gestoerten Werten Parameter
def genDataAndOptimize():
  def residuum(theta):
    I_res = weightI * (noisyI - modelValues(theta, ts)[:, 1])
    R_res = weightR * (noisyR - modelValues(theta, ts)[:, 2])
    total = np.concatenate([I_res,R_res])
    return I_res


   # erzeuge Daten und stoere diese
  genuineSIR  = genGenuineSIR(x0,timeSteps,alphaOrig,betaOrig)
  noisyI    = addNoise(var, timeSteps, genuineSIR[:, 1])
  noisyR    = addNoise(var, timeSteps, genuineSIR[:, 2])

   # Optimierung
  theta0 = [0.2,0.6]
  res = least_squares(residuum, theta0, verbose=0, ftol= 1e-12,
                      xtol= 1e-12, gtol= 1e-12, max_nfev=200)
  return res
```

Abbildung 5: Listing: Funktion zum Anpassen, der Parameter des SIR-Modells an die gestörte Lösung

Messdaten hat zufriedenstellend funktioniert. Es wird deutlich, dass bei einer größeren Varianz σ^2 der Störung, auch die Streuung der optimierten Parameter größer ist.
An den Scatterplots für beide Varianzen ist zu erkennen, dass es eine starke positive, lineare Abhängigkeit zwischen beiden Parametern gibt, dass heißt wird einer der beiden Parameter α oder β zu klein oder zu groß geschätzt, so wird auch der andere Parameter in nahezu linearer Abhängigkeit zu klein beziehungsweise zu groß geschätzt. Diese Abhängigkeit entspringt der Differentialgleichung des Kompartiments I des SIR-Modells: $\frac{\mathrm{d}I}{\mathrm{d}t} = \frac{\beta}{S_0} \cdot S \cdot I - \alpha I$, da die optimale Änderungsrate $\frac{\mathrm{d}I}{\mathrm{d}t}$ (im Sinne der Minimierung der Fehlerquadrate) für einen bestimmten Zeitpunkt t durch die Lage der gestörten Messdaten stark eingeschränkt ist. Der lineare Zusammenhang der Differentialgleichung lässt sich näherungsweise im Schätzer identifizieren.

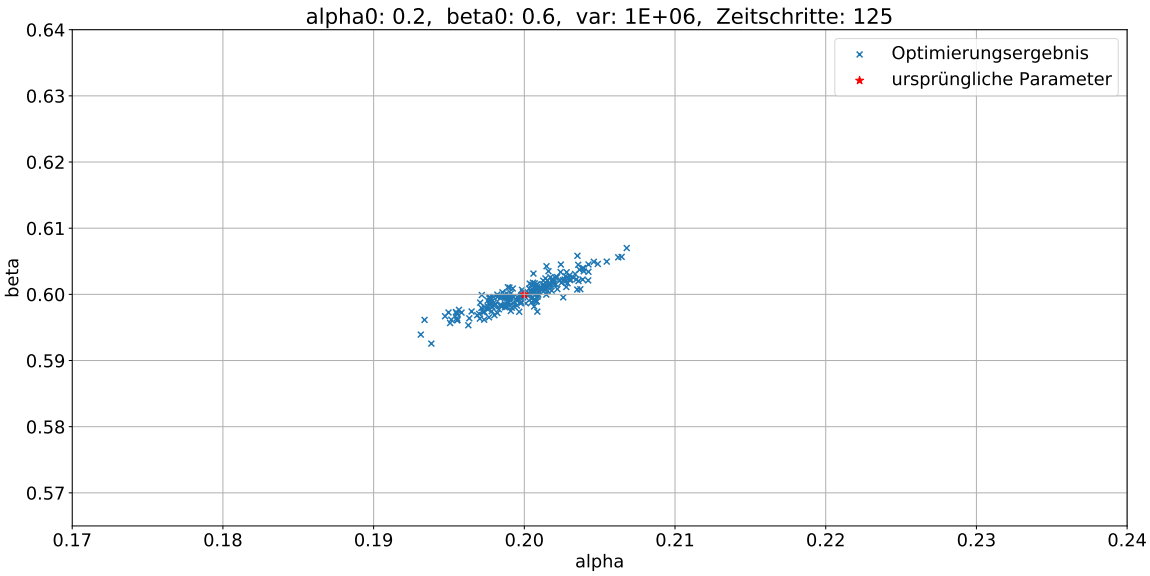

Abbildung 6: Scatterplot α_{opt} und β_{opt} für die Anpassung an normalverteilt gestörte Messdaten mit Störungsvarianz $\sigma^2 = 1 \cdot 10^6$

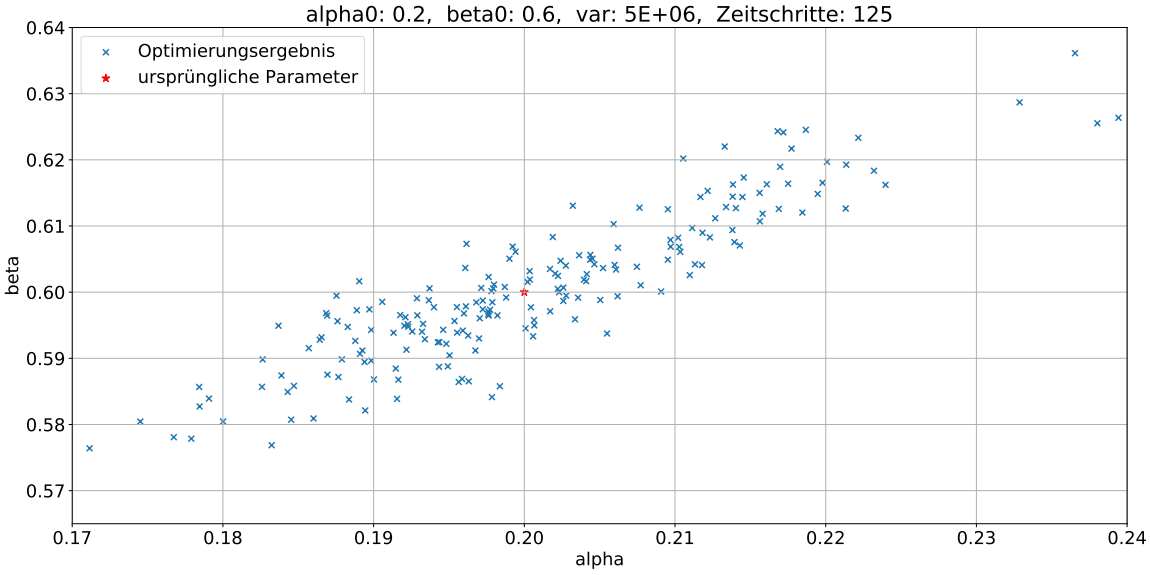

Abbildung 7: Scatterplot α_{opt} und β_{opt} für die Anpassung an normalverteilt gestörte Messdaten mit Störungsvarianz $\sigma^2 = 5 \cdot 10^6$

3.2 eindimensionale Optimierung - gewöhnliche Kleinste-Quadrate-Methode

3.2.1 Beschaffung von geeigneten Daten und Nutzbarmachung

Die Zahl der täglichen Neuinfektionen mit dem neuartigen Coronavirus, die Zahl der Todesfälle und zum Teil auch die Zahl der Genesung werden in der Bundesrepublik Deutschland von den Gesundheitsämtern erhoben und vom Robert-Koch-Institut verarbeitet. Der Informationsdienstleister ArcGIS veröffentlicht täglich eine CSV-Datei, die Daten zu Neuinfektionen, Todesfällen und Genesungen im Bezug auf COVID-19 beinhaltet.
Für die Bestimmung der der zu einem bestimmten Zeitpunkt aktiven Infektionen sind alle Dateien des gewählten Zeitraums nötig. Eine Sammlung der Dateien seit Januar ist online abrufbar.[2]
Genesungen werden vom Robert-Koch-Institut erst seit dem 01. Mai 2020 verzeichnet, dabei wird in nicht hospitalisierten Fällen die Dauer der Krankheit auf 14 Tage geschätzt, in hospitalisierten Fällen ist die genaue Dauer zwischen Infektion (*Referenzdatum*) und Genesung oder Tod angegeben. Zudem fehlen für den 04. und 05. Mai und für den 06. Oktober alle Daten, am 18.01.2021 kam es zu technischen Problemen, weshalb die Zahl der gemeldeten Neuinfektionen an diesem Tag zu niedrig ist.
Aufgrund des Meldeverzugs zwischen Gesundheitsämtern und Robert-Koch-Institut an Wochenenden und Feiertagen, gelegentlich fehlenden Daten und Schwankungen der Anzahl von Testungen von potenziell Infizierten ist es sinnvoll die Daten wochenweise zusammenzufassen, um Ausreißer zu verhindern.[3][4] Die verminderte Anzahl der Datenpunkte genügt trotzdem für eine verlässliche Schätzung der Parameter.
Die Zahl der aktuellen Infektionen $I_{mes}(t_i)$ zum Zeitpunkt t_i ergibt sich durch Addieren der Veränderung ΔI_i zum vorhergehenden Wert $I_{mes}(t-1)$:

$$I_{mes}(t) = I_{mes}(t-1) + \Delta I_i \tag{4}$$

Die Veränderungen zur Vorwoche ΔI_i können aus den vorliegenden Daten bestimmt werden. Der Ausgangswert $I_{mes}(0) = 23\,233$ muss aus den Daten zu Neuinfektionen und Sterbefällen und den Schätzungen zu Genesungen ermittelt werden. Aus dem Situationsbericht des Robert-Koch-Instituts vom 01.05.2020 geht hervor, dass bereits 133 381 Menschen zu R_{mes} zu zählen sind, da sie vor dem 01.05.2020 verstorben oder genesen sind.[6]
Die Struktur der Quelldateien ist in Tabelle 1: Struktur CSV-Dateien des RKI im Anhang beschrieben.[5] Das Entfernen der für Das SIR-Modell nicht benötigten Spalten (1 bis 6, 9

bis 11 und 17) und das Umsortieren der einzelnen Datensätze nach Meldedatum, statt nach Bundesland und Landkreis wurde wie das Summieren der täglichen Anzahl der Neuinfektionen, Todesfälle und Genesungen zur Gewinnung der ΔI und ΔR durch ein C-Programm erledigt. Der Quellcode des Programms ist im Anhang zu finden Abbildung 12: Listing: Summation.
Das Minimum an aktiven Infektionen lag in Woche 13 vor, deshalb wird als Ausgangspunkt der Modellierung Woche 13 (rote Markierung in beiden Plots) gewählt.

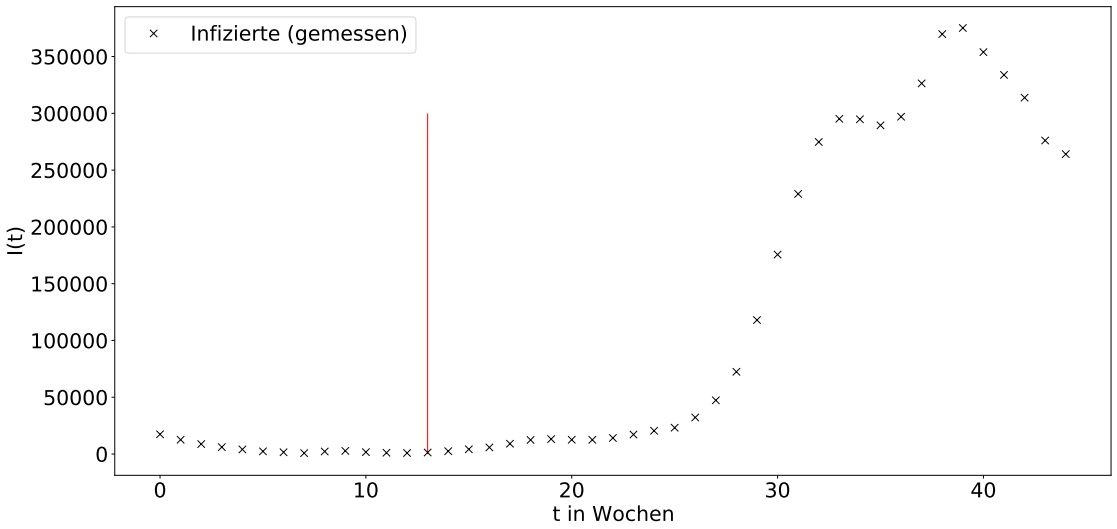

Abbildung 8: Darstellung des Messwertevektors I (wochenweise zusammengefasst)

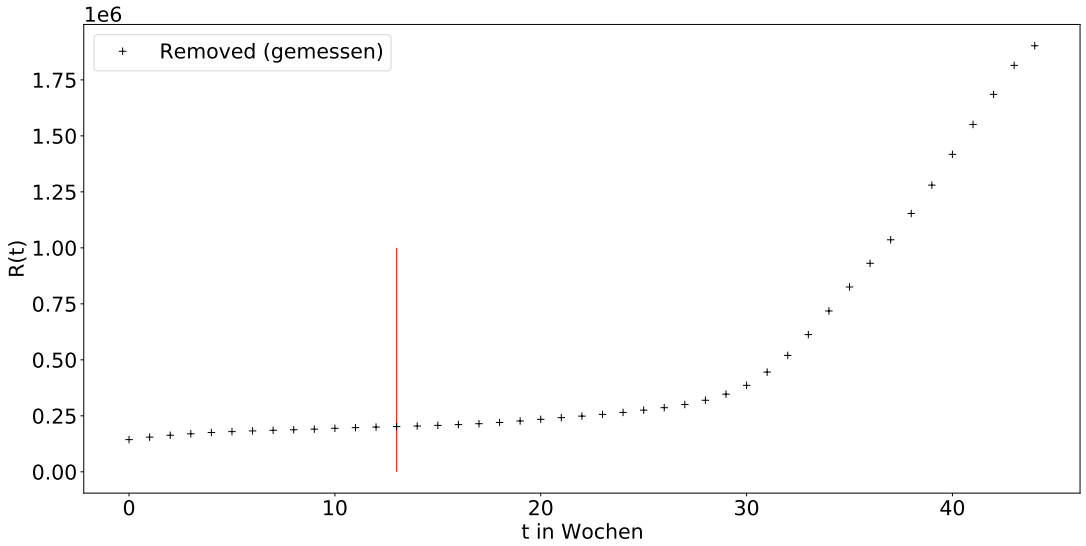

Abbildung 9: Darstellung des Messwertevektors R (wochenweise zusammengefasst)

3.2.2 Anpassung der Modellparameter an die Messwerte für I

```
from scipy.integrate import odeint
from scipy.optimize import least_squares

def y(theta,time):
    modelValues = odeint(model, x0, time,
            args=(theta[0], theta[1]))
    return modelValues[: ,1]

def model(x, t, alpha, beta):
    S = x[0];I = x[1];R = x[2];
    dSdt = (-beta/S0) * S * I
    dIdt = ((beta/S0) * S * I) - (alpha * I)
    dRdt = alpha * I
    return [dSdt, dIdt, dRdt]

ts = range(0,33,1)

def fun(theta):
    return I_measured[12:45] - y(theta,ts)[0:33]

theta0 = [0.4,0.5]
res = least_squares(fun, theta0)
```

Abbildung 10: Listing: gewöhnliche Kleinste-Quadrate-Methode in PYTHON *zur Optimierung von θ*

Die Größe des gesamten Messwertvektors I_{gem} beträgt $n = 45$. Es werden Daten ab der 13. Woche verwendet, also die 33 Datenpunkte die zwischen dem 23.07.2020 und dem 25.01.2020 liegen. Der 13. Eintrag im Messwertvektor ist das Minimum aller Einträge. Es wird davon ausgegangen, dass hier die Grenze zwischen den beiden Infektionswellen liegt. I_{gem_i} ist die Zahl der zum Zeitpunkt t_i (Einheitszeitintervall: 1 Woche) infektiösen Individuen.
Es wurden folgende Werte als Startwerte für das SIR-Modell festgelegt:

- $S_0 = 8 \cdot 10^7$, weil es der Bevölkerungszahl des betrachteten geographischen Gebiets (Bundesrepublik Deutschland) entspricht
- $I_0 = I_{gem_{13}} = 896$, entspricht dem ersten verwendeten Messwert für I
- $R_0 = R_{gem_{13}} = 63\,766$, entspricht dem ersten verwendeten Messwert für R

Das Residuum r_i für den i-ten Datenpunkt ist:

$$r_i = I_{gem_i} - I_i(\theta) \qquad \text{für } i = 1, ..., n \tag{5}$$

Die Funktion *fun* liefert die Residuen. $I(\theta)$ ist das Kompartiment I des SIR-Modells, das durch Lösen des Anfangswertproblems mit dem Parametervektor

$$\theta = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$$

entsteht.
Die zu optimierende Zielfunktion ist die Summe der Quadrate aller Residuen

$$f(\theta) = \frac{1}{2} \sum_{i=1}^{n} r_i^2 \tag{6}$$

Gesucht ist also $\underset{\theta}{\operatorname{argmin}}\, f(\theta)$. Die Optimierungsaufgabe wurde mit der PYTHON-Funktion SCIPY.OPTIMIZE.LEAST_ SQUARES nummerisch gelöst.
SCIPY.OPTIMIZE.LEAST_ SQUARES wählt standardmäßig ein Trust-Region-Verfahren (*Trust Region Reflective algorithm*) zum Finden des globalen Minimums.

3.2.3 Bewertung der Ergebnisse und numerischen Parameter

Das Programm liefert $\alpha \approx 2.68$ und $\beta \approx 2.97$. Der Algorithmus terminiert mit der Meldung „*xtol termination condition is satisfied.*“, dass heißt das die Änderung im Parametervektor θ zwischen

zwei Optimierungsschritten kleiner als die gewählte Toleranz für die unabhängige Variable ist. Verwendet wurde als Toleranz die Maschinengenauigkeit, in der PYTHON-Laufzeitumgebung für 64-bit Systeme beträgt sie circa $2.22 \cdot 10^{-16}$.
Die Supremumsnorm des Gradienten der Zielfunktion an dieser Stelle beträgt 92 451. Durch minimales Verändern des Startvektors θ_0 lässt sich diese Norm zu $9.08 \cdot 10^{-5}$ verbessern. Der entsprechende Parametervektor lautet dann

$$\theta = \begin{pmatrix} 2.6798 \\ 2.9664 \end{pmatrix} \tag{7}$$

Am Graphen der optimierten Funktionen und Messwerte wird allerdings deutlich, dass zwar die Anpassung von $I(\theta)$ an I_{mes} funktioniert hat, aber $R(\theta)$ nicht gut an R_{mes} angepasst ist.

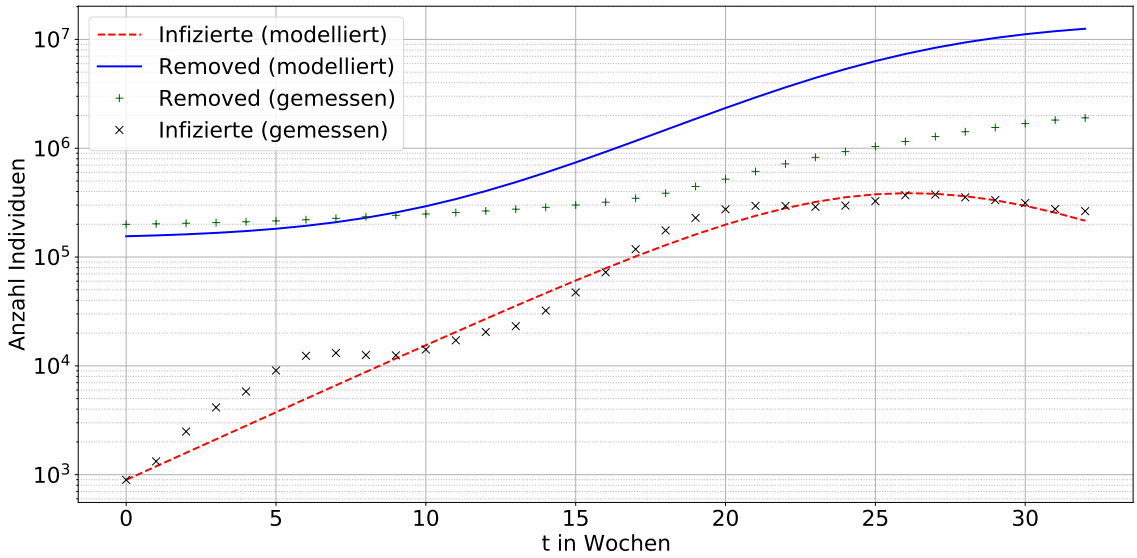

Abbildung 11: Graph der optimierten Funktion für $I(\theta)$ (blau), $R(\theta)$ (rot) und Messwerte I_{gem} ($\times$) und R_{mes} ($+$) in einfach logarithmischer Darstellung

3.3 Bewertung und Ausblick

Durch die Untersuchung der Optimierung der Beispieldaten in Abschnitt 3.1 und die Ergebnisse der Optimierung mit realen Daten in Abschnitt 3.2 lässt sich feststellen, dass ein SIR-Modell zur Modellierung des Verlaufs einer natürlichen Pandemie nur bedingt geeignet ist. Folgende Aspekte blieben bei der Modellierung unberücksichtigt, sind aber für einen Pandemieverlauf wesentlich:

- Der Parameter β ist in der Realität anders als hier angenommen zeitlich veränderlich. Sowohl die Kontaktrate der Individuen, als auch die Übertragungsrate des Krankheitserregers aus denen sich β zusammensetzt, können beeinflusst werden. Nennenswerte Einflussgrößen sind zum Beispiel allgemeine Kontaktbeschränkungen, Hygienemaßnahmen, und die Jahreszeit. In einem verbesserten Modellansatz müsste der Parameter β als Funktion der Zeit modelliert werden.

- Die Simulation in Abschnitt 3.1 geht von einer normalverteilten Störung der Messdaten aus, da diese nummerisch leichter zu handhaben ist, als andere Wahrscheinlichkeitsverteilungen. Jedoch ist nicht bekannt, ob die realen Messdaten die aus den Quellen des Robert-Koch-Instituts gewonnen wurden tatsächlich einen normalverteilten Fehler aufweisen. Insbesondere ist im Gegensatz zu den Annahmen der theoretischen Betrachtungen davon auszugehen, dass die Varianz des Messfehlers realer Daten relativ zu den basierend Werten ist. Ebenso vereinfachend war die Annahme des Erwartungswerts der Störung mit 0.

- Die Messwerte für die durch Genesung aus dem Modell entfernten Individuen wurden mehrheitlich nicht durch Tests bestimmt, sondern in allen nicht hospitalisierten Krankheitsfällen auf 14 Tagen durch das Robert-Koch-Institut geschätzt.

- Weiterhin wurde das Isolieren von Verdachtsfällen und infektiösen Individuen nicht modelliert, das Einfluss auf die Kontaktrate isolierter Individuen und die Wahrscheinlichkeit der Infektion durch isolierte Erkrankte hat. Diese Aspekte können zum Beispiel durch die Erweiterung des Modells, um eigene Klassen für isolierte Verdachtsfälle und isolierte Infektiöse modelliert werden. (vgl. SEQIJR-Modell [7])

4 Anhang

Spalte	Name	Bedeutung oder Bemerkung
1	Object ID	Fallnummer
2 3 4	ID Bundesland Bundesland Landkreis	räumliche Verteilung nicht betrachtet
5	Altersgruppe	Klassierung zu folgenden Gruppen: 0-4, 5-14, 15-34, 35-59, 60-79, 80+
6	Geschlecht	M=Männlich, W=Weiblich, unbekannt
7	**Anzahl Fall**	Anzahl Neuinfektionen
8	**Anzahl Todesfall**	Anzahl Todesfälle
9	Meldedatum	Datum, an dem der Fall dem Gesundheitsamt bekannt wurde
10	ID Landkreis	räumliche Verteilung nicht betrachtet
11	Datenstand	Datum der letzten Aktualisierung
12	**neuer Fall**	0: in aktueller und Publikation des Vortags 1: nur in aktueller Publikation -1: nur in Publikation des Vortags
13	**neuer Todesfall**	0: in aktueller und Publikation des Vortags ein Todesfall 1: in aktueller Publikation ein Todesfall, nicht in Publikation des Vortages -1: in aktueller Publikation kein Todesfall, jedoch in Publikation des Vortags Todesfall -9: weder in aktueller noch in Publikation des Vortages Todesfall
14	**Referenzdatum**	wenn bekannt Erkrankungsdatum, sonst Meldedatum

15	**neu genesen**	0: in aktueller und Publikation des Vortags genesen 1: in aktueller Publikation genesen, nicht in Publikation des Vortages -1: in aktueller Publikation nicht genesen, jedoch in Publikation des Vortags genesen -9: weder in aktueller noch in Publikation des Vortages genesen
16	**Anzahl genesen**	Anzahl der Genesenen
17	Erkrankungsbeginn	1: Referenzdatum ist Erkrankungsbeginn 0: sonst

Tabelle 1: Struktur der CSV*-Dateien des Robert-Koch-Institut*

```c
#include <string.h>
#include <stdlib.h>
#include <stdio.h>

#define LINESIZE 512

const char* fileExtension = ".csv";
const char* file    = "Data/RKI_COVID19_2020-";
const char* file2    = "Data/RKI_COVID19_2021-";

const char* month[]={
 "05-", "06-", "07-", "08-", "09-", "10-", "11-",  "12-"
};

const int month_Lenght[]={
 31, 30, 31, 31, 30, 31, 30, 31
};

int main(int argc, char* argv[]){

 char* name = malloc(sizeof(char) * 64);

 FILE* pDatei;

 char* line = malloc(sizeof(char) * LINESIZE);
 char * dayPart = malloc(sizeof(char*) * 3);

 int i, j;

 for(i = 0; i < 8; i++){
  for(j = 1; j <= month_Lenght[i]; j++){ //Tag
   sprintf(dayPart, "%02d", j);

   strcpy(name, file);
   name = strcat(name, month[i]);
   name = strcat(name, dayPart);
   name = strcat(name, fileExtension);
   FILE* pDatei = fopen(name, "r");

   if(pDatei == NULL){printf("Dateifehler"); exit(1);}

   fgets(line, LINESIZE, pDatei);
   int sum = 0 , sumTod = 0, sumHealed = 0;
```

```
   while(fgets(line, LINESIZE, pDatei)){

    char* part = malloc(sizeof(char*) * 16);

    part = strtok(line, ",");
    for(int l = 0; l < 5; l++)
     part = strtok(NULL, ",");

    part = strtok(NULL, ","); int anzahlFall = atoi(part);
    part = strtok(NULL, ","); int anzahlTodesfall = atoi(part);

    for(int l = 0; l < 3; l++)
     part = strtok(NULL, ",");

    part = strtok(NULL, ",");
    if(strcmp(part,"1") == 0)
     sum += anzahlFall;
    part = strtok(NULL, ",");
    if(strcmp(part,"1") == 0)
     sumTod += anzahlTodesfall;
    part = strtok(NULL, ",");
    part = strtok(NULL, ",");
    if(strcmp(part,"1") == 0){
     part = strtok(NULL, ",");
     sumHealed += atoi(part);
    }
   }

   printf("%s%s: neue: %d, tot: %d, genesen: %d, delta I: %d\n",
month[i],dayPart,sum,sumTod,sumHealed,sum-(sumTod + sumHealed));

   fclose(pDatei);
  }
 }

 for(j=1; j<=25; j++){  //Januar 2021
  sprintf(dayPart, "%02d", j);

  strcpy(name, file2);
  name = strcat(name, "01-");
  name = strcat(name, dayPart);
  name = strcat(name, fileExtension);

  FILE* pDatei = fopen(name, "r");

  if(pDatei == NULL){printf("Dateifehler2"); exit(1);}

   fgets(line, LINESIZE, pDatei);
   int sum = 0;
   int sumTod = 0;
   int sumHealed = 0;
   while(fgets(line, LINESIZE, pDatei)){

    char* part = malloc(sizeof(char*) * 16);
```

```
part = strtok(line, ",");
    for(int l = 0; l < 5; l++)
     part = strtok(NULL, ",");

    part = strtok(NULL, ","); int anzahlFall = atoi(part);
    part = strtok(NULL, ","); int anzahlTodesfall = atoi(part);

    for(int l = 0; l < 3; l++)
     part = strtok(NULL, ",");

    part = strtok(NULL, ",");
    if(strcmp(part,"1") == 0)
     sum += anzahlFall;
    part = strtok(NULL, ",");
    if(strcmp(part,"1") == 0)
     sumTod += anzahlTodesfall;
    part = strtok(NULL, ",");
    part = strtok(NULL, ",");
    if(strcmp(part,"1") == 0){
     part = strtok(NULL, ",");
     sumHealed += atoi(part);
    }
   }

   printf("%s%s: neue: %d, tot: %d, genesen: %d, delta I: %d\n",
"01-", dayPart, sum, sumTod, sumHealed, sum-(sumTod + sumHealed));

   fclose(pDatei);
 }

}
```

Abbildung 12: Listing: Summation der Neuinfektionen, Todesfälle und Genesungen

Literatur

[1] Robert-Koch-Institut. “Epidemiologischer Steckbrief zu SARS-CoV-2 und COVID-19“, https://www.rki.de/DE/Content/InfAZ/N/Neuartiges_Coronavirus/Steckbrief.html abgerufen am 31.01.2021

[2] CharlesStr. “CSV-Dateien-mit-Covid-19-Infektionen“, GitHub, https://github.com/CharlesStr/CSV-Dateien-mit-Covid-19-Infektionen- abgerufen am 25.01.2021

[3] Robert-Koch-Institut. “Tabellen zu Testzahlen, Testkapazitäten und Probenrückstau pro Woche (27.1.2021)“, https://www.rki.de/DE/Content/InfAZ/N/Neuartiges_Coronavirus/Daten/Testzahlen-gesamt.html abgerufen am 29.01.2020

[4] Robert-Koch-Institut. “Warum sind die Fallzahlen am/nach dem Wochenende geringer als an Arbeitstagen?“, https://www.rki.de/SharedDocs/FAQ/NCOV2019/gesamt.html abgerufen am 31.01.2020

[5] ArcGIS. “Beschreibung der Daten des RKI Covid-19-Dashboards“, https://www.arcgis.com/home/item.html?id=dd4580c810204019a7b8eb3e0b329dd6 abgerufen am 25.01.2020

[6] Robert-Koch-Institut. “Täglicher Lagebericht des RKI zur Coronavirus-Krankheit-2019 (COVID-19) 01.05.2020 – AKTUALISIERTER STAND FÜR DEUTSCHLAND“, https://www.rki.de/DE/Content/InfAZ/N/Neuartiges_Coronavirus/Situationsberichte/2020-05-01-de.pdf abgerufen am 01.02.2021

[7] Brauer F., Castillo-Chavez C. (2012). “Mathematical Models in Population Biology and Epidemiology“ (2.Aufl.). Springer.