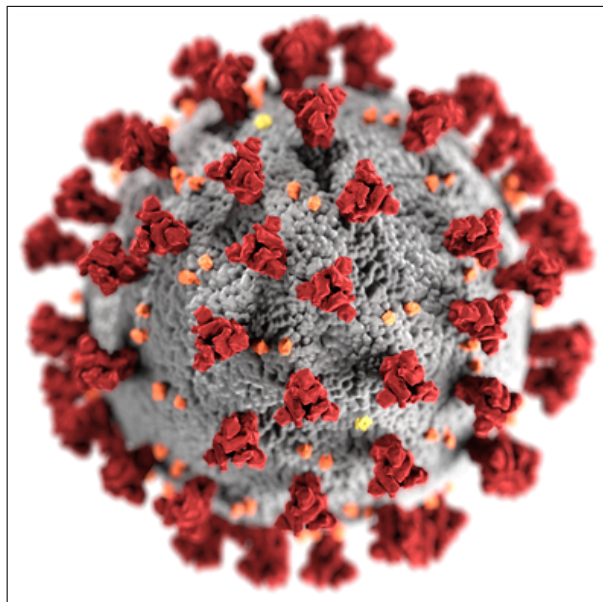


Epidemie

Modellierung und Simulation

Prof. Dr. Rebekka Axthelm (HTWG)



Tutorial

–1–

Ein epidemischer Verlauf

Wir wollen die Verbreitung eines Virus in einer bewegten Menschenmenge simulieren. Kurven $S, I, R : [0, T_{\max}] \rightarrow \mathbb{R}$ beschreiben jeweils die Anzahl Personen, die zu einem bestimmten Zeitpunkt $t \in [0, T_{\max}]$ anfällig ($S(t) = \text{susceptible}$), infektiös ($I(t) = \text{infected}$) oder genesen bzw. gestorben ($R(t) = \text{removed}$) sind. Genesen oder gestorben unterscheiden wir nicht, da jede*r Genesene endgültig immun sein soll. Die Gesamtzahl der Personen $\text{NumPerson} = S(t) + I(t) + R(t)$ ist konstant in der Zeit und soll im Programm als Parameter angegeben werden können. Ziel ist es die Kurven aus der Simulation, welche diskrete Funktionen sind, durch kontinuierliche Funktionen möglichst gut zu approximieren. Das ist eine typische Problemstellung wenn man mit Messwerten arbeitet.

Die Aufgaben teilen sich in drei Abschnitte:

1. Aufstellen des Programms.
2. Untersuchung der angegebenen kontinuierlichen Funktion auf Extremwerte und Wendepunkte.
3. Untersuchung der diskreten Funktionen aus der Simulation auf Extremwerte und Wendepunkte. Diese Untersuchung liefert Zahlen für die Parameter der kontinuierlichen Funktion.

Das Programm `HumanBrown` berechne die Bewegung einer Menschengruppe. Kommen sich zwei Personen näher als eine vorgegebene Distanz (critical distance `CD`, Parameter) so steckt die infizierte die noch nicht infizierte Person (falls das jeweils die Situation ist) mit einer gewissen Wahrscheinlichkeit (infection rate `IR`, Parameter) an. Jede Person hat 5 Eigenschaften zu jedem Zeitpunkt t_i :

1. x-Koordinate der Position
2. y-Koordinate der Position
3. infektiös (0/1)
4. Anzahl infizierte Tage
5. genesen (1/0)



Abbildung 1:
DemoVideo

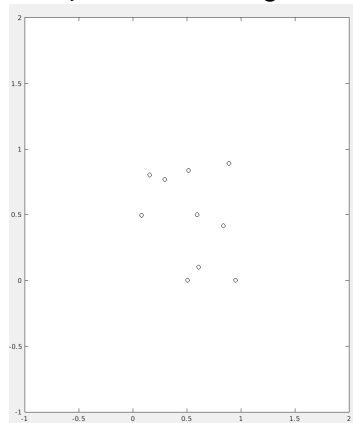
Einen Plot der "Kurven" $S(t)$, $I(t)$, $R(t)$ können Sie sich im DemoVideo (QR-Code in Abbildung 1) anschauen. Was Sie erhalten sind diskrete Funktionen. Die Aufgabe besteht am Ende darin, analytische Funktionen zu bestimmen, bzw. die Parameter von gegebenen Kurven zu finden, die den Krankheitsverlauf des Virus bestmöglich darstellt. Im ersten Teil erstellen Sie das Simulationsprogramm, im zweiten Teil untersuchen Sie vorgeschlagene analytische Funktionen und im dritten und letzten Schritt finden Sie typische Merkmale in den diskreten Kurven. Diese Merkmale helfen Ihnen dann bei der Wahl der Parameter in den analytischen Funktion, so dass diese Ihre Form "gut" an die Simulationsdaten anpassen. Solche Funktionen erlauben es dann Zukunftsprognosen zu stellen. Sie können es auch gerne mit Coronadaten statt der simulierten Werte versuchen.

1.1 Arbeitsanleitung: Programm

Schritt 1: Erstellen Sie für 10 (=NumPerson) Personen zufällige Positionen im Raum der Größe $[0, 1]^2$ mit dem Befehl

```
> Persons = rand(NumPerson,2)
```

und ploten Sie ihre Protagonisten schon mal. Es sollte ungefähr so aussehen (wegen der Zufallszahlen variiert es bei jeder Rechnung):



Die erste Spalte in `Persons` beinhaltet die x- und die zweite Spalte die y-Koordinate.

Schritt 2: Jede Personen P_i soll sich nun bewegen. Zu jedem Zeitschritt k mit einer Geschwindigkeit von CV (=Parameter) in eine zufällige Richtung, d.h. sie bewegen sich in Richtung

$$v_i^k = CV \begin{pmatrix} \cos \varphi_i^k \\ \sin \varphi_i^k \end{pmatrix},$$

wobei

```
> phi = rand*2*pi;
```

und

```
> P = P + CV*[cos(phi), sin(phi)];
```

Für jede Person wird zu jedem Zeitschritt ein zufälliges φ berechnet. Diesen Vorgang wiederholen Sie LOOPmax (=Parameter) mal.

```
for loop=1:LOOPmax
    phi = rand*2*pi;
    P = P + CV*[cos(phi), sin(phi)];
    plot(P(:,1),P(:,2),'ko');
end
```

Schritt 3: Eine bestimmte Anzahl (NumIllInit=1, Parameter) von zufällig gewählten Personen zum Startzeitpunkt $k = 1$ ist bereits infiziert.

Tipp: Die Matlabfunktion randperm(N) erzeugt ein Feld mit Zahlen 1 bis N in zufälliger Reihenfolge. Die ersten NumIllInit Nummern sind die zu Beginn kranken Personen.

```
RP = randperm(NumPersons);
Person(RP(1:NumIllInit),3:4) = 1;
```

Infizierte Personen ploten Sie rot.

Tipp: Die dritte Spalte in Person beinhaltet eine 1 wenn die entsprechende Person krank ist sonst eine 0. Die vierte Spalte zählt die "Tage" (Zeiteinheiten) der Krankheitsdauer. Die fünfte Spalte enthält eine 1 wenn die Person bereits genesen und damit immun ist sonst eine 0.

Genesene Personen werden in grün geplottet.

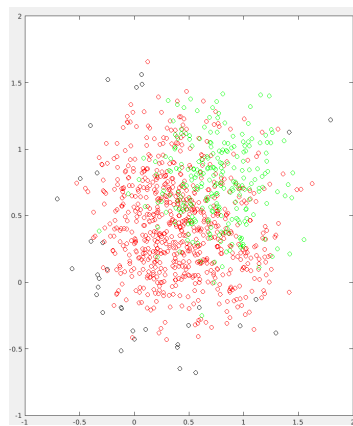
Tipp:

```
> Ind = find(Person(:,3)==1)
```

Ind beinhaltet die Personennummern der Genesenen.

```
IndI = find(Person(:,3)==1);  
IndS = find(Person(:,3)==0 & Person(:,5)==0);  
IndR = find(Person(:,5)==1);  
plot(Person(IndS,1),Person(IndS,2),'ko',...  
      Person(IndI,1),Person(IndI,2),'ro',...  
      Person(IndR,1),Person(IndR,2),'go');
```

Im Laufe der Zeit könnte es sich für 1000 Personen so darstellen:



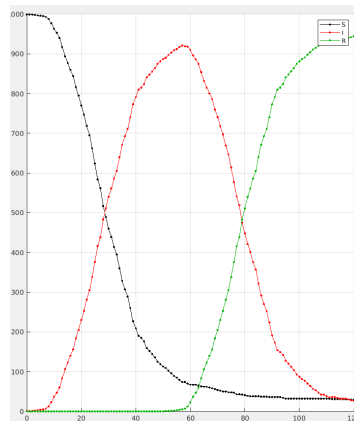
Dazu müssen sich die Personen auch gegenseitig anstecken und auch wieder genesen können.

Schritt 3: In diesem Schritt berechnen Sie paarweise die Abstände der Personen.

```
for j=1:NumPerson  
    for i=j+1:NumPerson  
        Dist = norm(P(i,1:2)-P(j,1:2))  
        if Dist > CD ...  
        end  
    end  
end
```

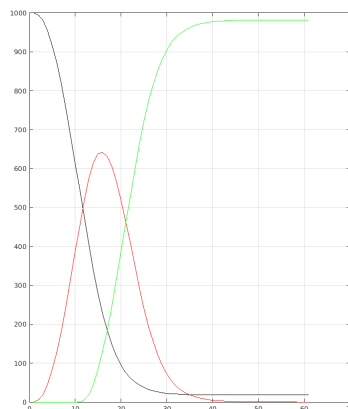
In der if-Abfrage geht es darum ob eine infizierte (I) Person P^i auf eine noch nicht infizierte (S) Person P^j trifft. Dann wird letztere mit einer Wahrscheinlichkeit von IR (infection rate, Parameter) angesteckt. In jedem Zeitloop müssen die "Krankheitstage" der Infizierten hochgezählt werden. Nicht vergessen.

Schritt 4: Die resultierende Kurven (Anzahl Personen in S, I und R zu jedem Zeitpunkt) ploten Sie. Könnte so aussehen:



Das sind die diskreten Funktionen mit denen wir arbeiten wollen.

Schritt 5: Da viele Zufallszahlen verwendet wurden ist unsere Gesamtmessung etwas verrauscht, was das postprocessing etwas schwierig macht. In der Praxis ist genau das auch ein generelles Problem. Wir können dieser Schwierigkeit schon einmal so begegnen, dass wir die Berechnungen von Schritt 1 bis 4 in eine Funktion `HumanBrown` setzen, die die Kurven S, I und R als Rückgabewert liefert. Wiederholen Sie die Berechnung nun mehrmals (`LOOPmax=100`, Parameter) und berechnen von allen Ergebnissen den Mittelwert. Sie erhalten dann ein wenig "glattere" Ergebnisse.



Am besten Sie speichern Ihr Ergebnis, also die diskreten Funktionen S, I und R in eine Datei.

```
fp = fopen('datei.dat','w');  
for i=1:TimeLOOPmax  
    fprintf(fp,'%f %f %f \n',...  
        SIR(i,1),SIR(i,2),SIR(i,3));  
end  
fclose(fp);
```

1.2 Untersuchung der analytischen Funktion

Schritt 6: Machen Sie sich dem "Anblick" der angegebenen Funktion f (logistische Funktion) ein wenig vertraut:

$$f(t) = G \cdot \frac{1}{1 + e^{-k G t} \left(\frac{G}{f(0)} - 1 \right)}.$$

Graph:

<https://www.geogebra.org/m/cfammtpe#material/tzxkeq5>

Schritt 7: Berechnen Sie

$$f'(t) =$$

$$f''(t) =$$

und untersuchen f' auf Extrema, bzw. f'' auf Wendepunkte.

1.3 Untersuchung diskreter Funktionen in einem separaten Programm Postprocessing

Schritt 8: Laden Sie Ihre diskreten Funktionen aus der Datei data.dat.

```
>SIR = load('data.dat');
```

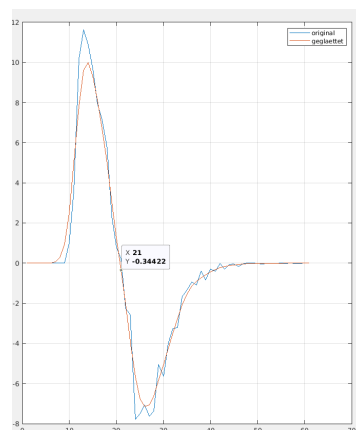
Berechnen Sie erste und zweite Ableitungen (im Programm) Ihrer diskreten Funktionen aus der Simulation.

```
% Matrix zur Erzeugung der zweiten Ableitung  
D2 = -2*diag(ones(M,1))...  
      +diag(ones(M-1,1),1)...  
      +diag(ones(M-1,1),-1)  
% Berechnung der zweiten Ableitung von R=SIR(:,3)  
d2R = D2*R
```

Unter Umständen werden die Ableitungen der diskreten Funktionen trotz Mittelung nach wie vor viel Rauschen aufweisen. Sie können eine Glättung vornehmen durch eine Art Tiefpassfilter:

```
for l=1:10 % oder mehr  
    % delta sollte klein sein, max. etwa 0.1  
    R = R + delta*d2R  
end
```

Probieren Sie das mal aus.



Schritt 9: Untersuchen Sie die diskreten Funktionen S , I , R aus Ihrer Simulation auf Maxima und Wendepunkte.

Schritt 10: Bestimmen Sie nun geeignete Parameter für die analytischen Funktion f , so dass diese die diskreten Funktionen möglichst gut approximieren. Sie entscheiden was genau was approximieren soll.

Schritt 11: Stellen Sie Ihr Ergebnis als plot vor und geben Sie alle verwendeten Parameter und ermittelten Parameter von f an.