

Übung 03: Pandemieausbruch

Tobias Blesgen und Leonardo Thome

09.06.2021

Im Folgenden wollen wir eine simulierte Pandemieausbreitung von SARS-CoV-2 für verschiedene Parameter im SIR Modell betrachten. Das SIR Modell beschreibt den zeitlichen Verlauf mit folgendem Differentialgleichungssystem:

$$S'(t) = -\frac{\beta}{N}S(t)I(t) - \Gamma(t) + \delta V(t) \quad (1)$$

$$I'(t) = \frac{\beta}{N}S(t)I(t) - \alpha I(t) \quad (2)$$

$$R'(t) = \alpha I(t) \quad (3)$$

$$V'(t) = \Gamma(t) - \delta V(t) \quad (4)$$

Hierbei entspricht S der Zahl der noch nicht Infizierten, I der aktuell Kranken, R der Genesenen und V der Geimpften. Wir werden im Verlauf der Simulationen die Geimpfenzahl mit der Anzahl der Personen mit Zweitimpfung gleichsetzen, da ab diesem Punkt der vollständige Schutz eintritt. Desweiteren ist N die Gesamtbevölkerung, R_0 entspricht der Basisreproduktionszahl, α der inversen Infektiositätszeit, $\beta = R_0 * \alpha$, Γ die Impfungsrate und δ die Rate mit der Geimpfte erkranken.

Runge-Kutta 2 Verfahren

Um die differenziellen Gleichungssysteme auszuwerten, verwenden wir das Runge-Kutta Verfahren nach

$$x_{i+1} = x_i + \frac{h}{2} [f(t_i, x_i) + f(t_i + h, x_i + hf(t_i, x_i))]. \quad (5)$$

Wobei sich unser f aus den vier Anteilen von S, I, R und V zusammensetzt. Wir werden die Auswirkungen der Schrittweite am Ende der Auswertung genauer betrachten.

Vorerst wählen wir die Schrittweite $h = 1$, da dies direkt den Tagen entspricht. Dabei bietet uns das Runge-Kutta Verfahren numerische Stabilität und weist mit einem Verfahrensfehler von $\mathcal{O}(h^2)$ einen kleineren Fehler als das Eulerverfahren auf.

Implementation des DGS nach dem Runge-Kutta 2 verfahren.

```

#include<Rcpp.h>
#include<stdlib.h>
#include<vector>
#include<algorithm>

using namespace Rcpp;

// Wir verwenden Strukturen um Funktionsargumente uebersichtlicher zu halten
typedef struct
{
    double S, I, R, V;
} Status;

typedef struct
{
    double alpha, beta, delta, N, gamma;
} Parameter;

// Template zum Zerschneiden der verwendeten Vektoren
template<typename T>
std::vector<T> slice(std::vector<T> const &v, int m, int n)
{
    auto erste = v.cbegin() + m;
    auto letzte = v.cbegin() + n + 1;

    std::vector<T> vektor(erste, letzte);
    return vektor;
}

// Berechnungsschritt der Ableitungen nach dem DGS
void f(Status alterStatus, Parameter parameter, Status& neuerStatus){
    double Gamma = std::min(parameter.gamma, alterStatus.I);
    neuerStatus.S = -parameter.beta * alterStatus.S * alterStatus.I/parameter.N
        - Gamma + parameter.delta * alterStatus.V;

    neuerStatus.I = parameter.beta * alterStatus.S * alterStatus.I/parameter.N
        - parameter.alpha * alterStatus.I;

    neuerStatus.R = parameter.alpha * alterStatus.I;

    neuerStatus.V = Gamma - parameter.delta * alterStatus.V;
}

// Ein Intergrationsschritt nach Runge-Kutta
void rkSchritt(Status& status, Parameter parameter, double h){
    Status fStatus;                // Standart Ableitung
    f(status, parameter, fStatus);

    Status f2Status;              // Mischterm Ableitung
    Status gemischt = {.S = status.S + h*fStatus.S,
                       .I = status.I + h*fStatus.I,
                       .R = status.R + h*fStatus.R,
                       .V = status.V + h*fStatus.V};
}

```

```

    f(gemischt, parameter, f2Status);
    status.V = status.V + h/2*(fStatus.V + f2Status.V);
    status.R = status.R + h/2*(fStatus.R + f2Status.R);
    status.S = status.S + h/2*(fStatus.S + f2Status.S);
    status.I = status.I + h/2*(fStatus.I + f2Status.I);
}

//[[Rcpp::export]]
Rcpp::List durchlauf(const int maxSchritte, const double h,
                    const double S, const double I, const double R,
                    const double V, const double alpha,
                    const double beta, const double delta,
                    const double N, const double gamma,
                    const double x0, const int I_min = 1){
    // Arrays der Werte zur späteren Ausgabe
    std::vector<double> xWerte(maxSchritte);
    std::vector<double> SWerte(maxSchritte);
    std::vector<double> IWerte(maxSchritte);
    std::vector<double> RWerte(maxSchritte);
    std::vector<double> VWerte(maxSchritte);
    // Quelltext
    Status status = {.S = S, .I = I, .R = R, .V = V};
    Parameter parameter = {.alpha = alpha, .beta = beta, .delta = delta, .N = N, .gamma = gamma};
    // Schleife bis zur Abbruchsbedingung
    for (int i = 0; i < maxSchritte; i++){
        xWerte[i] = x0 + i*h;
        SWerte[i] = status.S;
        IWerte[i] = status.I;
        RWerte[i] = status.R;
        VWerte[i] = status.V;
        rkSchritt(status, parameter, h);
    // Vorzeitige Abbruchbedingung für Sonderbedingung von I
        if(status.I < I_min && i>10){
            // Rückgabe für eine grafische Wiedergabe
            return List::create(Named("x") = slice(xWerte,0,i),
                               Named("S") = slice(SWerte,0,i),
                               Named("I") = slice(IWerte,0,i),
                               Named("R") = slice(RWerte,0,i),
                               Named("V") = slice(VWerte,0,i)
                               );
        }
    }
    // Rückgabe für eine grafische Wiedergabe
    return List::create(Named("x") = xWerte, Named("S") = SWerte,
                       Named("I") = IWerte, Named("R") = RWerte,
                       Named("V") = VWerte
                       );
}

```

Auswertung

Pandemieverlauf Ohne Gegenmaßnahmen

Für den Ausbruch einer Krankheit in Deutschland sind die Anfangswerte gegeben wie folgt. $N = 8.3 * 10^7$, $S(0) = N - 5000$, $I(0) = 5000$, und $R(0) = V(0) = 5000$. Betrachten wir nun den Verlauf ohne Gegenmaßnahmen, also $\Gamma = \delta = 0$ und den SARS_CoV-2 spezifischen Daten $R_0 \approx 2.9$ und $\alpha = \frac{1}{7 \text{ Tage}}$, so ergibt sich Abb. 1.

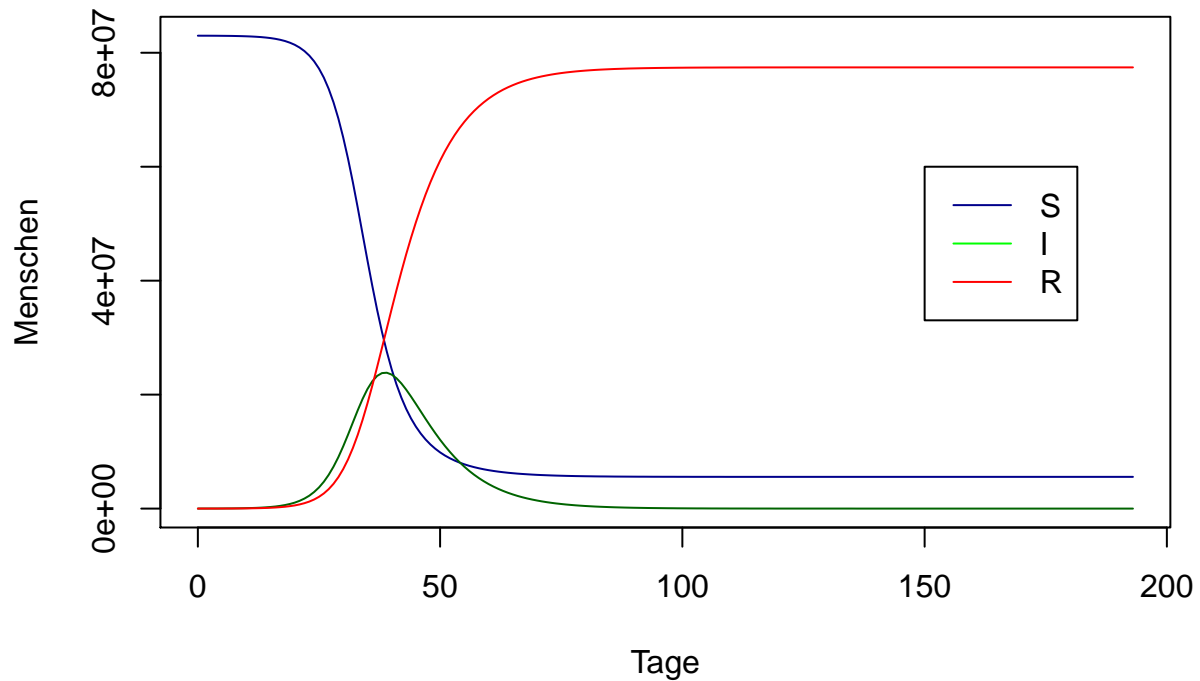
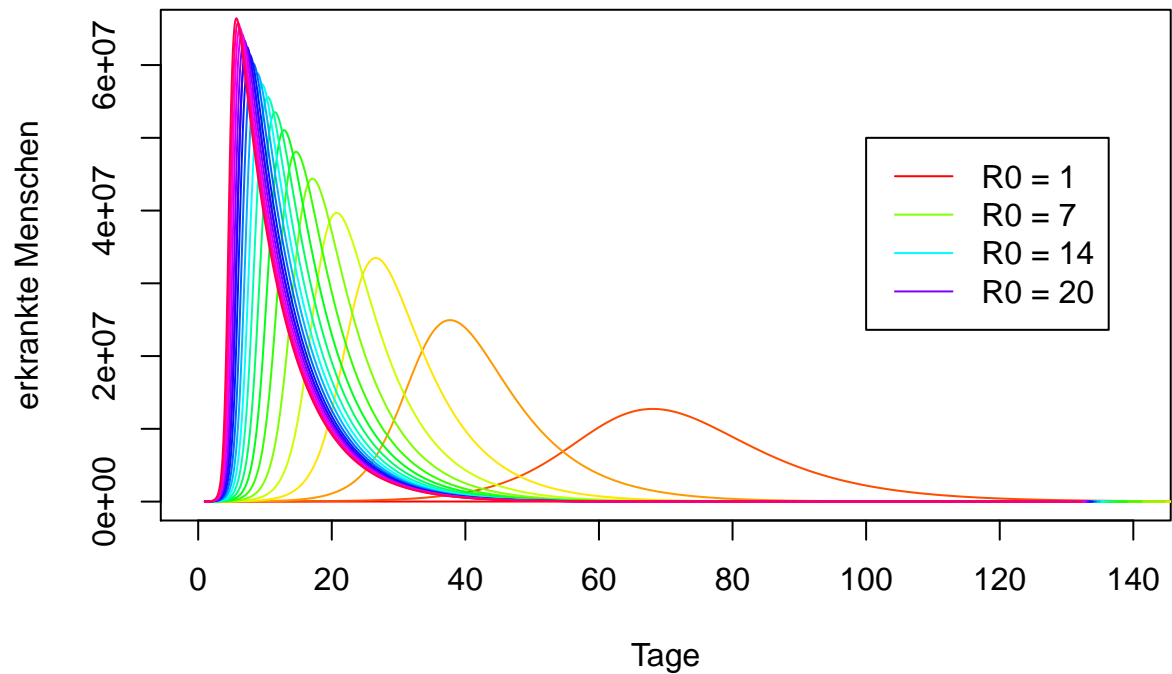


Abbildung 1: Entwicklungskurven für $R_0 = 2.9$

Abb. 1 bestätigt bereits, dass unsere Funktion die erwarteten Infektionskurven ausgibt. Betrachten wir nun die Kurven für $R_0 \in [1, 20]$, so können wir in Abb. ?? sehen, wie sich die Krankheit für große Basisreproduktionszahlen auf wenige Tage fokussiert und die Infektionszahlen in die Höhe steigen. Im Vergleich dazu sind die Kurven für kleine Basisreproduktionszahlen deutlich flacher, strecken sich jedoch auch über ein Vielfaches der Zeit. Für einen echten Krankheitsverlauf würde ersteres die Kapazitäten von Krankenhäusern und anderen Institutionen extrem belasten, während zweiteres lang andauernde Beschränkungen für soziale Interaktionen mit sich bringen könnte.



wollen wir noch prüfen, ob sich die Verläufe nach der Herdenimmunität verhalten. Diese ist gegeben durch:

$$\frac{S}{N} = \frac{1}{R_0} \quad (6)$$

Dazu tragen wir am Ende der Pandemie (d.h. $I(t) < 0$) das Verhältnis der noch nicht Infizierten $S(t)$ gegenüber der Gesamtbevölkerung N in Abhängigkeit von R_0 auf. Zum Vergleich betrachten wir auch den nach der Herdenimmunität erwarteten Kurvenverlauf.

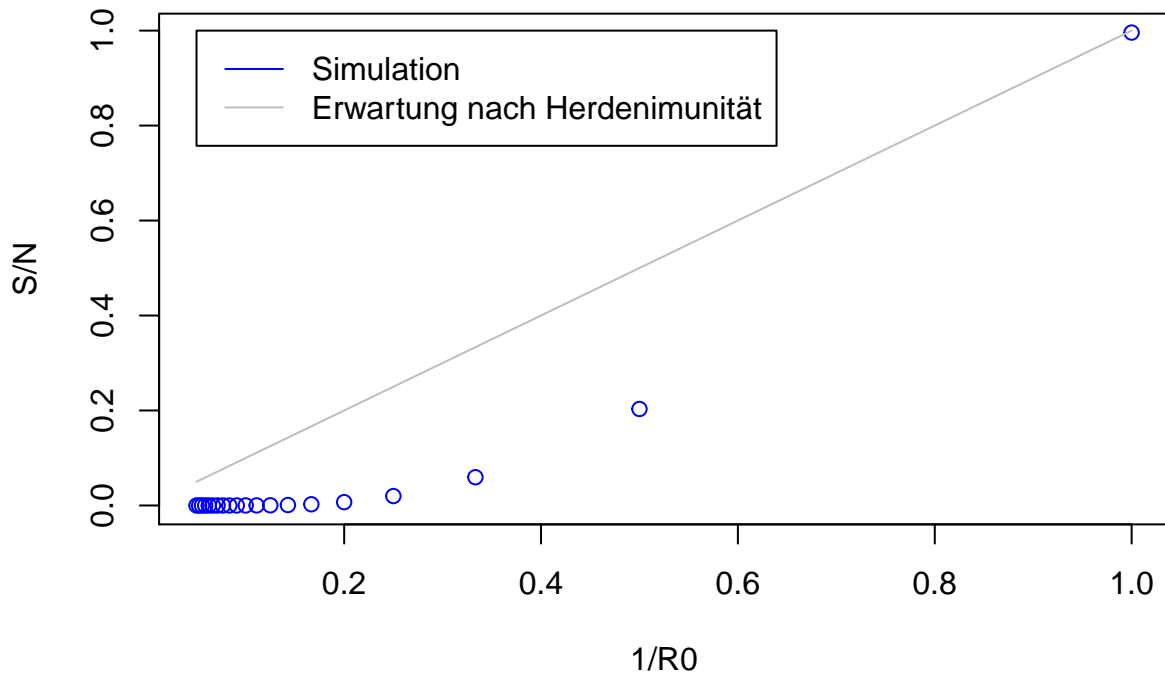


Abbildung 2: R_0 abhängige Herdenimmunität

In Abb. 2 ist zu erkennen, dass der von uns simulierte Verlauf, deutlich von der Herdenimmunität abweicht. Dies sollte an der sehr schnellen Verbreitung der Krankheit für große R_0 Werte liegen. Nach Abb. ?? ist

zu erkennen, dass nach wenigen Wochen, die Krankheit wieder abklingt, dies liegt jedoch daran, dass alle gesunden Menschen schon angesteckt wurden und damit keinen neuen Wirte mehr vorhanden sind. Diese kleine Menge an "gesunden" Menschen nach der Pandemie erklärt die große Abweichung zur Herdenimmunität, bei der größere Teile der Bevölkerung nach der Pandemie nie erkrankt waren. Es ist jedoch zu erwähnen, dass die Simulation ihre Schwächen hat, da durch keine Gegenmaßnahmen das Infektionswesen während der ganzen Zeit für große R_0 Werte auch groß bleibt. Unter Einbeziehung von Gegenmaßnahmen sollte man sich der Erwartung nach Herdenimmunität nähern.

Mit Gegenmaßnahmen

Unter der Verwendung der Daten des RKI versuchen wir den COVID-19-Verlauf zu modellieren. Als Startwert wurden die Daten des 03.01.2021 [1] mit $I = 349500$, $R = 1765666$, $V = 238809$ verwendet. Desweiteren wurde γ und N_0 als Treppenfunktion mit den Werten

Monat	Tag im Jahr	Impfungsrate γ	aktive Fälle I
Januar:	3	40665	349500
Februar:	37	61132	193200
März:	71	60776	127100
April:	110	57895	279400
Mai:	142	406455	167100
Juni:	152	337478	95900

Tabelle 1: RKI Daten zur Covid-19 Entwicklung von 3.1.21- 1.6.21 , [1] bis [6]

verwendet. $\frac{1}{\delta}$ wird auf 150 Tage gesetzt.

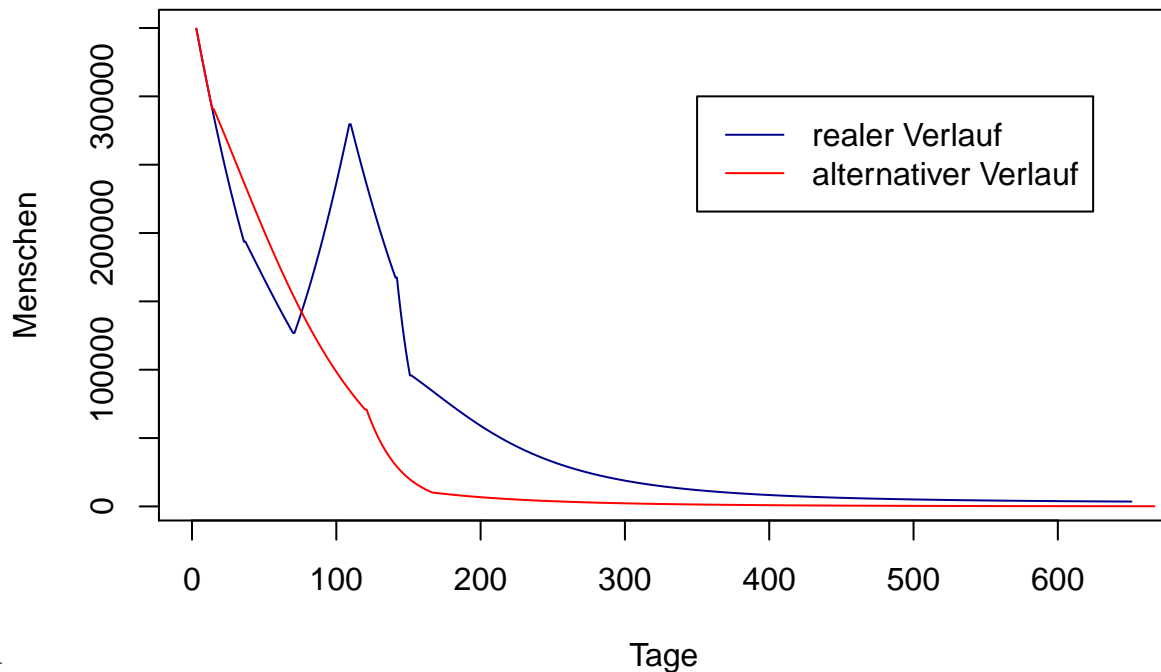
[1] 193693.4

[1] 126912.2

[1] 279589.8

[1] 167387.6

[1] 95831.58



ok

Fazit

Machen wir sowas überhaupt?

Literatur

- [1] *Täglicher Lagebericht des RKI zur Coronavirus-Krankheit-2019(COVID-19)*, Robert Koch Institut, https://www.rki.de/DE/Content/InfAZ/N/Neuartiges_Coronavirus/Situationsberichte/Jan_2021/2021-01-03-de.pdf?__blob=publicationFile, Stand: 24.05.2021.
- [2] *Täglicher Lagebericht des RKI zur Coronavirus-Krankheit-2019(COVID-19)*, Robert Koch Institut, https://www.rki.de/DE/Content/InfAZ/N/Neuartiges_Coronavirus/Situationsberichte/Feb_2021/2021-02-06-de.pdf?__blob=publicationFile, Stand: 24.05.2021.
- [3] *Täglicher Lagebericht des RKI zur Coronavirus-Krankheit-2019(COVID-19)*, Robert Koch Institut, https://www.rki.de/DE/Content/InfAZ/N/Neuartiges_Coronavirus/Situationsberichte/Maerz_2021/2021-03-12-de.pdf?__blob=publicationFile, Stand: 24.05.2021.
- [4] *Täglicher Lagebericht des RKI zur Coronavirus-Krankheit-2019(COVID-19)*, Robert Koch Institut, https://www.rki.de/DE/Content/InfAZ/N/Neuartiges_Coronavirus/Situationsberichte/Apr_2021/2021-04-20-de.pdf?__blob=publicationFile, Stand: 24.05.2021.
- [5] *Täglicher Lagebericht des RKI zur Coronavirus-Krankheit-2019(COVID-19)*, Robert Koch Institut, https://www.rki.de/DE/Content/InfAZ/N/Neuartiges_Coronavirus/Situationsberichte/Mai_2021/2021-05-22-de.pdf?__blob=publicationFile, Stand: 24.05.2021.
- [6] *Täglicher Lagebericht des RKI zur Coronavirus-Krankheit-2019(COVID-19)*, Robert Koch Institut, https://www.rki.de/DE/Content/InfAZ/N/Neuartiges_Coronavirus/Situationsberichte/Jun_2021/2021-06-01-de.pdf?__blob=publicationFile, Stand: 06.06.2021.