

SIR model

(Susceptible-Infected-Resistant/Removed)

Janusz Szwabiński

Outlook

- Introduction into SIR model
- Analytical approximation
- Numerical solution
- Numerical solution on a grid
- Simulation on networks
- Bibliography

Introduction into SIR model

The impact of epidemics

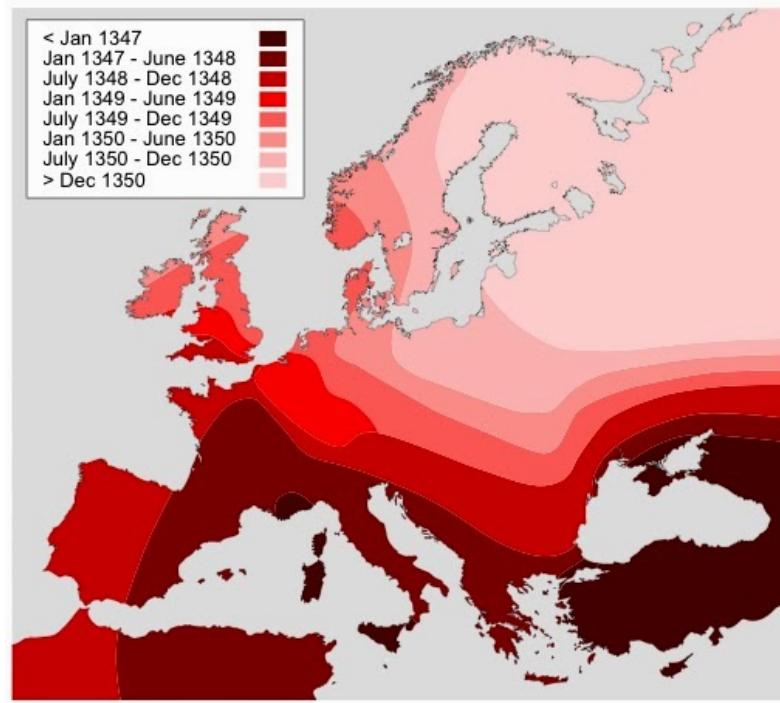
Plague of Athens

- epidemic which devastated the city-state of Athens in ancient Greece during the second year of the Peloponnesian War (430 BC)
- it is believed to have entered Athens through Piraeus
- much of the eastern Mediterranean also saw outbreak of the disease, albeit with less impact
- the plague returned twice more, in 429 BC and in the winter of 427/426 BC
- some 30 pathogens have been suggested as causing the plague, including typhus and typhoid



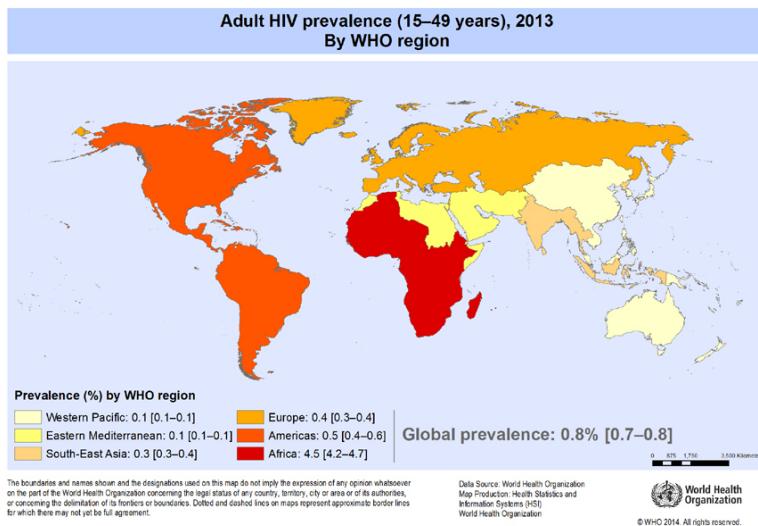
Black death in Europe

- one of the most devastating pandemics in human history
- estimated 75 to 200 million people in Eurasia
- peaking in Europe in the years 1346–1353
- estimated to have killed 30–60% of Europe's total population
- the plague may have reduced the world population from an estimated 450 million down to 350–375 million in the 14th century



HIV (Human Immunodeficiency Virus)

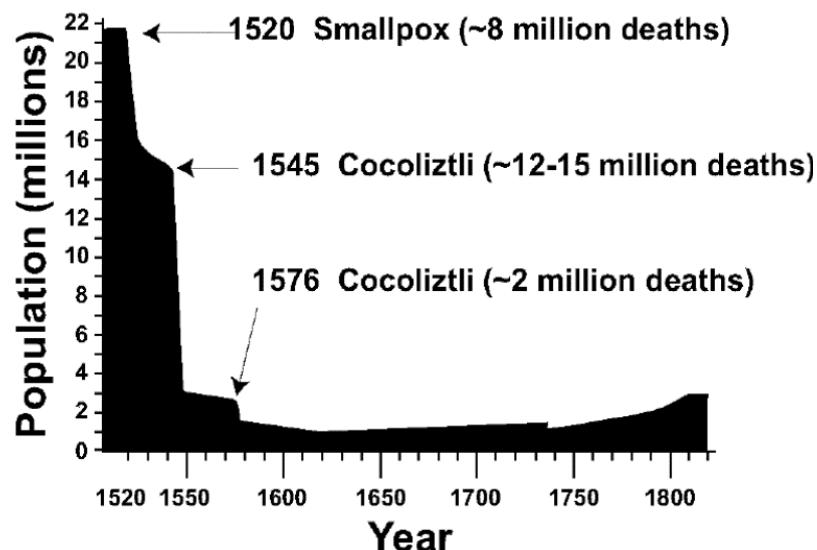
- estimated to have killed 20 million people in Africa since 1970s
- 1.1 million people died of HIV-related illnesses worldwide in 2015 (according to WHO)



Native American disease and epidemics

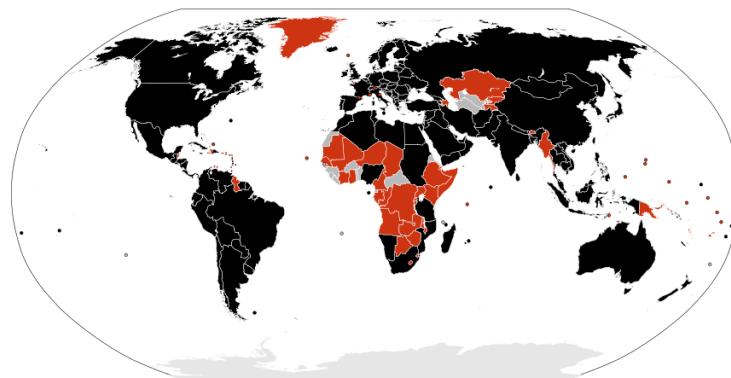
- numerous diseases were brought to North America by European, including Bubonic plague, chickenpox, cholera, the common cold, diphtheria, influenza, malaria, measles, scarlet fever, sexually transmitted diseases, typhoid, typhus, tuberculosis, and pertussis
- each of these brought destruction through sweeping epidemics
- native Americans had not built up internal immunities to the diseases or formed any medicines to combat them

Population Collapse in Mexico



1918 flu pandemic

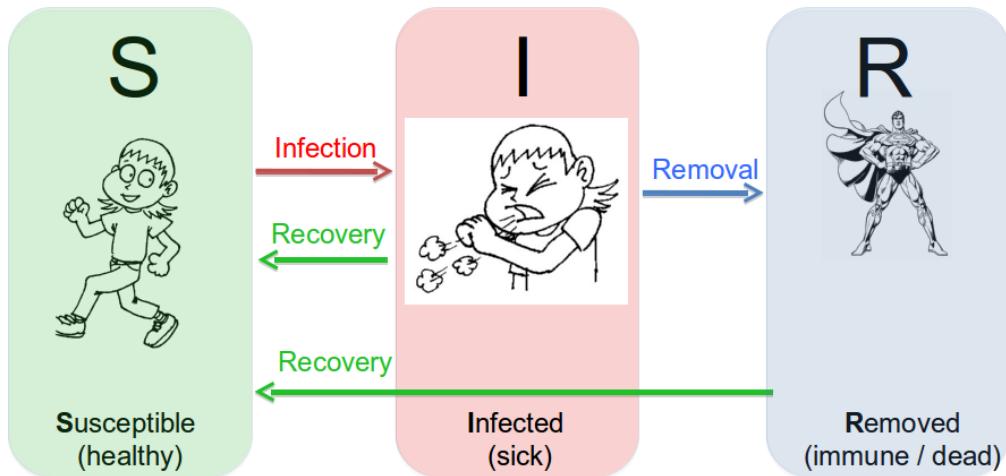
- January 1918 – December 1920
- the first of the two pandemics involving H1N1 influenza virus
- it infected 500 million people across the world and resulted in the deaths of 50 to 100 million (three to five percent of the world's population)
- it reduced life expectancy (e.g. by 12 years in the USA)



SIR model

- models are important for our understanding of epidemic spreading
- first appeared already in 1920s
- first models based on differential equations
- modern approaches relies on computer simulations (microscopic analysis)
- SIR model was proposed by W. O. Kermack and A. G. McKendrick in 1927
- one of the simplest models of epidemics
- applicable to specific diseases like **measles** (pl. odra), **pertussis** (pl. krztusiec) or **HIV**, i.e. to diseases for which individuals die or develop immunity after they recover from the infection
- relies on the concepts of **compartmentalization** and **homogenous mixing**
- compartmentalization means classifying each individual based on the stage of the disease affecting them:
 - Susceptible (S) - healthy individuals who have not yet contacted the pathogen
 - Infectious (I) - contagious individuals who have contacted the pathogen and hence can infect others

- Recovered (R) - individuals who have been infected before, but have recovered from the disease, hence are not infectious
- homogenous mixing hypothesis assumes that each individual has the same chance of coming into contact with an infected individual.



(Source: A.-L. Barabasi, "Network science")

$$\begin{aligned}\frac{dS(t)}{dt} &= -\beta I(t)S(t) \\ \frac{dI(t)}{dt} &= \beta I(t)S(t) - \gamma I(t) \\ \frac{dR(t)}{dt} &= \gamma I(t)\end{aligned}$$

where:

- β - transmission rate
- γ - recovery rate

Analytical approximation

Taking the time derivative of the above equations and performing multiple insertions we arrive at

$$\frac{d^2 S(t)}{dt^2} - \frac{1}{S} \left(\frac{dS(t)}{dt} \right)^2 = \frac{dS(t)}{dt} (\beta S(t) - \gamma)$$

We obtained a nonlinear differential equation for $S(t)$, which cannot be solved exactly. However, by making some assumptions we may still extract some useful informations from the original set of equations.

First of all, it should be noted that just after the outbreak the numbers of susceptibles and infected will be very close to their initial values S_0 and I_0 , respectively. Thus

$$\begin{aligned} S(t) &\simeq S_0 - \delta_S(t) \\ I(t) &\simeq \delta_I(t) \end{aligned}$$

where $\delta_S/S_0 \ll 1$ and $\delta_I/S_0 \ll 1$.

Inserting the above into the model and neglecting the terms of higher orders we obtain:

$$\begin{aligned} \frac{d\delta_S(t)}{dt} &= \beta S_0 \delta_I(t) \\ \frac{d\delta_I(t)}{dt} &= \beta S_0 \delta_I(t) - \gamma \delta_I(t) \end{aligned}$$

Since $I(t=0) = I_0$, from the second equation it follows:

$$I(t) = \delta_I(t) = I_0 e^{(\beta S_0 - \gamma)t}$$

This approximation is reasonable only if δ_I stays small, i.e. if

$$\beta S_0 - \gamma \ll 0 \quad \Rightarrow \quad \beta \ll \frac{\gamma}{S_0}$$

In other words, the disease cannot be very contagious, i.e. the number of susceptibles will change only little with time. Using the solution for δ_I from the first equation we get:

$$\delta_S(t) = \frac{\beta S_0 I_0}{\beta S_0 - \gamma} \left(e^{(\beta S_0 - \gamma)t} - 1 \right)$$

Note that for $t \rightarrow \infty$ the number of healthy individuals goes to some fixed value S_∞ ,

$$S_\infty \simeq S_0 \left(1 + \frac{\beta I_0}{\beta S_0 - \gamma} \right)$$

Now we want to plot S_∞ as a function of β :

In [26]:

```
%matplotlib inline
import matplotlib.pyplot as plt
import numpy as np
```

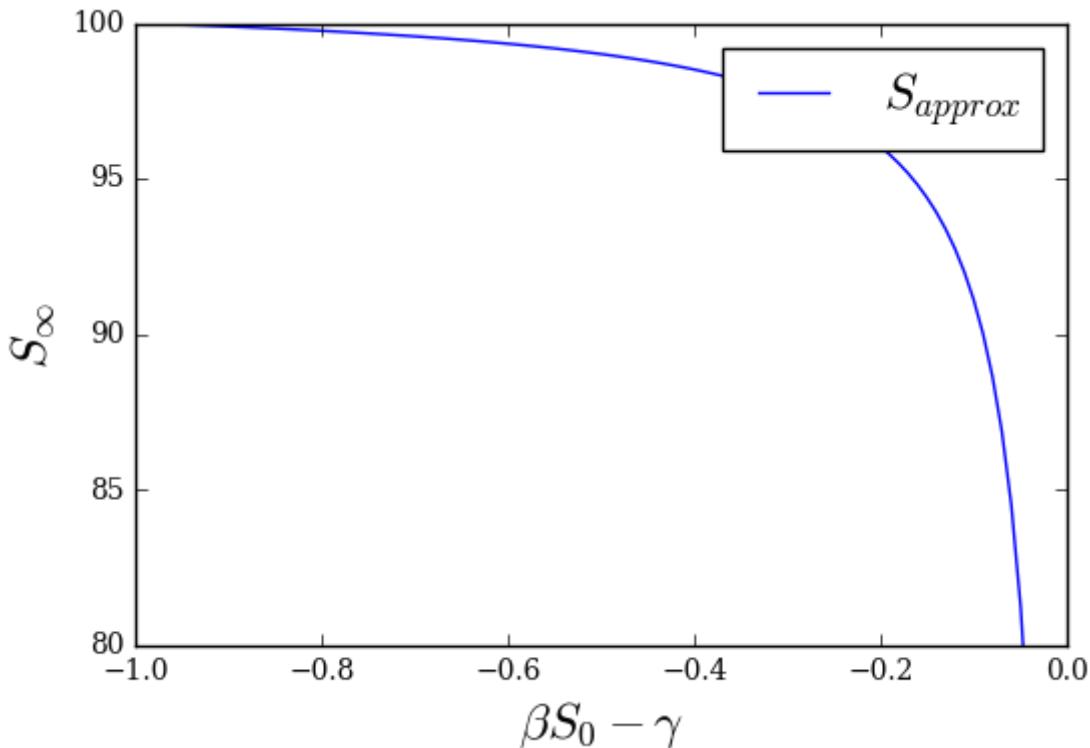
In [31]:

```
gamma = 1.0
S0 = 100.0
I0 = 1.0

bvals = np.linspace(0.0,0.01,100)
x = bvals*S0-gamma
Sapp = S0*(1+bvals*I0/(bvals*S0-gamma))

plt.plot(x,Sapp,label=r"$S_{approx}$")
plt.xlabel(r"$\beta S_0 - \gamma$ ", fontsize=18)
plt.ylabel(r"$S_{approx}$", fontsize=18)
plt.xlim([-1.0,0])
plt.ylim([80,100])
plt.legend(fontsize=18)
plt.show()
```

/usr/local/lib/python3.5/dist-packages/ipykernel/_main_.py:7: RuntimeWarning: divide by zero encountered in true_divide



Numerical solution

Usually we have to resort to numerical methods in order to solve sets of nonlinear differential equations. To this end, we write down our set in the following matrix form:

$$u'(t) = \begin{pmatrix} S'(t) \\ I'(t) \\ R'(t) \end{pmatrix} = \begin{pmatrix} -\beta I(t)S(t) \\ \beta I(t)S(t) - \gamma I(t) \\ \gamma I(t) \end{pmatrix}$$

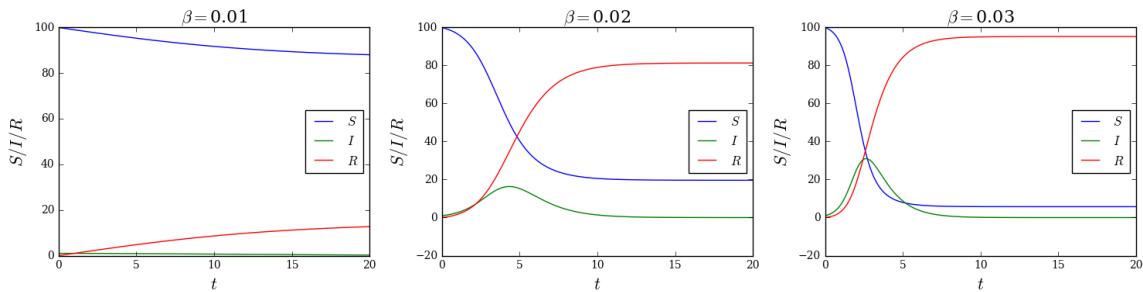
In [32]:

```
def rhs(u,t):
    S = u[0]
    I = u[1]
    R = u[2]
    return np.array([-beta*I*S,beta*I*S-gamma*I,gamma*I])
```

In [40]:

```
from scipy.integrate import odeint
gamma = 1.0
I0 = 1.0
S0 = 100.0
init= [S0,I0,0.0]
bvals = [0.01,0.02,0.03]
t = np.linspace(0,100,1000)

fig, axes = plt.subplots(1, 3, figsize=(15,4))
for i in range(len(bvals)):
    beta = bvals[i]
    u = odeint(rhs,init,t)
    axes[i].plot(t,u[:,0],label="$S$")
    axes[i].plot(t,u[:,1],label="$I$")
    axes[i].plot(t,u[:,2],label="$R$")
    axes[i].set_title(r"$\beta = $" + str(beta), fontsize=16)
    axes[i].set_xlabel(r"$t$", fontsize=16)
    axes[i].set_ylabel(r"$S/I/R$ ", fontsize=16)
    axes[i].legend(loc=5, fontsize=12)
    axes[i].set_xlim([0,20])
plt.tight_layout()
plt.show()
```



Some experimental data for comparisons:

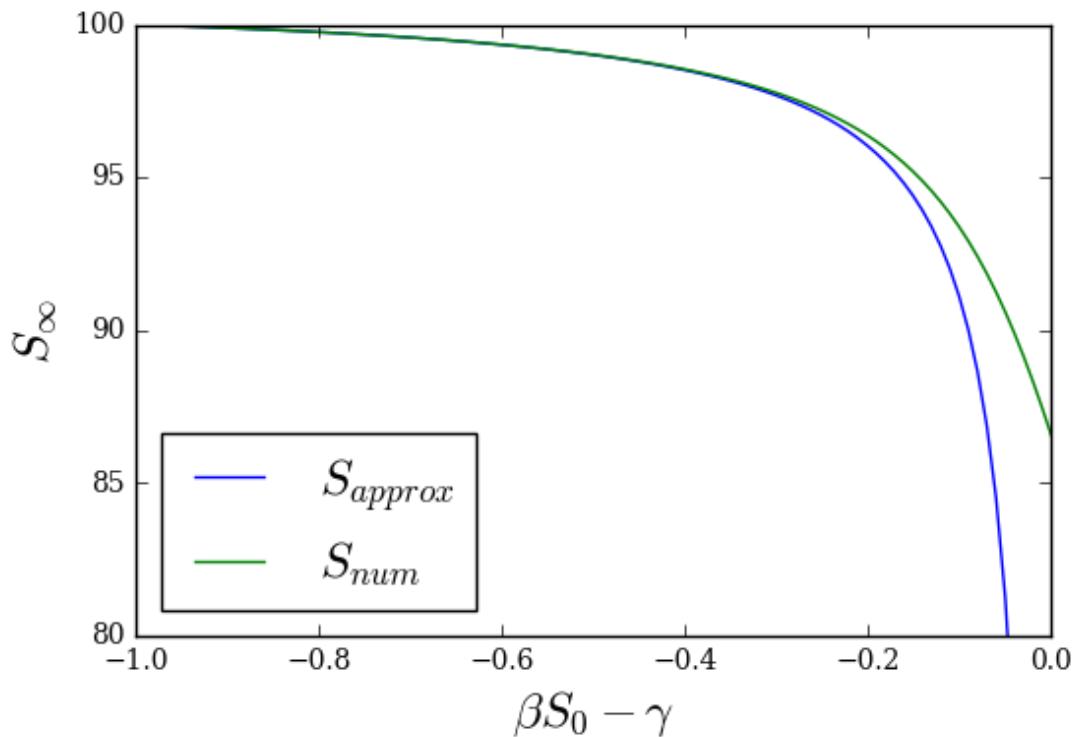
</table>

A. M. Correia, F. C. Mena, and A. J. Soares, An Application of the SIR Model to the Evolution of Epidemics in Portugal, in "Dynamics, Games and Science II"

Now we want to compare the analytic approximation with the numerical results:

In [41]:

In [42]:

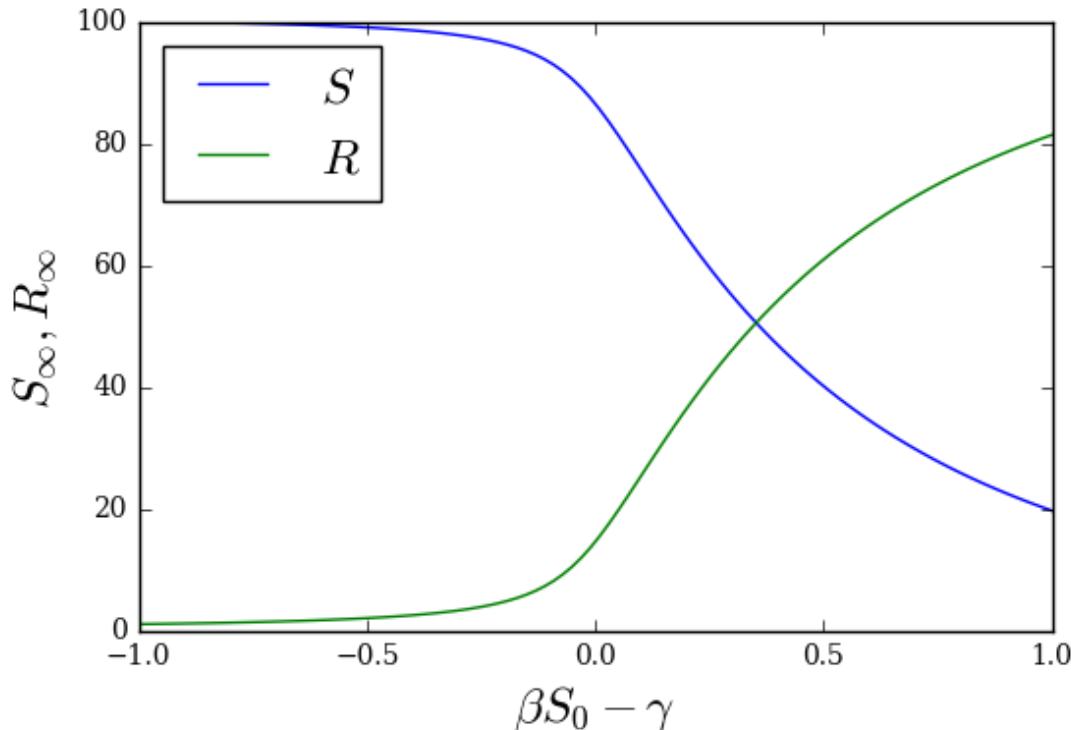


In line with our assumptions, the agreement is very good for small values of β .

It is interesting to look at the numerical solution for larger values of β as well:

In [43]:

In [44]:



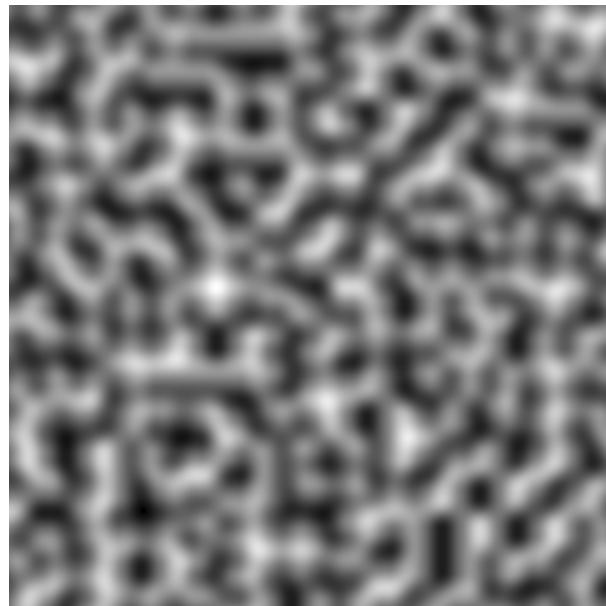
The point $\beta S_0 - \gamma = 0$ marks the transition between an initially decaying and growing number of infected agents, and this is marked by a relatively sudden change in behaviour. The disease can properly take hold and gain some momentum in the growing case, and the number of infections increases rapidly.

Numerical solution on a grid

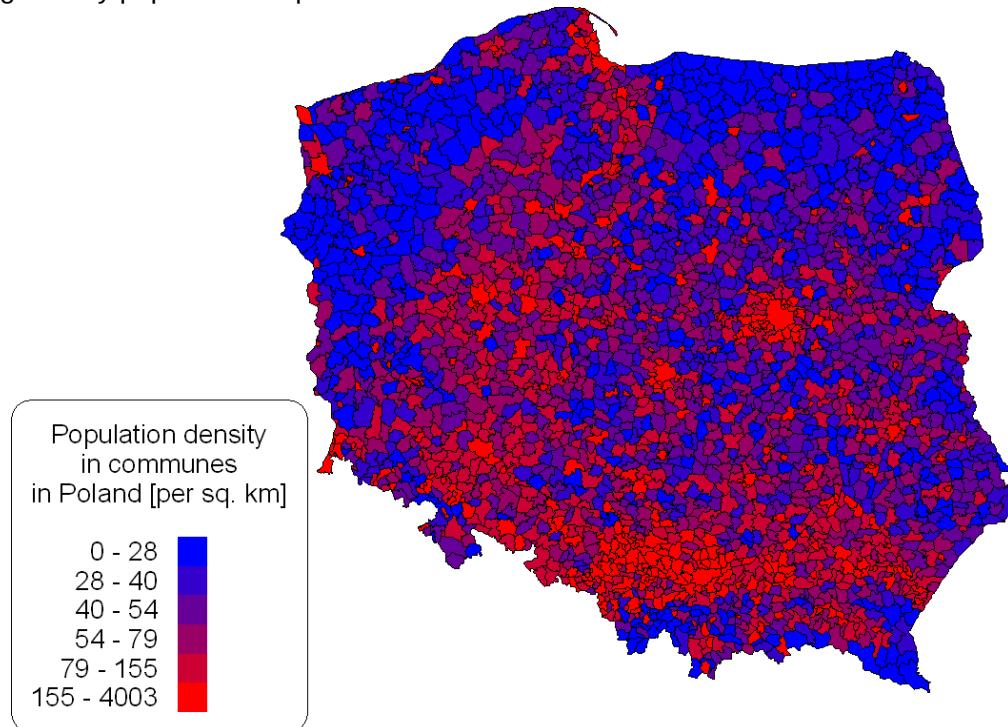
The above approach give already some insight into the dynamics of epidemics, but it is fairly limited in scope. All of the agents are assumed to be able to interact with one another, perhaps representing a population trapped in the same room. To extend this model, lets assume there are populations spread out on a grid, $S_{i,j}$, $I_{i,j}$, $R_{i,j}$ where (i, j) represent indices in a 2D array. The model equations are the same, except we allow neighbouring grid cells to interact with one another so infection can spread from cell to cell, i.e.

$$u'_i(t) = \begin{pmatrix} S'_i \\ I'_i \\ R'_i \end{pmatrix} = \begin{pmatrix} -\beta(S_{i,j}I_{i,j} + S_{i,j}I_{i-1,j} + S_{i,j}I_{i+1,j} + S_{i,j}I_{i,j-1} + S_{i,j}I_{i,j+1}) \\ \beta(S_{i,j}I_{i,j} + S_{i,j}I_{i-1,j} + S_{i,j}I_{i+1,j} + S_{i,j}I_{i,j-1} + S_{i,j}I_{i,j+1}) - \gamma I_{i,j} \\ \gamma I_{i,j} \end{pmatrix}$$

One often uses Perlin noise to generate random maps of initial population densities:



However, it is possible to use demographical data for the initial conditions. Let us consider for instance the following density population map for Poland:



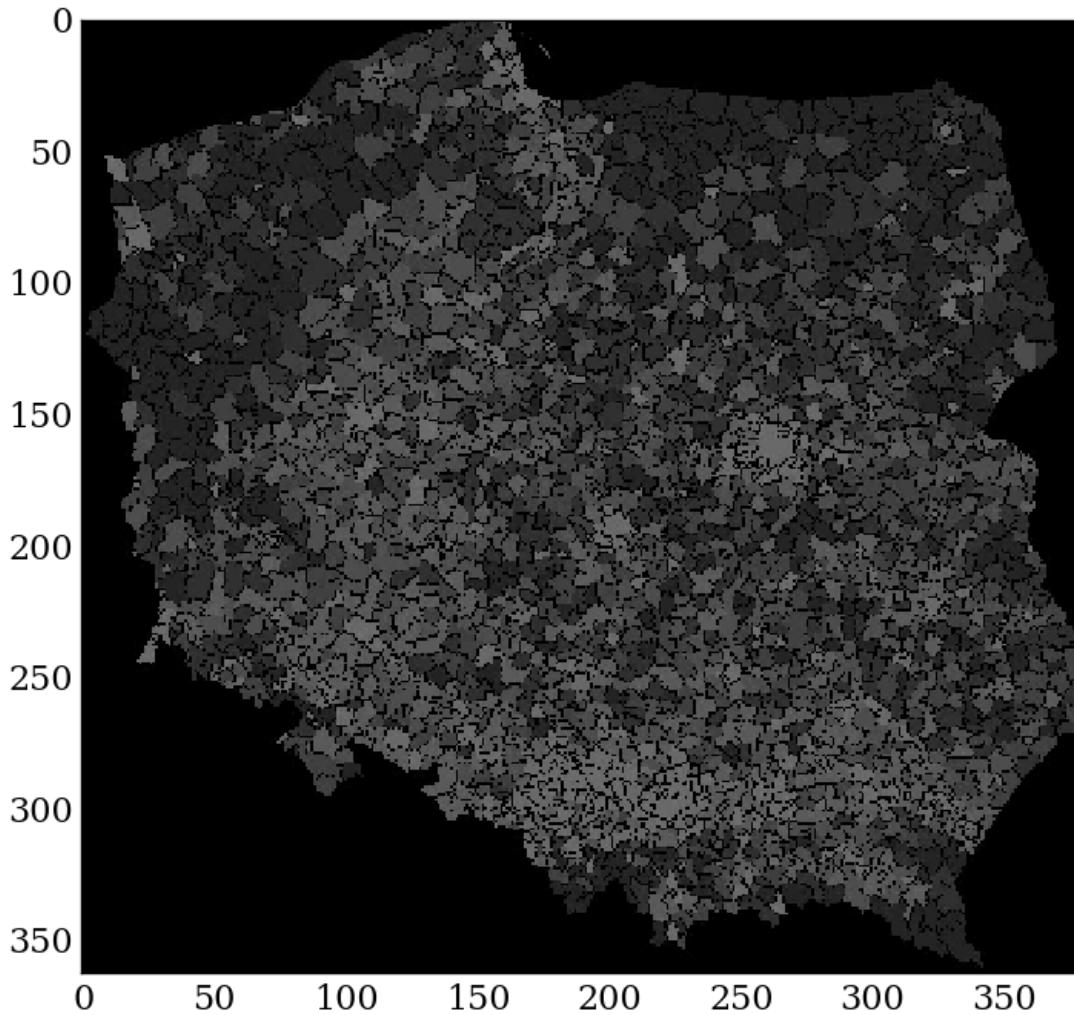
We remove the legend and convert it into grayscales with a software like GIMP:



We are ready to read the map and convert it into a numpy array:

In [45]:

(362, 379, 3)



The function `odeint` used before allows us to find a solution of a given set of differential equation over some time interval. However it is not the best choice for monitoring the time evolution of a system step by step. In this case it is better to use a method which provides a solution at time step $n + 1$ given the solution at previous time step/steps.

The simplest (and worst) possibility is the so called Euler method. Remember the definition of a derivative?

$$u'(t) = \lim_{\Delta t \rightarrow 0} \frac{u(t + \Delta t) - u(t)}{\Delta t}$$

With some rearranging it can actually be used to approximate the next step of the function when the derivative is known and Δt is assumed to be small. Since our set of equations may be written as

$$u'(t) = f(u)$$

then for small Δt we get

$$u_{n+1} = u_n + \Delta t * f(u)$$

The biggest advantage of this method is its simplicity:

In [55]:

Unfortunately, this method is only conditionally stable and inaccurate. Thus it is better to resort to the modified Euler method, also known as the Runge-Kutta method of the second order:

$$\begin{aligned} u_{n+1}^* &= u_n + \Delta t * f(u) \\ u_{n+1} &= u_n + \frac{\Delta t}{2} \{f(u_n) + f(u_{n+1}^*)\} \end{aligned}$$

In [56]:

Recall the equation to solve:

$$u'_i(t) = \begin{pmatrix} S'_i \\ I'_i \\ R'_i \end{pmatrix} = \begin{pmatrix} -\beta(S_{i,j}I_{i,j} + S_{i,j}I_{i-1,j} + S_{i,j}I_{i+1,j} + S_{i,j}I_{i,j-1} + S_{i,j}I_{i,j+1}) \\ \beta(S_{i,j}I_{i,j} + S_{i,j}I_{i-1,j} + S_{i,j}I_{i+1,j} + S_{i,j}I_{i,j-1} + S_{i,j}I_{i,j+1}) - \gamma I_{i,j} \\ \gamma I_{i,j} \end{pmatrix}$$

First we need a function defining the right hand side of the above set:

In [48]:

At $t = 0$ all population belongs to the susceptible category S . There are neither infected nor recovered individuals in the system:

In [57]:

We inject an infected person somewhere in central Poland:

In [58]:

Let us initialize the simulation:

In [60]:

Then we simulate the system

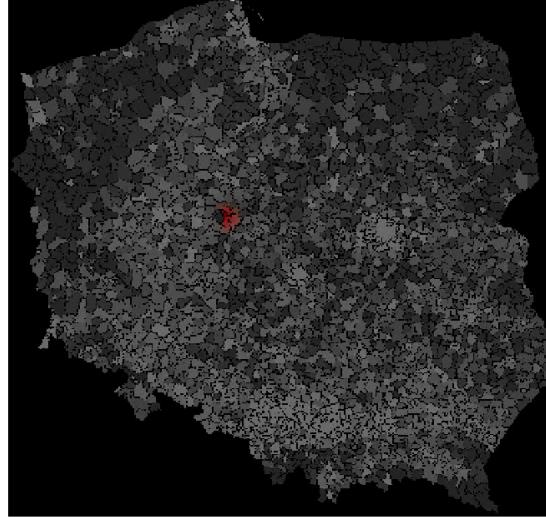
In [61]:

and save the solution for each step into a png file:

In [62]:

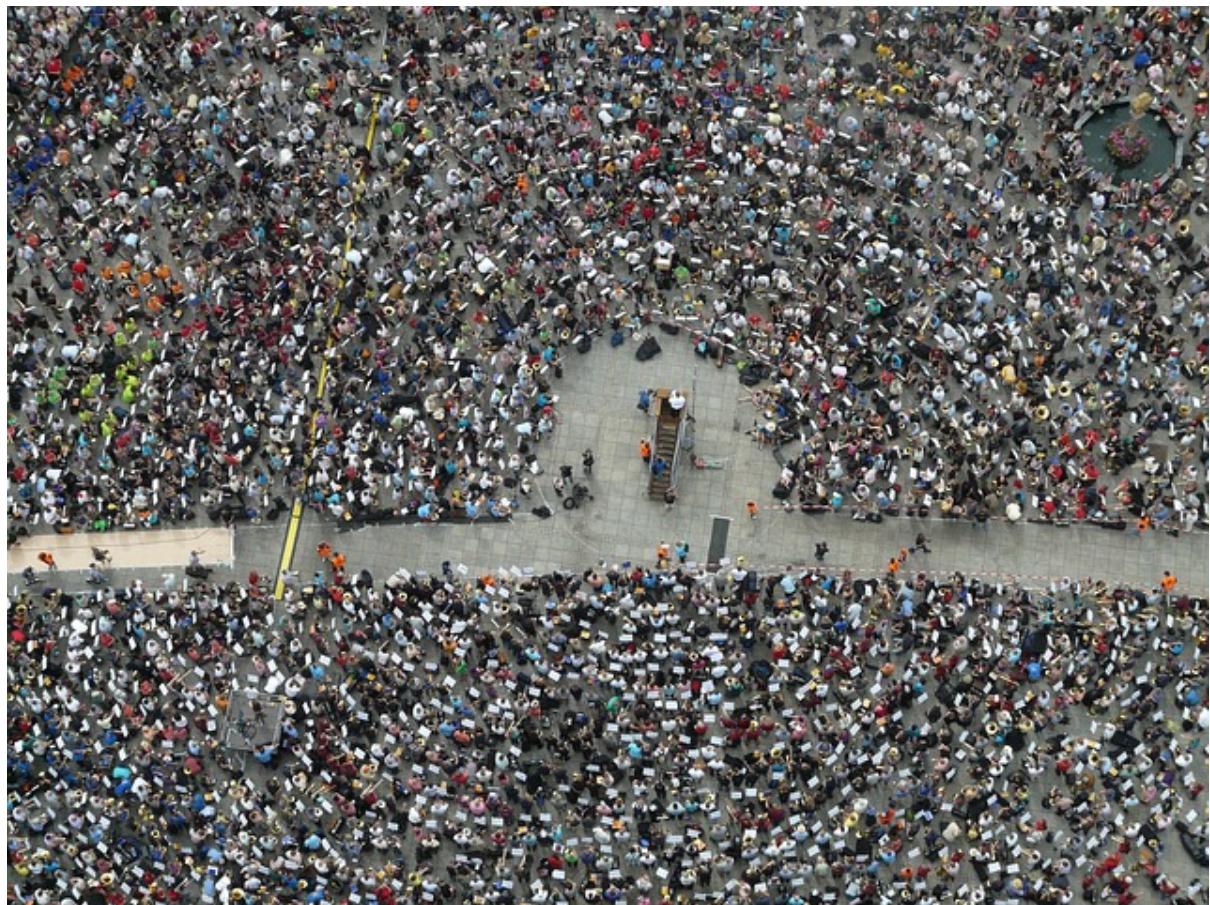
The png files are used to prepare an animated gif:

In [63]:



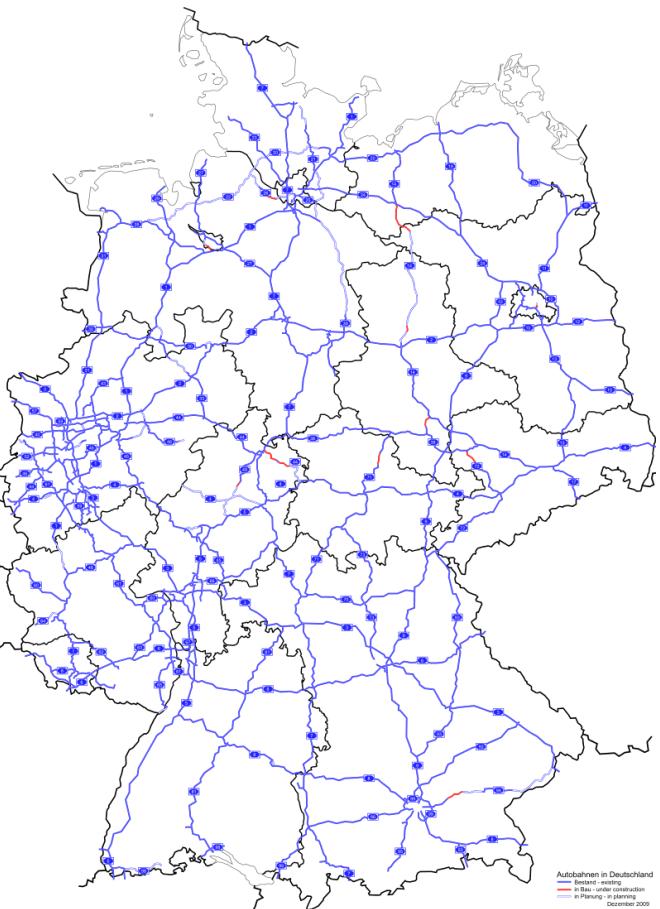
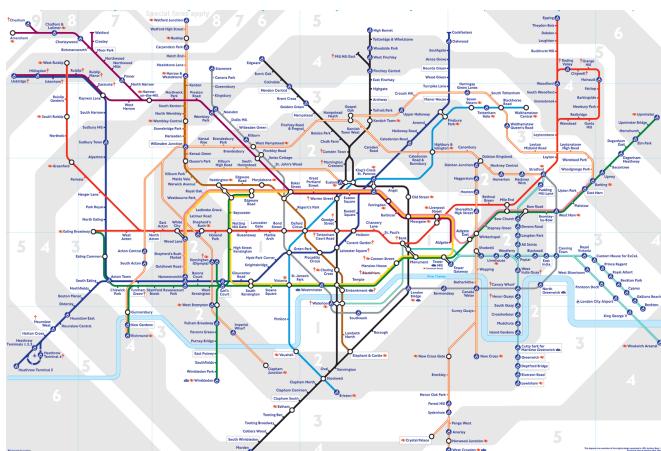
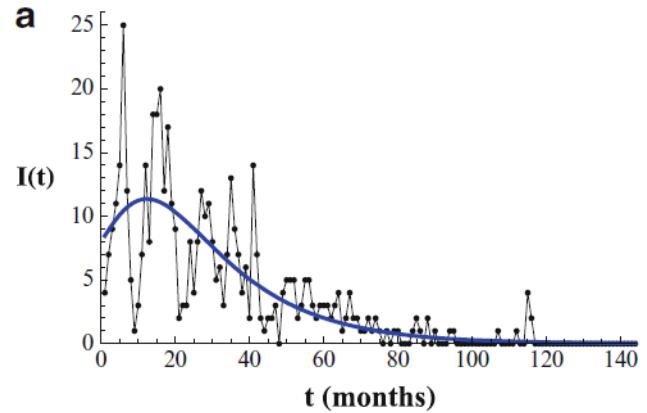
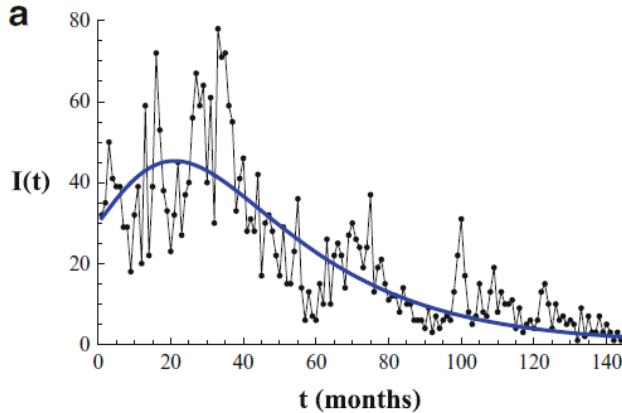
Simulations on networks

Beside the characteristics of a pathogen, the interactions within a population are very important for the disease spreading.



Measles

Viral hepatitis





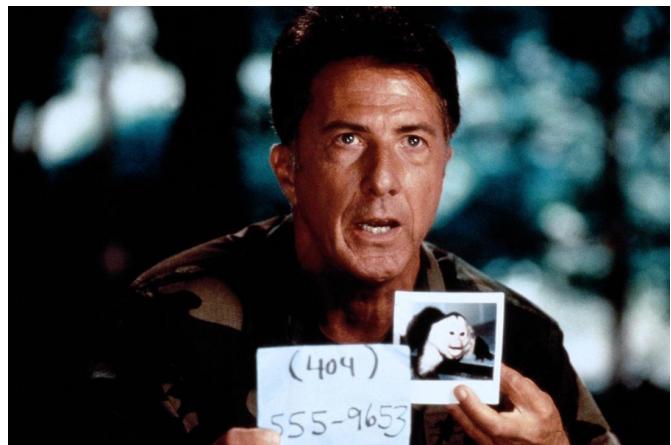
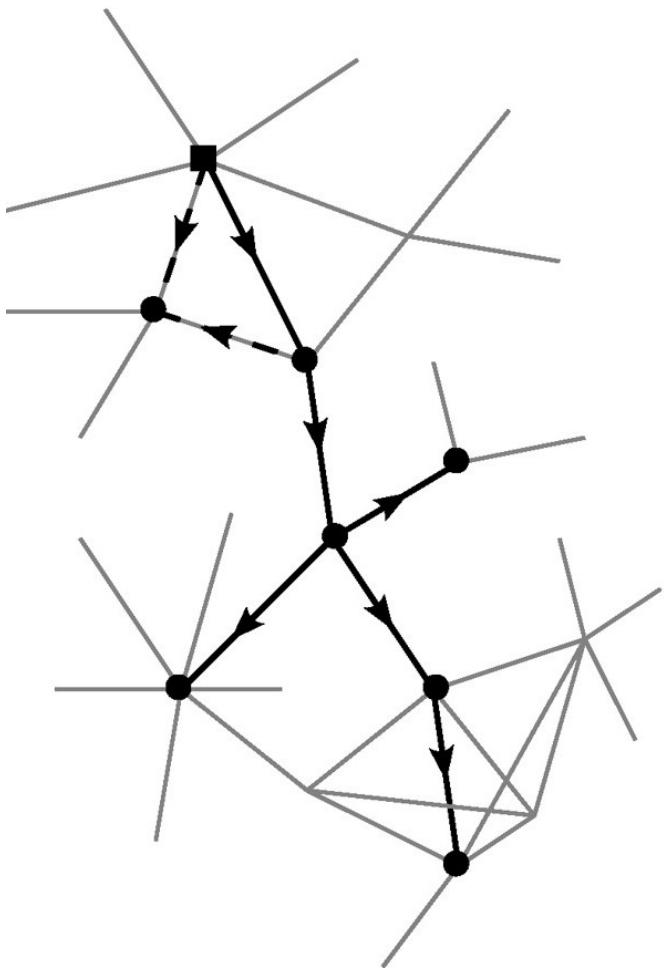
Social and transport networks give some insights into the directions of spreading. However, it is important to find the contact network for a given pathogen, because it defines the transmission channels of the disease.

Challenges in determining a contact network

- requires knowledge of all individuals and their interactions within a population
- possible only for small populations in a finite time
- if sampling possible, two problems:
 - people are reluctant to disclose some information (e.g. sexual contacts)
 - how to define an edge? For instance, how long does it take to catch a flu virus if in contact with an infected person?

Common techniques to determine contact networks:

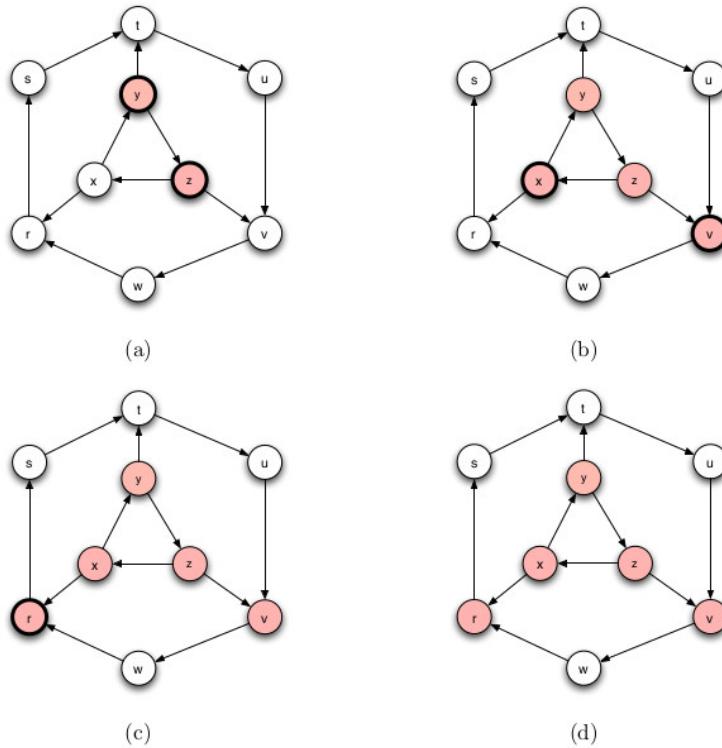
- **infection tracking** - identifying the source of infection (patient "zero"), then people who caught the infection from him etc. → tree-like network (cascading process)



- **contact tracking** - for a given person you look for all potential contacts (again, problems with the definition of a contact)
- **diary analysis** - ill people try to chronologically reproduce the list of their contacts

SIR model on networks

After determining the network structure, the actual simulation is pretty simple.



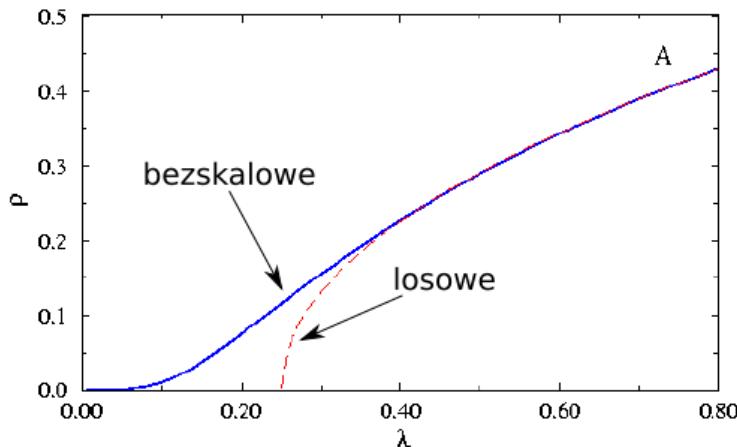
D. Easley, J. Kleinberg, "Networks, Crowds and Markets: Reasoning about a Highly Connected World"

Very often one uses artificial network models to understand the impact of some network characteristics on the spreading:

</table>

Keeling MJ, Eames KTD (2005) Networks and epidemic models. J R Soc Interface 2: 295–307.

One of the most important results of such simulations is that one concerning the **epidemic threshold** - there is no such threshold on scale-free networks. In this case the disease spreads quickly over the entire network.



R. Pastor-Satorras and A. Vespignani, PRL 86, 3200 (2001)

Here, the parameter λ is the *basic reproduction number*, defined as

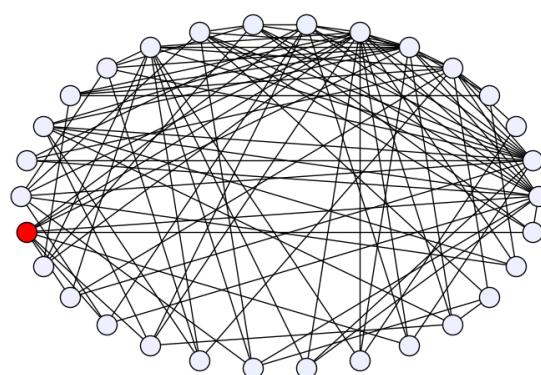
$$\lambda = \frac{\beta}{\gamma}$$

Such simulations are of course possible in Python, in particular in connection with the `networkx` module. It should be noted however that due to efficiency limitations of Python one may be forced to resort to compiled languages like C/C++ or Fortran in case of very large networks.

In [64]:

In [75]:

In [76]:



Bibliography

1. Daley, D. J. & Gani, J. (2005). *Epidemic Modeling: An Introduction*. NY: Cambridge University Press.
2. *Going Viral*, <http://jasmcole.com/2014/10/19/going-viral/> (<http://jasmcole.com/2014/10/19/going-viral/>)
3. Max Berggren, *Model of a zombie outbreak in Sweden, Norway and Finland (Denmark is fine)*, <http://maxberggren.se/2014/11/27/model-of-a-zombie-outbreak/> (<http://maxberggren.se/2014/11/27/model-of-a-zombie-outbreak/>)
4. <http://matplotlib.org/> (<http://matplotlib.org/>)
5. <https://networkx.github.io/> (<https://networkx.github.io/>)

