ETHlogo

**Lecture with Computer Exercises:**

**Modelling and Simulating Social Systems with MATLAB**

Project Report

|  |
| --- |
| Stochastic Simulation of Epidemic Outbreak in a large Network |

Jordi Christian, Schmid Yannick & Stücheli Pascal

Zürich

10. Mai 2012

**IMPORTANT**

**You MUST include the ETH declaration of originality here; it is available for download on the course website or at**

**http://www.ethz.ch/faculty/exams/plagiarism/index\_EN;**

**It can be printed as pdf and should be filled out in handwriting.**

**Agreement for free-download**

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

|  |  |
| --- | --- |
| Jordi Christian | Schmid Yannick |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
| Stücheli Pascal |  |

**Table of contents**

Evtl die funktionen nicht alle aufzählen -> überschrift ändern

[Abstract 4](#_Toc325584822)

[1 Individual contributions 5](#_Toc325584823)

[2 Introduction and Motivations 6](#_Toc325584824)

[3 Description of the Model 7](#_Toc325584825)

[3.1 The continuous disease model 7](#_Toc325584826)

[3.2 The stochastic disease model 7](#_Toc325584827)

[3.3 The network 8](#_Toc325584828)

[3.4 The transport 9](#_Toc325584829)

[4 Implementation 11](#_Toc325584830)

[4.1 Network generation 11](#_Toc325584831)

[4.2 Simulation of the infection 11](#_Toc325584832)

[4.3 Transport 11](#_Toc325584833)

[4.4 Final Simulation 12](#_Toc325584834)

[5 Simulation Results and Discussion 13](#_Toc325584835)

[Summary and Outlook 14](#_Toc325584836)

[Outlook 14](#_Toc325584837)

[References 15](#_Toc325584838)

[MATLAB code 16](#_Toc325584839)

[Initialisation.m 16](#_Toc325584840)

[networkepid2loops.m 16](#_Toc325584841)

[city\_size.m 18](#_Toc325584842)

[generate\_fixed\_tot\_T 19](#_Toc325584843)

# Abstract

# 1 Individual contributions

We initially divided the programming part which was easily feasible since there were three main tasks. Pascal Stücheli was responsible for the epidemic model and its implementation, Christian Jordi familiarized himself in with GEPHI and was responsible for the network generation. Yannick Schmid wrote the transport function. At any time we assisted each other because we had to make sure that the functions worked well together. Further we had to increase the overall performance of our simulation, mainly by making approximation or changing the structure of certain functions. At this point it was important to discuss the possibilities and limitations of the approximations and all of us contributed ideas and concerns. The analysis and interpretation of the results and also the writing of the report was done in a cooperative manner.

# 2 Introduction and Motivations

Last winter, the movie Contagion*[[1]](#footnote-1)* came to the cinemas. In this movie the fates of some characters are shown during the outbreak of a deadly viral infection disease. The first infected person in the movie was a travelling business woman, who carried the virus across two continents such that the disease could spread over the whole world very rapidly. After seeing this movie we asked ourselves how a realistic disease would behave if no remedy or cure could be found. Inspired by this movie, we thought about a large network representing the big cities on the world, which are either connected by air route or other means of public transportation.

Surveillance and simulation of infectious disease are necessary to predict possible outbreaks of epidemics. At the moment there are of course already several epidemic simulation tools. Some of them also contain stochastic effects like example given FluTE. Therefore we do not claim to invent something new, but we wanted to see if we would be able to implement an epidemic simulation in a large network with ordinary computational power.

The aim of this study is to follow the time evolution of discrete numbers of persons who are either susceptible or infected and the spreading of infected persons among a large network. We want to emphasize that we tried to approximate the stochastic simulation in our own way.

We wanted to investigate the following questions:

* How does the number of total infected in a network vary for a given time of simulation?
* How does the time vary among the different simulation until the whole population in the network has been infected?
* What are the variations for certain cities with different degrees? What are the differences for hubs and leaves in the network?
* Are there tendencies in the regional spreading like for example towards hubs?
* What happens if we implement a healing rate?????

Additionally we were also interest if we could handle such an amount of data. Actually we spent a lot of time improving the performance of our simulation and exploring approximations.

# 3 Description of the Model

3.1 The continuous disease model

Our model is basically a simplification of the SIR model proposed by Kermack and McKendrick[[2]](#footnote-2). The major difference is we did not consider the state R that represents removed people which became immune or died after being infected.

To derive the disease model a city with N inhabitants is considered. The inhabitants are either susceptible for the disease or they already carry the disease and can therefore infect susceptible persons. In the equations below susceptible persons are labelled S and infected persons I. At a constant birth rate B healthy (and therefore susceptible) children are born while dyeing is a first order process with a death rate d. The amount of newly infected persons per time unit depends on the availability of susceptible persons and their meetings with infected persons. The contact rate β is the rate at which susceptible persons become infected after a meeting with an infected person.

For further simplification we assume that the number of newborn balances the number of dyeing people per time unit in order to neglect the birth and death in the city.

3.2 The stochastic disease model

Since the goal of this study is to follow discrete numbers of infected persons and to integrate noise, we obviously ended up with a stochastic model. For a proper stochastic simulation spatial homogeneity is assumed. This means the population in a city is well mixed and there are no spatial accumulations of either susceptible or infected persons.

In the SI model there is only one transition possible.

Where s denotes a susceptible person j or k and i denotes an infected person j or k. The propensity a is the probability per time unit for a successful infection. The propensity depends on the meetings between susceptible and infected persons as well as on the rate of successful infections after meetings.

Hypergeometrische Verteilung erklären...

Ansteckung bei Zusammentreffen von einem S und einem I -> mean +/- std \* error (resp. randn).

3.3 The network

Instead of modelling an existing city network an artificial scale free network was created. A scale free network was chosen, because its capability to form hubs, which we compared with metropoles that are highly interconnected and surrounded by more isolated cities of smaller size. Hence the network would at least be a sufficient model for the air traffic between the cities, as stated by M. Beliant et al.

The scale free network was generated using an implementation of the Barabási-Albert algorithm, which we learned in an exercise of Prof. D.Iber’s lecture on mathematical modelling. []

The idea of this algorithm is to create a network around a couple of seed nodes by adding new ones step by step. Thereby the probability to form an edge to target existing node j is given by:

In this way nodes with a high degree kj are more likely to be selected and a scale free network is created. [mamo]

Initially a loop free version of the algorithm was tested but the created network did not look interconnected enough. Therefore a loop forming version was used in the simulation (Fig. 1)

|  |
| --- |
| 100unlooped.png100loop.png |
| **Fig. 1** Two examples of 100 node networks generated by a Barabási-Albert Model. A: loop free version. B: loop forming version  Gephi 0.81 beta was used for graph visualization |

Erklärung der Formel und Kennzahlen des NetzwerksIt is well known that the population of cities can be described by a power law (Zipf’s Law). Prof. Guido Spars et al [] determined the Zipf parameters of german cities with high accuracy. We assigned a population to each node according to its rank in the network and the value predicted by Zipf’s Law

Our final network consisted out 10 000 cities and had a population of ??? Millions

3.4 The transport

People from a city x are able to travel to city y if the nodes x and y are connected by an edge. However, we want the total population of a city to be constant. Therefore the same number of persons travel from city x to city y as vice versa while the composition of the travellers (susceptible or infected) can vary. If a fixed number of total travellers per simulation day are assumed – like for example seats in airplanes – the portion of infected travellers is hyper geometrically distributed. Txy corresponds to the total travellers on the edge between city x and city y and is equivalent to the number of drawings without laying back out of a population with Sx,y + Ix,y people and Ix,y as the number of infected persons[[3]](#footnote-3).

|  |
| --- |
|  |
| **Fig. 2** Sketch of two cities x and y where S and I correspond to the number of susceptible or infected persons. Edges are indicated as arrows. T denotes the total number of travellers along an edge and is the same in both directions of the edge. |

The number of total travellers along an edge should obviously not exceed the population of either city connected by the specific edge. It must also be ensured that cities with multiple edges do not have higher traveller fluxes than populations. Also a factor depending on the degrees k of the connected cities compared to the average degree in the network was included to contribute more transports between cities with many connections – intuitively that is like cities with more airports and stations have more transport capacity. Finally we included a factor to reduce the transport of infected people assuming that people who feel sick travel less.

Actually, it would have been nicer if the degree factor would be divided by two. However, since the transport rate is further reduced[[4]](#footnote-4), this flaw is corrected.

# 4 Implementation

*4.1 Network generation*

The function *networkepid2loops* was used to generatet he scale free network. After the definition of the seed nodes and a seed edge list, a network of 10‘000 nodes was created, by introducing new nodes one after another. Each new node was enabled to form two edges to existing nodes. Hence loops could be formed.

The targeted loops were chosen randomly and their pj was calculated. Therafter a random number between zero and one was generated and compared with pj. If the random number was smaller the edge was formed, otherwise a new target was chosen.

The edge list was saved as a nx2 matrix. Every row contains the two nodes connected by the edge. To import the matrix into the graph visualization tool Gephi, it had to be converted using the function *cell2csv*[[5]](#footnote-5)

The nodes were stored in *cities.txt*, a mx3 matrix, where every row stands for a city in the network. The degrees of the nodes were stored in the first column of this matrix and then passed to the function *city\_size*

*city\_size* ranked the cities according to their degree and used Zipf’s law to assign a population size to each city. Thereby a variation of 1% was added to the predicted size to reduce the impact of the sequence on cities that have the same degree.

The whole process of generating the network was collected in the function *initialization.*

4.2 Simulation of the infection

As the name indicates, the function Simulate\_Infection computes the number of infections per city and per time step. Therefore the number of meetings between infected and healthy persons is calculated according to an arbitrarily chosen mean and standard deviation of daily meetings with other persons. Further, a Gaussian distributed error was included. As long as the number of infected people in a city is below a certain threshold the infection is simulated in a stochastic manner. The number of meetings which result in an infection is a hyper geometrically distributed number, as mentioned in section 3.1. Because MATLAB is very slow in drawing hyper geometrically distributed random numbers, a binomial distributed random number was used as an approximation. The drawback of the binomial distribution in this situation is that a person can be infected more than one time. However, when the number of infected people exceeds the threshold, the simulation works in a deterministic manner and is equal to the expectation value.

4.3 Transport

According to section 3.4 the total number of travellers along every edge is calculated before the actual simulation. Then for every city and time step to number of infected persons travelling is simulated. The transport function uses the edges matrix and in order to avoid constraints from calculating always in the same order along the edges matrix, the rows of the matrix are shuffled in every time step. Again, we have hyper geometrical distribution for the number of infected travellers. So, the mean and the standard deviation of the hyper geometrical distribution are computed. This is done with the MATLAB function hygestat. A Gaussian distributed error is included to introduce noise in the transport.

4.4 Final Simulation

# 5 Simulation Results and Discussion

# Summary and Outlook

Outlook

Obviously, the next step would be to implement real data and an existing network. For the network it would make sense to take a continent or even the whole world as long as only the important cities are considered, such that the performance does not suffer too much. It would be interesting to weight the edges according to geographic distances and to include real transportation data.

Furthermore, the integration of either a healing rate or an immunisation would extend the simulation and make it more realistic.

# References

# MATLAB code

Ha mou oni gross z formatiere d funktione vo mir do igfüegt.

Initialisation.m

function[]=Initialisation()

%Summary of the functions needed to create our start network. bcs

%everything needed is stored in .txt or .csv files. -> has only to be

%executed once

%This function generates: -cities.txt w/o infection

% -edges.txt

% -network.csv

% -tot\_T.txt

ncit = 10000; %number of cities in the Network

seeded = [1 5; 2 3; 4 5;4 3];%seed for the edges

seedct = [1 0 0;1 0 0;1 0 0;1 0 0;2 0 0]; %seed for the cities

dt=2; %2hours (timesteps of the simulation) -> needed to scale the

%passenger volumina

%Generation of edge and node lists of a network with loops

[cities,edge]=networkepid2loops(ncit,seeded,seedct);

%Generation of the city populations

cities=city\_size(cities); %output is cities.txt

%Generation of the traffic volumina on each edge

tot\_T = generate\_fixed\_tot\_T(cities,edge,dt);

%Stores the traffic volumina for the simulation. The other data sets were

%already stored by the functions generating them.

dlmwrite('tot\_T.txt',tot\_T);

end

networkepid2loops.m

function[cities,edge]=networkepid2loops(ncit,seeded,seedct)

%Generation of a scale free network, with the possibility to form loops

%Input parameters are:

%ncit: integer, number of nodes in the generated network

%seeded: seed for the edges matrix. array of size (nx2)

%seedct: seed for the city matrix. array of size (mx3)

%Output is:

%cities: matrix of size ncitx3, the first column is filled, the two others

%are blank

%edge: matrix of size yx2, undirected edge list of the graph {city,city2;..]

%Additional output: -a .txt file of the edge list to be used by other

% functions

% -a .csv file of the edge list, to be used by gephi

cities=zeros(ncit,3);

pos= length(seedct(:,1));

cities(1:pos,1:3)=seedct; %Stores (degree,Inhabitants,Infected)

edge=seeded; %Connections btwn the cities, store lower index first

pass=0; %number of passengers on an edge %%%%%%not used yet%%%%%%%%%

mlinks=2; %number of links that are fixed at each step increased to two

sumlinks=length(edge(:,1)); %total number of edges

linkage=0; %Determines wether a new edge has been set

%functions

while pos < ncit

pos= pos+1;

posed=length(edge)+1;

linkage=0;

while linkage ~= mlinks %true as long as the new edge has not been set

%pos= pos+1;

%posed=length(edge)+1;

rnode = ceil(rand \* pos); %randomly choose a node in front of the current position

deg = cities(rnode,1); %degree of the chosen node

rlink = rand; %random number to determine wether an edge is formed

if rlink < deg / sumlinks && isequal(edge(length(edge(:,1)),:),[rnode,pos])~=1%&& Net(pos,rnode) ~= 1 && Net(rnode,pos) ~= 1 as long as only one linkage is set per step

% (mlinks=1) the oher two conditions do not matter

cities(pos,1)=cities(pos,1)+1; %degree of the new introduced city is increased by one

cities(rnode,1)=cities(rnode,1)+1; %The deg of the chosen city is increased by one

edge(posed,1)=rnode; %stores the new edge

edge(posed,2)=pos;

sumlinks=length(edge(:,1));

linkage=linkage+1;

posed=posed+1;

end

end

end

%bring data in a form gephi can understand -> convert into a cell to use

%the provided function cell2csv.

edgecell=num2cell(edge);

edgecell2{length(edgecell),3}=0;

%Beside the Source and Target nodes an additional column is needed so that

%gephi generates an undirected graph

for i=1:length(edgecell)

edgecell2{i+1,1}=edgecell{i,1};

edgecell2{i+1,2}=edgecell{i,2};

edgecell2{i+1,3}='undirected';

end

%The columns need titles, so that gephi recognizes them

edgecell2{1,1}='source';

edgecell2{1,2}='target';

edgecell2{1,3}='type';

%Save the data as a .csv for gephi and as .txt for the simulation

cell2csv('smallernetwork10kfinalloop.csv', edgecell2, [], 2007, [])

dlmwrite('edges.txt',edge);

end

city\_size.m

function[cities]=city\_size(cities)

%Code to determine the size of the cities

%Input parameters:

%cities: city matrix of format mx3, where the first column is filled with

%the degree of the city in the Network

%Output:

%cities: modified version of the input matrix. The city population

%(cities(:,2)) has been added

%Additional output: cities.txt stores the city matrix, so it can be reused

%rank the cities depending on degree

rank=zeros(length(cities(:,1)),1);

rank(1)=1;

for i=2:length(cities(:,1))

%n=i;

y=1;

while y<i

if cities(i,1)>=cities(rank(y),1)

for k=(i-1):-1:y

rank(k+1)=rank(k);

end

rank(y)=i;

y=i; %to break while loop

elseif y==i-1

rank(i)=i;

y=i;

else

y=y+1;

end

end

end

%Now use zipfs law to generate the city sizes.

%the parameters ln(a)= 18.6 b=1.23 are used (see 3.3 for further info)

a=exp(18.6);

b=1.23;

for i=1:length(cities(:,1))

zipf=exp(-log(i/a)/b);

cities(rank(i),2)= round(zipf+randn\*zipf\*0.01); %Added 1% normal distributed noise

end

%Store modified file

dlmwrite('cities.txt',cities);

end

ha de no es paar sache vom video här...

generate\_fixed\_tot\_T

...

...

1. Contagion. Dir. Steven Soderbergh. Perf. Matt Damon, Kate Winslet, Jude Law. Warner Bros. Pictures, 2011. [↑](#footnote-ref-1)
2. Kermack, W.O. and McKendrick, A.G. "A Contribution to the Mathematical Theory of Epidemics." Proc. Roy. Soc. Lond. A **115**, 700-721, 1927. [↑](#footnote-ref-2)
3. The colon of Sx,y and Ix,y means „or“. [↑](#footnote-ref-3)
4. The factor is not shown in the equation above, but was included in the simulation (see the FINAL\_SIMULATION.m files). [↑](#footnote-ref-4)
5. Provided on the lecture site [↑](#footnote-ref-5)